



HAL
open science

Co-conception d'un système commandé en réseau sans fil à l'aide de réseaux bayésiens distribués

Amine Mechraoui

► **To cite this version:**

Amine Mechraoui. Co-conception d'un système commandé en réseau sans fil à l'aide de réseaux bayésiens distribués. Automatique / Robotique. Institut National Polytechnique de Grenoble - INPG, 2010. Français. NNT: . tel-00594366

HAL Id: tel-00594366

<https://theses.hal.science/tel-00594366v1>

Submitted on 19 May 2011

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.

UNIVERSITE DE GRENOBLE
INSTITUT POLYTECHNIQUE DE GRENOBLE

No. attribué par la bibliothèque

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|

Rapport

pour obtenir le grade de

DOCTEUR DE L'Université de Grenoble
délivré par l'Institut polytechnique de Grenoble

Spécialité : AUTOMATIQUE-PRODUCTIQUE

préparée au laboratoire Grenoble Images Parole Signal Automatique

dans le cadre de l'École Doctorale :

Électronique, Électrotechnique, Automatique, Traitement du Signal

présentée et soutenue publiquement

par

Amine MECHRAOUI

le 14 Décembre 2010

Titre :

**Co-conception d'un système commandé en réseau sans fil à
l'aide de réseaux bayésiens distribués**

Directeurs de thèse :

M. Jean-Marc THIRIET (GIPSA-Lab - UJF)
Mme. Sylviane GENTIL (GIPSA-Lab - INPG)

JURY :

| | |
|-----------------------------|--------------------|
| Mme. Françoise SIMONOT-LION | Présidente |
| M. Abdessamad KOBI | Rapporteur |
| M. Thierry VAL | Rapporteur |
| M. Christophe SIMON | Examineur |
| M. Jean-Marc THIRIET | Directeur de thèse |
| Mme. Sylviane GENTIL | Co-directrice |

*A mes parents Amar et Assia,
à ma femme Nabila,
à mes sœurs Amira, Amani, Fedoua et Hala,
et à ma petite nièce Line.*

Remerciements

Cette thèse n'a qu'un seul auteur mais elle est le fruit du travail et des encouragements de plusieurs personnes que je voudrais remercier chaleureusement.

Je tiens d'abord à exprimer mes remerciements et ma gratitude à Madame Sylviane GENTIL et Monsieur Jean-Marc THIRIET qui m'ont permis, grâce à leur confiance et leur soutien, de mener à terme ce travail. Je les remercie aussi pour leur extrême gentillesse et leur bonne humeur communicative.

Je tiens à exprimer ma sincère reconnaissance à mes deux rapporteurs M. Abdessamad KOBI de l'Université d'Angers et M. Thierry VAL de l'Université du Mirail, pour avoir accepté d'être rapporteurs de mes travaux de thèse ainsi que pour leur lecture minutieuse du manuscrit. Merci également à Madame Françoise SIMONOT-LION, Professeur à l'Institut National Polytechnique de Lorraine, de m'avoir fait l'honneur de présider le jury de cette thèse.

Un très grand merci à M. Christophe SIMON, Maître de conférence à l'Universités de Nancy, pour ses conseils et pour avoir accepté d'examiner ce travail de thèse.

Mes remerciements les plus sincères à Marie-Thérèse, Marie-Rose, Patricia, Jonathan et Daniel pour leur amabilité et leur travail remarquable.

J'ai été heureux de travailler en compagnie des doctorants du laboratoires Gipsa-Lab département automatique. Je pense particulièrement à mes collègues de bureau Hieu Trong DO et Zeashan Hameed KHAN. A mes amis Irfan AHMAD, Cédric BERBRA, Marouane ALMA, Mohammad HAFFAR, Hoang Van NGUYEN et sa femme, Anh Lam DO, Van Thang PHAM, Antoine LEMARCHAND, Gabriel RODRIGUES DE CAMPOS, Andra Ioana VASILIU, Oumayma OMAR, Flor Lizeth TORRES ORTIZ, Joumana HERMASSI, Adriana Simona MIHAITA, Emilie ROCHE, Valentina CIARLA et Latéfa GHOMRI. Je pense aussi aux deux roumains sympathiques Claudiu BALAN et Razvan DORDEA.

Je tiens aussi à remercier mes chers amis Ahmed BENZERROUK, Sofiane MESSAI et Haithem DERBEL pour leur soutien et d'avoir accepté de venir de loin pour assister à ma soutenance.

Je remercie aussi mes amis en Algérie, Fouaz SID, Lotfi BOUMAIDA, Moustapha CHIAD, Sami BOUCHAREB, Yazid BOUDAREN, Riad BOUGHRARA, Salah ZEROUAL, Aziz TADJINE, Brahim BAZEMLAL et Belkacem MAMEN.

Enfin, je tiens à remercier vivement mon père et ma chère mère, ma femme Nabila, mes soeurs si particulières Amira, Amani, Fadwa et Hala pour leur soutien. Ils ont toujours été derrière moi à me transmettre de l'énergie. Sans eux, beaucoup de choses n'auraient pas été possibles.

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction générale | 27 |
| 2 | Diagnostic et commande de systèmes en réseau | 31 |
| 2.1 | Introduction | 31 |
| 2.2 | Systèmes commandés en réseau (NCS) | 32 |
| 2.3 | Analyse de stabilité pour les NCSs | 34 |
| 2.4 | Estimation pour les NCSs | 35 |
| 2.5 | Synthèse de commande des NCSs | 36 |
| 2.6 | Co-conception (Commande/Communication) | 36 |
| 2.7 | Diagnostic des systèmes commandés à travers un réseau | 38 |
| 2.7.1 | Diagnostic des systèmes | 38 |
| 2.7.2 | Classification des méthodes de diagnostic | 39 |
| | Méthodes quantitatives : | 40 |
| | Méthodes qualitatives : | 42 |
| 2.7.3 | Architectures du diagnostic | 44 |
| 2.8 | Conclusion | 47 |
| 3 | Architecture de commande via un réseau d'un robot unicycle auto- nome | 49 |
| 3.1 | Introduction | 49 |
| 3.2 | Le robot Khepera III et son modèle | 50 |
| 3.2.1 | Le robot | 50 |
| | Processeur et micro-contrôleur | 51 |
| | Les moteurs | 51 |
| | Les encodeurs | 52 |
| | Les capteurs | 53 |
| | Les protocoles de communication | 57 |
| 3.2.2 | Modèle du robot | 57 |
| | Modèle cinématique | 57 |
| | Modèle dynamique | 59 |
| 3.3 | Architecture de commande pour robot autonome | 59 |

| | | |
|----------|--|-----------|
| 3.3.1 | Schéma général de la commande | 60 |
| 3.3.2 | Implantation | 60 |
| | Commande en vitesse du robot en boucle ouverte | 60 |
| | Commande de position | 64 |
| 3.4 | Architecture de commande temps réel du robot | 65 |
| 3.4.1 | Programmation sous TrueTime | 65 |
| 3.4.2 | Architecture de commande temps réel | 66 |
| 3.5 | L'architecture de la commande du robot via un réseau sans fil | 67 |
| 3.5.1 | Simulation, expérimentation et résultats | 69 |
| | Atteinte d'une cible | 69 |
| | Suivi de trajectoire | 72 |
| | Évitement d'obstacles | 74 |
| 3.6 | Conclusion | 77 |
| 4 | Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC | 79 |
| 4.1 | Introduction | 80 |
| 4.2 | Analyse de performance du réseau sans fil Wi-Fi fonctionnant sous DCF | 81 |
| 4.2.1 | Fonctionnement du DCF (Distributed Coordination Function) | 82 |
| 4.3 | Evaluation des performances avec un réseau saturé | 83 |
| 4.3.1 | Analyse du débit | 83 |
| | Probabilité de transmission de paquet | 84 |
| | Débit | 87 |
| 4.3.2 | Retard d'un paquet avec le protocole IEEE 802.11 | 87 |
| 4.4 | Adaptation du réseau aux contraintes dynamiques de la commande | 88 |
| 4.4.1 | Handoff | 89 |
| | Pourquoi changer de station ? | 89 |
| 4.4.2 | Types de Handoff | 89 |
| | Soft Handoff | 90 |
| | Hard Handoff | 90 |
| 4.4.3 | Aide à la décision pour déclencher le Handoff | 90 |
| | Résultats de simulation | 93 |
| 4.5 | Adaptation de la commande aux contraintes dynamiques du réseau | 96 |
| 4.6 | Paramètres de QoS | 97 |
| 4.6.1 | Débit (<i>Throughput</i>) | 98 |
| 4.6.2 | Retard (<i>Delay</i>) | 98 |
| 4.6.3 | Gigues (<i>Jitter</i>) | 100 |
| 4.6.4 | Pertes de paquets (<i>Packet Loss</i>) | 100 |
| 4.7 | Étude de l'effet de la dégradation de la QoS du réseau sur la QoC du robot | 101 |
| 4.7.1 | Effet de la charge du réseau sur la QoC du robot | 102 |
| | Étude du temps de transmission | 102 |

| | | |
|----------|--|------------|
| | Analyse du retard d'un paquet pour le protocole 802.11 | 103 |
| | Spécifications | 104 |
| | Etude de la charge en fonction de la période d'envoi | 105 |
| | Etude de la charge en fonction de la longueur de la trame | 106 |
| | 4.7.2 Effet des pertes de paquet sur la QoC du robot | 110 |
| 4.8 | Environnement multi-robots communicants | 113 |
| | 4.8.1 Architecture de communication | 113 |
| 4.9 | Conclusion | 114 |
| 5 | Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués | 117 |
| 5.1 | Introduction | 118 |
| 5.2 | Pourquoi les réseaux bayésiens? | 118 |
| 5.3 | Les réseaux bayésiens (RBs) | 119 |
| | 5.3.1 Approche qualitative | 120 |
| | Identification des variables et de leurs espaces d'états | 120 |
| | Détermination de la structure du réseau bayésien | 122 |
| | 5.3.2 Approche quantitative | 122 |
| 5.4 | Expertise et étude statistique | 123 |
| | 5.4.1 Influence des pertes de paquets sur la QoC du robot : | 124 |
| | 5.4.2 Influence de la distance et de la présence d'obstacles sur la puissance du signal reçu (RSSI) : | 129 |
| | 5.4.3 Influence du retard et du RSSI sur le pourcentage des pertes de paquets : | 129 |
| | 5.4.4 Étude du délai de transmission : | 130 |
| | 5.4.5 Détermination des probabilités marginales des nœuds sans parent : | 131 |
| 5.5 | Aide à la décision et diagnostic | 131 |
| | 5.5.1 Aide à la décision | 132 |
| | 5.5.2 Diagnostic | 133 |
| 5.6 | Apprentissage de paramètres | 133 |
| | 5.6.1 Critère de la QoS | 133 |
| | 5.6.2 Critère de la QoC | 135 |
| | 5.6.3 Validation du réseau bayésien trouvé par apprentissage | 137 |
| 5.7 | Réseaux bayésiens distribués | 139 |
| | 5.7.1 Prise de décisions et diagnostic distribué | 140 |
| | 5.7.2 Priorités dynamiques | 141 |
| | 5.7.3 Résultats et simulation | 142 |
| | 5.7.4 Scénario 1 | 143 |
| | 5.7.5 Scénario 2 | 144 |
| 5.8 | Conclusion | 145 |

| | | |
|----------|--|------------|
| 6 | Conclusion générale | 147 |
| A | Réseaux bayésiens | 151 |
| A.1 | Introduction | 151 |
| A.2 | Définition des réseaux bayésiens : | 152 |
| A.3 | Approche qualitative | 153 |
| A.3.1 | La représentation graphique de la causalité : | 153 |
| A.3.2 | D-séparation (bloquage) | 155 |
| A.4 | Approche quantitative | 156 |
| A.4.1 | Probabilités | 159 |
| A.4.2 | Circulation d'information et indépendance conditionnelle | 163 |
| A.4.3 | Conclusion | 163 |
| A.5 | Inférence | 164 |
| A.5.1 | Introduction | 164 |
| A.5.2 | Algorithme de Pearl | 165 |
| | Généralisation de l'algorithme | 167 |
| | Exemple | 168 |
| | Calculs des probabilités marginales | 169 |
| | Calculs des probabilités a posteriori | 171 |
| A.5.3 | Algorithme de l'arbre de jonction dit JLO | 173 |
| | La moralisation du graphe | 175 |
| | La triangulation | 176 |
| | L'arbre de jonction | 177 |
| B | Réseaux sans fil | 179 |
| B.1 | Les technologies | 180 |
| B.1.1 | Les WWAN (IEEE 802.20) | 181 |
| B.1.2 | Les WRAN (IEEE 802.22) | 182 |
| B.1.3 | Les WMAN (IEEE 802.16) | 182 |
| B.1.4 | Les WLAN (IEEE 802.11) | 182 |
| B.1.5 | Les WPAN(IEEE 802.15) | 183 |
| B.1.6 | Autres types de réseaux sans fil | 183 |
| B.2 | Fonctionnement | 183 |
| B.2.1 | Fonctionnement en mode Ad-Hoc | 184 |
| B.2.2 | Fonctionnement en mode infrastructure | 184 |
| B.2.3 | L'accès au médium | 185 |
| B.3 | IEEE 802.11 Wi-Fi | 186 |
| B.3.1 | Le protocole Wi-Fi | 186 |
| B.3.2 | Les modes d'accès au médium | 187 |
| B.4 | 802.11e | 189 |

| | | |
|-------|--|-----|
| B.4.1 | EDCF (Enhanced DCF) ou EDCA (Enhanced Distribution Channel Access) | 190 |
| B.4.2 | EPCF (Enhanced PCF) ou HCF (Hybrid Coordination Function) | 191 |
| B.5 | IEEE 802.15.4 | 191 |
| B.6 | Le protocole Zigbee | 192 |
| B.6.1 | La couche physique | 192 |
| B.6.2 | La sous couche d'accès au médium (MAC) | 193 |
| B.7 | Le mode d'accès au médium | 194 |

Bibliographie

Liste des abbréviations

| | |
|-------------------|---|
| AD | Arbre de Décision |
| ACK | Acknowledgement |
| AIFS | Arbitration Inter Frame Space |
| AP | Access Point |
| BER | Bit Error Rate |
| CA | Collision Avoidance |
| CD | Collision Detection |
| CDF | Complementary Cumulative Distribution function |
| CFP | Contention Free Period |
| CF-Poll | Contention Free-Poll |
| CF-End | Contention Free-End |
| CP | Contention Period |
| CPT | Conditional Probability Table |
| CSMA | Carrier Sense Multiple Access |
| CW | Contention Window |
| CW _{max} | Contention Window maximum |
| CW _{min} | Contention Window minimum |
| DCF | Distributed Coordination Function |
| DIFS | DCF Inter Frame Space |
| DLL | Delay Lower Limit |
| EDCF | Enhanced DCF |
| FEC | Forward Error Correction |
| GTS | Guaranteed Time Slot |
| HC | Hybrid Coordination |
| HCF | Hybrid coordination function |
| HHO | Horizontal HandOff |
| IEEE | Institute of Electrical and Electronics Engineers |
| ISM | Industrial, Science, Medical |
| LRE | Limited Relative Error |
| MAC | Medium Access control |
| MD | Minimum Delay |
| MSDU | MAC Service Data Unit |

| | |
|----------|------------------------------------|
| MT | Minimum Throughput |
| NAV | Network Allocated Vector |
| NCS | Networked control system |
| NL | Network Load |
| NRF | Network rate Function |
| PAN | Personal Area Network |
| PC | Point Coordinator |
| PCF | Point Coordinator Function |
| PF | Persistence Factor |
| PHY mode | Physical layer mode |
| PIFS | PCF Inter Frame Space |
| QoC | Quality of Control |
| (Q)BSS | (QoS-supporting)Basic Service Set |
| QoS | Quality of Service |
| RàPC | Raisonnement à Partir de Cas |
| RB | Réseaux Bayésiens |
| AP | Access Point |
| RN | Réseaux de neurones |
| RSSI | Request Signal Strength Indicator |
| SE | Systemes Experts |
| SIFS | Short Inter Frame Space |
| TBTT | Target Beacon Transmission Time |
| TC | Traffic Category |
| TXoP | Transmission Opportunity |
| URF | Use Request Factor |
| VHO | Vertical HandOff |
| WLAN | Wireless Local Area Network |
| WPAN | Wireless Personal Area Network |
| WNCS | Wireless Networked Control Systems |

Table des figures

| | | |
|------|--|----|
| 1.1 | Problème de co-conception | 28 |
| 1.2 | Scénario général | 30 |
| 2.1 | Système commandé en réseau | 32 |
| 2.2 | Problème de co-conception | 37 |
| 2.3 | Classification des méthodes de diagnostic | 40 |
| 2.4 | Architecture générale de la détection de défauts à base de modèles | 41 |
| 2.5 | Diagnostic centralisé | 44 |
| 2.6 | Diagnostic décentralisé | 45 |
| 2.7 | Diagnostic distribué | 46 |
| 2.8 | Diagnostic distribué dans des systèmes commandés en réseau | 47 |
| 3.1 | Robot Khepera III | 51 |
| 3.2 | Réponse du moteur à un échelon de tension | 52 |
| 3.3 | Odométrie | 53 |
| 3.4 | Capteurs du Khepera III | 54 |
| 3.5 | Position des capteurs infrarouges | 55 |
| 3.6 | Réponse d'un capteur IR en fonction de la distance | 55 |
| 3.7 | Position des capteurs infrarouges | 56 |
| 3.8 | Zones de détection des capteurs ultrasons | 56 |
| 3.9 | Modèle cinématique du robot | 58 |
| 3.10 | Contrôle du robot | 59 |

Table des figures

| | |
|---|----|
| 3.11 Schéma général de la commande du robot simulé en Matlab/Simulink (Bloc continu) | 60 |
| 3.12 Commande en vitesse en boucle ouverte | 61 |
| 3.13 Schéma de la partie dynamique du système | 61 |
| 3.14 Réponse simulée du système en boucle ouverte en appliquant un échelon à la tension d'entrée du moteur | 62 |
| 3.15 Réponse indicielle du système en boucle fermée avec $K_{pk} = 3.5$ et $T_{ik} = 0.1$ | 63 |
| 3.16 Schéma général réduit de l'architecture de commande du robot | 65 |
| 3.17 Blocs TrueTime | 66 |
| 3.18 Schéma général de la régulation via un réseau sans fil | 68 |
| 3.19 Schéma bloc de la régulation locale des actionneurs du robot (Bloc B) . . | 68 |
| 3.20 Scénario général | 69 |
| 3.21 Trajectoire du robot avec un point d'accès WLAN | 70 |
| 3.22 Evolution des vitesses linéaire, angulaire et de l'orientation du robot avec réseau et une station | 70 |
| 3.23 Trajectoire du robot en simulation | 71 |
| 3.24 Évolution des vitesses linéaire, angulaire et de l'orientation du robot . . . | 71 |
| 3.25 Comparaison entre le robot en simulation et celui en expérimentation . . | 72 |
| 3.26 Trajectoire du robot en simulation | 73 |
| 3.27 Évolution des vitesses linéaire, angulaire et de l'orientation du robot en simulation | 73 |
| 3.28 Evitement d'obstacles | 74 |
| 3.29 Architecture de commande hybride d'atteinte d'une cible avec évitement d'obstacle | 75 |
| 3.30 Stratégie de commutation de contrôleur pour l'évitement d'obstacles . . . | 76 |
| 3.31 Trajectoires du robot pour atteindre une cible avec un seul obstacle . . . | 76 |
| 3.32 Trajectoires du robot pour atteindre une cible avec deux obstacles | 77 |
| 4.1 Modèle de chaîne de Markov du Backoff | 85 |

Table des figures

| | | |
|------|---|-----|
| 4.2 | Retard en fonction de la probabilité de collision et avec différentes valeurs de CW_{min} | 88 |
| 4.3 | Soft Handoff | 90 |
| 4.4 | Hard Handoff | 90 |
| 4.5 | Diagramme d'état du HHO | 92 |
| 4.6 | Trajectoire du robot avec ($HHO_1 : NL_{S_1} = 0$) et ($HHO_2 : NL_{S_1} = 0.8$ et $NL_{S_2} = 0.2$) | 94 |
| 4.7 | Évolution de la communication avec différentes stations avec $NL=0$. . . | 95 |
| 4.8 | Evolution de la communication avec différentes stations avec $NL_{S_1} = 0.8$ et $NL_{S_2} = 0.2$ | 96 |
| 4.9 | Trajectoire du robot avec la commutation de contrôleur sur la partie non couverte | 97 |
| 4.10 | Evolution des vitesse linéaire et angulaire et de l'orientation avec la commutation de contrôleur | 98 |
| 4.11 | Retard bout en bout | 99 |
| 4.12 | Retard dû à la file d'attente (Buffer) | 100 |
| 4.13 | Gigues (<i>Jitter</i>) | 101 |
| 4.14 | Trajectoire du robot en simulation pour atteindre une cible | 105 |
| 4.15 | Evolution des vitesses linéaire, angulaire et de l'orientation du robot en simulation | 106 |
| 4.16 | Retard de transmission (contrôleur/capteur) et ordonnancement de différentes tâches en simulation avec 100% de charge | 107 |
| 4.17 | Retards (capteur-contrôleur et contrôleur-actionneur) et ordonnancement des différentes tâches avec 50% de charge | 108 |
| 4.18 | Trajectoire du robot en simulation pour atteindre une cible en fonction de la charge du réseau | 108 |
| 4.19 | Retards (capteur-contrôleur et contrôleur-actionneur) et ordonnancement des différentes tâches avec 90% de charge | 109 |
| 4.20 | Retards (capteur-contrôleur et contrôleur-actionneur) et ordonnancement des différentes tâches avec 95% de charge | 109 |
| 4.21 | Distance entre le robot et l'obstacle en fonction de la charge du réseau . . | 110 |

Table des figures

| | | |
|------|---|-----|
| 4.22 | Trajectoire du robot avec obstacle en fonction de la charge du réseau . . . | 110 |
| 4.23 | Dégradation de la QoC en fonction des pertes de paquets | 111 |
| 4.24 | Différentes trajectoires du robot en simulation en fonction de taux de pertes de paquets fixé dans TrueTime | 112 |
| 4.25 | Évolution des vitesses linéaire, angulaire et de l'orientation du robot avec réseau et une station | 112 |
| 4.26 | Architecture de commande de deux robots dans un mode infrastructure . | 113 |
| 4.27 | Ordonnancement des différentes tâches avec deux robots | 114 |
| 4.28 | Ordonnancement des différentes tâches avec deux robots et des priorités . | 115 |
| 4.29 | Trajectoires de deux robots (en simulation) | 115 |
| 5.1 | Réseau bayésien du système | 121 |
| 5.2 | Dégradation de la QoC en fonction des pertes de paquets dans le cas d'atteinte d'une cible | 124 |
| 5.3 | Différentes trajectoires du robot, évolution des vitesses linéaire, angulaire et de l'orientation en simulation en fonction du taux de pertes de paquets fixé dans TrueTime | 125 |
| 5.4 | Dégradation de la QoC en fonction des pertes de paquets dans le cas de suivi de trajectoire | 126 |
| 5.5 | Trajectoire du robot en fonction des pertes de paquets dans le cas de suivi de trajectoire | 127 |
| 5.6 | Distance entre le robot et l'obstacle en fonction des pertes de paquets . . | 128 |
| 5.7 | Trajectoire du robot avec obstacle en fonction des pertes de paquets . . . | 128 |
| 5.8 | Etat du RB après convergence avec une connaissance sur la distance et le nombre de nœuds communicants | 132 |
| 5.9 | Etat du RB après convergence avec une connaissance sur la distance et le nombre de nœud communicants et le RSSI | 133 |
| 5.10 | QoS en fonction du nombre de noeuds | 134 |
| 5.11 | QoC en fonction de la QoS avec la mission atteinte de cible et aussi évitement d'obstacle | 135 |
| 5.12 | Probabilité de collision et retard en fonction des nombre de noeuds voulant communiquer | 136 |

Table des figures

| | | |
|------|---|-----|
| 5.13 | Evolution des probabilités de chaque noeud en fonction des observations . | 137 |
| 5.14 | Evolution des probabilités de chaque noeud en fonction des observations . | 138 |
| 5.15 | Réseau bayésien distribué | 139 |
| 5.16 | P(QoC) des deux robots dans le meilleur cas | 141 |
| 5.17 | Trajectoires des deux robots suivant l'axe des X et des Y avec positions initiales des deux robots $(x_0^1, y_0^1, \theta_0^1) = (0, 0, 0)$ et $(x_0^2, y_0^2, \theta_0^2) = (5, 5, -\frac{\pi}{2})$ sous les limites du réseau | 142 |
| 5.18 | P(QoC) des deux robots dans le cas du réseau chargé et collisions | 143 |
| 5.19 | Trajectoires des deux robots pour atteindre leurs cibles sous les limites du réseau et avec la reconfiguration de la commande et l'architecture de communication | 145 |
| 5.20 | Ordonnancement des deux tâches capteur de chaque robot et la tâche contrôleur | 146 |
| 5.21 | Trajectoires des deux robots avec un défaut capteur du <i>Robot</i> ₁ à l'instant $t = 5s$ | 146 |
| A.1 | Problème de démarrage d'une voiture | 154 |
| A.2 | Problème de démarrage d'une voiture | 156 |
| A.3 | Circulation de l'information dans un graphe causal | 157 |
| A.4 | problème de démarrage d'une voiture | 158 |
| A.5 | Algorithme de Pearl | 165 |
| A.6 | Exemple des trois vannes | 168 |
| A.7 | Réseau bésien du système trois vannes | 169 |
| A.8 | Regroupement de nœuds | 171 |
| A.9 | Moralisation du graphe | 175 |
| A.10 | Triangulation du graphe | 176 |
| B.1 | Différentes catégories de réseaux sans fil | 180 |
| B.2 | Différents réseaux sans fil selon leur vitesse et leur portée | 181 |
| B.3 | Différentes topologies | 184 |
| B.4 | Mode de fonctionnement Ad-Hoc | 185 |

Table des figures

| | | |
|------|---|-----|
| B.5 | Mode de fonctionnement infrastructure | 185 |
| B.6 | Méthode d'accès CSMA/CA | 186 |
| B.7 | Mode RTS-CTS | 188 |
| B.8 | Mode PCF | 189 |
| B.9 | Cycle PCF-DCF | 189 |
| B.10 | Cycle EPCF-EDCF | 192 |
| B.11 | Réseau ZigBee | 193 |
| B.12 | Modèle protocolaire | 193 |
| B.13 | Structure de la super trame | 196 |

Liste des tableaux

| | | |
|------|--|-----|
| 2.1 | Tableau récapitulatif des différents papiers traitants les NCSs | 38 |
| 4.1 | Signification des différents temps de transmission | 103 |
| 5.1 | Comparaison des réseaux bayésiens avec d'autres méthodes | 120 |
| 5.2 | Signification des différents nœuds du RB et leurs modes | 122 |
| 5.3 | Probabilités <i>a priori</i> de la QoS en fonction des pertes de paquets | 123 |
| 5.4 | CPT <i>a priori</i> de la QoC en fonction de l'état du robot et de la QoS du réseau sans fil dans le cas de l'atteinte d'une cible | 126 |
| 5.5 | CPT <i>a priori</i> de la QoC en fonction de l'état du robot et de la QoS du réseau sans fil dans le cas du suivi de trajectoire | 126 |
| 5.6 | CPT <i>a priori</i> de la QoC en fonction de l'état du robot et de la QoS du réseau sans fil dans le cas d'évitement d'obstacles | 129 |
| 5.7 | CPT <i>a priori</i> de la RSSI en fonction de la distance entre le robot et la station et la présence d'obstacles | 129 |
| 5.8 | CPT <i>a priori</i> des pertes de paquet en fonction du retard et du RSSI | 130 |
| 5.9 | CPT <i>a priori</i> du retard en fonction de la probabilité de collision | 131 |
| 5.10 | CPT <i>a priori</i> de la collision en fonction du nombre de nœuds voulant communiquer | 131 |
| 5.11 | Observations | 132 |
| A.1 | La table de probabilités jointes des différents modes | 160 |
| A.2 | Distribution de probabilités des 3 vannes | 169 |
| A.3 | Distribution de probabilités de la disponibilité du fluide | 169 |

Liste des tableaux

| | | |
|-----|---|-----|
| A.4 | Distribution de probabilités de RO et RF | 170 |
| A.5 | λ – <i>message</i> entre D et P | 172 |
| A.6 | $\lambda(P)$ | 172 |
| A.7 | Table de probabilités de V sachant P | 173 |
| A.8 | Table de probabilités de $\lambda_P(V)$ | 174 |
| A.9 | Différentes étapes de triangulation | 177 |

Liste des publications

1. 2010

- 7 **On-line Distributed Bayesian Decision and Diagnosis of Wireless Networked Mobile Robots** (A. Mechraoui, J.-M. Thiriet, S. Gentil), *In 18th Mediterranean Conference on Control and Automation, Marrakech, Maroc*, 2010.
- 6 **Aide à la décision et diagnostic par réseaux bayésiens d'un robot mobile commandé en réseau** (A. Mechraoui, J.-M. Thiriet, S. Gentil), *In Sixième Conférence Internationale Francophone d'Automatique Nancy, France*, 2010.

2. 2009

- 5 **Effect of Packet Loss on the Quality of Control of a Networked Mobile Robot** (A. Mechraoui, Z. H. Khan, J.-M. Thiriet), *In 30th IFAC Workshop on Real-Time Programming and 4th International Workshop on Real-Time Software, Mragowo, Poland*, 2009.
- 4 **Co-design for wireless networked control of an intelligent mobile robot** (A. Mechraoui, Z. H. Khan, J.-M. Thiriet, S. Gentil), *In International Conference on Informatics in Control, Automation and Robotics (ICINCO), Milan, Italy*, 2009.

3. 2008

- 3 **Diagnostic et pronostic de défaillances par réseaux bayésiens** (K. Medjaher, A. Mechraoui, N. Zerhouni), *In 4^{mes} Journées Francophones sur les Réseaux Bayésiens, Lyon, France*, 2008.
- 2 **Bayesian based fault diagnosis : application to an electrical motor** (A. Mechraoui, K. Medjaher, N. Zerhouni), *In 17th Triennial World Congress of the International Federation of Automatic Control, Seoul, South Korea*, 2008.
- 1 **Control of an intelligent robot system over a wireless network** (A. Mechraoui, J.-M. Thiriet, S. Gentil), *In Proceeding of the 23rd IAR Workshop on Advanced Control and Diagnosis, Coventry, United Kingdom*, 2008.

Chapitre 1

Introduction générale

Le besoin actuel de développer des véhicules autonomes et multi-tâches a favorisé l'orientation vers les systèmes commandés en réseau sans fil (Wireless Networked Control systems (WNCS)) qui sont des systèmes distribués dans lesquels la communication entre les capteurs, les actionneurs et le contrôleur est assurée par un réseau de communication sans fil partagé [Hespanha et al., 2007]. Outre la robotique industrielle, aujourd'hui la robotique s'applique aussi à d'autres domaines, nous pouvons citer principalement :

- Les autoroutes intelligentes dans lesquelles le véhicule mobile autonome peut communiquer avec les stations de contrôle d'une part (réseau infrastructure), et avec d'autres véhicules d'autre part (réseau ad-hoc), pour assurer la sécurité routière, le bon fonctionnement du trafic et aussi un bon ordonnancement de la circulation pour minimiser le temps et le trajet des véhicules ;
- L'intervention ou l'exploration dans des milieux hostiles à l'homme (incendie, zones radioactives, décharges de produits dangereux...), ou simplement difficiles d'accès (fond de mer, espace, planètes extraterrestres...);
- Les opérations militaires délicates ;
- Les missions de surveillance et/ou de protection.

Dans tous ces cas, la robotique devient de plus en plus indispensable pour limiter les risques encourus par les humains. En revanche, la communication entre des robots et des stations s'avère capitale pour assurer un bon fonctionnement, c'est-à-dire, une bonne exécution de la mission planifiée et une bonne maîtrise des perturbations (obstacles rencontrés, coordination entre robots ...). Cependant, l'intégration d'un réseau dans la boucle de commande des robots peut entraîner des problèmes qui peuvent influencer les performances du système.

Dans le contrôle des systèmes traditionnellement connus, les données capteurs sont valables instantanément pour la commande et les données de la commande sont délivrées aussi instantanément aux actionneurs. Cependant, dans les systèmes commandés en réseau, le comportement stochastique des réseaux, notamment des réseaux sans fil, rend

difficile de s'assurer que les données sont transmises correctement et entièrement aux actionneurs et aux contrôleurs. En effet, beaucoup de problèmes peuvent être introduits dans la boucle de commande durant la communication [Li & Wang, 2008] : la période d'envoi des informations, la capacité de communication du réseau, les perturbations introduites dans la communication, le retard et/ou les gigues et aussi les pertes de paquets peuvent influencer la stabilité, la robustesse et la performance du système. Les réseaux sans fil ajoutent d'autres problèmes liés à la mobilité (qui peut faire atténuer le signal reçu et éventuellement perdre totalement la connexion), au déséquencement des données (ce qui veut dire que les données ne sont pas reçues dans l'ordre de transmission), et au retard (qui peut être variable suivant plusieurs paramètres tels que la distance entre les différents composants communicants).

Dans le cadre de cette thèse, nous nous intéressons aux problèmes de co-conception (voir Fig.1.1) dans les systèmes commandés en réseau sans fil. Des travaux proches avaient été initiés en laboratoire par [Berbra, 2009] sur une application de réseaux filaires dans le cadre d'un projet national.

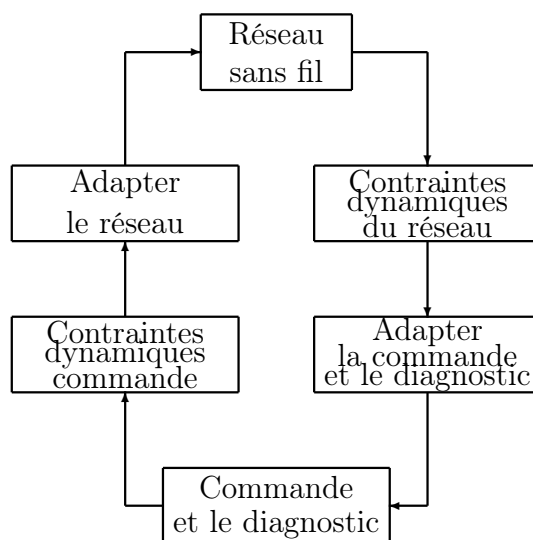


Figure 1.1 – Problème de co-conception

Nous proposons dans nos travaux d'utiliser des robots unicycles mobiles pour tester nos approches et comparer les résultats de simulation avec les résultats de l'expérimentation. En conséquence, deux domaines complètement indépendants seront traités. Le premier concerne la commande du robot. Le deuxième concerne la communication entre les robots et la station de contrôle. L'objectif principal de cette étude est de trouver un compromis entre la commande et le diagnostic distribués d'un côté et le réseau de

l'autre côté, c'est-à-dire qu'il faut adapter le réseau aux contraintes dynamiques de la commande et du diagnostic et aussi adapter l'algorithme de commande et de diagnostic aux contraintes dynamiques du réseau (voir Fig.1.1) ce qui nous amène à résoudre ce qu'on appelle un problème de Co-conception. Pour cela, on va définir une architecture de communication pour des véhicules mobiles se déplaçant dans un espace 2D et plusieurs stations au sol (réseau infrastructure). Ces mobiles ont pour objectif de se déplacer pour atteindre une cible, suivre une trajectoire ou bien éviter des obstacles. Les robots peuvent communiquer entre eux via une station de contrôle pour échanger les informations critiques liées à leur mission et aussi les informations nécessaires pour faire du diagnostic et réagir en cas de défauts. Outre les défauts matériels liés à chaque robot, on peut aussi avoir des problèmes de communication qui peuvent influencer le bon déroulement de la mission. En effet, en fonction de la position du mobile et du signal reçu de chaque station, le véhicule pourra être en regard d'une ou plusieurs stations, ou d'aucune (Fig.1.2). Lorsqu'il ne sera plus en face d'une station, il sera complètement autonome (il fonctionnera avec un contrôleur embarqué). Dans ce cas on peut imaginer aussi qu'il se sert d'un autre robot comme routeur pour communiquer avec la station. Lorsqu'il sera en face d'une ou plusieurs stations, le mobile doit pouvoir communiquer avec l'une de ces stations en prenant en compte deux critères principaux, à savoir : la Qualité de Service (QoS) et la Qualité de Contrôle (QoC). La QoS désigne *la capacité à fournir un support de communication conforme aux exigences en matière de temps et de bande passante et aussi de garantir un niveau acceptable de pertes de paquets défini pour un usage particulier*. La QoC désigne *la capacité de garantir la stabilité, la précision, le temps de réponse, l'allure des transitoires et la robustesse d'un système*. La QoC dépend directement de la QoS du réseau. La quantification de ces deux critères est liée fortement à l'usage du système, au nombre d'informations échangées et aussi au réseau sans fil utilisé.

L'aspect stochastique et non déterministe des réseaux sans fil rend difficile de trouver un modèle analytique qui traduit fidèlement leur comportement. Donc, il est difficile de déterminer avec certitude le niveau de la QoS et donc de faire le choix de la meilleure station qui assure une QoS pouvant garantir une QoC suffisante au bon fonctionnement du véhicule mobile. La théorie des probabilités peut être une bonne solution pour déterminer la QoS de la station courante, ce qui peut après aider à décider de déclencher ou non la procédure de changement de la station courante pour une autre station qui assure plus de garanties de transmission de mesures et de commandes via le réseau.

La causalité entre les différents paramètres du réseau d'une part et la QoS et la QoC d'autre part nous a poussé à utiliser la théorie des graphes pour modéliser le comportement du réseau. L'outil qui permet le mariage entre la théorie des graphes et la théorie des probabilités s'appelle les réseaux bayésiens et entre dans le cadre de l'intelligence artificielle, très utilisée ces dernières années vu la complexité des modèles mathématiques des systèmes à étudier.

L'utilisation de cet outil est justifiée par le besoin d'aide à la décision pour choisir la

Chapitre 1. Introduction générale

bonne station, de diagnostic pour identifier la cause de la dégradation de la QoS et de la QoC et aussi de pronostic pour anticiper la mauvaise qualité de contrôle et de service, afin d'éviter la déstabilisation du système à commander avant qu'elle ne se produise.

Ce document est organisé en chapitres structurés de la manière suivante :

- Le deuxième chapitre est dédié à l'état de l'art des systèmes commandés en réseau ainsi que du diagnostic et du diagnostic des systèmes commandés en réseau
- Dans le troisième chapitre, nous présentons une étude d'un robot unicycle ainsi que l'architecture de commande utilisée. Nous présentons aussi l'intégration du réseau dans la boucle de commande. L'outil de simulation TrueTime (ToolBox sous Matlab) est utilisé. C'est un outil de simulation temps réel qui permet de se rapprocher le plus possible de la réalité et d'éviter de mauvaises surprises lors de l'implémentation sur le robot réel.
- Ensuite, dans le chapitre 4, nous traitons l'influence de la QoS sur la QoC et des stratégies d'évitement de la dégradation de QoC. Nous commençons ce chapitre par une évaluation de performance du protocole de communication utilisé et nous étudions ensuite les effets du retard, de la charge du réseau et des pertes de paquets sur la QoC du robot. Nous proposons aussi une stratégie pour éviter la dégradation de la QoC dans un environnement multi-stations.
- Le chapitre 5 propose un nouveau modèle basé sur les réseaux bayésiens distribués pour l'aide à la décision et le diagnostic distribué, en ligne, d'un robot mobile commandé en réseau. Nous proposons à partir de résultats fournis par le réseau bayésien une stratégie de co-conception pour garantir une bonne QoS et aussi une bonne QoC.
- Enfin, une conclusion présente un bilan et des perspectives.

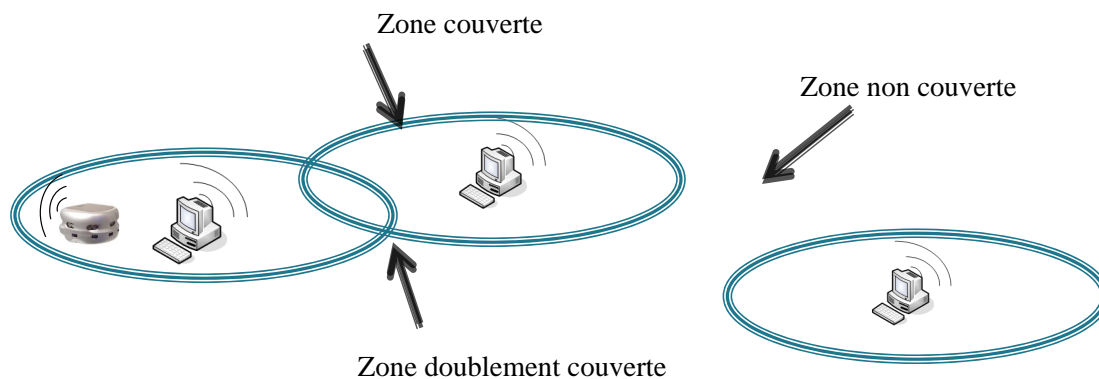


Figure 1.2 – Scénario général

Chapitre 2

Diagnostic et commande de systèmes en réseau

Sommaire

| | | |
|------------|--|-----------|
| 2.1 | Introduction | 31 |
| 2.2 | Systèmes commandés en réseau (NCS) | 32 |
| 2.3 | Analyse de stabilité pour les NCSs | 34 |
| 2.4 | Estimation pour les NCSs | 35 |
| 2.5 | Synthèse de commande des NCSs | 36 |
| 2.6 | Co-conception (Commande/Communication) | 36 |
| 2.7 | Diagnostic des systèmes commandés à travers un réseau | 38 |
| 2.7.1 | Diagnostic des systèmes | 38 |
| 2.7.2 | Classification des méthodes de diagnostic | 39 |
| 2.7.3 | Architectures du diagnostic | 44 |
| 2.8 | Conclusion | 47 |

2.1 Introduction

Dans une architecture de systèmes où l'on a besoin de faire communiquer plusieurs entités mobiles (dans notre cas un groupe de robots), l'intégration d'un réseau sans fil dans la boucle de commande entraîne des problèmes qui influencent les performances du système. Le diagnostic des systèmes commandés en réseau (NCSs) est influencé aussi par la présence du réseau entre les différents diagnostiqueurs. Dans ce chapitre, un état de l'art des systèmes commandés en réseau et aussi des différentes méthodes du diagnostic est présenté. Après un bref survol des problèmes posés par l'introduction d'un réseau dans un système de commande, nous commençons par aborder le problème de la stabilité des

NCSs, la section suivante est consacrée à l'estimation pour les NCSs puis à la synthèse de commande des NCSs. Ensuite, la co-conception en terme de commande / communication est abordée tout en définissant le principe de co-conception. Pour finir, le diagnostic des systèmes et sa classification est présenté. Ensuite, les différentes architectures de diagnostic centralisé, décentralisé ou distribué sont étudiées dans la dernière partie.

2.2 Systèmes commandés en réseau (NCS)

Les systèmes commandés en réseau (Networked Control Systems NCSs) entrent dans le cadre des systèmes distribués dans lesquels les informations entre le nœud contrôleur et le nœud système sont échangées via un réseau de communication filaire ou sans fil comme le montre la Fig.2.1. Par rapport aux systèmes commandés traditionnellement connus, la QoC dans les NCS dépend essentiellement de la QoS du réseau utilisé.

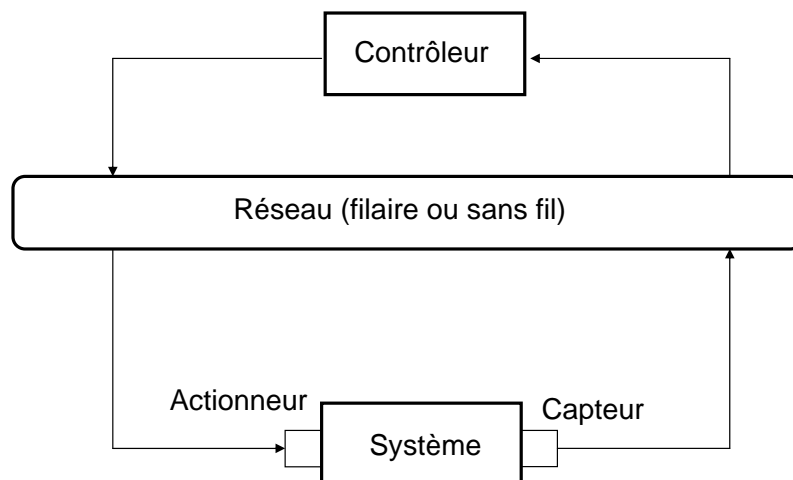


Figure 2.1 – Système commandé en réseau

L'utilisation d'un réseau pour interconnecter les différents nœuds d'un système distribué apporte des avantages qui se révèlent importants vu la complexité des systèmes actuellement connus. La réduction de la complexité en terme d'installation est l'un des avantages des NCSs. Le poids et l'espace réduit des NCSs peut être l'un des critères pour choisir ce type de systèmes. La simplicité du diagnostic et de la maintenance se révèle aussi un grand avantage pour ces systèmes. Ces avantages peuvent aussi accroître la souplesse du système. En conséquence, les NCSs ont trouvé leurs applications dans différents domaines tels que les réseaux de capteurs mobiles [Leonard et al., 2007], la robotique et la robotique mobile [Val et al., 2003; Cremean et al., 2006; Arzen et al., 2007; Mechraoui et al., 2008; Zacharie, 2009], les UAV (Unmanned Aerial Vehicles UAV) [Berbra

et al., 2009]. Cependant, l'intégration du réseau dans la boucle de commande introduit quelques problèmes difficiles à résoudre.

La commande des systèmes traditionnellement connus est basée sur l'hypothèse que toutes les informations transmises sont reçues sans erreur, sans pertes et sans retard, ce qui signifie que le lien entre les différents composants du système est dit "parfait". Par contre, dans les systèmes commandés en réseau, le réseau que ce soit filaire ou sans fil est "imparfait". C'est le réseau donc qui fait la différence entre un système commandé classique et un système commandé à travers un réseau et rend la modélisation et l'analyse des NCSs plus complexe [Hespanha et al., 2007]. L'influence du réseau sur la QoC du système est due à :

1. **Echantillonnage et Retard** : la transmission d'un signal continu via un réseau nécessite l'échantillonnage du système, le codage en format binaire, l'encapsulation dans une trame de communication ensuite l'envoi et enfin la décapsulation et le décodage. Toute cette procédure prend du temps et donc introduit un retard qui s'avère variable à cause du temps d'accès au médium et aussi du temps de transmission qui dépend essentiellement des conditions dans lesquelles le réseau fonctionne à savoir : la charge du réseau et la qualité du canal de transmission. En général, les réseaux introduisent des retards [Hespanha et al., 2007; Richard & Divoux, 2007] dus au temps de transmission et de traitement de l'information entre deux ou plusieurs nœuds. Il est connu dans la théorie de la commande que le retard peut influencer la performance du système commandé et entraîner son instabilité [Zhang et al., 2001].
2. **Pertes de paquets** : Un autre problème qui peut influencer un NCS est la perte de paquets. En effet, envoyer les informations via un réseau peut entraîner leurs pertes. Ces pertes sont le résultat des erreurs au niveau de la couche physique du réseau (les pertes sont plus probables dans les réseaux sans fil que dans les réseaux filaires) ou bien de la charge des buffers (à cause de la charge du réseau). Un trop long retard peut être considéré comme une perte de paquets surtout pour les applications temps réel.
3. **Débit** : chaque réseau de communication est limité par une bande passante (un nombre limité d'envoi d'informations par seconde). Dans certaines applications, cette limite de la bande passante pose quelques problèmes au niveau de la performance des NCSs.

Les systèmes commandés à travers un réseau nécessitent une connaissance pluridisciplinaire, à savoir trois domaines de recherche complémentaires. Le premier concerne la commande des systèmes, le deuxième, les réseaux de communications et enfin l'informatique temps réel [Baillieul & Antsaklis, 2007]. En effet, la principale difficulté est de fusionner les résultats des trois domaines à cause du manque de communication entre les différentes communautés et donc des différences fixées entre les objectifs menés par

chacune des communautés. Prenons par exemple le problème du retard. Dans le domaine de la communication, les retards dans la transmission d'informations ne sont pas au centre des préoccupations, car il est plus important de transmettre le message avec exactitude, même si cela peut impliquer parfois des retards importants au cours de la transmission. De même, en informatique, le délai de transmission n'a pas été un problème central puisque les systèmes informatiques sont en interaction avec un autre ordinateur ou un opérateur humain et non pas directement avec un système physique. En revanche, la présence de retards ou de perte de paquets dans la boucle de commande influencent considérablement la stabilité et la performance du système. Par conséquent, l'analyse, la modélisation et le contrôle des NCSs sont beaucoup plus complexes que des systèmes traditionnellement connus. Beaucoup de travaux ont été faits récemment concernant les NCSs. Selon les différents aspects de l'analyse et la conception des NCSs, les travaux existants peuvent être généralement divisés en quatre catégories : l'analyse de stabilité, l'estimation d'état, la synthèse de commande et de la co-conception du contrôle et de la communication.

2.3 Analyse de stabilité pour les NCSs

La stabilité est d'une importance primordiale dans les systèmes en boucle fermée. De nombreux résultats ont été publiés dans la littérature pour analyser la stabilité en boucle fermée en présence d'une ou plusieurs caractéristiques influençant les systèmes commandés à travers un réseau. Deux grandes classes d'approches ont été utilisées, à savoir les approches déterministes ou stochastiques. Les approches déterministes proposées prennent l'hypothèse que le retard est variable mais borné et utilisent la théorie de Lyapunov pour trouver le retard maximal toléré [Hassibi et al., 1999; Walsh et al., 2001] et aussi le pourcentage maximal de pertes de paquets admis [Zhang et al., 2001; Liu et al., 2008]. Ces limitations sont des seuils pour garantir la stabilité du système. Les approches stochastiques prennent en compte explicitement des caractéristiques NCS dans le système, et ensuite prouvent la stabilité en utilisant la stabilité quadratique moyenne [Ma & Fang, 2005; Hespanha et al., 2007] ou bien la stabilité stochastique [Montestruque & Antsaklis, 2003; Xiaomei et al., 2008].

Dans [Walsh et al., 2001], les auteurs ont analysé la stabilité d'un système non-linéaire commandé à travers un réseau et ont trouvé la limite de stabilité en fonction du retard introduit par la présence du réseau. Dans [Zhang et al., 2001], la stabilité des NCSs avec des retards est analysée en définissant les régions de stabilité et en modélisant les NCSs avec des pertes de paquets comme étant des systèmes dynamiques asynchrones [Hassibi et al., 1999]. Cette méthode permet de déterminer si le NCS est stable à un certain taux de perte de paquets, et donc de rechercher le plus fort taux de perte pour préserver la stabilité. Dans [Liu et al., 2008], la stabilité a été aussi analysée avec une période d'échantillonnage variable et aussi un retard, dû au réseau, variable ; l'inégalité

matricielle linéaire (LMI) a été développée pour résoudre ce problème. [Ma & Fang, 2005] ont proposé un nouveau mode de contrôle où le retard induit par le réseau est supérieur à la période d'échantillonnage. Le retard est traité comme étant une variable stochastique. La tâche capteur est guidée par le temps, la tâche contrôleur est guidée par les événements et la tâche actionneur, quant à elle, est guidée par un temps de division en utilisant un algorithme de division de temps. Dans [Hespanha et al., 2007], une synthèse et comparaison des différentes recherches récentes sur les NCSs a été faite et a été analysée. Dans [Montestruque & Antsaklis, 2003], la stabilité stochastique dans les NCS sous la contrainte d'un retard variable a été abordée. Le modèle des NCSs avec retard variable proposé dans [Montestruque & Antsaklis, 2002] a été utilisé, ce modèle utilise la connaissance sur le système pour réduire le nombre de paquets échangés et donc la charge du réseau. Dans ce papier, ils se sont concentrés sur la stabilité stochastique sous certaines connaissances des propriétés stochastiques. En revanche, dans [Xiaomei et al., 2008], la stabilité stochastique a été analysée non seulement en présence du retard mais aussi des pertes de paquets. Dans [Zhang et al., 2005], le retard capteur-contrôleur et le retard contrôleur-actionneur sont modélisés comme étant des chaînes de Markov et l'approche LMI a été utilisée pour calculer le gain du retour d'état.

2.4 Estimation pour les NCSs

Comme cela a été discuté précédemment, dans un NCS, la transmission des données souffre donc des retards variables et des pertes de paquets. Certains auteurs ont estimé ces pertes [Seiler, 2003; Hadjicostis & Touri, 2002; Ling & Lemmon, 2003; Giorgiev & Tilbury, 2006; Elia & Eisebeis, 2004; Sinopoli et al., 2004; Imer et al., 2006]. Dans toutes ces recherches, les auteurs considèrent que le retard soit est zéro soit est infini. Dans [Seiler, 2003], l'auteur a considéré une estimation avec des pertes de paquets. Ils modélisent le système et les paquets perdus comme étant un processus de Markov à saut. Dans [Hadjicostis & Touri, 2002; Ling & Lemmon, 2003; Giorgiev & Tilbury, 2006], les auteurs proposent des techniques de compensation pour les paquets perdus. [Elia & Eisebeis, 2004] ont proposé une approche de perturbation stochastique pour les systèmes MIMO-LTI (Linear Time-Invariant) à temps discret et ont montré que la conception d'un contrôleur optimal est équivalent à résoudre un problème d'optimisation convexe d'un LMI. [Sinopoli et al., 2004] ont porté spécifiquement sur l'estimation à variance minimale en cas de pertes de paquets et ont montré que l'estimateur optimal est nécessairement variable dans le temps. Ces résultats ont été étendus à des contrôleurs LQG dans [Imer et al., 2006] qui permet de calculer le gain d'une commande par retour d'état.

2.5 Synthèse de commande des NCSs

Jusqu'ici, la synthèse de contrôleur a employé la méthode linéaire quadratique gaussienne (LQG) [Gupta et al., 2005; Varsakelis & Zhang, 2008], la commande basée sur un observateur pour compenser le retard [Naghshtabrizi & Hespanha, 2005; Zhang et al., 2001], la commande prédictive [Liu et al., 2007; Guo & Li, 2010], la commande robuste H_2 et H_∞ [Fu & Dai, 2009; Gao & Chen, 2008]. [Varsakelis & Zhang, 2008] montre l'utilisation d'un contrôleur LQG pour un système contrôlé en réseau partagé. Au lieu d'insister sur le contrôle conjointement optimal et les politiques d'accès au support, les auteurs ont réduit le nombre d'accès possibles pour garantir la stabilité et ne pas dépasser un retard maximal en rendant l'accès au médium périodique. Dans [Gupta et al., 2005], les auteurs ont utilisé le filtre de Kalman dans la partie capteur et un filtre de commutation linéaire dans la partie contrôleur du système. Ils ont considéré le problème comme étant un problème de contrôleur optimal LQG et ils ont simulé les pertes de paquets avec un commutateur qui génère des séquences aléatoires de pertes de paquets.

[Naghshtabrizi & Hespanha, 2005] ont proposé un contrôleur qui stabilise le système à la fois en présence de retards, et de pertes de paquets (les mesures et les commandes). Le contrôleur a été conçu en se basant sur le théorème de Lyapunov-Krasovskii. Dans [Liu et al., 2007; Guo & Li, 2010], les auteurs ont utilisé la commande prédictive pour compenser les retards et les pertes de paquets mais en général ces commandes sont assez lourdes surtout pour des systèmes embarqués. [Fu & Dai, 2009] ont proposé une architecture de commande basée sur la commande mixte H_2/H_∞ tout en posant comme hypothèse que le nombre de pertes de paquets est borné et aussi que les pertes de paquets suivent un processus Markovien.

2.6 Co-conception (Commande/Communication)

Le domaine de co-conception commande/communication est un nouvel axe de recherche et parmi les premiers travaux dans ce domaine on peut citer le travail mené dans [Branicky et al., 2002; Branicky et al., 2003] et récemment le livre [Aubrun et al., 2010]. Dans les approches co-conception, les problèmes issus du réseau tels que les retards, pertes de paquets et giges sont considérés simultanément avec les problèmes de commande tels que la stabilité et la performance du contrôleur. En règle générale, l'ordonnancement dans les NCSs consiste à attribuer un calendrier de transmission à chaque entité de transmission (capteur, contrôleur) basé sur un algorithme d'ordonnancement (un ensemble de règles qui, à tout moment, détermine l'ordre dans lequel les messages sont transmis). D'habitude, la conception du système de contrôle et l'ordonnancement du réseau sont étudiés séparément. Cette séparation a permis à la communauté du contrôle

de se concentrer sur son domaine sans se soucier particulièrement de l'ordonnancement. Mais avec la croissance de l'utilisation des NCS, le besoin de traiter le problème d'ordonnancement et de commande simultanément a favorisé l'orientation vers ce qu'on appelle un problème de co-conception (adapter le réseau aux besoins dynamiques de la commande et aussi adapter la commande aux contraintes dynamiques du réseau) Fig.2.2. Donc, comprendre l'impact de l'ordonnancement sur la stabilité et la performance de la commande est devenu crucial. En effet, ces dernières années ont permis à la communauté automatique d'étudier les réseaux de communication d'une façon plus élaborée et donc plusieurs papiers ont été publiés dans le domaine de la co-conception [Branicky et al., 2002; Cervin & Eker, 2005; Branicky et al., 2003; Aubrun et al., 2010; Huo et al., 2005; Zhang & Varsakelis, 2006]. Dans [Cervin & Eker, 2005], les auteurs ont proposé un modèle de serveur de commande. Ce serveur crée l'abstraction d'une tâche de commande avec une durée déterminée et un temps de latence d'entrées-sorties fixe plus court que la période ce qui exige un ordonnancement très strict et ainsi élimine les gigues et garantit un retard fixe. Dans [Huo et al., 2005], les auteurs ont proposé un ordonnancement et une commande tolérante aux fautes pour les NCSs dans le cadre d'une problématique de co-conception. Ce contrôleur est basé sur H_∞ . Les paramètres du contrôleur ont été définis à partir de la méthode LMI. Côté ordonnancement, les priorités sont attribuées en fonction d'un critère quadratique (IAE : Integral of the Absolute Error). [Zhang & Varsakelis, 2006] propose une stratégie commande/communication basée sur la stabilité exponentielle d'un NCS en identifiant les séquences de communication qui garantissent l'atteignabilité et l'observabilité et ensuite construire un observateur basé sur ces séquences. Dans [de Wit et al., 2007], les auteurs ont étudié l'impact de la stratégie de modulation RZ (retour à zéro) d'amplitude en relation avec la stabilité du système. Il a été montré que cette modulation introduit une petite latence arbitraire, mais qu'elle peut être compensé du côté du décodeur, en accueillant les horizons de prédiction de ce retard qui est connu et fixé par l'utilisateur. De cette façon, les conditions de stabilité, même dans le cas d'une transmission idéale, sont préservés.

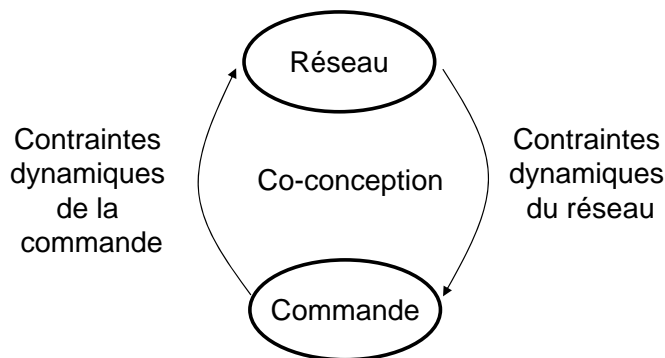


Figure 2.2 – Problème de co-conception

Le tableau suivant résume et classe les différents articles qui traitent les systèmes commandés en réseau et les différentes façon d'aborder le problème lié à l'intégration du réseau dans la boucle de commande.

Tableau 2.1 – Tableau récapitulatif des différents papiers traitants les NCSs

| | Stabilité | Estimation | Performance du contrôleur | Co-conception |
|-------------------|--|--|--|--|
| Retard | [Zhang et al., 2001] [Baillieul & Antsaklis, 2007] [Hassibi et al., 1999] [Walsh et al., 2001] [Carnevale et al., 2007] [Liu et al., 2008] [Ma & Fang, 2005] [Richard & Divoux, 2007] [Xiaomei et al., 2008] [Zhang et al., 2005] | [Chunmao & Jian, 2006] [Li et al., 2006] [Richard & Divoux, 2007] | [Guo & Li, 2010] [Varsakelis & Zhang, 2008] [Naghshtabrizi & Hespanha, 2005] | [Huo et al., 2005] [Cervin & Eker, 2005] [Zhang & Varsakelis, 2006] [Juanole et al., 2010] [Mechraoui et al., 2008] [Berbra et al., 2008] [Jia et al., 2007] [Khan et al., 2009] [Branicky et al., 2002] [Branicky et al., 2003] [?] [Liu et al., 2007] |
| Pertes de paquets | [Zhang et al., 2001] [Xiaomei et al., 2008] [Wu & Chen, 2007] | [Elia & Eisembeis, 2004] [Seiler, 2003] [Hadjicostis & Touri, 2002] [Ling & Lemmon, 2003] [Giorgiev & Tilbury, 2006] | [Fu & Dai, 2009] [Gao & Chen, 2008] [Gupta et al., 2005] | |
| Ordonnancement | | | | |

2.7 Diagnostic des systèmes commandés à travers un réseau

Cette thématique de recherche est plus récente que les précédentes, mais fait l'objet de beaucoup d'attention récemment, aussi nous allons la développer de façon approfondie.

2.7.1 Diagnostic des systèmes

Le diagnostic des systèmes complexes a suscité et continue à susciter un grand intérêt dans l'industrie : savoir localiser et identifier une défaillance suffisamment tôt peut permettre de prendre une décision rapide susceptible de ramener le processus dans un mode de fonctionnement normal. Le diagnostic est donc de nos jours un élément clef d'accroissement de la productivité en augmentant la disponibilité des systèmes complexes.

Le diagnostic de défaillance des systèmes industriels s'apparente au diagnostic médical. En effet, c'est à partir de symptômes externes observés sur la machine ou sur le patient que l'expert ou le médecin réalise son diagnostic. Donc, effectuer le diagnostic d'un système c'est identifier le mode de fonctionnement dans lequel il se trouve. La tâche de *diagnostic de défaut* consiste à déterminer le composant défaillant [Isermann, 1997], elle désigne aussi toute méthode permettant de déterminer si une machine est défaillante ou non et de discriminer l'origine du défaut à partir des informations relevées par observation, contrôles et/ou tests.

Une **défaillance** est une altération ou une cessation de l'aptitude d'un ensemble à accomplir sa ou ses fonctions requises avec les performances définies dans les spécifications techniques (autrement dit, une **défaillance** est un état qui peut mener à un dysfonctionnement du système [Isermann, 1997; Isermann, 2005]). Un défaut peut être caractérisé par une déviation, en dehors d'un intervalle acceptable, d'une variable observée ou d'un paramètre associé au procédé.

Un système est dit dans un état de **fonctionnement normal** lorsque les variables le caractérisant (variables d'état, variables de sortie, variables d'entrée, paramètres du système) demeurent cohérentes entre elles et avec un modèle de fonctionnement normal plus au moins précis.

2.7.2 Classification des méthodes de diagnostic

Les premières méthodes de sûreté furent basées sur la redondance de matériels jugés critiques pour le fonctionnement du système. La redondance matérielle est très répandue dans les domaines où la sûreté de fonctionnement est cruciale pour la sécurité des personnes et de l'environnement, comme dans l'aéronautique ou le nucléaire. Les principaux inconvénients de la redondance matérielle sont liés aux coûts dus à la multiplication des éléments ainsi que l'encombrement et le poids supplémentaires qu'elle génère. Les progrès spectaculaires réalisés dans le domaine des calculateurs numériques combinés à une baisse des coûts permettent aujourd'hui la mise en œuvre, dans le milieu industriel, des méthodes modernes de l'automatique et de l'intelligence artificielle. Cette nouvelle approche permet d'éliminer en partie, voire même en totalité, la redondance matérielle pour le diagnostic des systèmes industriels. On peut globalement distinguer deux grandes familles dans les méthodes de diagnostic [Isermann, 1997][Isermann, 2005] :

- les méthodes à bases de modèles analytiques (espace de parité, observateur, estimation de paramètres etc) " approches quantitatives " ;

- les méthodes sans modèles analytiques (basées sur l'intelligence artificielle) " approches qualitatives ".

Nous présentons à la Fig.2.3 les différentes méthodes de diagnostic.

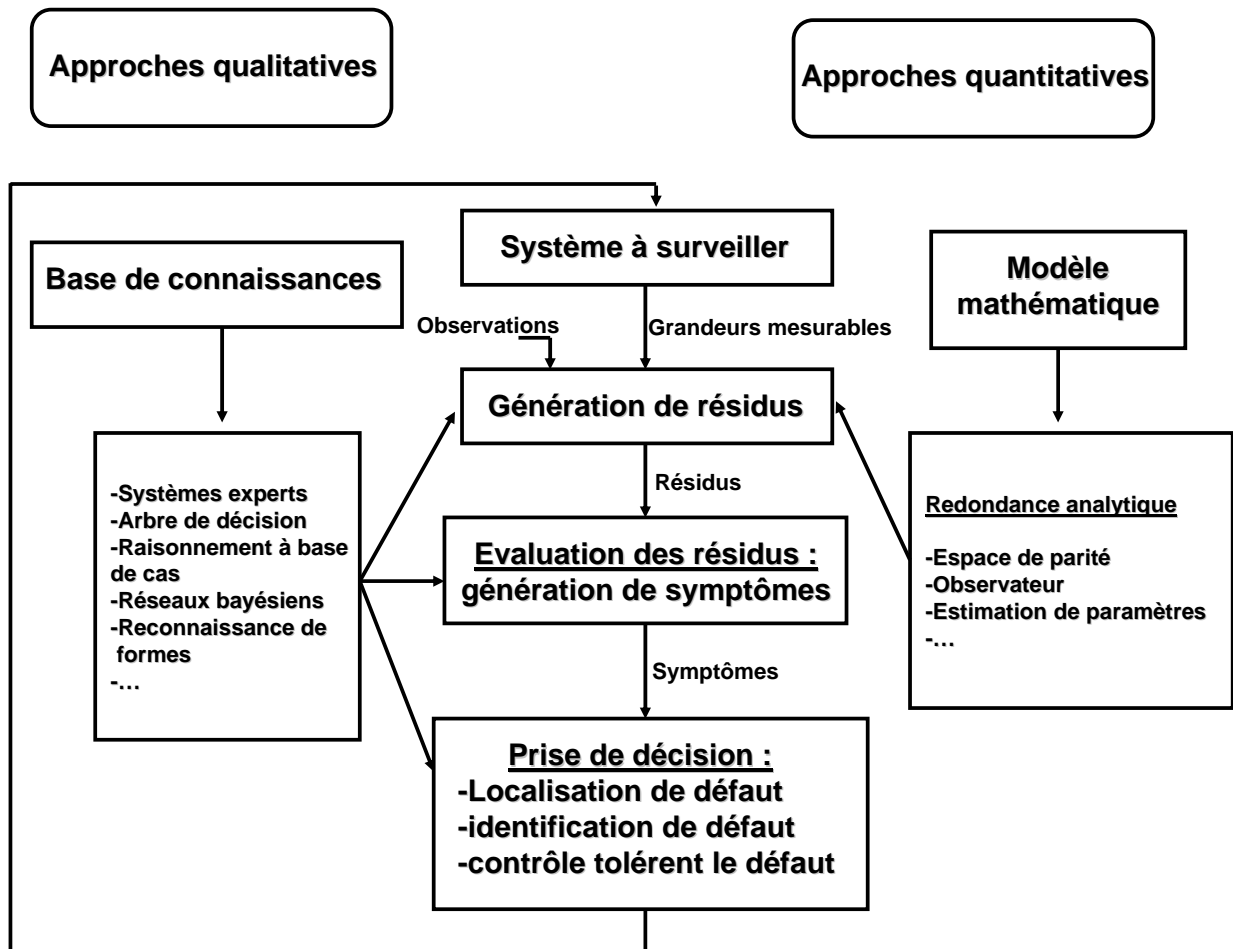


Figure 2.3 – Classification des méthodes de diagnostic

Méthodes quantitatives :

Ce sont des méthodes qui peuvent provenir de la connaissance quantitative du comportement entrée/sortie d'un processus. Il s'agit de générer un modèle analytique redondant. Ce modèle servant directement de référence pour la détection de défauts, la qualité du résultat dépend directement de la qualité des modèles. La mise en œuvre de ces méthodes nécessite donc une modélisation précise. La détection de défaut basée sur l'utilisation de modèles mathématiques peut être divisée en deux étapes principales : la

génération de résidus et la prise de décision. Lors de la première étape, les signaux d'entrée et de sortie du système sont utilisés pour générer un résidu, c'est-à-dire un signal mettant en évidence la présence d'un défaut. Théoriquement, en régime de fonctionnement normal, ce signal est nul et s'écarte notablement de zéro en présence de défaut. La génération de résidus est propre à la méthode utilisée. Durant la seconde étape, les résidus sont analysés pour décider s'il y a ou non présence de défaut (détection et génération de symptômes), sur quel composant du système il est intervenu (localisation) et pour déterminer la nature du défaut et sa cause (identification). La détection peut s'effectuer à l'aide d'un simple test de dépassement de seuil.

La Fig.2.4 présente l'architecture générale de la détection de défauts basée sur l'utilisation de modèles.

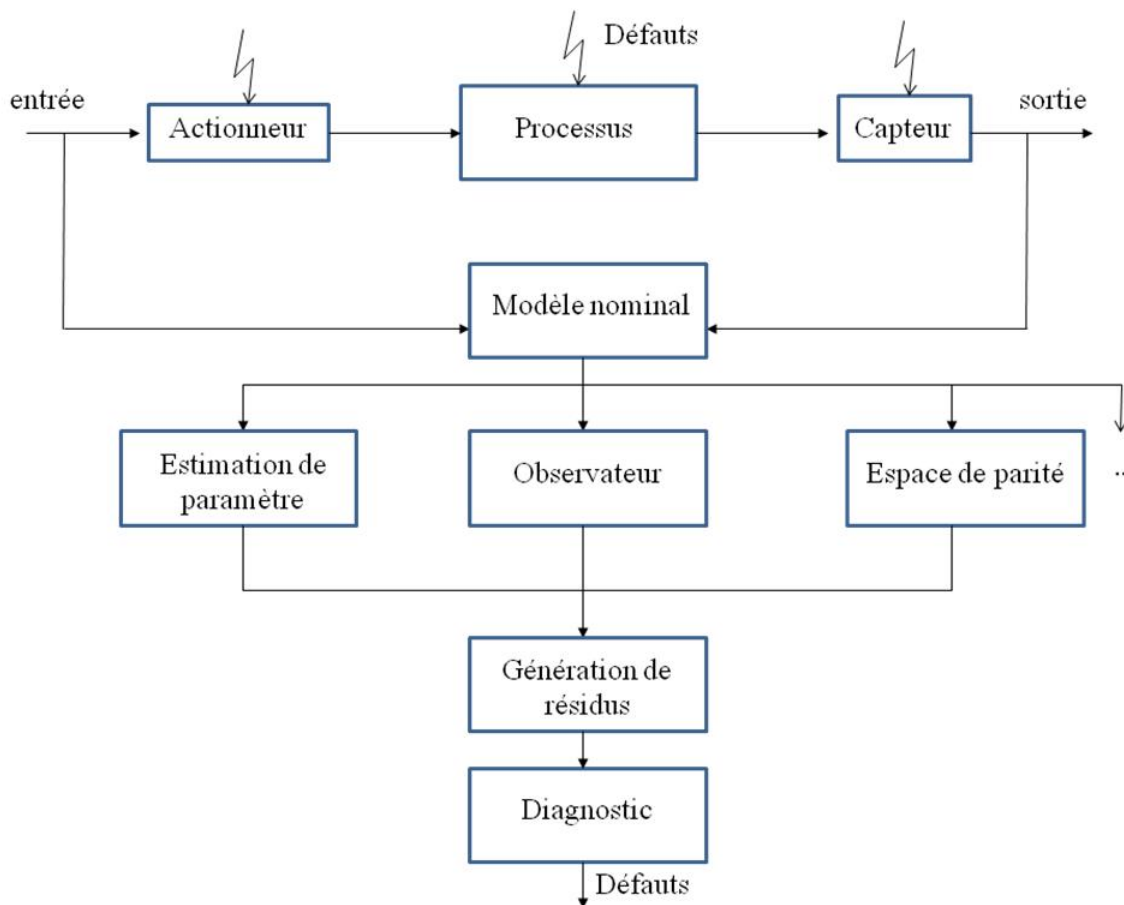


Figure 2.4 – Architecture générale de la détection de défauts à base de modèles

Parmi les différentes méthodes de détection et de diagnostic utilisant des modèles mathématiques, nous trouvons principalement l'espace de parité, les observateurs et l'estimation des paramètres (plus de détails dans [Isermann, 1997] et [Isermann, 2005]).

1. **Espace de parité** : une relation de redondance analytique est une équation dans laquelle toutes les variables sont connues. La génération de telles relations permet

d'engendrer des résidus. Pour la détection de défauts basée sur l'utilisation de modèles, un résidu est une fonction des entrées et des sorties du processus. En l'absence de défauts, ce résidu est statistiquement nul. Lors de l'apparition d'un défaut, son amplitude évolue de manière significative. L'approche la plus classique est celle dite de l'espace de parité. Les relations de parité utilisent la redondance directe au moyen de relations algébriques.

2. **Observateur** : la génération de résidus à l'aide d'une estimation d'état consiste à reconstruire l'état ou, plus généralement, la sortie du processus à l'aide d'observateurs et à utiliser l'erreur d'estimation comme résidu.
3. **Estimation de paramètres** : l'approche d'estimation paramétrique considère que l'influence de défauts se reflète sur les paramètres du modèle et non pas uniquement, comme c'est le cas des observateurs, sur les variables du système physique. Le principe de cette méthode consiste à estimer en continu des paramètres du procédé en utilisant les mesures d'entrée/sortie et en évaluant la distance qui les sépare des valeurs de référence de l'état normal du procédé. L'estimation paramétrique possède l'avantage d'apporter de l'information sur l'importance des déviations.

La majorité de ces méthodes sont déterministes, toutefois il est difficile, voire impossible d'obtenir un modèle global d'une installation complexe en raison de la complexité des phénomènes physiques mis en jeu. De plus, ces approches sont incapables de prendre en compte par elles-mêmes la révision des modèles (la mise à jour).

Méthodes qualitatives :

Parmi les différentes méthodes de détection et de diagnostic utilisant l'intelligence artificielle, nous trouvons principalement les suivantes :

1. **Systèmes experts (SE)** : un système expert est un programme capable d'explorer une expertise. Cette expertise rassemble non seulement des connaissances directement liées à ce domaine mais également des capacités de raisonnement acquises par les spécialistes au cours de leur activité. Dans ces systèmes, les connaissances expertes sont formalisées par des règles de la forme :

Si conditions alors conclusions

où conditions est une expression qui précise les conditions de déclenchement de la règle. Déclencher une règle, c'est admettre ce qui est précisé dans les conditions. On parle également d'inférence. Cela revient à modifier l'état des connaissances actuelles sur le problème en cours de résolution (appelées mémoire de travail) permettant d'avancer dans cette résolution.

2. **Réseaux de neurones (RN)** : Les RNs sont des modèles mathématiques constitués d'un nombre de processeurs élémentaires (ou nœuds) interconnectés entre eux qui traitent l'information qui leur arrive à partir des signaux extérieurs. De manière

générale, l'utilisation des RNs se fait en deux phases. Tout d'abord, la synthèse du réseau est réalisée et comprend plusieurs étapes : le choix du type de réseau, du type de neurones, du nombre de couches, des méthodes d'apprentissage. L'apprentissage permet alors, sur la base de l'optimisation d'un critère, de reproduire le comportement du système à modéliser. Il consiste dans la recherche des poids. Quand les résultats d'apprentissage obtenus par le RN sont satisfaisants, il peut être utilisé pour la généralisation. Il s'agit ici de la deuxième phase où de nouveaux exemples - qui n'ont pas été utilisés pendant l'apprentissage - sont présentés au RN pour juger le diagnostic -dans notre cas- du système ainsi modélisé.

3. Raisonnement à Partir de Cas (RàPC) :

Dans un système de raisonnement à partir de cas (RàPC), les connaissances sont emmagasinées sous la forme de cas. L'ensemble des cas connus est indexé en fonction de critères de similarité. Ainsi face à un problème, la recherche d'un ou de plusieurs cas similaires se fait de manière efficace. Le calcul de la similarité se fait à partir des attributs pertinents sur les situations rencontrées auparavant. Le raisonnement à partir de cas se fait en plusieurs étapes. On distingue cinq grandes étapes :

- L'évaluation du problème à résoudre.
- La recherche dans la mémoire de cas des candidats potentiels puis la sélection des meilleurs d'entre eux.
- L'adaptation éventuelle des candidats retenus et la proposition d'une solution.
- L'évaluation de la solution proposée.
- L'apprentissage qui entraîne la mise à jour de la base de cas.

Quand un nouveau problème doit être traité, il est d'abord formaté, c'est-à-dire représenté par un ensemble de descripteurs, véritables attributs pertinents. Cette représentation va permettre de comparer ce nouveau problème avec l'ensemble des cas connus et classés dans la base de cas. Parmi tous les cas connus, seuls ceux ayant un lien avec le nouveau problème sont retenus. Puis ils sont classés en utilisant une fonction de similarité. Ensuite, à partir du ou des meilleurs cas obtenus, la solution au problème est déterminée par adaptation ou correction des solutions connues. Enfin, la session est conservée dans la base de cas : c'est la phase d'apprentissage.

4. **Arbre de Décision (AD)** : Un arbre de décision est composé de feuilles correspondant aux noms des classes (les diagnostics) et de nœuds internes correspondant à des tests avec une branche pour chacun des résultats possibles du test. Les arcs entre les nœuds représentent des valeurs possibles pour les différents tests. Typiquement, les arbres de décision sont utilisés lorsque l'on traite un problème de diagnostic comme un problème de décision ou de classification.
5. **Réseaux bayésiens (RB)** : Les réseaux bayésiens fournissent une méthode pour représenter les relations entre des propositions ou des variables, même si ces relations comportent de l'incertitude ou de l'imprécision. Ils peuvent être générés

automatiquement à partir de données fournies par des experts ou par apprentissage. Utiliser les réseaux bayésiens en diagnostic revient à calculer les probabilités conditionnelles d'événements liés les uns aux autres par des relations de cause à effet. Connaître la (les) cause(s) la (les) plus probable(s) d'un événement (nœud) en fonction des informations données (les observations). Plusieurs travaux ont été publiés dans l'utilisation des réseaux bayésiens pour le diagnostic des systèmes complexes [Verron et al., 2008; Weber & Simon, 2008; Verron et al., 2009]. Ils proposent des nouveaux modèles basés sur les réseaux bayésiens pour détecter et localiser les défauts d'un système.

2.7.3 Architectures du diagnostic

Plusieurs approches pour le diagnostic des NCSs peuvent être envisagées :

1. **Approche centralisée** : le diagnostiqueur est une entité centralisée qui à partir d'un modèle global du système et de toutes les observations reçues, génère le diagnostic global (voir Fig.2.5).

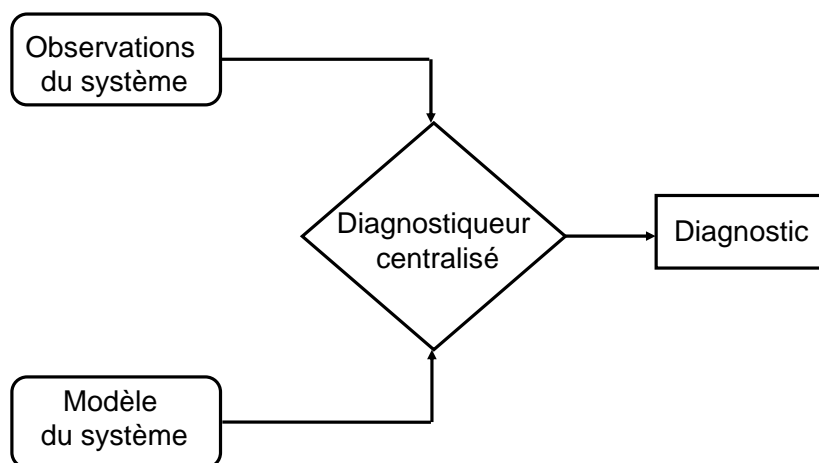


Figure 2.5 – Diagnostic centralisé

2. **Approche décentralisée** : le diagnostic est réalisé par plusieurs diagnostiqueurs locaux qui envoient les informations au superviseur pour générer le diagnostic global (voir Fig.2.6).
3. **Approche distribuée** : [Su et al., 2002] proposent une méthode de diagnostic distribué à partir de diagnostiqueurs locaux. Chaque diagnostiqueur est associé à un composant et le diagnostic est élaboré à partir du comportement de ce composant et de ses interactions avec les autres composants. Le diagnostic local détermine l'état normal ou fautif du composant d'après les observations locales. Les diagnostiqueurs communiquent entre eux pour partager leur information de diagnostic

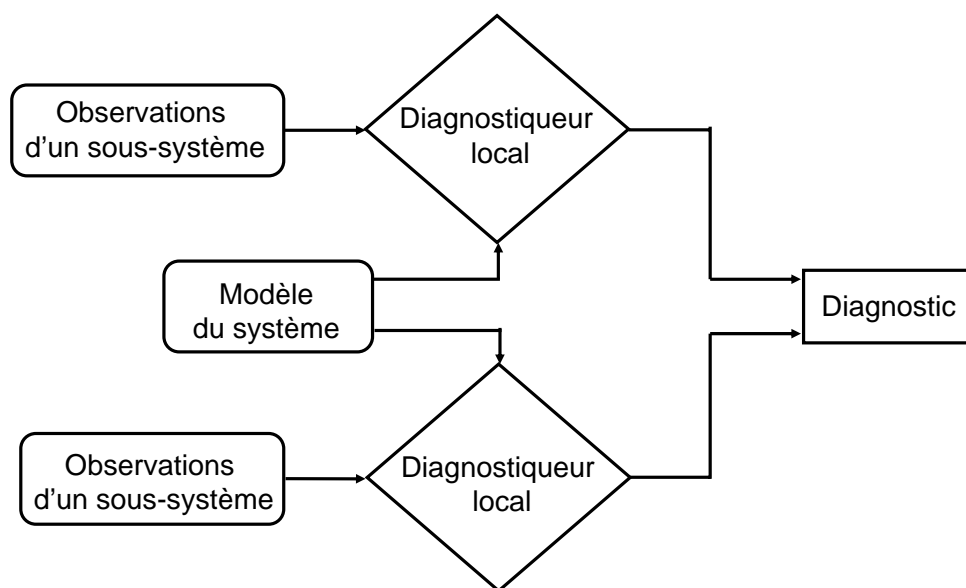


Figure 2.6 – Diagnostic décentralisé

avec l'objectif d'améliorer leurs diagnostics locaux. Le diagnostic global est déterminé par recherche de la consistance entre les diagnostics locaux. Les problèmes liés aux systèmes distribués, sujets à des défaillances de composants, sont notamment les difficultés à diagnostiquer le système. En diagnostic distribué chaque nœud en fonctionnement doit donner des informations précises sur l'état des composants qu'il supervise (en fonctionnement ou endommagé) [Subbiah & Blough, 2002] (voir Fig.2.7). Le diagnostic distribué dans les systèmes commandés en réseau ressemble au diagnostic distribué des systèmes habituellement connus sauf que la communication entre les différentes entités de diagnostic est assurée par un réseau filaire ou sans fil (Fig.2.8).

Les NCSs deviennent de plus en plus répandus pour des applications en robotique. Avec la rapidité accrue des techniques de communication, notamment sans fil, la distribution du contrôle et du diagnostic est devenue nécessaire pour réduire le traitement d'information en embarqué et son effet sur la consommation d'énergie, mais aussi le poids et la taille du robot qui sont les contraintes clés lors de la conception du système. Aussi dans les systèmes distribués, la communication est un paramètre important à prendre en considération, étant donné que les différents composants du système n'ont pas nécessairement une mémoire partagée.

Dans [Berbra et al., 2008] une comparaison entre deux approches pour le diagnostic d'un système commandé en réseau est proposée, l'une ne prend pas en compte la désynchronisation de données, l'autre considère le synchronisme. Cette comparaison a été réalisée en simulant un moteur à courant continu, un contrôleur débarqué et un diagnostiqueur en utilisant les outils de simulation Matlab/Simulink. La toolbox TrueTime, dédiée à la

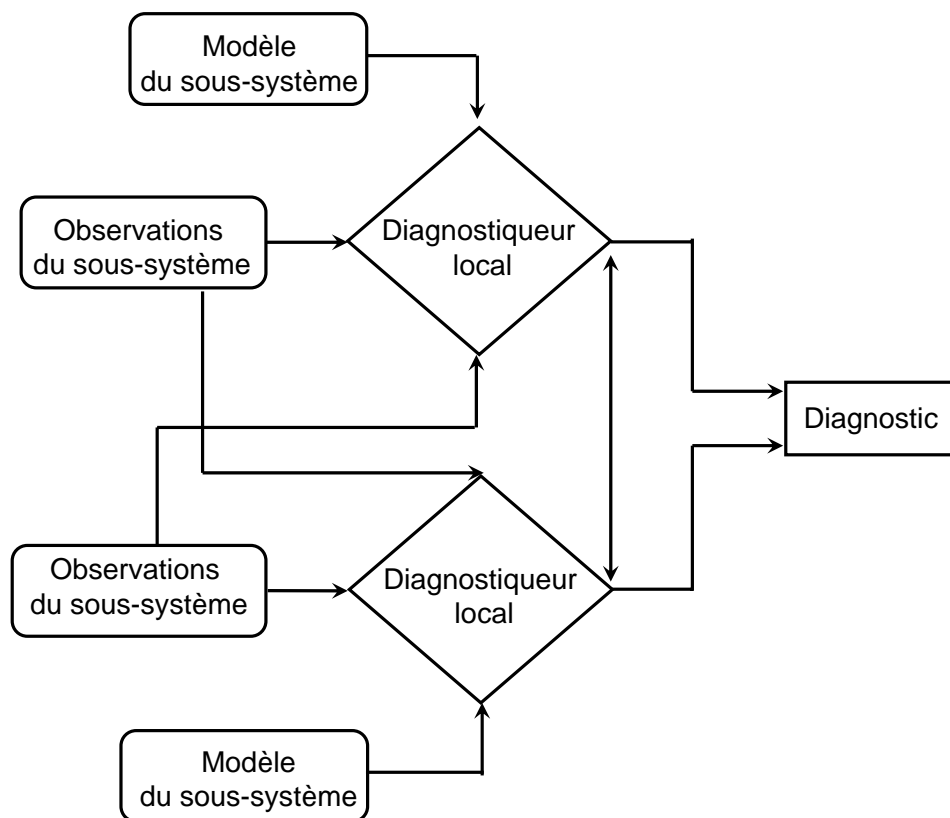


Figure 2.7 – Diagnostic distribué

simulation de systèmes temps-réel distribués, a été utilisée pour simuler le réseau.

Dans [Mechraoui et al., 2010a] une approche d'aide à la décision et de diagnostic off-line par réseaux bayésiens a été proposée. Ce papier prend en compte les contraintes apportées par le réseau sur les algorithmes de diagnostic, et les manières dont le réseau peut s'adapter dynamiquement aux besoins ponctuels des algorithmes de diagnostic, en fonction des contraintes d'environnement, de l'état du système et de sa criticité.

Dans [Mechraoui et al., 2010b] la question du diagnostic et de l'aide à la décision a été abordée, pour un système commandé en réseau intégrant plusieurs robots mobiles communiquant via un réseau sans fil. L'utilisation d'un réseau sans fil dans la boucle de contrôle influence la qualité de contrôle (QoC) du système en termes de qualité de service (QoS) du réseau sans fil. Un nouveau modèle (basé sur les réseaux Bayésiens) de la QoS et son influence sur la QoC a été proposé. L'objectif principal de ce modèle est de diagnostiquer, *on-line*, les problèmes liés à la QoS et la QoC. En fonction du diagnostic et de la situation, une décision est prise en temps-réel pour reconfigurer le réseau sans fil dans un environnement multi-stations et le contrôleur de manière à éviter la dégradation de la QoC ainsi que de la QoS.

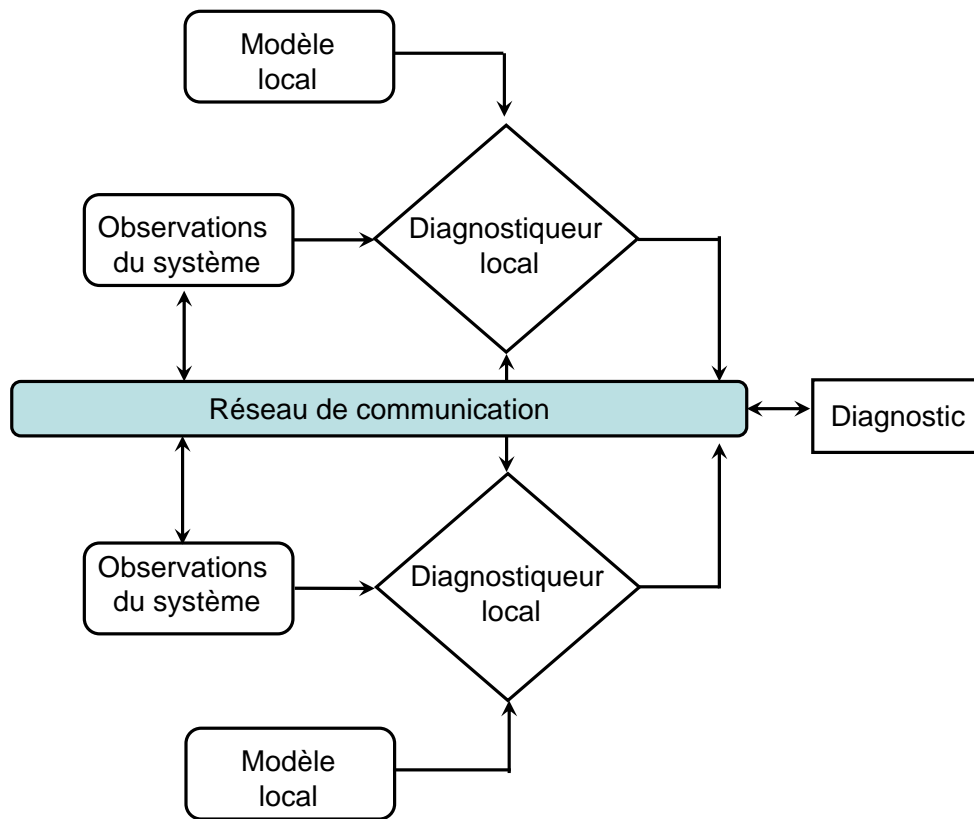


Figure 2.8 – Diagnostic distribué dans des systèmes commandés en réseau

2.8 Conclusion

Ce premier chapitre a permis de positionner notre travail dans les travaux menés par la communauté des systèmes commandés à travers un réseau. Nous avons alors présenté au début du chapitre les NCSs et les différentes approches appliquées jusqu'à maintenant pour résoudre les problèmes liés à l'intégration du réseau dans la boucle de commande (Analyse de stabilité, Synthèse de commande, Estimation et co-conception). Ensuite, une présentation des différentes approches pour le diagnostic des systèmes ainsi que le diagnostic des systèmes commandés en réseau (Approches qualitatives et quantitatives). Dans un premier temps, on a abordé les différentes méthodes du diagnostic des systèmes et ensuite le diagnostic des réseaux filaires et sans fil. Dans le contexte du diagnostic, nous pouvons avoir plusieurs architecture à savoir : diagnostic centralisé, décentralisé ou bien distribué.

Chapitre 3

Architecture de commande via un réseau d'un robot unicycle autonome

Sommaire

| | | |
|------------|--|-----------|
| 3.1 | Introduction | 49 |
| 3.2 | Le robot Khepera III et son modèle | 50 |
| 3.2.1 | Le robot | 50 |
| 3.2.2 | Modèle du robot | 57 |
| 3.3 | Architecture de commande pour robot autonome | 59 |
| 3.3.1 | Schéma général de la commande | 60 |
| 3.3.2 | Implantation | 60 |
| 3.4 | Architecture de commande temps réel du robot | 65 |
| 3.4.1 | Programmation sous TrueTime | 65 |
| 3.4.2 | Architecture de commande temps réel | 66 |
| 3.5 | L'architecture de la commande du robot via un réseau sans fil | 67 |
| 3.5.1 | Simulation, expérimentation et résultats | 69 |
| 3.6 | Conclusion | 77 |

3.1 Introduction

Les travaux menés dans ce chapitre concernent l'organisation, le développement et l'intégration de l'ensemble des fonctions opératoires dont doit disposer un robot mobile afin d'appréhender l'environnement et d'agir selon la mission qui lui a été confiée. Il s'agit donc des fonctions de perception, de commande et de modélisation. Un robot mobile peut être capable d'accomplir une variété de tâches dans un environnement mal connu

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

et évolutif, nécessite des capacités décisionnelles qui vont lui permettre de déterminer les actions adéquates selon l'environnement et la mission. Ainsi les fonctions opératoires sont au service du niveau décisionnel et doivent s'intégrer dans une structure comportant différentes entités communicantes que l'on appelle architecture de commande du robot. C'est pourquoi dans ce chapitre, nous avons pour objectif de présenter le prototype du robot unicycle, son modèle ainsi que l'architecture de commande. En l'occurrence, nous détaillerons la partie mécanique (actionneurs), la partie électronique (Micro-contrôleurs, capteurs), la partie informatique (le système d'exploitation) du robot ainsi que la partie automatique (l'architecture de commande).

L'autonomie d'une machine peut être définie comme la capacité d'exécuter une action ou une séquence d'actions sans intervention. Il y a naturellement une limite au delà de laquelle le système s'avère impuissant à satisfaire le but fixé. Le degré d'autonomie se caractérise par la variété de situations que le système peut appréhender et aussi la précision de la nature des tâches qui lui sont confiées.

Ce chapitre est organisé de la façon suivante. La deuxième section présente une brève description du robot Khepera III [Lambercy & Caprari, 2007] et de son modèle cinématique et le modèle dynamique de ses deux moteurs. Dans la section suivante, l'architecture de commande pour suivre une trajectoire sera présentée et illustrée avec quelques simulations. Celles-ci utilisent l'outil de simulation temps réel "TrueTime" [Anderson et al., 2007] pour s'approcher de la réalité en utilisant des blocs de calcul qui prennent en compte le temps de calcul ainsi que l'ordonnancement des tâches. Les résultats d'implémentation sur le robot réel seront aussi présentés et comparés avec la simulation. Et enfin une conclusion terminera ce chapitre.

3.2 Le robot Khepera III et son modèle

3.2.1 Le robot

Le robot Khepera III (voir Fig. 3.1) est un véhicule autonome, muni de deux roues motrices indépendantes, qui a été développé par le Laboratoire des systèmes autonomes de l'EPFL à Lausanne. Le couple est transmis par deux moteurs à courant continu identiques. Les moteurs sont équipés d'encodeurs magnétiques (12 impulsions par millimètre du déplacement du robot) pour donner la position angulaire et linéaire du robot par rapport à un repère fixe.



Figure 3.1 – Robot Khepera III

Processeur et micro-contrôleur

Afin de faire fonctionner le robot, celui-ci est composé d'un processeur Motorola 68331 @25MHz, RAM : 512 KB, Flash : 512 KB programmable par le port série. Le Khepera III possède une extension Korebot qui lui ajoute des fonctionnalités, comme la possibilité d'exécution d'un programme en C. Elle est composée d'un processeur Intel PXA255 XScale ARM @400MHz, RAM : 64MB, Flash : 32MB. Le système d'exploitation du robot est un Kernel 2.6.25.7-kb1 et celui du Korebot un ARM Embedded Linux. Comme micro-contrôleur, le robot possède un dsPIC et un PIC18F4431. La programmation du robot se fait en langage C sous Linux.

Les moteurs

Le robot est actionné par deux moteurs à courant continu indépendants. Ces moteurs sont commandés grâce à une PWM (Pulse Width Modulation) générée par le PIC 18F à travers un double pont en H. Ils sont contrôlés par un PID programmé dans le PIC. Les contrôleurs des moteurs peuvent être utilisés pour contrôler soit la vitesse, soit la position, ils basculent automatiquement selon le type de consigne donnée. Dans la suite de notre travail, seul le contrôle de la vitesse sera utilisé. La vitesse maximale des moteurs est de 0.5m/s. Afin de choisir les bons coefficients pour le correcteur, une identification du moteur doit être faite. Pour cela, le moteur doit être en boucle ouverte et soumis à un échelon de tension. La réponse du moteur à cet échelon est représentée sur la Fig. 3.2. D'après la réponse du moteur enregistrée sur notre prototype, le système est un système du premier ordre. La période d'échantillonnage utilisée est $T_e = 20ms$ (voir section 3.3.2).

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

Mesurons le gain K :

$$K = \frac{V_{finale}}{Echelon} = 6.13 \quad (3.2.1)$$

A 95% de la vitesse finale, nous obtenons :

$$3\tau = 43 * T_e = 2.14s \Rightarrow \tau = 0.71s \quad (3.2.2)$$

Finalement, la fonction de transfert en boucle ouverte du moteur est :

$$H = \frac{K}{1 + \tau s} = \frac{6.13}{1 + 0.71s} \quad (3.2.3)$$

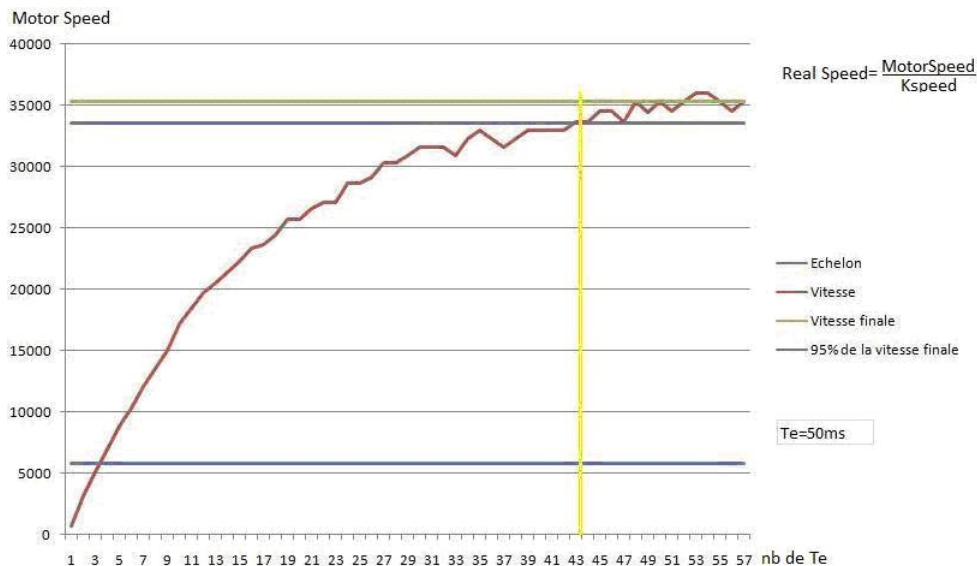


Figure 3.2 – Réponse du moteur à un échelon de tension

Les encodeurs

Chaque moteur est associé à un encodeur d'une résolution de 2764 points par tour. Les roues ont un diamètre de 41mm, ce qui permet d'avoir une précision d'environ 50 micromètres. Ces deux encodeurs permettent de connaître la position du robot par rapport à une origine.

Pour connaître la position du robot à tout moment, le principe de l'odométrie est utilisé. En effet, l'odométrie est une technique qui consiste à calculer la position du robot, en cumulant tous les déplacements effectués depuis la dernière position connue par les deux roues.

Calcul de la position :

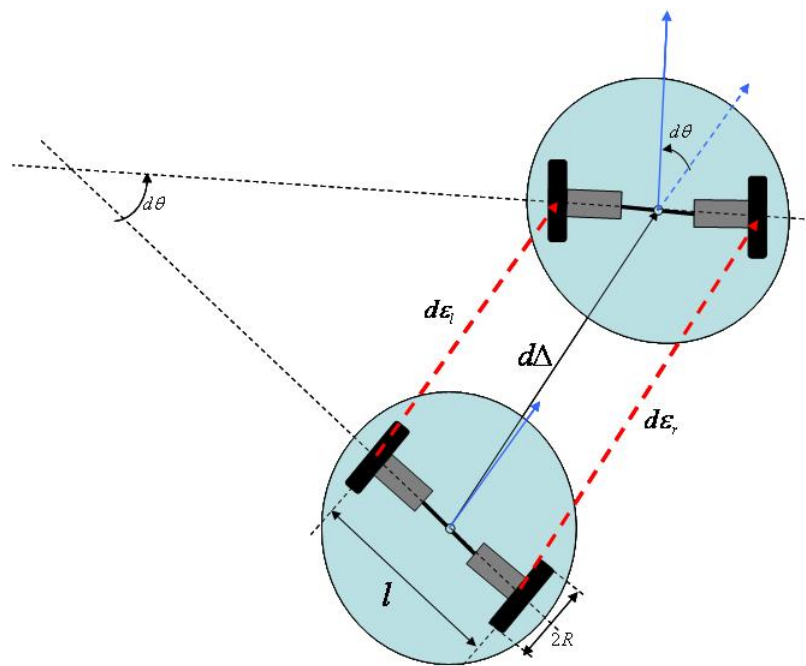


Figure 3.3 – Odométrie

On considère que la position initiale du robot est $(x, y, \theta) = (x_0, y_0, \theta_0)$ (voir 3.3).

La variation de l'angle $d\theta$:

$$d\theta = \frac{d\varepsilon_r - d\varepsilon_l}{l} \Rightarrow \theta = \theta_0 + d\theta \quad (3.2.4)$$

La variation de l'avancement $d\Delta$:

$$d\Delta = \frac{d\varepsilon_r + d\varepsilon_l}{2} \quad (3.2.5)$$

Calcul de la variation selon x et y :

$$dx = d\Delta \cos d\theta \quad (3.2.6)$$

$$dy = d\Delta \sin d\theta \quad (3.2.7)$$

Calcul de la position en mètres :

$$x = x_0 + dx \quad (3.2.8)$$

$$y = y_0 + dy \quad (3.2.9)$$

Les capteurs

Le robot Khepera III est muni d'un réseau de capteurs infrarouges destinés à la détection à courte distance d'obstacles et de capteurs ultrasons pour une détection d'obstacle longue distance (voir Fig.3.4)

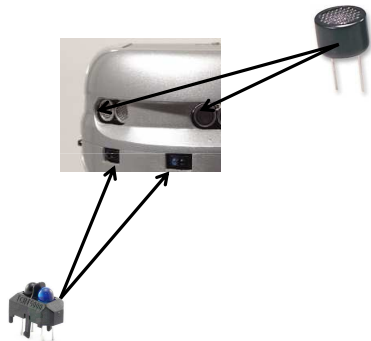


Figure 3.4 – Capteurs du Khepera III

- **Les capteurs infrarouges** : le robot Khepera possède 11 capteurs infrarouges TCRT5000, 9 répartis autour du robot pour permettre la détection et l'évitement d'obstacle et 2 en-dessous pouvant être utilisés pour suivre une ligne ou détecter la présence du sol. Chacun des capteurs est composé d'un émetteur et d'un récepteur. Ces capteurs permettent de faire deux mesures différentes. Ils peuvent détecter la lumière ambiante ; dans ce cas, seuls les récepteurs sont utilisés ; cette méthode ne sera pas utilisée. Ils peuvent aussi détecter les rayons émis par l'émetteur et réfléchis par un obstacle. Cette dernière permet de repérer un objet jusqu'à 12 cm environ si celui-ci est blanc, sinon plus l'objet sera sombre plus la distance maximale sera réduite. Chaque capteur renvoie comme information, un nombre compris entre 0 et 4000 suivant la distance de l'objet. La Fig. 3.6 montre les données fournies par les capteurs en fonction de la distance d'un objet de couleur bleue et rouge.

Les caractéristiques générales des capteurs infrarouges sont :

Type TCRT5000

Composition Emetteur (DEL) et récepteur (phototransistor)

Portée 5 à 12 cm

Directivité Cône d'environ 5

Précision Relativement précis, mais la précision diminue en fonction de la distance et de la couleur de l'obstacle.

Sensibilité aux interférences Sont sensibles aux fortes sources de lumière qui contiennent un fort rayonnement infrarouge, aux couleurs et à la nature des obstacles mais restent robustes.

Inconvénients majeurs Non-linéarité de la réponse en fonction du flux. Temps de réponse en général élevé et bande passante limitée.

La mesure est effectuée toutes les 33 ms. Pendant les 33 ms, les onze capteurs sont lus de manière séquentielle toutes les 3 ms. La lecture commence du capteur arrière gauche et se poursuit dans le sens horaire Fig.3.8. La valeur retournée à

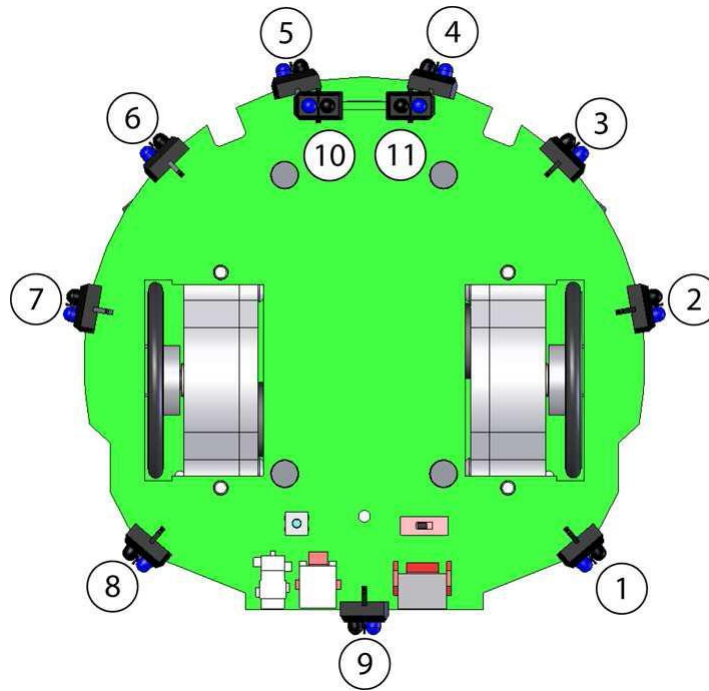


Figure 3.5 – Position des capteurs infrarouges

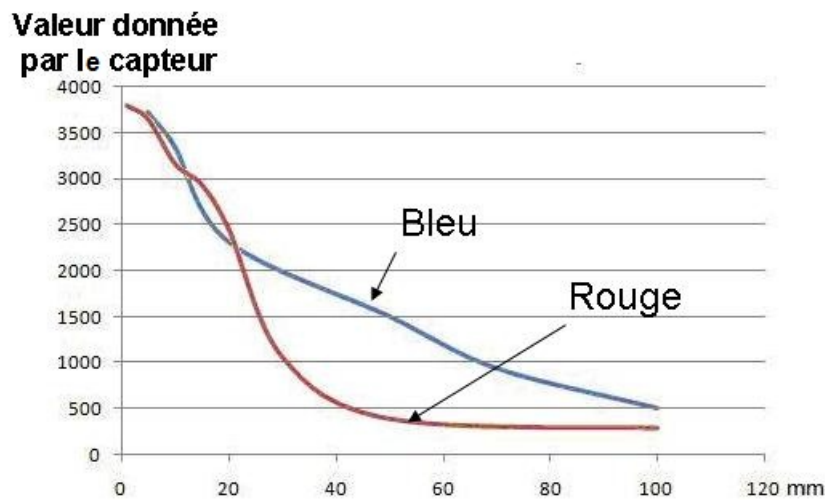


Figure 3.6 – Réponse d'un capteur IR en fonction de la distance

un moment donné est le résultat de la dernière mesure prise. La sortie de chaque mesure est une valeur analogique convertie en une valeur 12 bits numériques.

En ce qui concerne le principe de fonctionnement de ces capteurs, la DEL émet des infrarouges et le phototransistor fait varier la tension en fonction de la quantité de lumière qu'il reçoit. Cette tension est ensuite convertie en distance grâce à une loi physique Fig.3.7.

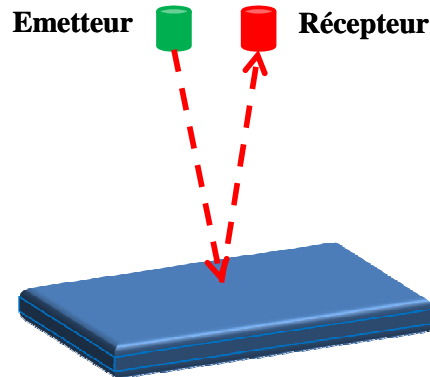


Figure 3.7 – Position des capteurs infrarouges

- **Les capteurs ultrasons** : le robot possède également 5 capteurs ultrasons positionnés comme sur la Fig. 3.8. Sur le même principe que les capteurs infrarouges, les capteurs ultrasons sont formés de paires, un émetteur 400ST100 et un récepteur 400SR100. Ces capteurs permettent de repérer les objets sur une plus grande distance que les précédents. En effet, ils peuvent repérer un objet entre 20cm et 2m. De plus, un capteur est capable de repérer jusqu'à trois objets, de mesurer la distance entre les objets et le robot en mesurant le temps qu'a mis le signal pour faire l'aller retour et l'amplitude du signal reçu. Cette dernière permet de savoir s'il y a vraiment un objet ou si c'est seulement du bruit.

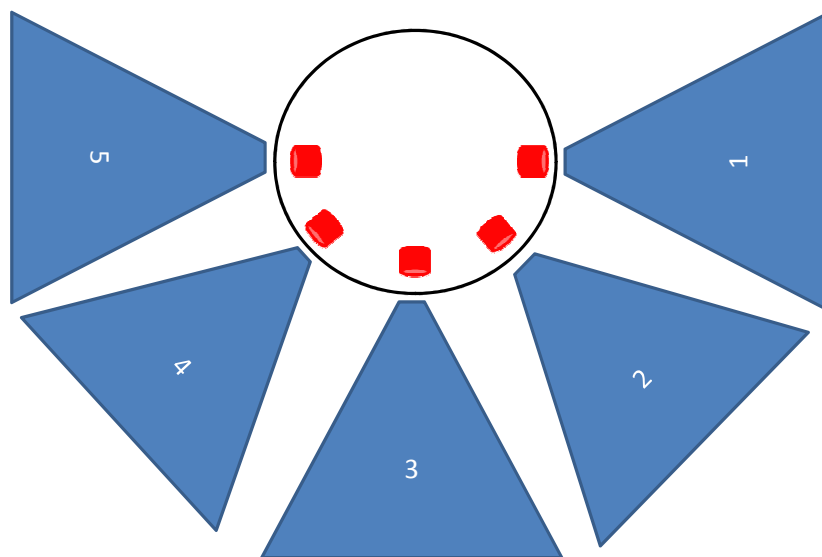


Figure 3.8 – Zones de détection des capteurs ultrasons

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

Les caractéristiques générales des capteurs ultrasons sont :

Type 400ST100 (Emetteur), 400SR100 (Récepteur).

Composition Emetteur et récepteur

Portée 20 à 200 cm

Directivité Cône d'environ 30

Précision Relativement précis mais la précision diminue avec la distance, l'angle de mesure et les conditions de température et de pression.

Sensibilité Sensible à la température et à la pression. Egalement sensible aux autres robots utilisant la même fréquence

Inconvénients majeurs Sensibles aux mêmes fréquences. Moins directif que les capteurs infrarouges.

L'émetteur et le récepteur sont situés dans le même boîtier. L'émetteur envoie un train d'ondes qui va se réfléchir sur l'objet à détecter et revenir à la source. Le temps mis pour parcourir un aller-retour permet de déterminer la distance de l'objet par rapport à la source. L'émetteur envoie un train d'onde puis le capteur passe en mode réception et attend le retour du signal. Si un signal est de retour avant 20 ms, l'électronique du capteur détermine le temps que le signal a mis pour faire l'aller retour puis donne la distance à laquelle l'objet se situe, sous la forme d'un signal analogique 4-20mA. Si aucun signal ne revient après 20 ms, le capteur repasse en mode émission et renvoie une onde.

Les protocoles de communication

La communication interne au robot se fait par liaison série I2C (Inter Integrated Circuit). L'envoi de commande, de programme et la communication entre un ordinateur et le robot peut se faire par plusieurs moyens. A l'origine, la communication se fait par liaison série. Cependant, la communication peut se faire aussi via USB, Bluetooth (802.15.4) ou Wifi (802.11 b/g). Il faut en revanche les installer sur le robot s'ils ne le sont pas et les configurer.

3.2.2 Modèle du robot

Modèle cinématique

Le robot étudié est unicycle. Il est actionné par deux roues indépendantes et possède éventuellement un certain nombre de roues folles assurant sa stabilité. Le schéma des robots de type unicycle est donné sur la Fig. 3.9. La roue folle, qui n'intervient pas dans la cinématique du robot, n'a pas été représentée. Le déplacement du robot est décrit

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

par un vecteur $M = [x \ y \ \theta]^T$ où le couple (x, y) désigne les coordonnées du centre de masse du robot par rapport au repère $R(O, \vec{X}, \vec{Y})$ et l'angle θ représente l'orientation du véhicule par rapport à l'axe (O, \vec{X}) de ce même repère.

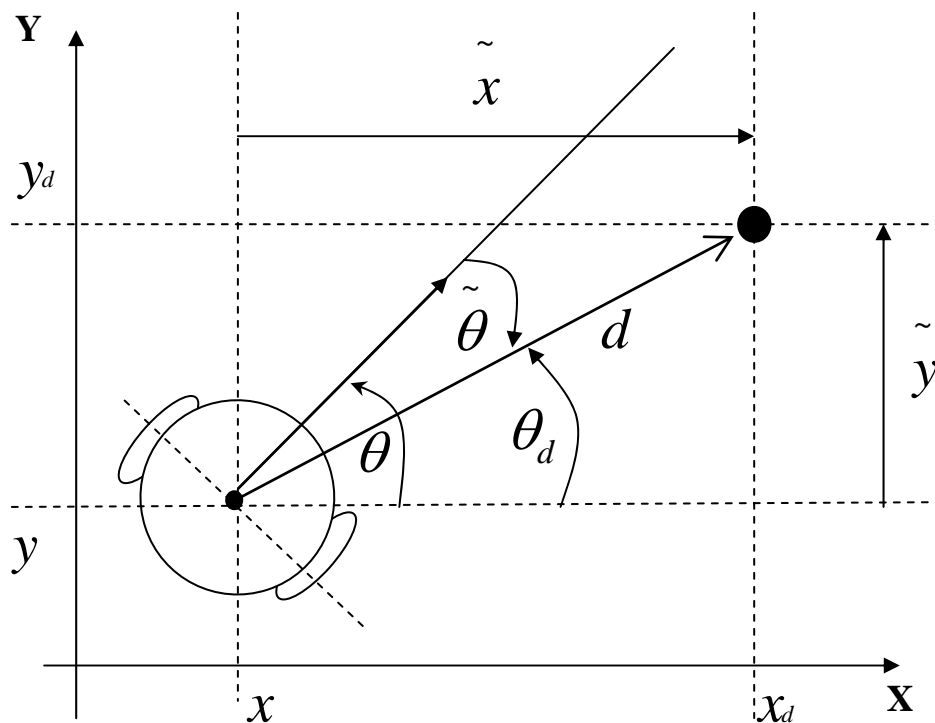


Figure 3.9 – Modèle cinématique du robot

On considère que l'axe de rotation passe par le centre de masse du robot. Le modèle cinématique est donné par l'équation non-linéaire suivante :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (3.2.10)$$

Le modèle cinématique du robot a deux entrées ω_{left} et ω_{right} , vitesses angulaires de la roue gauche et droite respectivement. Ces vitesses angulaires sont liées directement à la vitesse linéaire v et angulaire ω du robot suivant les deux équations suivantes :

$$\omega_{left} = \frac{v - \frac{\omega * l}{2}}{R}, \quad \omega_{right} = \frac{v + \frac{\omega * l}{2}}{R} \quad (3.2.11)$$

Où l est la distance entre les roues motrices, et R le rayon des roues.

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

Modèle dynamique

Le modèle dynamique du robot est caractérisé par les équations des deux moteurs. La fonction de transfert a été trouvée par identification (voir section 3.2.1).

$$\frac{\omega_{(left,right)}}{U} = \frac{K}{\tau s + 1} \quad (3.2.12)$$

Où U représente la tension appliquée à chaque moteur et $\omega_{(left,right)}$ représente la vitesse angulaire de chaque roue.

3.3 Architecture de commande pour robot autonome

L'objectif de cette section est de présenter la conception de la commande du robot pour qu'il puisse atteindre une cible. Pour cela, deux niveaux de commande sont implémentés. Le premier niveau (bas niveau) concerne la commande des deux roues en vitesse angulaire en utilisant un contrôleur PI pour chacune des roues motrices (Fig. 3.10). Le deuxième niveau de commande concerne la régulation en vitesse linéaire et angulaire du robot en utilisant les lois de commande non-linéaires proposées dans [Toibero et al., 2007].

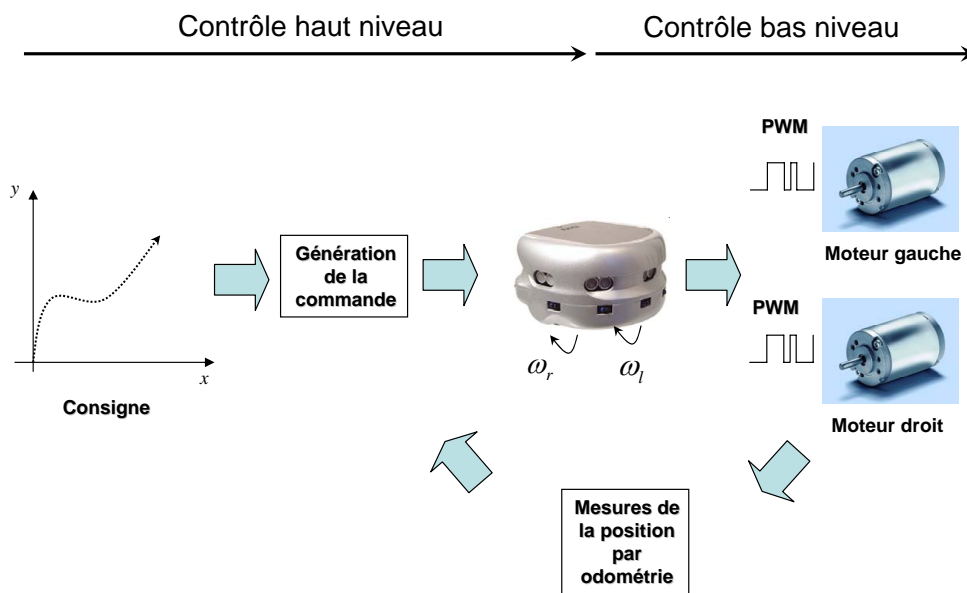


Figure 3.10 – Contrôle du robot

3.3.1 Schéma général de la commande

Pour simplifier la conception de la tâche de commande, on implémente deux régulateurs en cascade : le régulateur en vitesse, et le régulateur de trajectoire et/ou de position. Cette séparation entre la régulation en vitesse du robot et le suivi de trajectoire semble être une bonne solution pour plusieurs raisons :

- bonne séparation des tâches pour une meilleure lisibilité ;
- facilité de programmer lors de l'implémentation.

Le schéma de régulation est représenté sur la Fig.3.11.

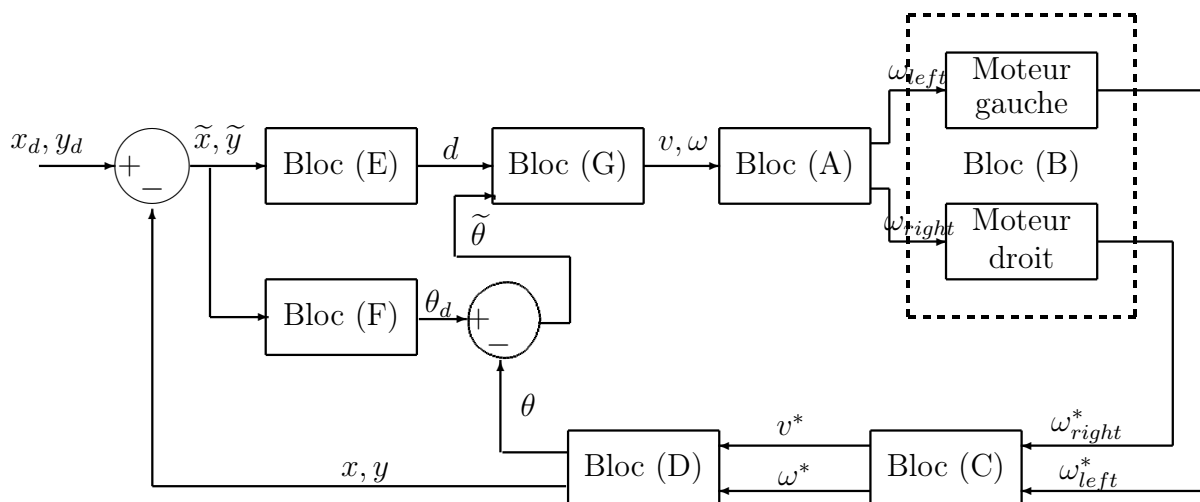


Figure 3.11 – Schéma général de la commande du robot simulé en Matlab/Simulink (Bloc continu)

3.3.2 Implantation

Commande en vitesse du robot en boucle ouverte

Une bonne commande en vitesse du robot est nécessaire pour espérer un suivi de trajectoire et une position corrects.

L'approche standard pour la commande en vitesse des roues d'un robot est l'utilisation d'un PI par roue (roue gauche et droite).

On a utilisé Matlab/Simulink pour la simulation du contrôle de ce robot en boucle fermée. La régulation en vitesse va être détaillée au fur et à mesure en expliquant chaque bloc du schéma général (Fig.3.11).

Détaillons chaque partie du schéma (Fig.3.12). Les entrées, intitulées v et ω , représentent

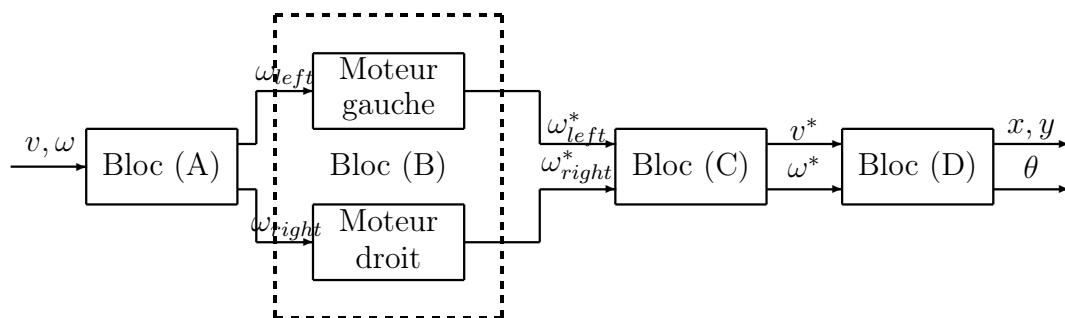


Figure 3.12 – Commande en vitesse en boucle ouverte

le profil des vitesses souhaitées. Le profil trapézoïdal a été choisi pour bien représenter la réalité. L'entrée ω peut être un profil trapézoïdal, ou une vitesse angulaire nulle. Ce choix permet de réaliser des tests en translation pure et en translation/rotation. Les roues étant indépendantes, et couplées chacune à un moteur, les vitesses linéaire v et angulaire ω désirées (ou de référence) pour le robot doivent être transformées en une vitesse applicable à chaque roue. C'est l'objet du bloc (A). Ces vitesses sont transformées selon les équations 3.3.1 et 3.3.2.

$$\omega_{left} = \frac{v - \frac{\omega^* l}{2}}{R} \quad (3.3.1)$$

$$\omega_{right} = \frac{v + \frac{\omega^* l}{2}}{R} \quad (3.3.2)$$

Où l est la distance entre les roues motrices, et R le rayon des roues.

Le bloc (B) reproduit la dynamique du système. Il comporte, pour chaque roue, donc pour chaque moteur, le contrôleur PI et la fonction de transfert correspondant à chaque moteur. La Fig.3.13 représente le schéma bloc de cette partie.

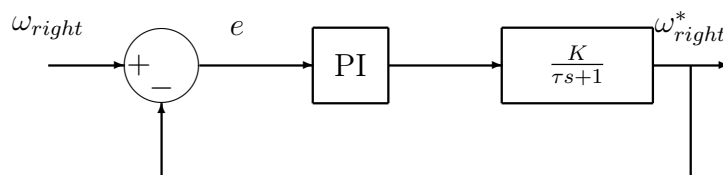


Figure 3.13 – Schéma de la partie dynamique du système

Le correcteur PI utilisé est caractérisé par la fonction de transfert de l'équation 3.3.3.

$$C(s) = K_p + \frac{1}{T_i s} \quad (3.3.3)$$

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

La réponse du système simulée en BO à un échelon unitaire sur la tension d'entrée d'un moteur en utilisant le modèle identifié expérimentalement à la Fig.3.2 est représentée sur la Fig.3.14.

Calcul du correcteur :

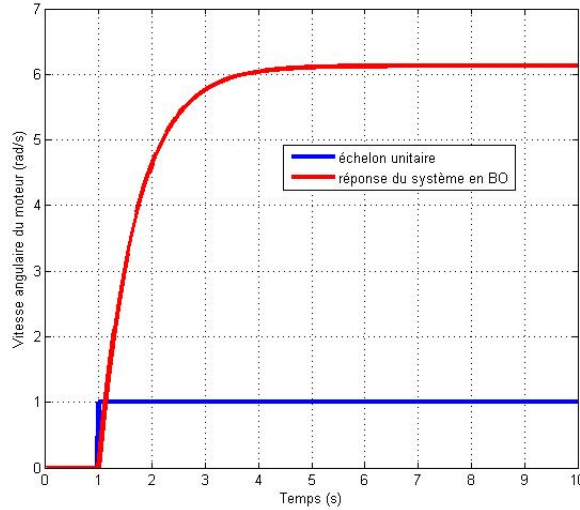


Figure 3.14 – Réponse simulée du système en boucle ouverte en appliquant un échelon à la tension d'entrée du moteur

Les coefficients du PI ont été choisis afin d'obtenir un temps de réponse en boucle fermée T_r de $100ms$, correspondant à une constante de temps en boucle fermée de $\tau_t = 34ms$. Calculons la fonction de transfert en boucle fermée :

$$H_{BF}(s) = \frac{H_{BO}}{1 + H_{BO}} \quad (3.3.4)$$

$$H_{BO}(s) = C(s) \frac{K}{\tau s + 1} = \frac{K_p T_i s + 1}{T_i s} \frac{K}{\tau s + 1} \quad (3.3.5)$$

Pour que la fonction de transfert en boucle fermée reste du premier ordre il faut que :

$$K_p T_i s + 1 = \tau s + 1 \Rightarrow K_p T_i = 0.71 \quad (3.3.6)$$

d'où :

$$H_{BF} = \frac{K}{T_i s + K} = \frac{1}{\tau_t s + 1} \quad (3.3.7)$$

Donc :

$$\frac{T_i}{K} = \tau_t \Rightarrow T_i = 0.2(s) \quad (3.3.8)$$

Par conséquent :

$$K_p = 3.5 \quad (3.3.9)$$

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

Pour passer du correcteur PI analogique au correcteur PI discret par simple numérisation, il faut dans un premier temps choisir une période d'échantillonnage ($T_r/10 < T_e < T_r/4$) où T_r est le temps de montée du système corrigé, ici égal au temps de réponse désiré. Nous obtenons les coefficients numériques K_{p_k} et T_{i_k} suivants en fonction des coefficients analogiques K_p et T_i :

$$K_{p_k} = K_p \quad (3.3.10)$$

$$K_{i_k} = \frac{T_e}{T_i} \quad (3.3.11)$$

$$C(z) = K_p + K_i \frac{z}{z-1} \quad (3.3.12)$$

Pour une période d'échantillonnage de $T_e = 20ms$:

$$K_{p_k} = 3.5 \quad (3.3.13)$$

$$T_{i_k} = 0.1 \quad (3.3.14)$$

La réponse du système en boucle fermée avec le régulateur PI discret choisi est représentée sur la Fig.3.15.

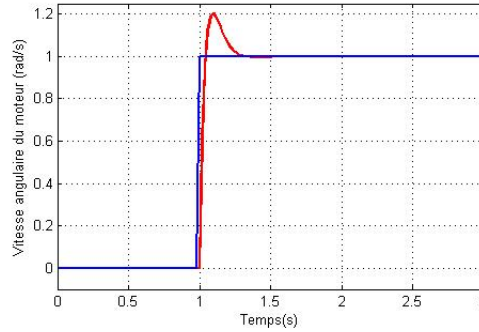


Figure 3.15 – Réponse indicielle du système en boucle fermée avec $K_{p_k} = 3.5$ et $T_{i_k} = 0.1$

Nous constatons que le système est stable avec une erreur statique en réponse à l'échelon nulle et un dépassement de 20% qui ne gêne pas vu la rapidité de la boucle locale par rapport à la boucle globale.

Le bloc (C) effectue l'opération inverse du bloc (A) : il transforme les vitesses des roues en deux vitesses : une vitesse linéaire et une vitesse angulaire. Ces dernières vont nous permettre d'obtenir la position du robot. Les formules utilisées sont les équations 3.3.15 et 3.3.16.

$$v^* = R \frac{\omega_{right}^* + \omega_{left}^*}{2} \quad (3.3.15)$$

$$\omega^* = R \frac{\omega_{right}^* - \omega_{left}^*}{l} \quad (3.3.16)$$

Le dernier Bloc (D) permet de calculer la position du robot à partir des vitesses linéaire et angulaire. Les équations utilisées sont celles du modèle cinématique (Équation 3.2.10).

Commande de position

Pour que le robot puisse rejoindre la position désirée définie par $[x_d \ y_d \ \theta_d]$, des erreurs de suivi d et d'orientation $\tilde{\theta}$ sont calculées.

$$\tilde{x} = x_d - x, \quad \tilde{y} = y_d - y \quad (3.3.17)$$

Les blocs (E) et (F) concernent le calcul de l'erreur de suivi (Eq. 3.3.18a) et de l'erreur d'orientation (Eq. 3.3.18b) respectivement.

$$d = \sqrt{\tilde{x}^2 + \tilde{y}^2} \quad (3.3.18a)$$

$$\tilde{\theta} = \theta_d - \theta = \tan^{-1}(\tilde{y}/\tilde{x}) - \theta \quad (3.3.18b)$$

Le bloc (G) est le contrôleur des vitesses linéaire et angulaire du robot. Selon [Toibero et al., 2007], les lois de commande suivantes sont considérées :

$$v = \frac{v_{max}}{1 + |d|} d \cos(\tilde{\theta}) \quad (3.3.19a)$$

$$\omega = \frac{v_{max}}{1 + |d|} \cos(\tilde{\theta}) \sin(\tilde{\theta}) + K_{\tilde{\theta}} \tanh(k_{\theta} \tilde{\theta}) \quad (3.3.19b)$$

v_{max} est la vitesse linéaire maximale que le robot peut atteindre ($v_{max} = 0.5m/s$) et $K_{\tilde{\theta}} = 0.5, k_{\theta} = 1$ sont des constantes. Dans [Toibero et al., 2007], les auteurs ont montré que ces lois de commande sont stables en utilisant la fonction du Lyapunov V_t .

$$V_t = \tilde{\theta}^2/2 + d^2/2 \quad (3.3.20)$$

Tout au long de ce mémoire, les blocs A, F, E, G de la Fig.3.11 seront considérés comme un seul bloc appelé le bloc "contrôleur". De plus, les bloc C et D seront considérés comme un seul bloc appelé le bloc "capteur". Le schéma allégé de la Fig.3.11 est présenté sur la Fig.3.16.

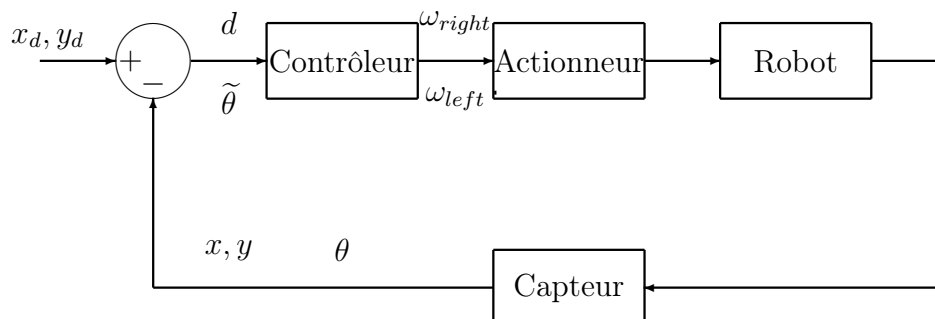


Figure 3.16 – Schéma général réduit de l'architecture de commande du robot

3.4 Architecture de commande temps réel du robot

L'objectif de cette partie est de présenter l'architecture de commande temps réel simulée en utilisant l'outil de simulation temps réel TrueTime développé à l'Université de Lund en Suède depuis 1999 [Anderson et al., 2007]. Cet outil est basé sur Matlab/Simulink. Il n'est pas un outil de co-simulation, le temps de calcul doit être spécifié pour chaque tâche.

3.4.1 Programmation sous TrueTime

Pour programmer une tâche sous TrueTime, on utilise un bloc dit Kernel (voir Fig.3.17) qui a pour objectif de simuler un microprocesseur en prenant en compte le temps de calcul et l'ordonnancement des différentes tâches. Le bloc Kernel a trois entrées et cinq sorties. Les entrées sont :

- l'entrée "A/D" sert à convertir un signal analogique en un signal numérique suivant la période d'échantillonnage prédéfinie dans le fichier d'initialisation.
- l'entrée "Interrupts" sert à interrompre ou réveiller les tâches.
- l'entrée "Rcv" est utilisée en cas de données reçues à travers un réseau.

Les sorties sont :

- la sortie "D/A" sert à convertir les données numériques en des données analogiques.
- "Snd" est une sortie pour envoyer les données à travers un réseau.
- "Schedule" sert à regarder ou à visualiser l'ordonnancement de la tâche.
- "Monitors" a pour objectif de surveiller l'évolution et le bon déroulement de la tâche.
- "P" permet de simuler la puissance utilisée ou consommée lors de l'exécution de la

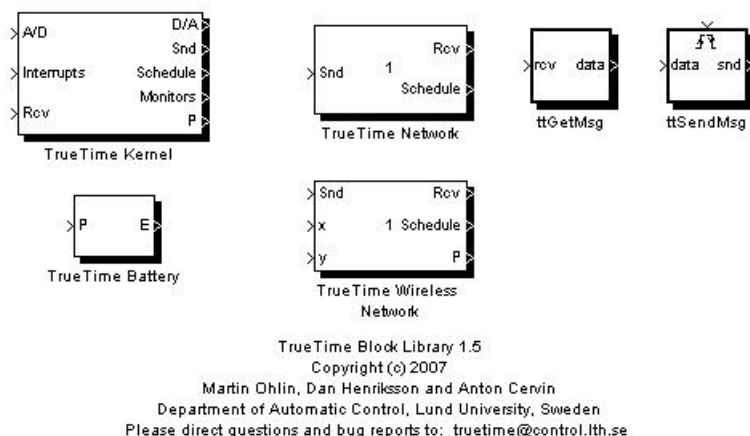


Figure 3.17 – Blocs TrueTime

tâche.

La programmation d'une tâche comprend trois étapes nécessaires :

1. La première étape est l'initialisation, elle est très importante pour initialiser tous les paramètres et aussi spécifier la nature de la tâche ainsi que toutes les ressources que la tâche peut utiliser (réseaux, batterie ...).
2. La deuxième étape concerne le code de programmation de la tâche en elle-même (les différentes actions de la tâche).
3. La troisième est la simulation du calculateur qui a pour but d'effectuer les calculs nécessaires pour accomplir la tâche.

Les autres blocs TrueTime qui peuvent être utilisés sont (voir Fig.3.17) :

1. Le bloc "TrueTime Network" permet de simuler un réseau filaire (CAN, Ethernet, Switched Ethernet).
2. Le bloc "TrueTime Wireless Network" permet de simuler les réseaux sans fils (Wi-Fi et Zigbee).
3. Le bloc "ttGetMsg" sert à envoyer des données ou des trames à travers un réseau filaire ou sans fil.
4. Le bloc "ttSendMsg" sert à recevoir des données ou des trames à travers un réseau filaire ou sans fil.
5. Le bloc "TrueTime Battery" permet la prise en compte de la consommation de l'énergie.

3.4.2 Architecture de commande temps réel

Le but de cette section est de créer des tâches pour chaque bloc de commande, à savoir les tâches contrôleurs (niveau haut et bas), actionneurs et capteurs. Dans toute la

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

partie qui suit, on considère que le temps de calcul est négligeable.

Dans la section précédente, au niveau bas de la commande des moteurs, le contrôleur PI a été discrétisé, toutefois il faut le considérer comme une tâche calculée par le microcontrôleur. Pour cela, un bloc kernel a été programmé pour calculer les commandes envoyées aux moteurs gauche et droit. Ce Kernel est équivalent dans notre prototype Khepera III au microprocesseur PIC18F4431 qui génère la commande envoyée à chaque moteur. La période d'échantillonnage de cette tâche est $T_e = 20ms$.

En ce qui concerne la tâche contrôleur (niveau haut), un bloc kernel a été aussi programmé et mis en place pour générer les consignes ω_{left} et ω_{right} , ce bloc comprend le calcul des erreurs (d et $\tilde{\theta}$), des commandes (v et ω) et aussi des consignes de vitesse des roues de chaque moteur (ω_{left} et ω_{right}). Cette tâche est guidée par les événements dans le cas d'une commande à travers un réseau, c'est-à-dire que le calcul ne se lance que quand les données capteurs sont reçues. On doit préciser que dans le cas où il n'y a pas le réseau, cette tâche est considérée comme guidée par le temps parce qu'on est sûr de recevoir les données capteur.

La tâche capteur, en revanche, a été programmée pour simuler les capteurs (à savoir les encodeurs sur le prototype) qui sont utilisés pour estimer la position du robot en utilisant le modèle cinématique inverse en simulation et le principe de l'odométrie sur le robot réel. Cette tâche intègre le calcul de la vitesse linéaire et angulaire du robot à partir des vitesses angulaires de chaque roue et aussi le calcul de la position et de l'orientation du robot. Elle est guidée par le temps et la période d'échantillonnage choisie est de $T'_e = 400ms$. C'est cette période qui détermine l'échantillonnage de la boucle principale (Fig.3.16). Elle est égale à 20 fois la période T_e de la boucle locale de commande de l'actionneur (Fig.3.13).

3.5 L'architecture de la commande du robot via un réseau sans fil

Le réseau sans fil a été intégré dans la boucle de commande entre le contrôleur et le robot d'un côté et entre les capteurs et le contrôleur d'un autre côté comme représenté sur la Fig. 3.18.

Pour pouvoir étudier le système, les paramètres suivants sont considérés :

- Un débit de 800 *kb/s* pour WiFi dans la couche physique.
- Le flux contrôleur et capteur utilisent 64 *bits* chacun.

L'outil de simulation des systèmes temps réel TrueTime [Anderson et al., 2007] a été utilisé pour simuler le comportement du réseau sans fil WiFi. Deux tâches ont été programmées. La première est la tâche contrôleur qui génère un flux contrôleur et la deuxième concerne la tâche capteur qui, à son tour, génère le flux capteur. La tâche contrôleur est guidée par les événements, c'est-à-dire que le contrôleur calcule et envoie

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

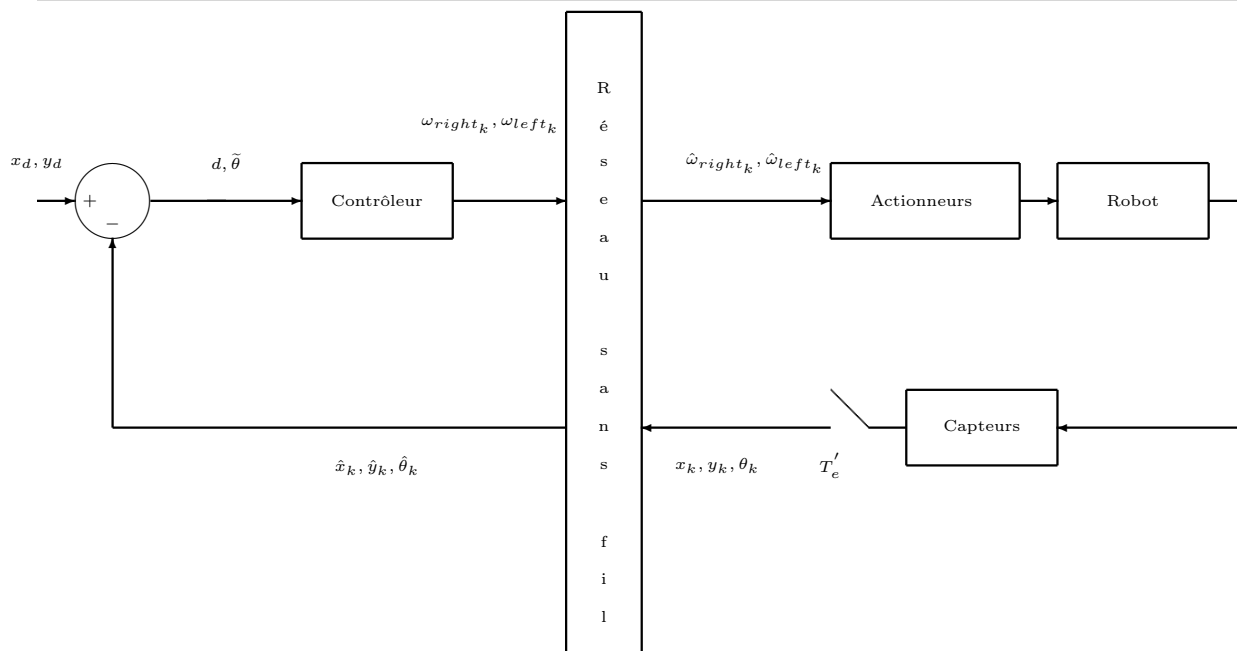


Figure 3.18 – Schéma général de la régulation via un réseau sans fil

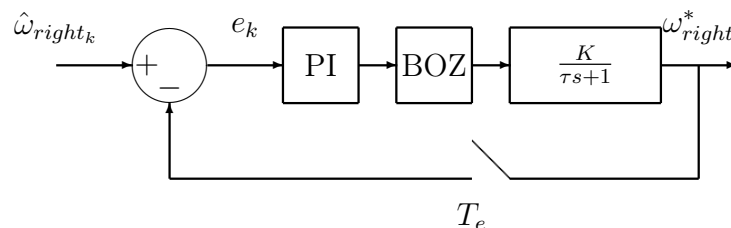


Figure 3.19 – Schéma bloc de la régulation locale des actionneurs du robot (Bloc B)

la commande $(\omega_{right_k}, \omega_{left_k},$ consignes des boucles locales de vitesse angulaire des roues) à condition qu'il ait reçu toutes les mesures qui sont envoyées par le capteur $(\hat{x}_k, \hat{y}_k, \hat{\theta}_k)$. Les données capteur et contrôleur sont critiques pour maintenir la stabilité du système, donc les pertes de paquets sont indésirables. Pratiquement, en utilisant une architecture distribuée et des réseaux sans fil, les pertes de paquets ne peuvent pas être totalement éliminées. Mais beaucoup de méthodes ont été proposées pour réduire la probabilité de pertes de paquets, comme la méthode de "Forward Error Correction (FEC)". La retransmission des paquets perdus est l'une des solutions proposées mais dans les systèmes temps réel, cette solution est exclue parce que le paquet perdu est considéré comme un paquet ancien. En effet, lorsque le paquet est perdu et qu'il est retransmis, les capteurs ont déjà mesuré de nouvelles valeurs; or le paquet renvoyé est basé sur

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

des données anciennes et le calcul correspondant de la commande est obsolète. Une solution pour assurer une bonne qualité de service robuste aux pertes de paquets est de *sur-échantillonner*, avec une période d'échantillonnage plus petite que la période nécessaire. Cependant, cette solution augmente le "Use Request Factor (URF)" et introduit davantage de retard et de pertes [Mechraoui et al., 2008]. La Fig. 3.20 et la Fig. 3.21 montrent que le robot peut communiquer avec la station si et seulement si il est dans la zone de couverture de cette dernière. Lorsque le robot s'éloigne, il perd en effet la communication, et il garde la dernière valeur de sa vitesse linéaire et angulaire (Fig.3.22); par conséquent il dérive de la trajectoire de référence (voir Fig.3.21 et Fig.3.22).

Le développement important actuel de l'infrastructure des réseaux sans fil (WPAN) permet au robot de trouver une autre station (PAN : Personal Area Network) pour communiquer. En conséquence, on suppose qu'on a différentes stations avec lesquelles le robot peut communiquer (voir Fig. 3.20). On suppose aussi que les stations peuvent communiquer entre elles via un réseau filaire de type infrastructure.

Dans le cas où le robot est en regard de deux stations qui assurent chacune un certain degré de QoS, le robot choisit en fonction du paramètre QoS la station qui lui assure une bonne QoS. Pour cela, on propose dans la section suivante la stratégie de changement de station pour reconfigurer et adapter le réseau aux contraintes dynamiques de la commande.

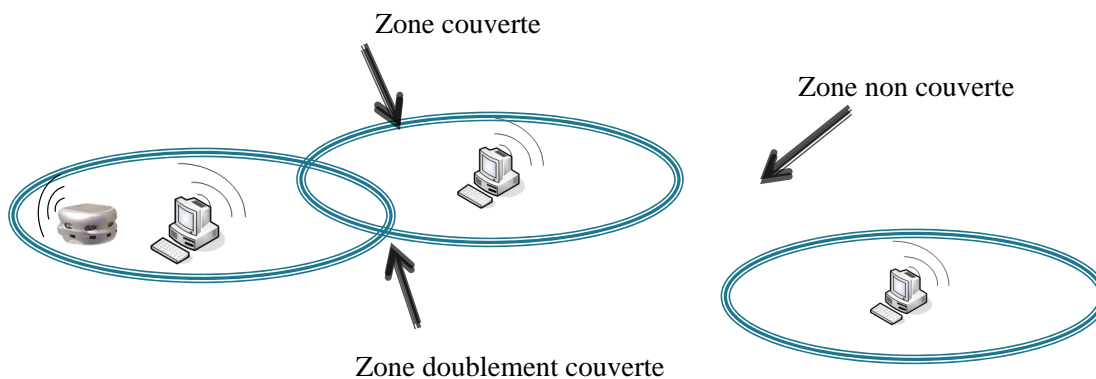


Figure 3.20 – Scénario général

3.5.1 Simulation, expérimentation et résultats

Atteinte d'une cible

Après avoir identifié, modélisé et proposé une architecture de commande du système à contrôler, nous avons appliqué une consigne de position $(x_d, y_d) = (1, 1)$. Les résultats

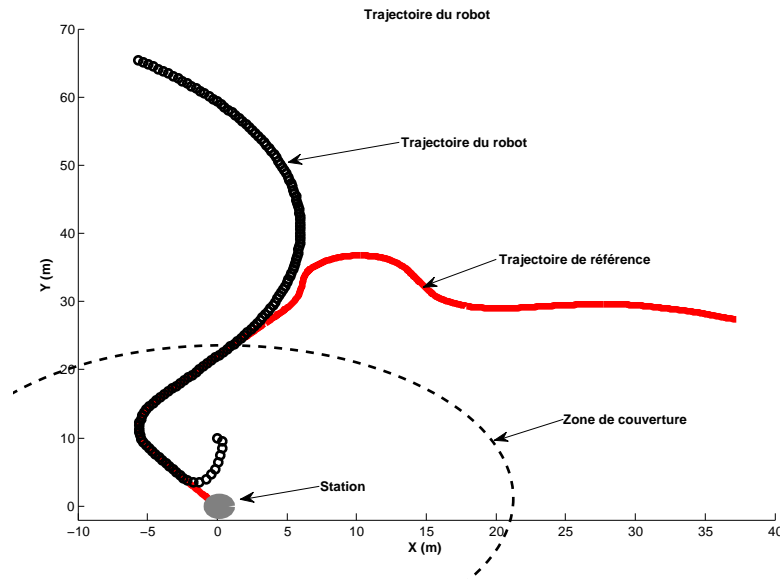


Figure 3.21 – Trajectoire du robot avec un point d'accès WLAN

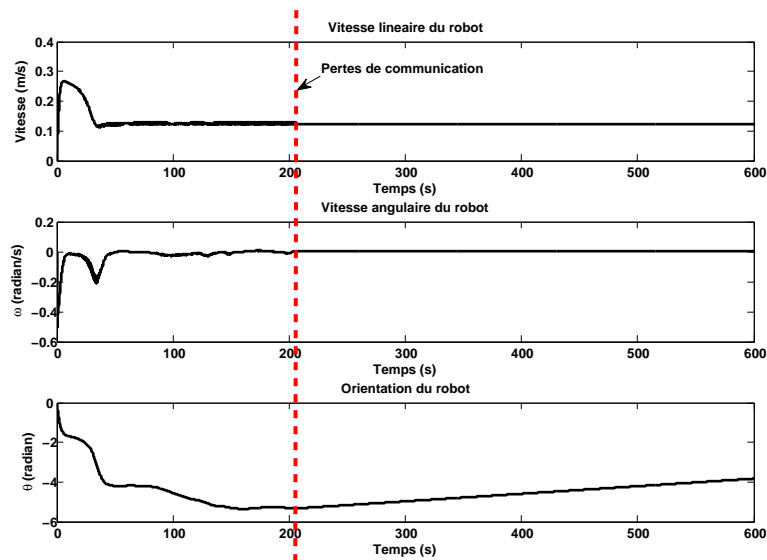


Figure 3.22 – Evolution des vitesses linéaire, angulaire et de l'orientation du robot avec réseau et une station

de simulation sont montrés sur les Fig.3.23 et Fig.3.24.

La Fig.3.23 montre la cible à atteindre avec la condition initiale $[X_0, Y_0, \theta_0] = [0, 0, 0]$ et aussi la trajectoire simulée du robot mobile en utilisant les contrôleurs (3.3.19a) et (3.3.19b). L'image de la position du robot a été prise à chaque période d'échantillonnage $T'_e = 0.4s$. Les résultats sont obtenus en utilisant le simulateur Matlab/TrueTime. La Fig. 3.24 montre l'évolution de la vitesse linéaire, de la vitesse angulaire et aussi

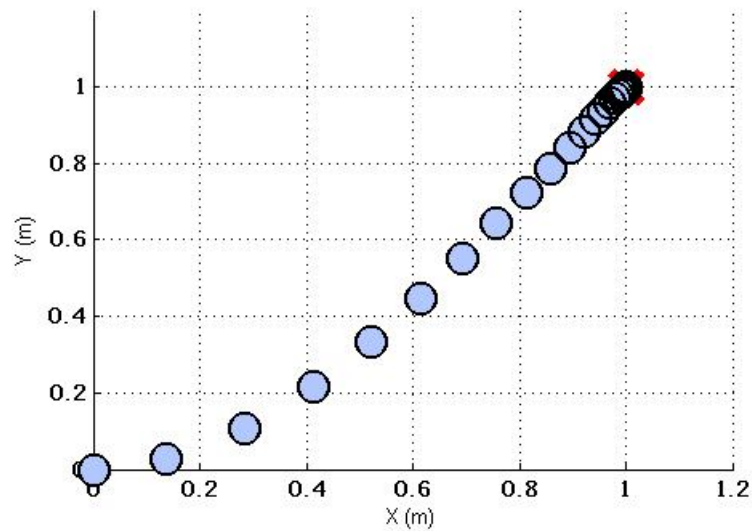


Figure 3.23 – Trajectoire du robot en simulation

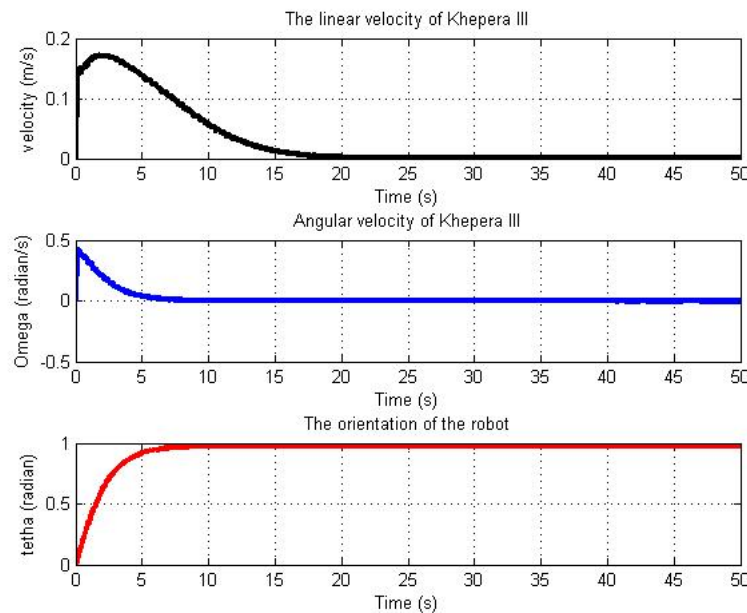


Figure 3.24 – Évolution des vitesses linéaire, angulaire et de l'orientation du robot

l'orientation du robot tout au long de sa trajectoire. On remarque que le robot a rejoint la cible désirée au bout de 20s.

En implémentant l'algorithme de commande embarqué sur le robot réel, nous avons pu comparer les résultats expérimentaux avec ceux de la simulation dans le cas de l'atteinte d'une cible. Dans ce cas, on considère que la position initiale du robot est $(x_0, y_0, \theta_0) = (0, 0, 0)$ et la cible $(x_d, y_d) = (0.5, 0.5)$ (Fig. 3.25). On remarque que

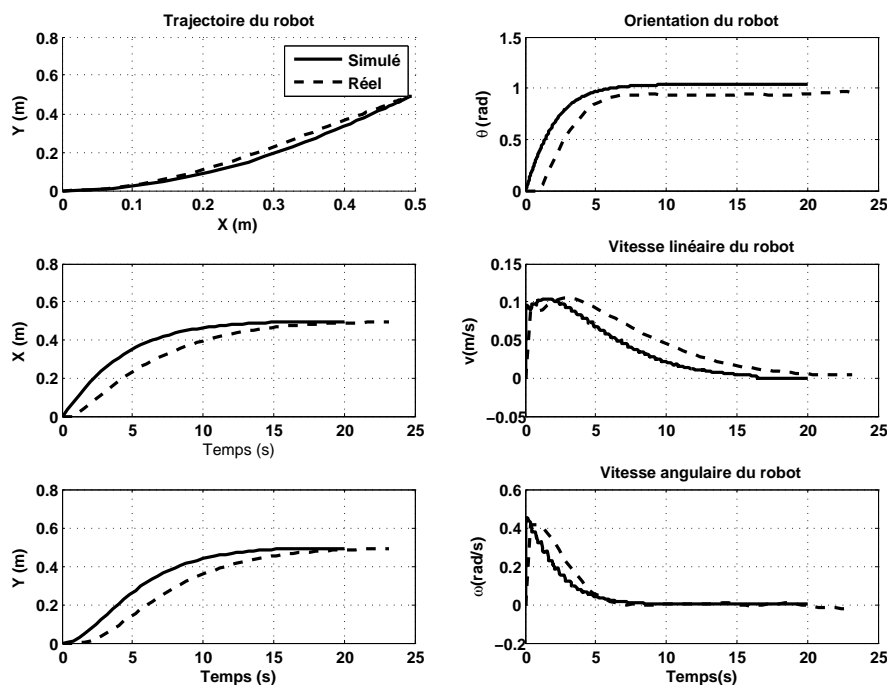


Figure 3.25 – Comparaison entre le robot en simulation et celui en expérimentation

la trajectoire simulée et la trajectoire réelle sont très proches. La différence est due essentiellement aux frottements et aussi aux erreurs d'odométrie.

Suivi de trajectoire

Dans ce qui suit, nous avons appliqué une consigne qui est une trajectoire quelconque définie par un ensemble de points. Les résultats de simulation sont montrés sur les Fig.3.26 et Fig.3.27.

La Fig.3.26 montre la trajectoire de référence appliquée et aussi la trajectoire simulée du robot mobile avec une position initiale $(x_0, y_0, \theta_0) = (0, 10, 0)$. x , y et θ sont mesurés à chaque période d'échantillonnage $T'_e = 0.4s$. La trajectoire est connue à l'avance et les consignes x_d , y_d et θ_d sont mises à jour chaque période.

La Fig.3.27 montre l'évolution de la vitesse linéaire, de la vitesse angulaire et aussi de l'orientation du robot tout au long de sa trajectoire. On remarque que le robot a rejoint la trajectoire de référence et qu'ensuite, il l'a suivie. En fonction de la période de mise à jour de la consigne, la vitesse linéaire varie pour atteindre le plus vite possible la consigne. Vu que cette période est calculée d'avance pour que le robot roule avec une vitesse moyenne de $0.18m/s$, on remarque qu'à partir de $40s$, le robot garde une vitesse fixe. En revanche, de 0 à $40s$, le robot a accéléré pour atteindre la trajectoire à suivre.

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

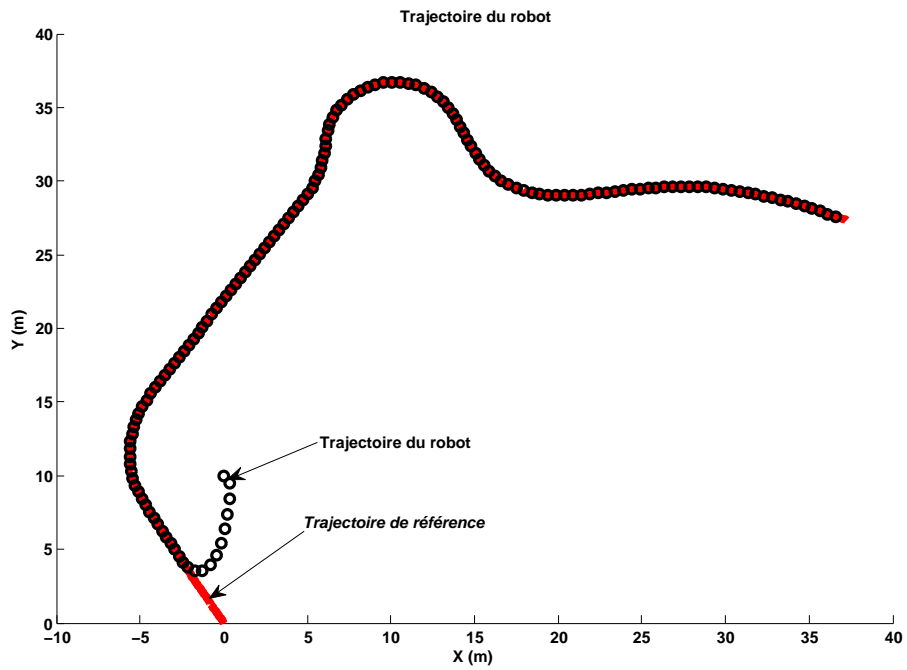


Figure 3.26 – Trajectoire du robot en simulation

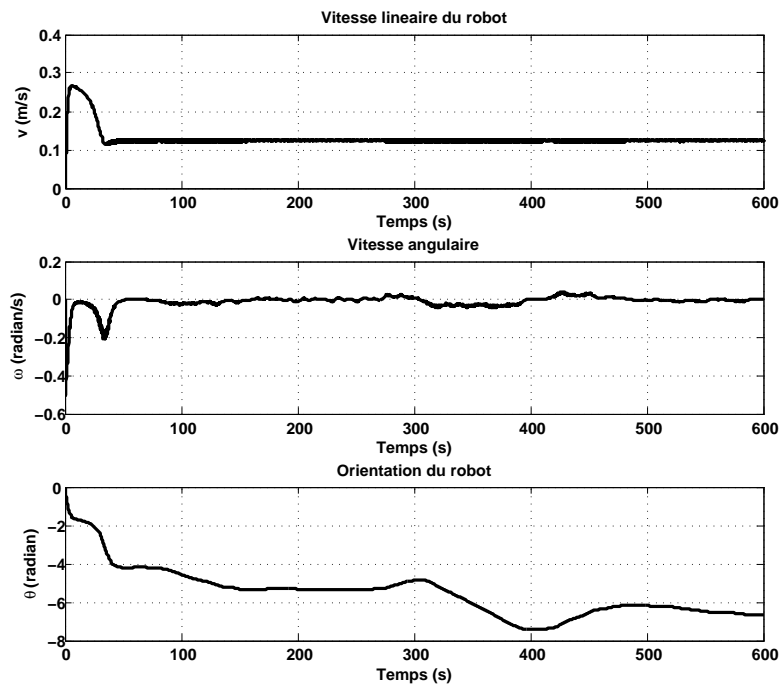


Figure 3.27 – Évolution des vitesses linéaire, angulaire et de l'orientation du robot en simulation

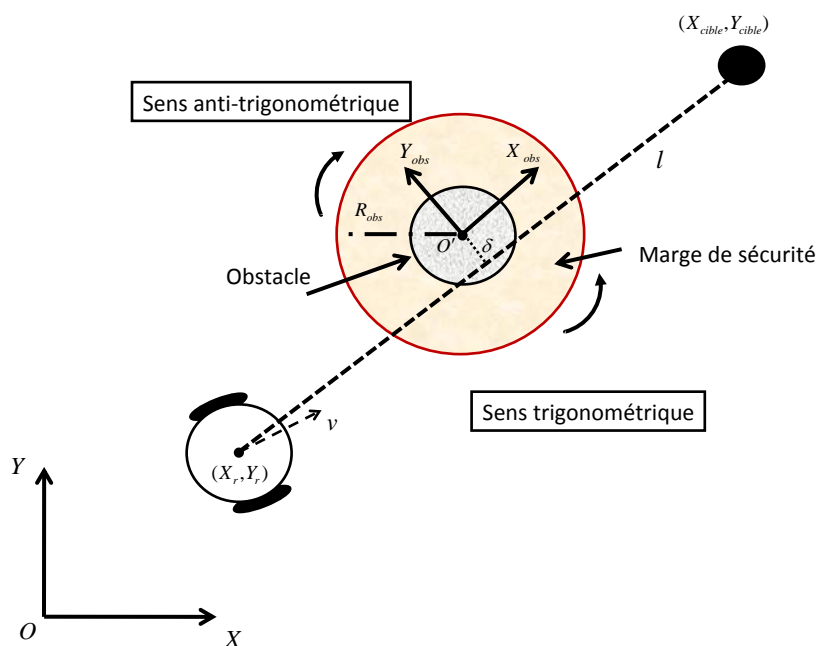


Figure 3.28 – Evitement d'obstacles

Évitement d'obstacles

Pour éviter un ou plusieurs obstacles, l'algorithme de cycle limite proposé dans [Hassan, 1996] et adapté dans [Kim & Kim, 2003] pour les robots unicycles a été utilisé. On considère que le robot doit atteindre une cible avec comme coordonnées (X_{cible}, Y_{cible}) . On considère aussi qu'il y a un obstacle circulaire avec les coordonnées (X_{obs}, Y_{obs}) et un rayon de R_{sec} avec :

$$R_{sec} = R_{obs} + D_{sec} \quad (3.5.1)$$

R_{obs} est le rayon réel de l'obstacle circulaire et D_{sec} représente la marge de sécurité (D_{sec} est égal au rayon du robot ($65mm$) augmentée d'une constante, dans notre expérimentation $100mm$).

En fonction des données fournies par les différents capteurs du robot, la décision de commuter entre le contrôle pour atteindre une cible et celui pour l'évitement d'obstacle sera prise. Si $D_{robot-obstacle}$ (distance entre les centres de gravité du robot et de l'obstacle) est inférieure à R_{sec} le robot décide de commuter son contrôleur vers celui d'évitement d'obstacles (Fig. 3.29 et Fig. 3.30). Les différentes étapes de l'algorithme d'évitement d'obstacles sont :

1. la droite l (la droite qui relie le robot à la cible Fig.3.28) est représentée par

Chapitre 3. Architecture de commande via un réseau d'un robot unicycle autonome

l'équation (3.5.2) :

$$ax + by + c = 0. \quad (3.5.2)$$

2. la distance δ entre le centre de l'obstacle et la droite l (Fig.3.28) est calculée en utilisant la formule 3.5.3.

$$\delta = \frac{a * X_{obs} + b * Y_{obs} + c}{\sqrt{a^2 + b^2}}. \quad (3.5.3)$$

Démonstration. le vecteur \vec{n} de coordonnée (a, b) est un vecteur normal de l , H est le projeté orthogonal de $O'(X_{obs}, Y_{obs})$ sur la droite l . Pour tout point M du plan on a : $\vec{n} \begin{pmatrix} a \\ b \end{pmatrix} * \overrightarrow{O'M} \begin{pmatrix} x - X_{obs} \\ y - Y_{obs} \end{pmatrix} = \vec{n} * \overrightarrow{O'H} \Leftrightarrow$

$$a(x - X_{obs}) + b(y - Y_{obs}) = \|\vec{n}\| * \|\overrightarrow{O'H}\| \cos(\vec{n}; \overrightarrow{O'H}) \Leftrightarrow$$

$$ax + by - aX_{obs} - bY_{obs} = \sqrt{a^2 + b^2} O'H \cos(\vec{n}; \overrightarrow{O'H}) \text{ or } M(x, y) \in l \Rightarrow c = -ax - by$$

$$\text{donc } -aX_{obs} - bY_{obs} - c = \sqrt{a^2 + b^2} O'H \cos(\vec{n}; \overrightarrow{O'H})$$

$$\text{d'où } |aX_{obs} + bY_{obs} + c| = \sqrt{a^2 + b^2} O'H$$

$$\Rightarrow O'H = \frac{|aX_{obs} + bY_{obs} + c|}{\sqrt{a^2 + b^2}} \quad \square$$

3. La direction désirée du robot unicycle est calculée chaque instant en utilisant les équations (3.5.4) et (3.5.5).

$$\dot{x} = \frac{\delta}{|\delta|} y + x(R_{sec} - x^2 - y^2) \quad (3.5.4)$$

$$\dot{y} = -\frac{\delta}{|\delta|} x + y(R_{sec} - x^2 - y^2) \quad (3.5.5)$$

x et y sont les coordonnées du robot dans le repère relatif $(O', \vec{X}_{obs}, \vec{Y}_{obs})$ lié à l'obstacle. Si δ est positif, le robot évite l'obstacle selon la direction qui minimise la distance à

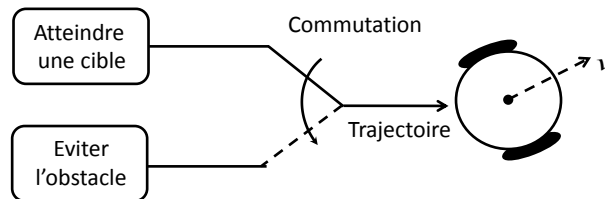


Figure 3.29 – Architecture de commande hybride d'atteinte d'une cible avec évitement d'obstacle

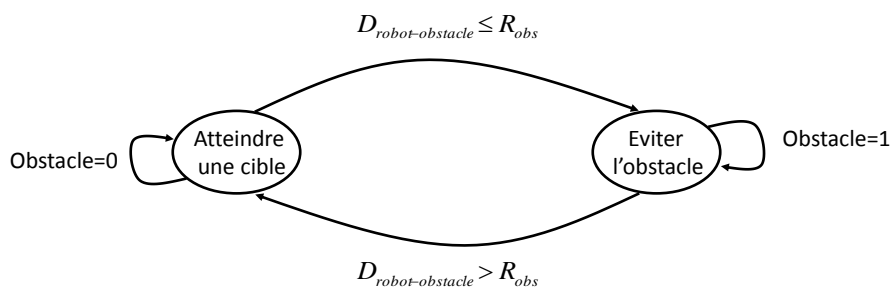


Figure 3.30 – Stratégie de commutation de contrôleur pour l'évitement d'obstacles

parcourir (sens trigonométrique), sinon il l'évite selon la direction anti-trigonométrique. Le contrôleur utilisé dans le cas de l'évitement d'obstacles est :

$$v = K_v \tag{3.5.6a}$$

$$\omega = K_\omega v \tilde{\theta}_{obs} \tag{3.5.6b}$$

$K_v = 0.1m/s$ et $K_\omega = 1$ sont des constantes.

$\tilde{\theta}_{obs}$ est calculée suivant l'équation 3.5.7.

$$\tilde{\theta}_{obs} = \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \tag{3.5.7}$$

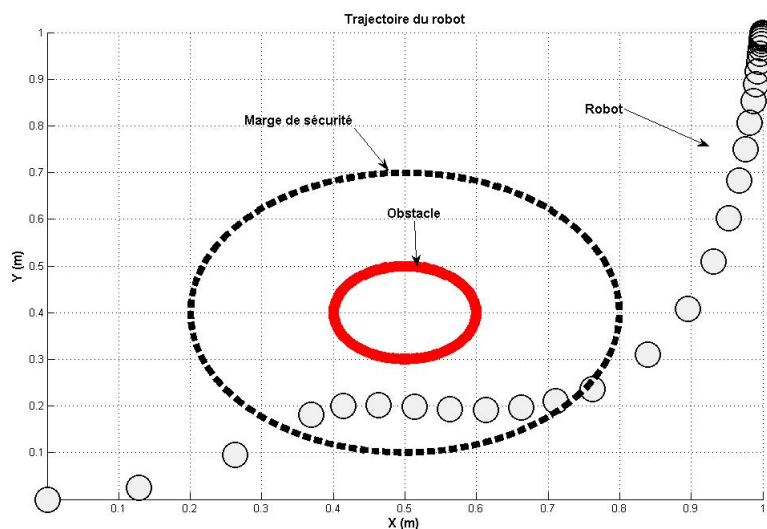


Figure 3.31 – Trajectoires du robot pour atteindre une cible avec un seul obstacle

Sur la Fig.3.31, on remarque que le robot a évité l'obstacle circulaire tout en dépassant un peu la marge de sécurité d'où l'intérêt de celle-ci. Sur cette même figure, on constate que le robot a choisi la direction optimale (sens trigonométrique) pour rejoindre la cible

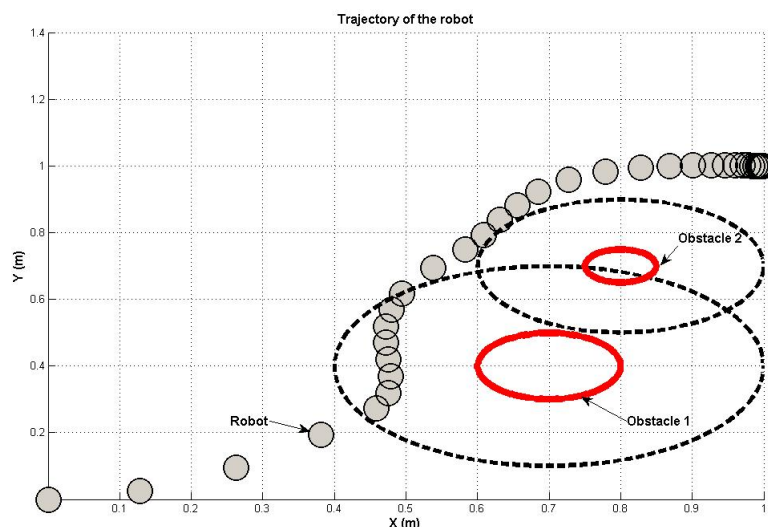


Figure 3.32 – Trajectoires du robot pour atteindre une cible avec deux obstacles

de coordonnées $(X_{cible}, Y_{cible}) = (1, 1)$. La fig.3.32 montre un évitement de deux obstacles consécutifs. Le robot a évité le premier obstacle dans le sens anti-trigonométrique (optimal dans ce cas). Dès qu'il a évité le premier obstacle, il a détecté la présence d'un autre et il l'a évité en commutant une autre fois son contrôleur vers un contrôleur d'évitement d'obstacles.

3.6 Conclusion

Dans ce chapitre, une étude du robot Khepera III a été faite d'une façon exhaustive pour la partie mécanique, électronique et informatique. Une architecture de commande a été proposée et un simulateur, utilisant Matlab/TrueTime, a été réalisé pour simuler le comportement du robot dans le cas de l'atteinte d'une cible et d'un suivi de la trajectoire. De plus, un algorithme d'évitement d'obstacles a été utilisé pour éviter un ou plusieurs obstacles statiques ou dynamiques.

Ces différents scénarios (atteinte d'une cible, suivi de trajectoires et évitement d'obstacles) peuvent être vu comme autant de missions que le système peut accomplir, avec des contraintes de QoC variables, et des niveaux de risques différents, ceci sera discuté dans le chapitre 5. Les résultats de la simulation ont été comparés avec ceux de l'expérimentation et ont été validés. Les écarts sont faibles.

Dans le prochain chapitre, un réseau sans fil sera intégré entre le capteur et le contrôleur d'une part et entre le contrôleur et les actionneurs d'autre part. L'influence du réseau sur la performance de la boucle de commande sera étudiée.

Chapitre 4

Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

Sommaire

| | | |
|------------|---|------------|
| 4.1 | Introduction | 80 |
| 4.2 | Analyse de performance du réseau sans fil Wi-Fi fonctionnant sous DCF | 81 |
| 4.2.1 | Fonctionnement du DCF (Distributed Coordination Function) | 82 |
| 4.3 | Evaluation des performances avec un réseau saturé | 83 |
| 4.3.1 | Analyse du débit | 83 |
| 4.3.2 | Retard d'un paquet avec le protocole IEEE 802.11 | 87 |
| 4.4 | Adaptation du réseau aux contraintes dynamiques de la commande | 88 |
| 4.4.1 | Handoff | 89 |
| 4.4.2 | Types de Handoff | 89 |
| 4.4.3 | Aide à la décision pour déclencher le Handoff | 90 |
| 4.5 | Adaptation de la commande aux contraintes dynamiques du réseau | 96 |
| 4.6 | Paramètres de QoS | 97 |
| 4.6.1 | Débit (<i>Throughput</i>) | 98 |
| 4.6.2 | Retard (<i>Delay</i>) | 98 |
| 4.6.3 | Gigues (<i>Jitter</i>) | 100 |
| 4.6.4 | Pertes de paquets (<i>Packet Loss</i>) | 100 |
| 4.7 | Étude de l'effet de la dégradation de la QoS du réseau sur la QoC du robot | 101 |
| 4.7.1 | Effet de la charge du réseau sur la QoC du robot | 102 |

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

| | | |
|------------|--|------------|
| 4.7.2 | Effet des pertes de paquet sur la QoC du robot | 110 |
| 4.8 | Environnement multi-robots communicants | 113 |
| 4.8.1 | Architecture de communication | 113 |
| 4.9 | Conclusion | 114 |

4.1 Introduction

Les systèmes commandés en réseaux (NCS) dans la robotique mobile sont devenus très populaires ces dernières années vu l'importance de distribuer les tâches. Avec l'évolution que le domaine des technologies de la communication connaît, spécialement dans les réseaux sans fil, la commande distribuée et aussi l'aide à la décision sont indispensables pour éviter d'embarquer le contrôleur, l'intelligence, le diagnostic, etc. Comme précisé dans le chapitre précédent, distribuer les tâches est essentiel surtout pour réduire la consommation d'énergie des robots notamment les robots de petite taille dont le volume et le poids représentent les atouts capitaux. Cependant, intégrer un réseau, que ce soit sans fil ou filaire, dans la boucle de commande introduit quelques problèmes tels que : une bande passante limitée, un retard et éventuellement des pertes de paquets [Hespanha et al., 2007]. De plus, les réseaux sans fil ajoutent d'autres problèmes liés à la mobilité et aussi à la sensibilité à l'environnement. La mobilité change la distance entre le robot et le contrôleur ce qui peut augmenter le pourcentage de pertes de paquets à cause de l'atténuation du signal reçu et aussi la probabilité d'avoir des erreurs dans la trame transmise (BER) [Zhu et al., 2005].

Dans la robotique mobile, l'architecture de communication peut être centralisée (dans ce cas, il y a un nœud qui communique avec tous les autres nœuds), ou bien décentralisée (dans ce cas le nœud mobile peut opérer de manière autonome) [Schwager et al., 2007]. Dans l'architecture décentralisée, chaque sous-système traite une partie du problème et partage la mémoire sans avoir une vue globale sur la mission. Dans la commande distribuée, la communication est un paramètre important et les sous-systèmes n'ont pas besoin de partager la mémoire [Martinez et al., 2007]. Dans les travaux liés à ces problèmes, beaucoup d'approches ont été proposées. Dans [Fierro & Lewis, 1996] le modèle dynamique du robot mobile est contrôlé par des réseaux de neurones. Dans [Aicardi et al., 1995] et [de Wit & Sordalen, 1992], un contrôleur non-linéaire a été proposé et étudié. Une architecture de contrôle hybride pour une navigation autonome des robots est étudiée dans [Benzerrouk et al., 2008].

L'objectif de ce chapitre est de définir une architecture de communication en vue de la commande et du diagnostic distribués d'un véhicule mobile se déplaçant dans un espace où existent plusieurs stations au sol (infrastructure). En fonction de la position du mobile, il pourra être en regard d'une ou plusieurs stations, ou d'aucune (voir Fig. 1.2).

- Lorsqu'il ne sera plus en face d'une station, le mobile devra être complètement

autonome (contrôleur embarqué).

- Lorsqu'il sera face à une ou plusieurs stations, le mobile doit choisir la station (Horizontal Handoff (HHO) [Wang et al., 2007]) qui lui garantit un niveau de Qualité de Service (QoS) suffisant pour une bonne qualité de Contrôle (QoC) (contrôleur distribué).
- Les transitoires (passage d'une station à une autre) devront être pris en compte.

Nous allons ici rappeler la définition de la QoC et la QoS données en introduction. La QoC concerne *la capacité de garantir la stabilité, la performance et la robustesse d'un système*. Elle dépend essentiellement de :

- la QoS du réseau
- l'algorithme de commande
- la période d'échantillonnage.

En revanche la QoS désigne *la capacité de fournir un support de communication conforme aux exigences en matière de temps, de bande passante et aussi de garantir un niveau acceptable de pertes de paquets défini pour un usage défini*.

Dans la littérature, des stratégies de HHO ont été proposées surtout pour les téléphones mobiles. Elles sont essentiellement des stratégies d'aide à la décision qui prennent en compte le signal reçu (Received Signal Strength (RSS)), la consommation d'énergie et le coût de la communication pour calculer une fonction de coût [Chen et al., 2004]. Dans notre cas, on propose d'ajouter le paramètre perte de paquets qui est causé par la propagation du retard et/ou la distance entre le robot et la station qui a des conséquences sur la QoC du système mobile.

Ce chapitre est organisé de la façon suivante. La deuxième section présente l'analyse de performance du réseau sans fil utilisé (IEEE 802.11) (plus de détails sur les réseaux, les protocoles et les technologies sont donnés dans l'annexe B). Dans la troisième section, nous présentons une approche pour résoudre le problème de co-conception dans le cas de notre application qui permet de trouver un compromis entre le réseau et la commande. La section suivante traite de façon exhaustive les effets de la charge du réseau sur la QoC du robot ainsi que l'effet des pertes de paquets. Enfin, une conclusion termine ce chapitre.

4.2 Analyse de performance du réseau sans fil Wi-Fi fonctionnant sous DCF

L'analyse de performance du réseau étudié est primordiale pour comprendre et analyser le comportement du système commandé à travers un réseau. Pour cela, cette section présente une analyse du protocole IEEE 802.11 fonctionnant sous DCF (Distributed Coordination Function) en mode CSMA/CA (Carrier Sense Multiple Acces and Collision Avoidance). En fait, la norme 802.11 propose deux mode d'accès au médium à savoir le

DCF et le PCF (Point Coordination Function). Le mode PCF est destiné au application temps réel - comme notre cas - mais il est peu répandu et la carte Wi-Fi du robot Khepera III utilisé ne prévoit pas ce mode d'accès. En conséquence, nous sommes mené à utiliser le mode DCF qui ne prévoit pas d'amélioration au niveau de la QoS (voir Annexe B).

4.2.1 Fonctionnement du DCF (Distributed Coordination Function)

Cette section résume brièvement les DCF normalisés par le protocole 802.11. Pour une présentation plus complète et détaillée, se référer à l'annexe B.

Une station qui veut transmettre un paquet surveille l'activité du canal. Si le canal est inactif pendant une période égale à un intervalle inter-distribué (DIFS), la station transmet. Sinon, si le canal est détecté occupé (soit immédiatement, soit au cours de la DIFS), la station continue à surveiller le canal jusqu'à ce qu'elle le détecte libre pendant toute la DIFS. A ce stade, la station génère un temps de backoff aléatoire (temps aléatoire) avant de transmettre (c'est la fonction d'évitement des collisions du protocole), pour minimiser la probabilité de collision avec les paquets transmis par d'autres stations. En outre, pour éviter le monopole du canal, une station doit attendre un temps backoff aléatoire entre deux transmissions consécutives.

Pour des raisons d'efficacité, DCF utilise une échelle backoff à temps discret. Le temps qui suit immédiatement un DIFS libre est discrétisé par tranche de temps, et une station est autorisée à transmettre seulement au début de chaque tranche horaire (Slot Time). La taille des tranches de temps est égale au temps nécessaire à une station pour détecter la transmission d'un paquet de n'importe quelle autre station. Elle dépend de la couche physique du réseau.

Le mode d'accès au médium DCF adopte un régime de backoff exponentiel. A chaque transmission, le temps de backoff est uniformément choisi dans l'intervalle $(0, W - 1)$. La valeur W est appelée fenêtre de contention, et dépend du nombre de transmissions échouées par paquet. Lors de la première tentative de transmission, W est égale à une valeur appelée fenêtre de contention minimale CW_{min} . Après chaque transmission échouée, W est doublé, jusqu'à une valeur maximale $CW_{max} = 2^m CW_{min}$, ensuite le paquet est rejeté et ne sera pas transmis.

Le CSMA/CA, comme son nom l'indique, ne propose pas un mécanisme de détection de collision, un ACK (ACKnowledgement) est donc transmis par la station de destination pour signaler la réception de paquets avec succès. Le ACK est immédiatement transmis après réception du paquet, après une période de temps appelée l'espace inter-trames court (SIFS). Comme le SIFS plus le délai de propagation de la trame ACK sont infé-

rieurs à un DIFS, aucune autre station n'est capable de détecter le canal inactif pendant un DIFS jusqu'à la fin de l'envoi du ACK. Si la station émettrice ne reçoit pas le ACK dans le temps appelé $ACK_{Timeout}$, où si elle détecte la transmission d'un paquet différent sur le canal, elle reporte la transmission du paquet selon les règles de l'algorithme du backoff.

4.3 Evaluation des performances avec un réseau saturé

Cette section est consacrée à la "Charge du réseau". Il s'agit d'un indice de performance fondamental, il représente la charge maximale que le système peut supporter dans des conditions stables. Il est bien connu que plusieurs régimes d'accès aléatoires ont un comportement instable. En particulier, lorsque la charge offerte augmente, le débit augmente aussi jusqu'à une valeur maximale, appelée "débit maximum". Toutefois, de nouvelles augmentations de la charge offerte provoquent une diminution significative du débit du système. Il en résulte l'impossibilité en pratique d'utiliser le système d'accès aléatoire à son débit maximum pendant une longue période de temps. La formulation mathématique et l'interprétation de ce problème d'instabilité est l'objet d'un large débat dans [Bertsekas & Gallager, 1987]. En effet, le protocole 802.11 est connu pour donner lieu à une certaine forme d'instabilité (voir, par exemple, [Chhaya & Gupta, 1997], et [Bianchi et al., 1996]). Pour visualiser le comportement instable de 802.11, [Bianchi, 2000] a effectué des simulations dans lesquelles la charge offerte augmente linéairement avec le temps de simulation. Il a trouvé que le débit mesuré suit de près la charge mesurée offerte dans un premier temps de la simulation, alors qu'il sera en baisse asymptotiquement dans la seconde partie de la simulation.

4.3.1 Analyse du débit

La contribution de base de [Bianchi, 2000] est l'évaluation analytique du débit de saturation, dans l'hypothèse que le canal est idéal (pas de terminaux cachés). Dans cette analyse, le nombre de stations est supposé fixe, chacune ayant toujours un paquet disponible pour la transmission. En d'autres termes, des conditions de saturation ont été adoptées et la file d'attente de transmission de chaque station est supposée non vide.

L'analyse faite dans [Bianchi, 2000] se divise en deux parties distinctes. Tout d'abord, le comportement d'une seule station avec un modèle de Markov est étudié, la probabilité q que la station transmette un paquet dans une tranche de temps est obtenue : cette probabilité ne dépend pas du mécanisme d'accès employé.

Probabilité de transmission de paquet

Considérons un nombre fixe de stations qui veulent communiquer. Dans des conditions de saturation, chaque station a immédiatement un paquet disponible pour la transmission, à l'issue de chaque transmission réussie. De plus, tous les paquets étant consécutifs, chaque paquet doit attendre pendant un temps aléatoire backoff avant de transmettre. Soit le processus stochastique $b(t)$ qui représente le compteur du Backoff pour une station donnée. t et $t+1$ correspondent au début de deux tranches de temps consécutives, et le backoff de chaque station décrémente au début de chaque tranche horaire. Comme la valeur du backoff de chaque station dépend aussi de son historique de transmission (le nombre de retransmissions du même paquet), le processus stochastique $b(t)$ est donc non-Markovien.

Posons maintenant $W = CW_{min}$, m est appelé la valeur maximale de backoff telle que $CW_{max} = 2^m W$. Considérons enfin la notation $W_i = 2^i W$ telle que $i \in 0, \dots, m$ est appelé la valeur courante du backoff. Considérons aussi $s(t)$ un processus stochastique représentant la valeur courante du backoff $(0, \dots, m)$ de la station à l'instant " t ".

L'approximation clé du modèle de Bianchi est que, à chaque tentative de transmission, et quel que soit le nombre de retransmissions, chaque collision est associée à une probabilité constante et indépendante. Il est intuitif que les résultats sont plus précis lorsque " W " et " n " (Nombre de nœuds) sont plus grands. " p " est appelée la probabilité de collision vue par un paquet transmis sur le canal.

Une fois l'indépendance supposée, et " p " supposée constante, il est possible de modéliser le processus bidimensionnel $b(t), s(t)$ avec la chaîne de Markov à temps discret montrée à la Fig. 4.1. Dans cette chaîne de Markov, on a les probabilités suivantes :

$$\begin{cases} P\{i, k|i, k+1\} = 1 & k \in (0, W_i - 2) \\ P\{0, k|i, 0\} = \frac{1-p}{W_0} & k \in (0, W_0 - 1) \\ P\{i, k|i-1, 0\} = \frac{p}{W_i} & k \in (0, W_i - 1) \\ P\{m, k|m, 0\} = \frac{p}{W_m} & k \in (0, W_m - 1) \end{cases} \quad (4.3.1)$$

La première équation dans 4.3.1 tient compte du fait que, au début de chaque tranche horaire, le temps de backoff est décrémente. La seconde équation montre qu'un nouveau paquet qui suit un paquet transmis avec succès, commence avec une valeur du backoff de 0, et donc le backoff sera choisi dans $(0, W_0 - 1)$. Les autres cas modélisent le système après un échec de transmission. En particulier, comme pris en compte dans la troisième équation dans 4.3.1, lorsque se produit une transmission sans succès à la valeur du backoff $i - 1$, la valeur du backoff s'incrémente, et la valeur initiale du nouveau backoff est uniformément choisie dans $(0, W_i)$. Enfin, le quatrième cas prend en compte le fait qu'une fois que la valeur du backoff atteint la valeur m , le backoff ne sera plus augmenté

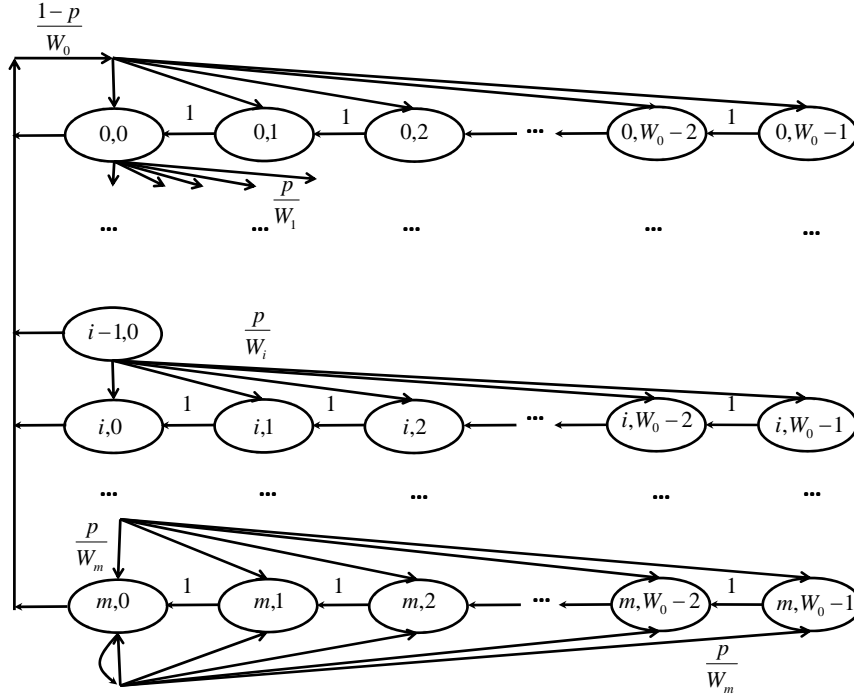


Figure 4.1 – Modèle de chaîne de Markov du Backoff

dans les transmissions de paquets ultérieures.

Considérons $b_{i,k} = \lim_{t \rightarrow \infty} P\{s(t) = i, b(t) = k\}$, $i \in (0, m)$, $k \in (0, W_i - 1)$ la distribution stationnaire de la chaîne. Dans [Bianchi, 2000] les auteurs ont montré qu'il est facile d'obtenir une solution analytique pour cette chaîne de Markov. On a :

$$\begin{aligned} b_{i-1} * p &= b_{i,0} \rightarrow b_{i,0} = k^i b_{0,0} \\ b_{m-1,0} * p^i &= (1-p)b_{m,0} \rightarrow b_{m,0} = \frac{p^m}{1-p} * b_{0,0} \end{aligned} \quad (4.3.2)$$

Grâce à la régularité de chaîne de Markov, pour chaque $k \in (1, W_i - 1)$, on a :

$$b_{i,k} = \frac{W_i - k}{W_i} * \begin{cases} (1-p) \sum_{j=0}^m b_{j,0} & i = 0 \\ pb_{i-1,0} & 0 < i < m \\ p(b_{m-1,0} + b_{m,0}) & i = m \end{cases} \quad (4.3.3)$$

En utilisant la relation 4.3.2, et en prenant en compte le fait que $\sum_{i=0}^m b_{i,0} = \frac{b_{0,0}}{1-p}$ de la relation 4.3.3, on déduit que :

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0} \quad i \in (0, m) \quad k \in (0, W_i - 1) \quad (4.3.4)$$

Ainsi, par les relations 4.3.2 et 4.3.4, toutes les valeurs de $b_{i,k}$ sont exprimées en fonction de la valeur $b_{0,0}$ et de la probabilité de collision p . $b_{0,0}$ est finalement déterminée par

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

l'utilisation de la condition de normalisation, qui la simplifie comme suit :

$$\begin{aligned}
 1 &= \sum_{i=0}^m \sum_{k=0}^{W_i-1} b_{i,k} = \sum_{i=0}^m b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} \sum_{i=0}^m b_{i,0} \frac{W_i + 1}{2} \\
 &= \frac{b_{0,0}}{2} \left[W \left(\sum_{i=0}^{m-1} (2p)^i + \frac{(2p)^m}{1-p} \right) + \frac{1}{1-p} \right]
 \end{aligned} \tag{4.3.5}$$

Ce qui donne :

$$b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{4.3.6}$$

Nous pouvons maintenant exprimer la probabilité q qu'une station transmette dans une tranche horaire choisie au hasard. Comme toute transmission se produit lorsque le temps de backoff est différent de zéro, quelle que soit la valeur courante du backoff :

$$q = \sum_{i=0}^m b_{i,0} = \frac{b_{0,0}}{1-p} = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{4.3.7}$$

Il est important de souligner que, lorsque $m = 0$, c'est-à-dire lorsqu'aucun backoff exponentiel n'est considéré, q est indépendant de p , et l'équation 4.3.7 se simplifie en la relation proposée par [Ho & Chen, 1996] de la fenêtre de backoff.

$$q = \frac{2}{W+1} \tag{4.3.8}$$

Cependant, en général, q dépend de la probabilité de collision p , qui est encore inconnue. Pour connaître la valeur de p , on peut noter que la probabilité qu'un paquet transmis rencontre une collision est la probabilité que, dans un intervalle de temps, au moins une des $n - 1$ stations restantes transmette. L'hypothèse d'indépendance fondamentale ci-dessus implique que chaque transmission voie le système dans le même état, c'est-à-dire, dans un état stable. Dans l'état stable, chaque station restante transmet un paquet avec une probabilité q . On obtient ainsi :

$$p = 1 - (1 - q)^{n-1} \tag{4.3.9}$$

Les équations 4.3.7 et 4.3.9 représentent un système non linéaire à deux inconnues q et p qui peut être résolu en utilisant des techniques numériques. Il est facile de prouver que ce système a une solution unique. En fait, en inversant l'équation 4.3.9, nous obtenons $q(p) = 1 - (1 - p)^{\frac{1}{n-1}}$. Il s'agit d'une fonction continue et monotone croissante pour tout $p \in (0, 1)$. L'équation 4.3.7 est aussi continue pour tout $p \in (0, 1)$, sauf le point critique $p = \frac{1}{2}$. Mais, en utilisant les propriétés des séries géométriques on peut simplifier l'équation 4.3.7 comme suit :

$$q = \frac{2}{(W+1) + pW \sum_i^{m-1} (2p)^i} \tag{4.3.10}$$

Débit

Bianchi dans [Bianchi et al., 1996; Bianchi, 2000], calcule le débit de saturation S qui est la charge maximale que le système peut supporter dans des conditions stables. S peut être trouvé en divisant la charge utile des informations transmises dans un intervalle de temps par la durée moyenne d'une tranche horaire :

$$S = \frac{P_{tr} * P_s * \varsigma}{M[slot]} = \frac{P_{tr} * P_s * \varsigma}{(1 - P_{tr})\sigma + P_{tr} * P_s * T_s + P_{tr}(1 - P_s)T_c} \quad (4.3.11)$$

où $M[slot]$ est la longueur moyenne d'une tranche horaire, σ est la durée d'un slot pendant lequel il n'y a pas de transmission, P_{tr} est la probabilité qu'il y ait au moins une transmission de paquets dans un intervalle de temps choisi au hasard. P_s est la probabilité qu'un paquet survenant de la transmission est réussie. ς est la longueur de la trame. T_c et T_s sont les durées moyennes pendant lesquelles le canal est occupé à cause d'une collision ou à cause d'une transmission réussie. Les probabilités P_{tr} et P_s sont données par :

$$\begin{aligned} P_{tr} &= 1 - (1 - q)^n \\ P_s &= \frac{n * q(1 - q)^{n-1}}{1 - (1 - q)^n} \end{aligned} \quad (4.3.12)$$

4.3.2 Retard d'un paquet avec le protocole IEEE 802.11

Basé sur l'étude faite dans [Bianchi et al., 1996], [Vu, 2006] a analysé le retard d'un paquet pour le protocole 802.11. Il a pu calculer le retard moyen appelé ici $M[D]$ pour une transmission réussie. Le retard d'un paquet est défini comme l'intervalle de temps entre le moment où le paquet est à la tête de sa file d'attente dans la couche MAC prêt pour la transmission, jusqu'à sa réception réussie à la destination. $M[D]$ est donné par :

$$M[D] = M[X] * M[slot] \quad (4.3.13)$$

où $M[X]$ est le nombre moyen d'un slot de temps pour une transmission de paquets avec succès. $M[X]$ peut être trouvé en multipliant le nombre de slots d_i où le paquet est retardé pour chaque étape du backoff par ρ_i qui est la probabilité de pouvoir parvenir à cette valeur de backoff. Plus précisément :

$$M[X] = d_i * \rho_i \quad (4.3.14)$$

$$d_i = \frac{W_i + 1}{2}, \quad i \in [0, m] \quad (4.3.15)$$

$$\rho_i = \begin{cases} p^i & i \in [0, m - 1] \\ \frac{p^m}{1-p} & i = m \end{cases} \quad (4.3.16)$$

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

Après calculs, $M[X]$ est donné par :

$$M[X] = \frac{(1 - 2p)(W + 1) + pW(1 - (2p)^m)}{2(1 - 2p)(1 - p)} \quad (4.3.17)$$

Ce résultat a été validé en utilisant le simulateur OpNet dans [Vu, 2006]. Il sera utile pour nous dans le chapitre 5 de ce rapport. La Fig.4.2 montre l'évaluation du retard en fonction de la probabilité de collision avec différentes valeurs de CW_{min} . Les valeurs de CW_{min} ont été choisies vu leurs utilisations fréquentes dans ce protocole. On remarque que plus la fenêtre de contention est petite plus la probabilité de collision est grande. Cela est dû au fait que la longueur de la fenêtre de contention influence la durée d'attente du nœud voulant émettre et donc influence directement la probabilité de collision. Et comme le retard est influencé directement par la longueur de la fenêtre de contention (retard d'attente) il augmente en augmentant la valeur de CW_{min} avec une probabilité de collision importante.

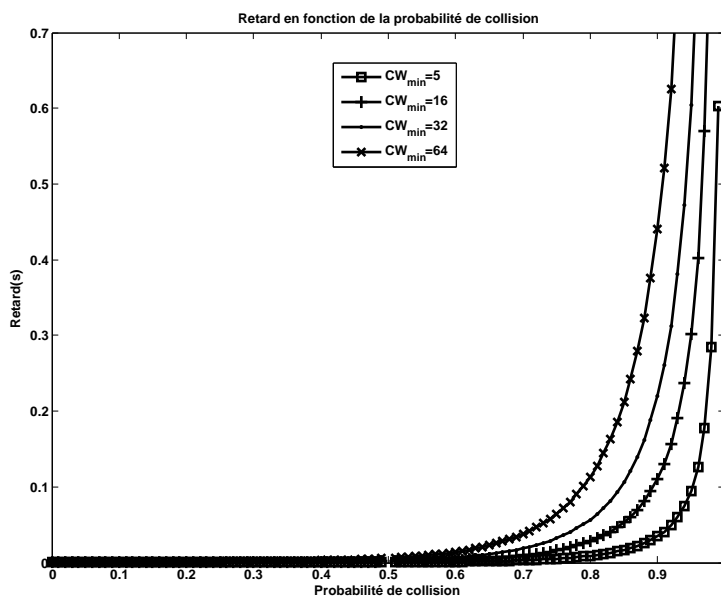


Figure 4.2 – Retard en fonction de la probabilité de collision et avec différentes valeurs de CW_{min}

4.4 Adaptation du réseau aux contraintes dynamiques de la commande

L'objectif de cette section est de reconfigurer le réseau pour assurer une bonne QoC au robot [Mechraoui et al., 2009]. Sous l'hypothèse de plusieurs stations disponibles, on

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

propose de changer de station (Handoff) lorsque la station courante ne peut pas assurer les besoins du robot, en terme de communication.

4.4.1 Handoff

Le Handoff représente l'ensemble des opérations permettant qu'un mobile puisse changer de station sans interruption de service [Tripathi et al., 1998]. Le processus consiste à ce qu'un terminal mobile maintienne la communication en cours, lors d'un déplacement qui amène le mobile à changer de cellule ou de station. En effet, lorsque le signal de transmission entre le mobile et une station de base s'affaiblit, le mobile doit trouver une autre station (ou cellule), qui est capable d'assurer à nouveau la communication. On peut dire aussi que c'est un mécanisme de sélection et/ou de resélection de stations ou de cellules et donc de gestion de localisation. La sélection de la station est nécessaire parce qu'elle permet d'enregistrer les informations diffusées par les stations au mobile et qu'elle permet aussi de signaler les mouvements des mobiles à la station. Les mobiles doivent être prêts à se connecter au réseau en cas de besoin de communication. Le mécanisme de Handoff a un grand impact sur la qualité de service (QoS) des réseaux sans fil et doit intégrer des éléments particuliers en relation avec la QoS :

- probabilité de rejet de communication,
- probabilité d'interruption de la communication,
- durée d'interruption.

Pourquoi changer de station ?

On peut résumer les causes qui incitent au changement de station en trois points principaux :

- maintien d'une qualité de communication suffisante entre le mobile et le réseau à travers un changement de fréquence ou de cellule,
- mesures physiques montrant qu'il existe une meilleure cellule ou station (conséquence de la mobilité),
- équilibrage du trafic,
- prédiction de futures positions.

4.4.2 Types de Handoff

Il existe plusieurs sortes de Handoff [Tripathi et al., 1998] :

Soft Handoff

Lorsqu'un nœud mobile se trouve dans une zone de couverture commune à deux stations de base, les communications du mobile empruntent simultanément deux canaux différents pour atteindre les deux stations de base (redondance) afin de ne pas perdre la communication lors du changement de station ; puis le nœud peut facilement couper la communication avec la première station de base (Fig. 4.3). Ce mécanisme est appelé *make-before-break*. Il est utilisé généralement avec les équipements à grande vitesse.



Figure 4.3 – Soft Handoff

Hard Handoff

Il permet à un nœud mobile de passer d'une station à une autre d'une façon directe, c'est à dire que le déclenchement du Handoff fait passer la communication d'une station à une autre sans garder la connexion avec la première station de base (Fig. 4.4). Ce mécanisme est appelé aussi *break-before-make*. Il est utilisé avec les équipements à petite vitesse. Le temps de déconnexion dépend d'un protocole de communication à un autre.

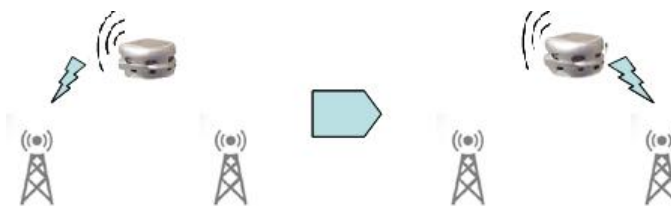


Figure 4.4 – Hard Handoff

4.4.3 Aide à la décision pour déclencher le Handoff

Dans un environnement multi-stations, la gestion de la mobilité est capitale pour assurer un niveau suffisant de QoS et donc une bonne QoC. Le but du Handoff est de maintenir la communication au moment où on change de couche physique. On distingue

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

deux types de Handoff, à savoir l'horizontal et le vertical. Le HandOff Horizontal (HHO) est un Handoff classique entre deux cellules homogènes de même niveau hiérarchique, tandis qu'un HandOff Vertical (VHO) se passe entre deux cellules hétérogènes. Les Handoffs Horizontal et Vertical sont proposés par plusieurs chercheurs dans le domaine des réseaux et télécommunications. [Chen et al., 2004] ont proposé un modèle de décision intelligent pour choisir la meilleure station du réseau et aussi le meilleur moment de commutation pour le Handoff Vertical. [Stevens-Navarro & Wong, 2006] ont fait une comparaison de la bande passante et du retard entre quatre algorithmes différents d'aide à la décision de VHO pour différentes classes de trafic. [Wang et al., 1999] ont présenté une architecture de Handoff qui est basée sur l'estimation des conditions actuelles du réseau et détermine le meilleur réseau à chaque période d'échantillonnage, en caractérisant le coût, la performance et aussi la consommation d'énergie. [Nkansa-Gyekye & Agbinya, 2007] ont proposé un mécanisme VHO distribué pour réduire les calculs dans le processeur du système. [Angermann & Kammann, 2002] a évalué différentes structures de fonctions de coût et a présenté des simulations traitant l'influence des différents paramètres.

Dans notre cas, une fonction de coût a été utilisée, cette fonction est appelée NRF ('Network Rating Function'). Cette dernière sera calculée pour chaque réseau dans le cas du VHO ou pour chaque point d'accès (AP) dans le cas du HHO.

Les paramètres qui sont critiques dans notre application et qui sont introduits pour calculer la NRF sont :

- Le Request Signal Strength Indicator (RSSI) ;
- Le retard (D) qui est dû au changement de station ou de point d'accès ;
- La charge du réseau (NL).

La fonction de coût NRF pour une station "j" est calculée de la façon suivante :

$$NRF_j = W_{RSSI}N_{RSSI,j} + W_{NL}N_{NL,j} + W_D N_{D,j} \quad (4.4.1)$$

avec :

$$W_{RSSI} + W_{NL} + W_D = 1 \quad (4.4.2)$$

où W_{RSSI} , W_{NL} et W_D sont respectivement les poids de la fonction de RSSI, de la charge du réseau (NL) et du retard (D). $N_{RSSI,j}$, $N_{NL,j}$ et $N_{D,j}$ sont fonction des paramètres agissant directement sur leurs valeurs. NRF_j est entre 0 et 1. W_i est le poids du paramètre $i \in \{RSSI, NL, D\}$, qui définit le degré d'importance de chaque paramètre et $N_{i,j}$ représente un taux normalisé de la station 'j' pour le paramètre 'i'.

Un algorithme embarqué sur le robot attribue les "poids" de chaque paramètre dans le but de déterminer l'importance de chaque paramètre. Cette décision de la valeur du poids sera prise en fonction des informations échangées entre robots d'une part et entre le robot et la station d'autre part.

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

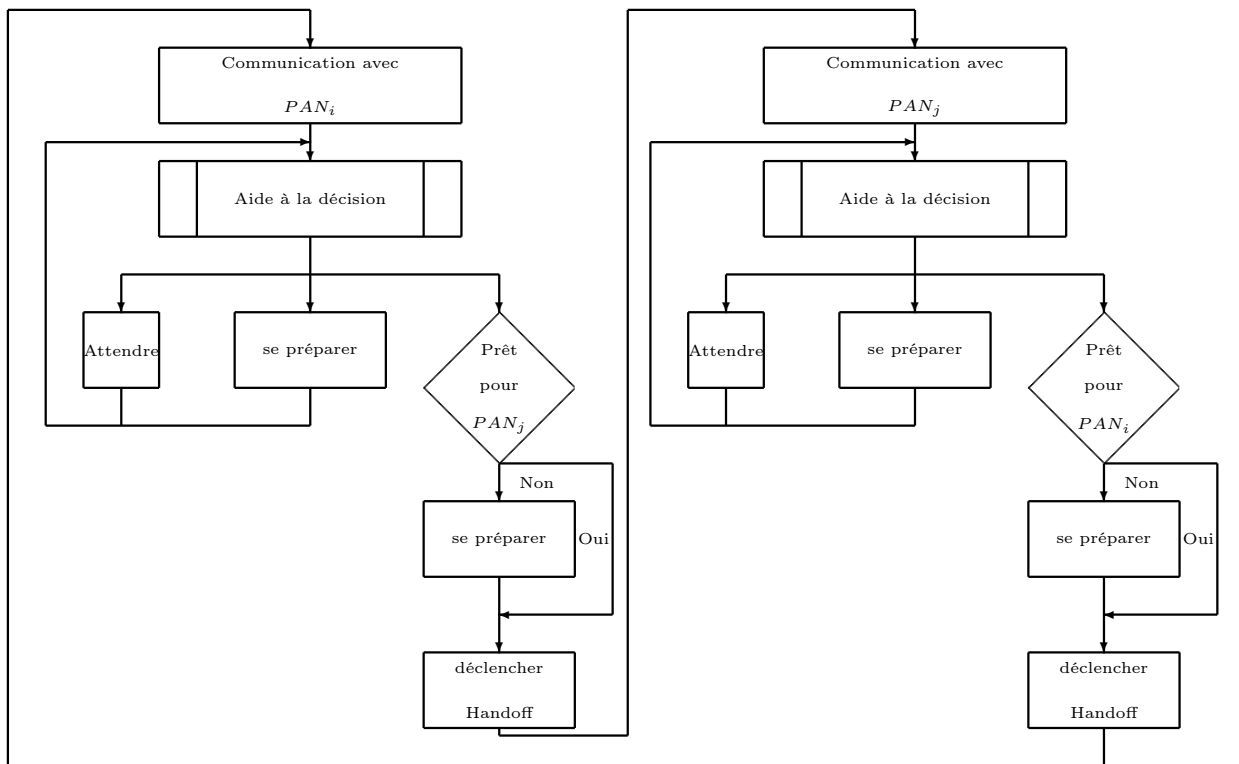


Figure 4.5 – Diagramme d'état du HHO

Dans notre cas, on considère $W_{RSSI} > W_{NL} > W_D$ ¹. Le robot choisit la meilleure station en calculant le NRF de la station courante et aussi ceux des stations candidates (voir Fig.4.5) à chaque période d'échantillonnage, ensuite il les compare. Cette comparaison présente trois cas à étudier :

1. Si le NRF de la station courante est supérieur à celui de la station candidate, le robot attend deux périodes d'échantillonnage (pour ne pas recalculer trop souvent) (bloc "attendre") et ensuite il recommence le calcul.
2. Si le NRF de la station courante est égal à celui de la station candidate, le robot recalcule le plus vite possible le prochain NRF (à la T'_e suivante (bloc "se préparer").
3. Si le NRF de la station courante est inférieur à celui de la station candidate, le robot demande une place à la station candidate qui a le plus grand NRF et il attend jusqu'à avoir l'acceptation ; ensuite il déclenche la procédure du Handoff (bloc "prêt pour PAN_j ").

¹Cette contrainte est justifiée par le fait que la diminution du RSSI influence beaucoup plus la QoS du réseau en comparant avec la charge du réseau et le retard généré par le déclenchement du handoff, de même, l'influence de la charge du réseau est plus importante que celle du retard généré par le Handoff.

Résultats de simulation

Des valeurs des poids ont été choisies en fonction de l'importance de chaque paramètre. Le calcul de NRF_i et NRF_j est lancé pour choisir la station qui maximise le critère de décision (Fig. 4.5).

Les fonctions $N_{RSSI,i}$, $N_{NL,i}$, et $N_{D,i}$ sont calculées en utilisant les équations (4.4.3).

$$N_{RSSI,i} = \frac{e^{P_r}}{e^{P_s}} \quad (4.4.3a)$$

$$N_{NL,i} = 1/e^{NL} \mid 0 \leq NL \leq 1 \quad (4.4.3b)$$

$$N_{D,i} = \begin{cases} 1 & \text{si } S_i \text{ est la station courante \& } RSSI > -48dBm \\ 0 & \text{sinon} \end{cases} \quad (4.4.3c)$$

Dans nos simulations, la puissance transmise est de $-3dBm$ et le seuil du signal reçu est de $-48dBm$. Le RSSI est calculé en utilisant l'équation suivante :

$$P_r = \frac{1}{d^\alpha} P_s \quad (4.4.4a)$$

$$RSSI(dBm) = 10 \log(P_r(mW)) \quad (4.4.4b)$$

P_r et P_s sont les puissances en mW reçue et transmise respectivement. d est la distance entre deux nœuds en m , et α est un paramètre pour modéliser les différents environnements.

La Fig.4.6 montre la trajectoire du robot avec un changement de station qui assure un certain niveau de QoS. Les positions des stations dans notre scénario sont :

- Pour la première station (S_1) (0, 5) (m)
- Pour la deuxième station (S_2) (-3, 30)
- Pour la troisième station (S_3) (40, 35).

Pour cette simulation, les valeurs des poids sont $W_{RSSI} = 0.8$, $W_{NL} = 0.15$ et $W_D = 0.05$.

On suppose que S_1 et S_2 ont une zone de couverture commune, et S_3 est un peu loin des deux autres (Fig. 4.6). Avec la position initiale du robot $(X_0, Y_0, \theta_0) = (0, 0, 0)$ le robot communique d'abord avec S_1 . Il continue sa trajectoire jusqu'à la détection de la deuxième station S_2 . Dans ce cas, le robot exécute l'algorithme 1.

Pour comprendre cet algorithme, des symboles ont été utilisés. \textcircled{R} est le robot mobile, S_i est la station courante, $S_{\{1,2,..k\}}$ sont toutes les stations et $S_{\{m,..,n\}}$ sont les stations candidates. On attribue le symbole \bowtie pour dire que le robot a détecté un ensemble de stations candidates.

Lorsque le robot est hors de portée de toutes les stations, il pourrait garder la dernière valeur de sa commande, mais ce n'est pas une bonne solution car elle engendre des erreurs

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

Algorithm 1 Decision algorithm

```

if  $\mathbb{R} \times \{S_m, \dots, S_n\} \mid m \in \{1, \dots, k\} \text{ and } n \in \{1, \dots, k\} \mid m \neq n$  then
  calculate  $NRF$  for all  $S_m, \dots, S_n$ 
  for all  $i$  and  $l \in \{m, \dots, n\}$  do
    if  $NRF_{S_i} < NRF_{S_l}$  then
       $i = l$ 
    else if  $NRF_{S_i} > NRF_{S_l}$  then
       $i = i$ 
      Wait
    else
       $i = i$ 
    end if
  end for
end if

```

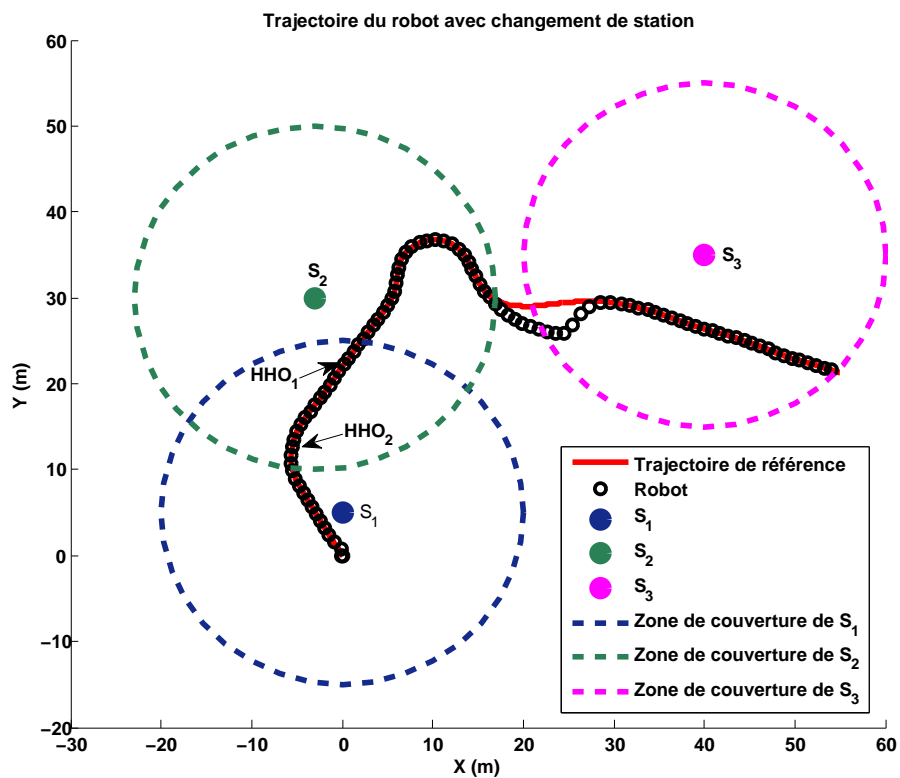


Figure 4.6 – Trajectoire du robot avec ($HHO_1 : NL_{S_1} = 0$) et ($HHO_2 : NL_{S_1} = 0.8$ et $NL_{S_2} = 0.2$)

par rapport à la trajectoire de référence (voir Fig.4.6). On propose dans ce cas de changer le contrôleur et d'utiliser un contrôleur embarqué dégradé qui sera décrit dans la section 4.5.

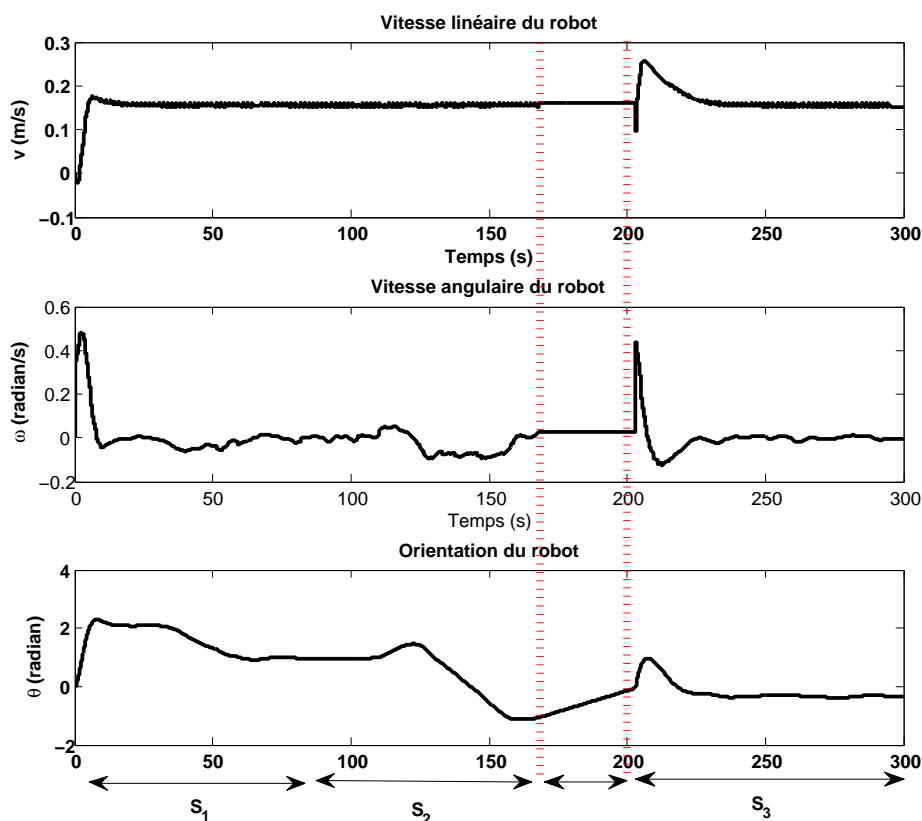


Figure 4.7 – Évolution de la communication avec différentes stations avec $NL=0$

Les figures Fig. 4.6 et Fig. 4.7 montrent la trajectoire et l'évolution des vitesses linéaire et angulaire du robot ainsi que son orientation avec $NL = 0$ pour toutes les stations. On peut aussi voir sur ces figures le moment de changement de station et la station courante le long de la trajectoire. La commutation de S_1 vers S_2 dépend de la distance entre le robot et la station et aussi du retard généré en passant d'une station à l'autre. Le robot change de station si et seulement si le RSSI de la station courante n'est pas suffisant.

Sur la Fig.4.8, la simulation a été faite avec $NL_{S_1} = 0.8$ qui veut dire que la station 1 est chargée à 80% et $NL_{S_2} = 0.2$. Dans ce scénario, le robot décide de changer de station suivant la valeur de RSSI des stations candidates ainsi que de leur charge. En comparant les Fig.4.7 et Fig.4.8, on voit que le déclenchement du HHO est sensible à la charge du réseau dans le but d'assurer une QoS suffisante et aussi pour diminuer la charge de la station courante : le robot commute beaucoup plus tôt sur la station " S_2 " pour assurer au robot une QoS suffisante au bon fonctionnement. Par contre, les vitesses et l'orientation ne diffèrent pas de celles de la Fig. 4.7 et donc la trajectoire est identique à celle de la Fig. 4.6.

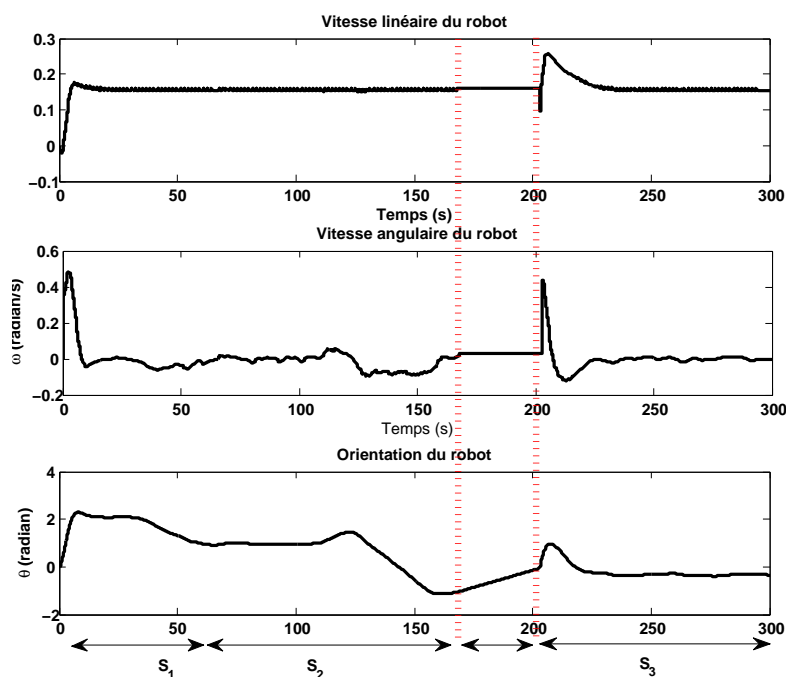


Figure 4.8 – Evolution de la communication avec différentes stations avec $NL_{S_1} = 0.8$ et $NL_{S_2} = 0.2$

4.5 Adaptation de la commande aux contraintes dynamiques du réseau

Dans le cas où le robot commence à s'éloigner de la station courante, le RSSI diminue. Dès que le RSSI atteint une valeur critique, le robot informe la station et demande de lui envoyer une partie de la trajectoire que la station estime suffisante pour arriver à la zone de couverture d'une autre station. Cette estimation se fait en prenant en compte la direction et la vitesse du robot. Donc, la station de contrôle envoie une partie de la trajectoire au robot et elle informe la ou les stations, dont elle estime qu'il est plus probable que le robot rentre dans leurs zones de couverture, de l'arrivée du robot et donc elle demande de lui réserver si possible une place pour qu'il puisse entrer directement en communication avec la station. Ceci a pour but de minimiser le temps de connection. Donc, un PAN slot (tranche de temps) pourrait être réservé en avance pour que le robot puisse communiquer le plus vite possible avec la première station trouvée. Cette proposition permet de maintenir une bonne QoC malgré la dégradation de la QoS.

Le contrôleur embarqué est un contrôleur dégradé avec une période d'échantillonnage la plus grande possible. Ce changement de période d'échantillonnage est très important

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

pour limiter la consommation d'énergie du robot et aussi le temps d'exécution du contrôleur. Les algorithmes 1 et 2 sont exécutés ("*poor*" dans l'algorithme 2 signifie que le RSSI de la station courante approche du seuil). La Fig.4.9 et la Fig.4.10 montrent la simulation en prenant en compte la commutation de contrôleur dans la zone non couverte. La commutation de contrôleur peut se déclencher dans la zone de couverture dans laquelle un certain niveau de QoS n'est pas assuré.

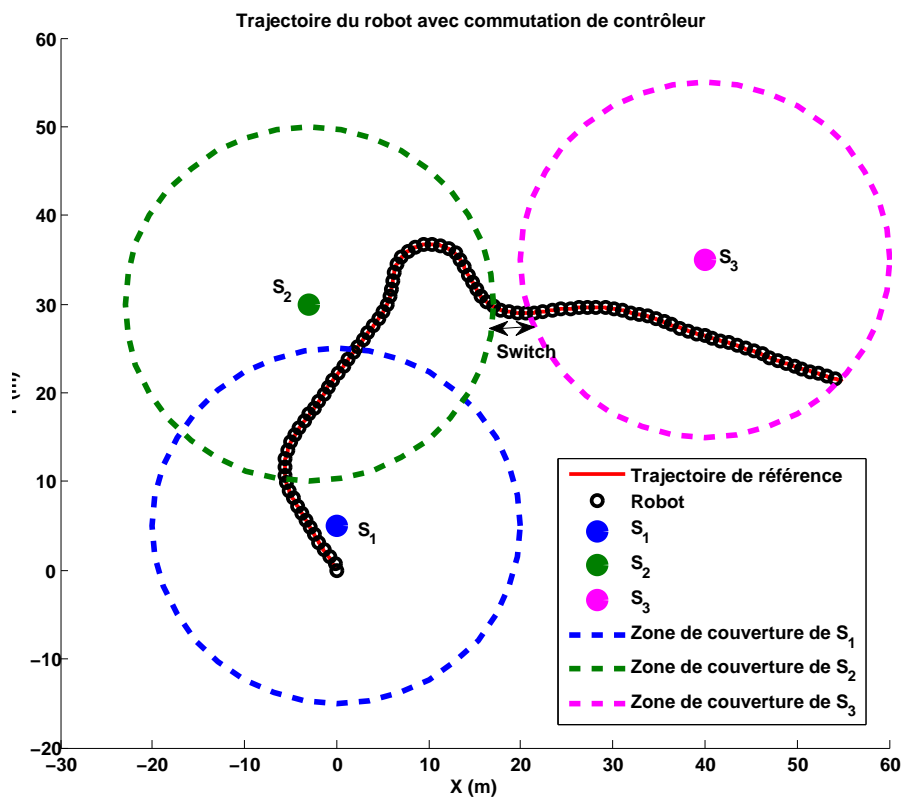


Figure 4.9 – Trajectoire du robot avec la commutation de contrôleur sur la partie non couverte

Algorithm 2 Decision algorithm

if $NR^F = poor$ and $\textcircled{R} \bowtie S_i$ then

 Switch controller

end if

4.6 Paramètres de QoS

Avant d'étudier l'influence de la QoS du réseau sur la QoC du mobile, les paramètres qui influencent la QoS doivent être étudiés d'une façon exhaustive. Les paramètres qui

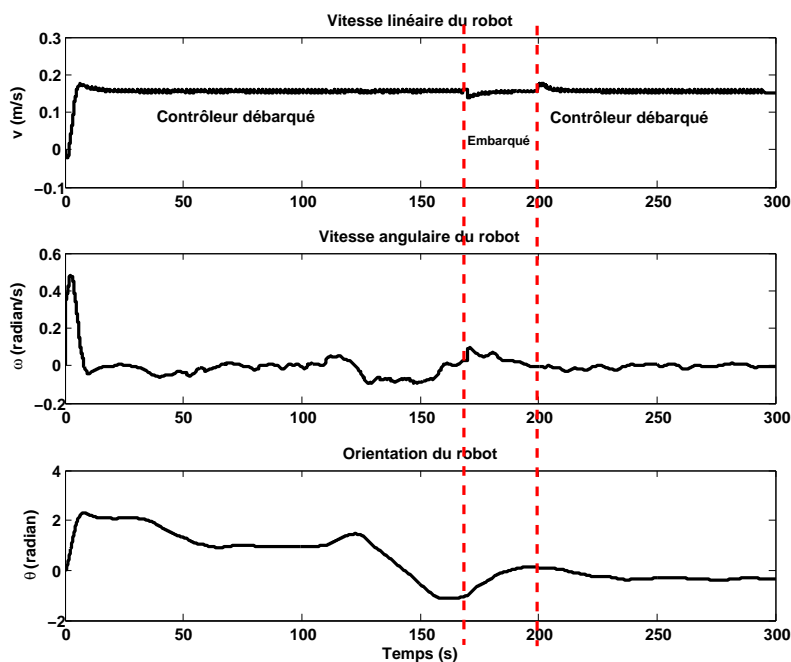


Figure 4.10 – Evolution des vitesse linéaire et angulaire et de l'orientation avec la commutation de contrôleur

influent la QoS sont :

1. Débit (*Throughput*) ou bande passante (*Bandwidth*) ;
2. Retard (*Delay*) ou latence ;
3. Variation du retard (*Delay jitter*) ;
4. Perte de paquets (*Packet loss*).

4.6.1 Débit (*Throughput*)

Le débit est mesuré par le nombre de bits par seconde, parfois appelé aussi bande passante (*Bandwidth*). La bande passante est considérée comme les ressources d'un réseau dédiées à une ou plusieurs applications.

Le débit demandé par une application dépend directement des besoins de cette application et aussi de la configuration des paramètres de l'application, à savoir la taille de la trame en bits et la stratégie d'ordonnancement.

4.6.2 Retard (*Delay*)

Le retard (appelé aussi retard bout en bout) (Fig. 4.11) a un impact direct sur la QoC spécialement lorsque la dynamique du système est rapide. Les applications temps

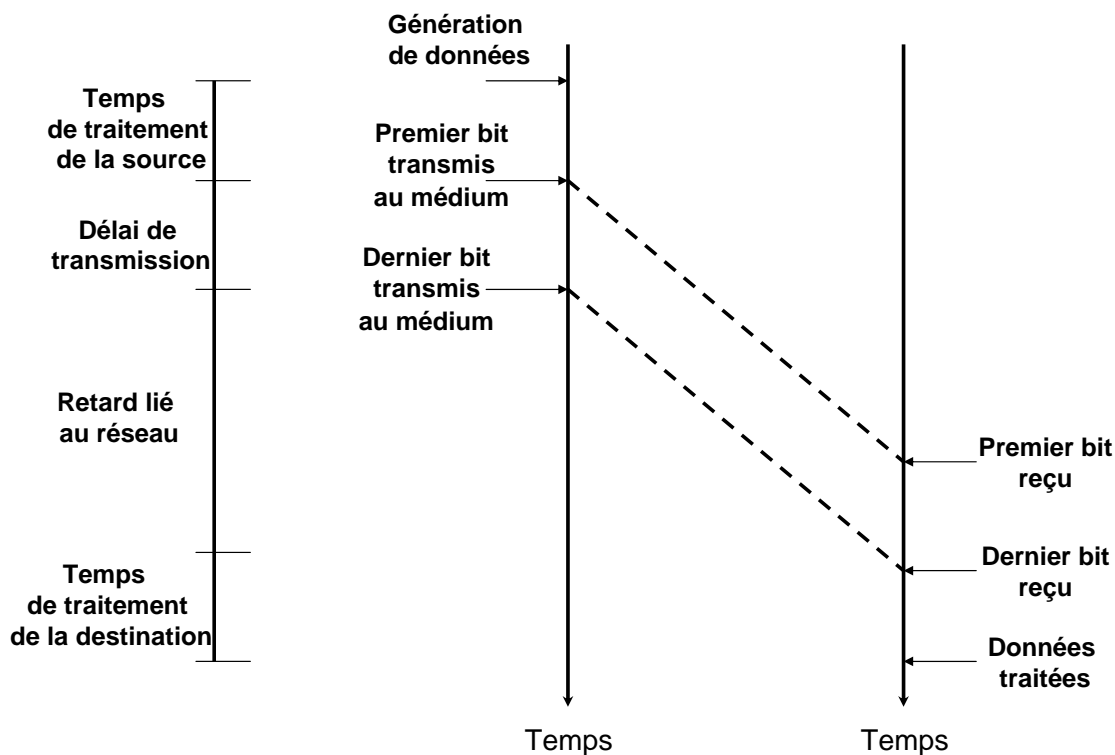


Figure 4.11 – Retard bout en bout

réel en général nécessitent la délivrance de l'information de la source à la destination en respectant les délais. Un grand retard peut causer des incidents ce qui influence directement la QoC durant les tâches avec dynamique rapide. Les principales causes du retard sont :

1. **Temps de traitement de la source** : Ce temps dépend des caractéristiques de la source en terme de processeur, de mémoire ...
2. **Retard de transmission (δ_{tr})** : il est fonction de la taille du paquet et aussi du débit du réseau (la vitesse du réseau). Ce retard est calculé en utilisant l'équation suivante :

$$\delta_{tr} = \frac{\Gamma}{\Upsilon} \quad (4.6.1)$$

Γ est la taille de la trame et Υ est le débit du réseau.

3. **Retard lié au réseau (*Network Delay*)** : Le retard dû au réseau est influencé par trois paramètres.
 - **Délai de propagation (δ_p)** : le délai de propagation d'une trame de la source vers une destination dépend de la distance physique entre la source et la destination. Ce délai est calculé en utilisant l'équation suivante :

$$\delta_p = \frac{d}{c} \quad (4.6.2)$$

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

d est la distance entre deux nœuds communicants et c est la vitesse de la lumière $c = 3 * 10^8 m/s$.

- **Délai lié aux protocoles utilisés** : Ce délai est causé principalement par les protocoles utilisés par les différents composants du réseau, à savoir les routeurs, gateways, et les cartes réseaux. Le retard est donc dû au protocole, à la charge du réseau, et aussi à la configuration du matériel qui exécute les protocoles.
- **Délai dû à la file d'attente** : Ce retard est causé par le temps que le paquet passe dans la file d'attente du composant du réseau (Fig. 4.12). Ce retard dépend essentiellement de la congestion du réseau, de la configuration du réseau et aussi de la vitesse du réseau.

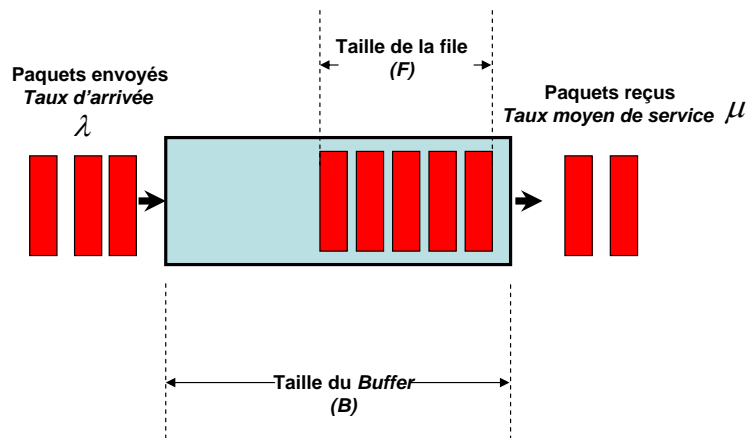


Figure 4.12 – Retard dû à la file d'attente (Buffer)

4. **Temps de traitement de la destination** : Ce temps dépend des caractéristiques matérielles (Rapidité, Mémoire ...).

4.6.3 Giges (*Jitter*)

La gigue désigne, dans le domaine des réseaux, la variation du retard introduite par le réseau (à cause de différents paramètres) tout au long de la communication. Si un paquet passe par plusieurs chemins, le retard bout à bout varie d'une façon remarquable en fonction du chemin pris (Fig. 4.13).

4.6.4 Pertes de paquets (*Packet Loss*)

Les pertes de paquets influencent d'une façon directe et notable la QoS du réseau. Elles compromettent l'intégrité des données ou interrompent totalement le service. Les

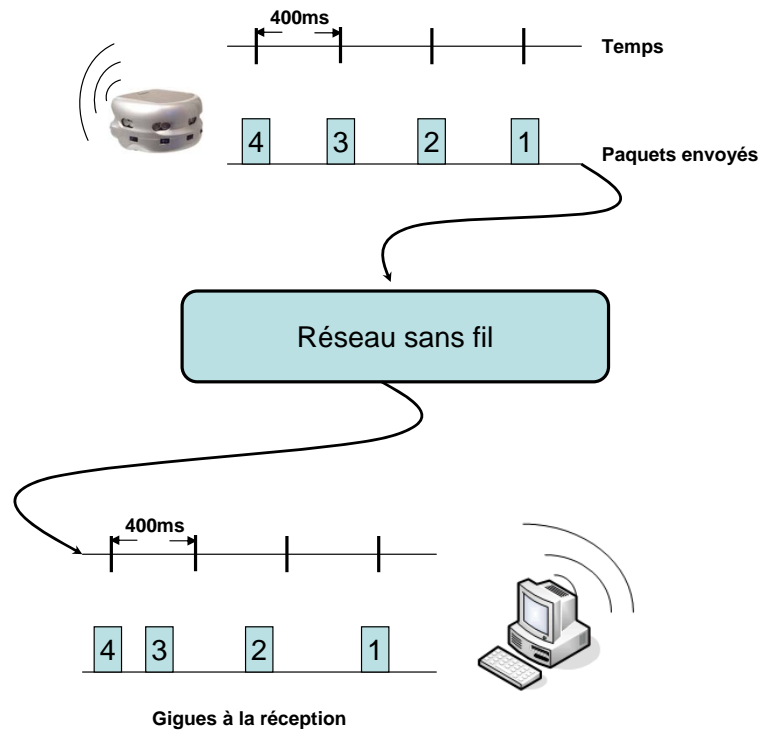


Figure 4.13 – Giges (*Jitter*)

perdes de paquets peuvent être influencées par la charge du réseau, le BER (*Bit Error Rate*) (ce type de pertes se produit dans les réseaux sans fil en présence de bruit sur le canal de communication), la distance entre les deux nœuds communicants ou bien la présence d'obstacles entre les deux nœuds. Pour essayer de résoudre ce problème, plusieurs méthodes ont été utilisées. La retransmission des données est la plus fréquente mais elle ne convient pas aux applications temps réel. La correction des données en cas d'erreurs au niveau de la couche physique est aussi utilisée dans le cas d'erreur de transmission.

4.7 Étude de l'effet de la dégradation de la QoS du réseau sur la QoC du robot

Cette section étudie les effets de la dégradation de la QoS du réseau sans fil sur la QoC du robot. Deux paramètres de la QoS sont pris en compte.

- Le premier paramètre concerne la charge du réseau qui influence le taux de pertes de paquets, et le retard ;
- le second paramètre concerne le taux de pertes de paquets.

4.7.1 Effet de la charge du réseau sur la QoC du robot

Étude du temps de transmission

Dans cette partie, nous nous intéressons à l'étude du délai de transmission qui est considéré dans notre cas comme un retard qui influence la QoC du robot. Le paramètre qui influence le délai de transmission sur un réseau est principalement la charge du réseau.

Le terme délai désigne le temps que met un paquet pour être acheminé d'une source vers une destination. Il est égal à la somme des temps sur les nœuds que compte ce chemin. Ce délai prend en considération deux délais : le délai dans la file d'attente des nœuds le long du chemin et le délai de transmission du paquet sur le médium. Il faut préciser ici qu'on s'intéresse seulement à la communication à un saut radio.

Le problème de synchronisation des horloges a été étudié dans la littérature. [Huang & Lai, 2002] et [Zhou et al., 2008] ont bien évalué le protocole de synchronisation des horloges (Timing Synchronization Function TSF) proposé par le standard IEEE 802.11 de base. Ils ont trouvé par analyse et aussi en simulation que, dans le cas où il y a un grand nombre de nœuds qui communiquent, la probabilité de désynchronisation est non-négligeable. Donc, plus il y a de stations, plus la probabilité d'asynchronisme est grande. Cependant, les méthodes de synchronisation sont difficiles à réaliser en pratique dans les réseaux filaires et les réseaux sans fils ne font qu'augmenter cette complexité. Malgré tout et bien que cela puisse poser un problème, nous supposons dans ce qui suit que toutes les horloges sont parfaitement synchronisées.

Pour étudier le temps limite minimal de transmission (Delay Lower Limit DLL) sur IEEE 802.11, on a besoin de calculer le retard minimum (Minimum Delay MD). Ce retard est calculé dans le cas du meilleur scénario : à chaque cycle de transmission, il y a un et un seul nœud actif qui a toujours un paquet à envoyer à la station qui à son tour n'envoie que la trame d'acquittement (ACK). On doit préciser aussi que dans le cas d'interférences sur le canal de transmission, le débit diminue et le retard augmente (supérieur à MD). Le cycle de transmission du mode DCF contient un DIFS, un backoff, la transmission des données, un SIFS et aussi un ACK (Fig. B.6). Le backoff moyen \overline{CW} est donné par :

$$\overline{CW} = \frac{CW_{min}T_{slot}}{2} \quad (4.7.1)$$

CW_{min} est la fenêtre de contention minimale et T_{slot} est la durée d'un slot de temps. Le temps de transmission de données TD_{DATA} , et le temps de transmission de la trame d'acquittement (TD_{ACK}) sont calculés comme suit :

$$TD_{DATA} = T_P + T_{PHY} + T_{H_{DATA}} + T_{DATA} \quad (4.7.2)$$

$$TD_{ACK} = T_P + T_{PHY} + T_{ACK} \quad (4.7.3)$$

$$MD = TD_{DATA} + \tau + T_{DIFS} + \overline{CW} \quad (4.7.4)$$

Ce temps de transmission est le temps minimal qu'un paquet met pour passer d'une source à une destination. Ce délai ne prend pas en compte le nombre de nœuds communicants, le nombre de collisions ni le nombre de retransmissions en cas de collision.

Pour calculer un délai de transmission dans IEEE 802.11 en utilisant le protocole DCF, plusieurs travaux ont pu estimer ce retard en utilisant les chaînes de Markov. [Carvalho & Garcia-Luna-Aceves, 2003; Chatzimisios et al., 2003; Vu, 2006] ont proposé des modèles basés sur les chaînes de Markov pour estimer les probabilités de collision et le retard moyen en fonction du nombre de nœuds communicants.

Analyse du retard d'un paquet pour le protocole 802.11

Pour analyser le retard subi par un paquet pour arriver à sa destination, on se place dans les hypothèses suivantes [Chatzimisios et al., 2003] :

1. n nœuds peuvent communiquer sur le même réseau ;
2. chaque nœud a toujours un paquet à transmettre ;

Tableau 4.1 – Signification des différents temps de transmission

| | | |
|-------------|--|--------------|
| T_{slot} | Temps d'un slot | $20\mu s$ |
| T_{SIFS} | Temps d'un SIFS | $10\mu s$ |
| T_{DIFS} | Temps d'un DIFS | $50\mu s$ |
| CW_{min} | Backoff minimal | 3 |
| T_P | Temps de transmission du préambule | $144\mu s$ |
| T_{PHY} | Temps de transmission de l'en-tête de la couche physique | $48\mu s$ |
| L_{HDATA} | Taille de l'en-tête MAC (en octets) | $28octets$ |
| L_{ACK} | Taille de la trame d'acquittement ACK (octets) | $14octets$ |
| T_{HDATA} | Temps de transmission de l'en-tête MAC | |
| L_{DATA} | Taille de la trame utile (payload) | |
| T_{DATA} | Temps de transmission de la trame utile | |
| τ | Temps de propagation des trames | $1\mu s$ |
| R_{DATA} | Débit | $800kbits/s$ |

Pour étudier en simulation et expérimentalement l'effet de la charge du réseau sur la QoC du robot, on a ajouté une tâche externe guidée par le temps soit avec une période d'échantillonnage variable soit émettant une trame de longueur variable. On suppose qu'on a une station (de coordonnées $[X_s, Y_s] = [0, 5]$) qui communique avec le robot. La

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

position initiale du robot est $[X_0, Y_0, \theta_0] = [0, 0, 0]$. Chaque trame est datée pour pouvoir calculer le retard de transmission.

Spécifications

1. Le réseau :
 - 800kbits/s
 - Retry limit=0, c'est à dire que la retransmission des données perdues est interdite.
2. Le flux capteur :
 - la tâche est guidée par le temps ($T'_e = 400ms$)
 - la longueur de la trame est de 64 bits pour les trois mesures
 - si le retard est supérieur à T'_e , la trame est considérée comme perdue.
3. Le flux contrôleur :
 - la tâche est guidée par les événements
 - la longueur de la trame est de 64 bits pour les deux consignes de vitesse des roues.
 - si le retard est supérieur à T'_e , la trame est considérée comme perdue.
4. Le flux externe (deux tâches) :
 - soit la tâche est guidée par le temps (période variable)
 - soit la longueur de la trame est variable.

En appliquant les équations ci-dessus, on trouve que le retard minimal (MD) pour envoyer une trame de 64bits est égal à $MD = 603\mu s$ pour le réseau spécifié ci-dessus.

La Fig.4.14 montre la trajectoire du robot pour atteindre la cible désirée $[X_d, Y_d] = [10, 10]$ à partir de $[X_0, Y_0] = [0, 0]$. Le long de son trajet, le robot est toujours dans la zone de couverture de la station. La Fig. 4.15 représente l'évolution des vitesse linéaire, angulaire et l'orientation du robot. Ces deux figures seront utilisées par la suite comme référence pour comparer la trajectoire du robot avec une charge de zéro et les autres trajectoires obtenues en augmentant la charge. Après avoir chargé le réseau à différents niveaux, on fait une étude sur le retard. Le retard est calculé de la façon suivante :

- on envoie dans la trame transmise le temps d'envoi de cette trame (étiquetage) ;
- dès l'arrivée de cette trame, on compare le moment d'envoi et le moment de réception.

Comme précisé avant, si le retard est supérieur à la période d'échantillonnage, la trame est considéré comme perdue. Le robot garde pendant cette période la dernière valeur des consignes de boucles locales reçues.

Pour effectuer cette étude, nous nous intéressons à deux cas différents pour le flux externe, à savoir la charge est simulée en changeant la période d'envoi ou bien en changeant la longueur de la trame.

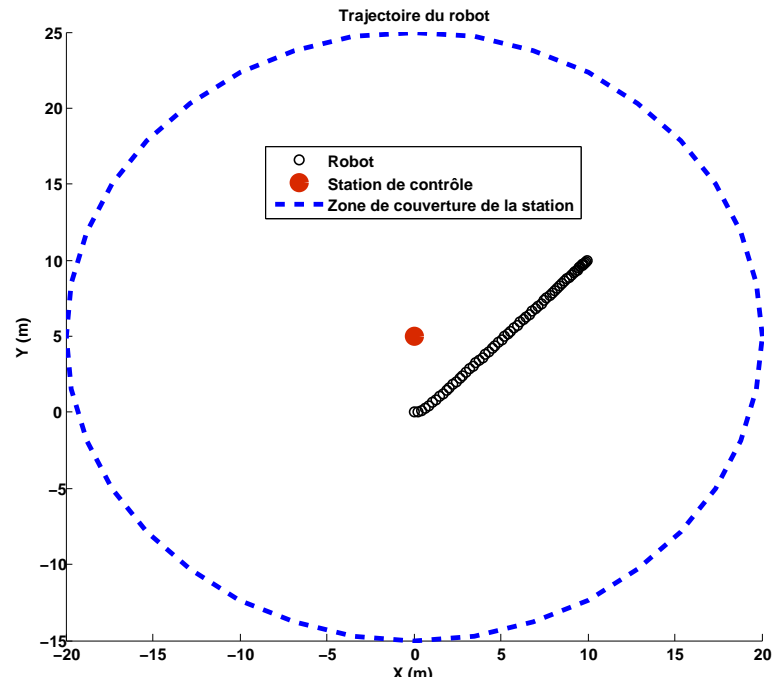


Figure 4.14 – Trajectoire du robot en simulation pour atteindre une cible

Etude de la charge en fonction de la période d'envoi

Pour étudier l'influence de la charge sur la performance de l'algorithme de commande, une tâche externe a été ajoutée. La variation de la charge dépend essentiellement de la période d'envoi de la trame (plus la période est petite plus le réseau est chargé). La longueur de la trame ici est constante (64 bits).

En analysant le protocole d'accès au médium (DCF expliqué dans la section précédente), et vu la nature stochastique de ce protocole, même en chargeant le réseau à 100%, le robot peut atteindre la cible (idem Fig. 4.14). La Fig.4.16 montre aussi que le retard est variable mais il reste toujours très inférieur à la période d'échantillonnage et la période d'échantillonnage est très inférieure au temps de réponse du robot, ce qui n'influence donc pas la QoC du robot. On peut expliquer cette non-influence de la charge du réseau, dans ce cas, par la robustesse du protocole d'accès au médium. En effet, le protocole donne la même importance à chaque trame voulant envoyer (il n'y a pas de priorités). Donc, le protocole attribue une bande passante à la trame capteur et à la trame contrôleur pour qu'elles puissent envoyer, et cela au détriment de la tâche externe. La variation du retard est justifiée par la présence des collisions et donc l'ajout d'un temps aléatoire (Backoff).

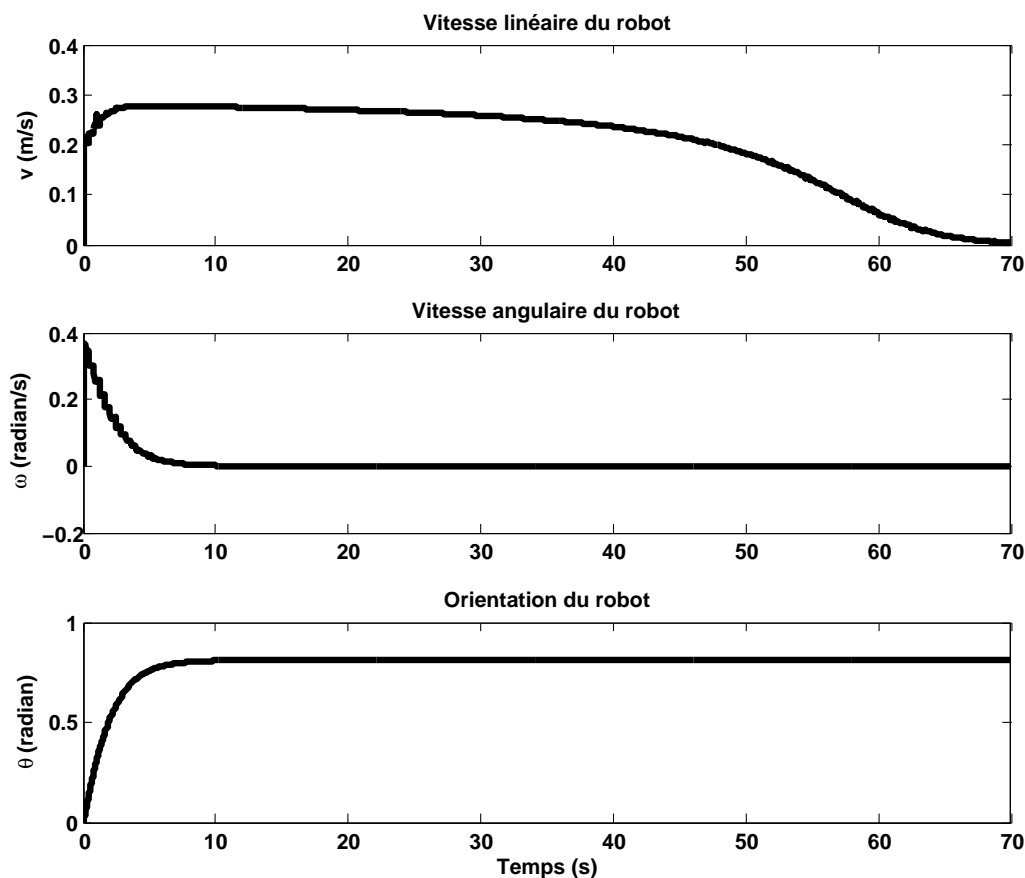


Figure 4.15 – Evolution des vitesses linéaire, angulaire et de l'orientation du robot en simulation

Etude de la charge en fonction de la longueur de la trame

Pour bien étudier l'influence de la charge du réseau sur la QoC du robot, on peut distinguer deux cas à savoir, l'atteinte d'une cible sans obstacle et avec obstacle.

a. Atteinte d'une cible sans obstacle

Dans ce cas de figure, le robot doit atteindre une cible ($[X_d, Y_d] = (1, 1)$) sans qu'un obstacle soit sur sa trajectoire. On charge le réseau avec une tâche externe (un autre robot qui envoie différents types de trames ie. trame vidéo, mesures ...). La charge a varié, en changeant la longueur de la trame externe, entre 0% et 100%. La Fig. 4.17 représente le retard entre le capteur et le contrôleur (flux capteur), le retard entre le contrôleur et l'actionneur (flux contrôleur) ainsi que l'ordonnancement des différents flux (externe, capteur et contrôleur) avec 50% de charge. Le retard de la trame capteur ne dépasse pas la période d'échantillonnage (entre 200 et 250ms), et le retard mesuré de

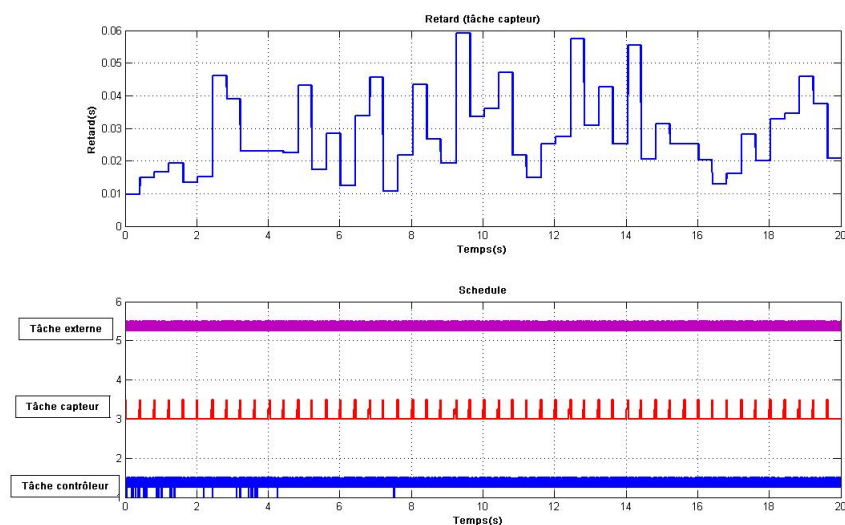


Figure 4.16 – Retard de transmission (contrôleur/capteur) et ordonnancement de différentes tâches en simulation avec 100% de charge

la trame contrôleur est entre 15 et 20ms ce qui n'influence pas d'une façon remarquable la performance du système (le robot rejoint la cible avec la trajectoire souhaitée Fig. 4.18). Dans ce cas, le retard du flux capteur est dix fois plus grand que le retard du flux contrôleur car la tâche capteur est guidée par le temps et la tâche contrôleur est guidée par les événements ce qui influence le moment d'envoi de la trame contrôleur et donc le retard.

La Fig. 4.19 montre le retard qu'a une trame entre le capteur et le contrôleur et une autre trame entre le contrôleur et l'actionneur avec 90% de charge du réseau. Le retard du flux capteur a augmenté (entre 350 et 400ms) ce qui est normal parce que la tâche externe a mis beaucoup plus de temps à être acheminée. Pour la tâche contrôleur, le retard reste constant vu qu'il n'y a pas de concurrence avec les autres tâches (pas de collision). Au delà de 95% de charge (Fig. 4.18 et Fig. 4.20), on remarque que la station perd complètement le contrôle et le robot ne peut pas atteindre la cible. Le retard dans ce cas dépasse dans certains cas la période d'échantillonnage, que ce soit pour le flux capteur ou le flux contrôleur ce qui dégrade la QoC du robot.

b. Atteinte d'une cible avec obstacle

Dans cette partie, on suppose qu'il y a un obstacle entre la position initiale du robot et la cible à atteindre. En changeant la charge du réseau, l'influence de cette charge sur la performance de l'algorithme d'évitement d'obstacle a été étudiée. Cette performance est caractérisée par la distance entre le centre du robot et le centre de l'obstacle (le robot ne doit pas toucher l'obstacle). La Fig.4.21 montre l'influence de la charge sur la distance

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

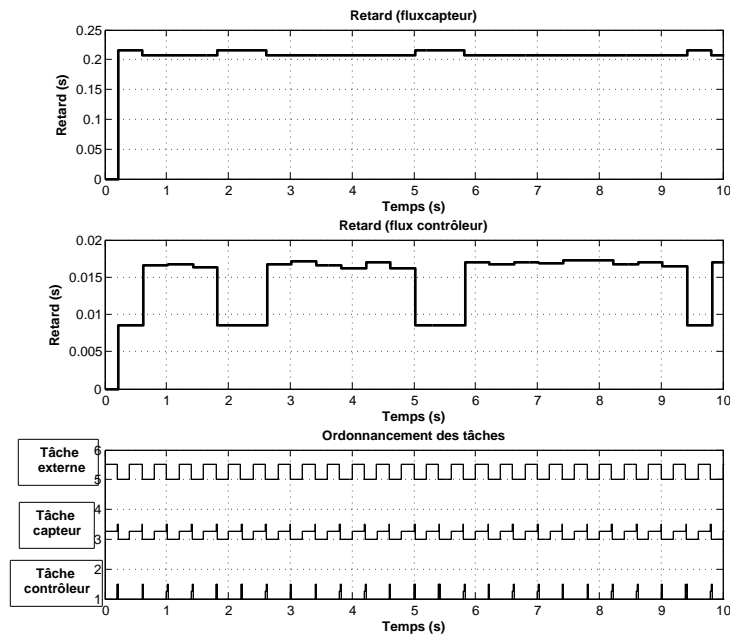


Figure 4.17 – Retards (capteur-contrôleur et contrôleur-actionneur) et ordonnancement des différentes tâches avec 50% de charge

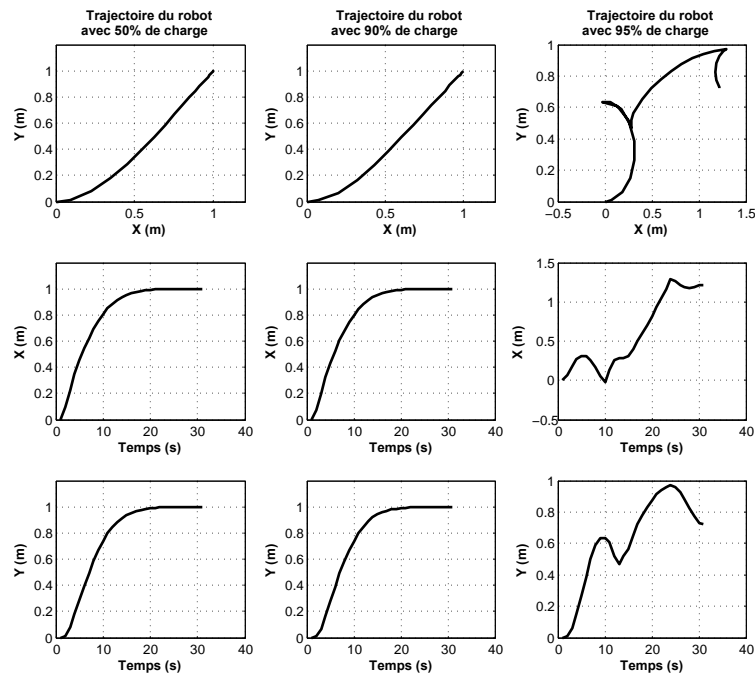


Figure 4.18 – Trajectoire du robot en simulation pour atteindre une cible en fonction de la charge du réseau

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

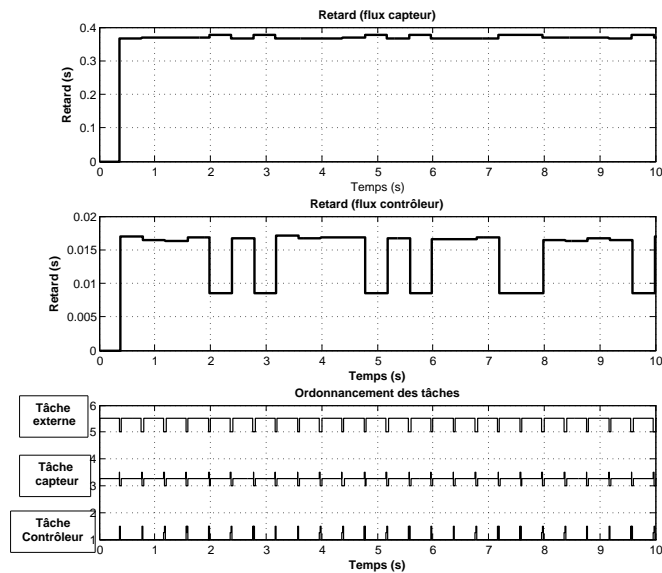


Figure 4.19 – Retards (capteur-contrôleur et contrôleur-actionneur) et ordonnancement des différentes tâches avec 90% de charge

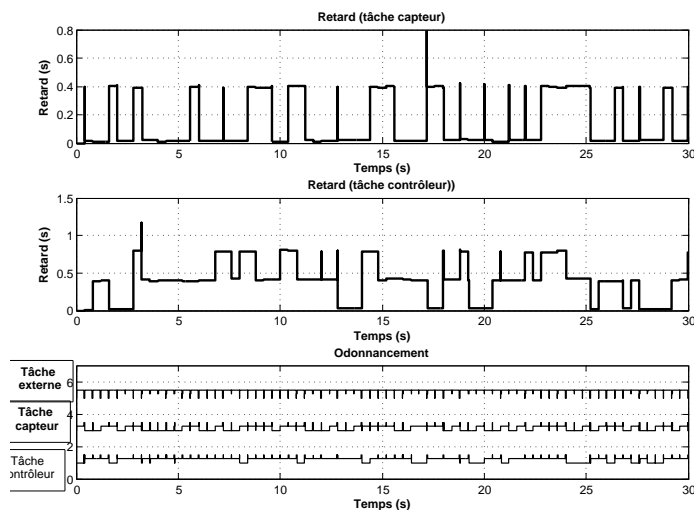


Figure 4.20 – Retards (capteur-contrôleur et contrôleur-actionneur) et ordonnancement des différentes tâches avec 95% de charge

entre le robot et l'obstacle. En effet, l'influence de la charge du réseau est beaucoup plus remarquable dans ce cas que dans le cas où il n'y a pas d'obstacle. Entre 0% et 54% de charge, le robot contourne l'obstacle sans le toucher (Fig.4.21 et Fig.4.22). Par contre, à partir de 55% de charge, le robot n'a pu l'éviter (Fig.4.22). Cela est dû au retard qui est généré par la tâche externe.

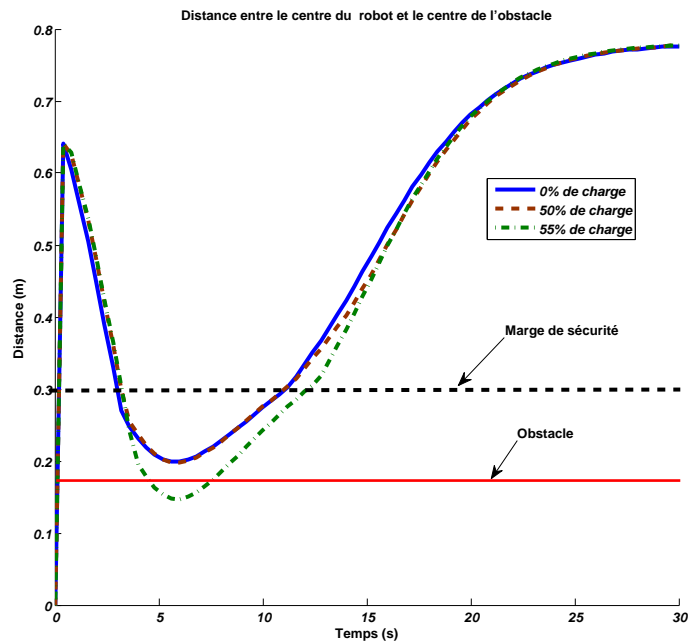


Figure 4.21 – Distance entre le robot et l'obstacle en fonction de la charge du réseau

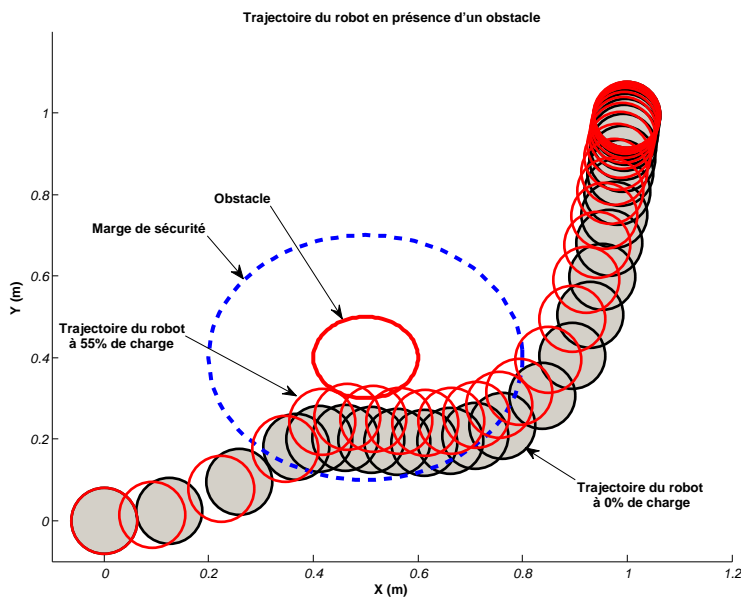


Figure 4.22 – Trajectoire du robot avec obstacle en fonction de la charge du réseau

4.7.2 Effet des pertes de paquet sur la QoC du robot

On va étudier l'influence des pertes de paquets sur la QoC du robot. Pour cela, on a simulé les pertes sur TrueTime suivant l'Algorithme 3.

rand est une fonction qui donne une valeur aléatoire η entre 0 et 1 suivant une loi

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

Algorithm 3 Algorithme pour simuler les pertes de paquets

```

PL = rand
if PL >  $\eta$  then
    Calcul la valeur de la commande et prend en compte le changement de cette valeur
end if

```

uniforme. η représente ici le taux de perte de paquet. Suivant un critère (équation 4.7.5) que nous avons fixé pour déterminer les différents intervalles des différents états de la QoC, on a pu tracer l'évolution de ce critère en fonction du taux de pertes de paquets sur la Fig. 4.23.

$$QoC_{crit\acute{e}rion} = \sum_{i=0}^k \epsilon_x(i)^2 + \epsilon_y(i)^2 \quad (4.7.5)$$

ϵ_x et ϵ_y sont les erreurs entre les points de la courbe de référence (à 0% de pertes de paquets) et les points des autres courbes (fonctions du pourcentage des pertes de paquets) à l'instant t suivant l'axe X et Y respectivement. k est le nombre de points représentant chaque courbe. La Fig.4.24 montre les différentes trajectoires du robot en fonction de

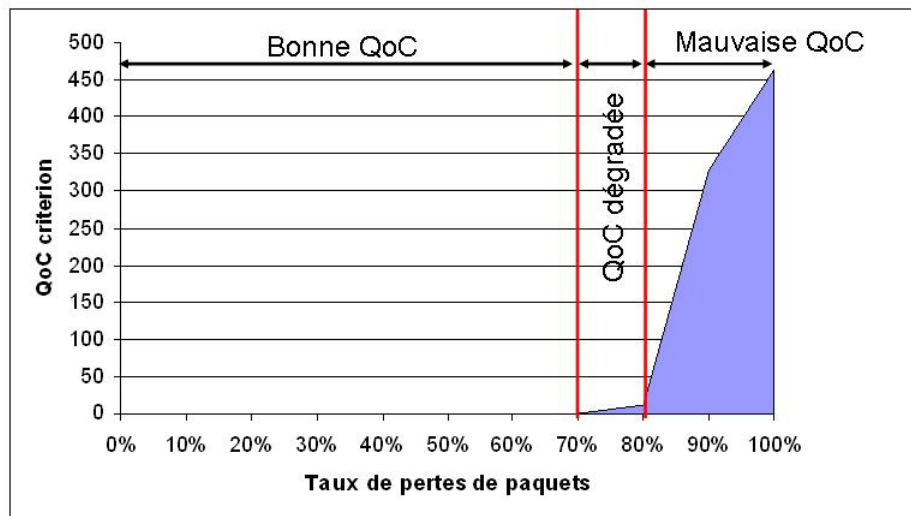


Figure 4.23 – Dégradation de la QoC en fonction des pertes de paquets

l'état de la QoC, à savoir : bonne, dégradée ou mauvaise, et l'évolution de ces trajectoires suivant l'axe X et Y . La Fig.4.25 montre l'évolution des vitesses linéaire, angulaire et de l'orientation du robot avec réseau et une station en fonction des trois états possibles de la QoC.

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

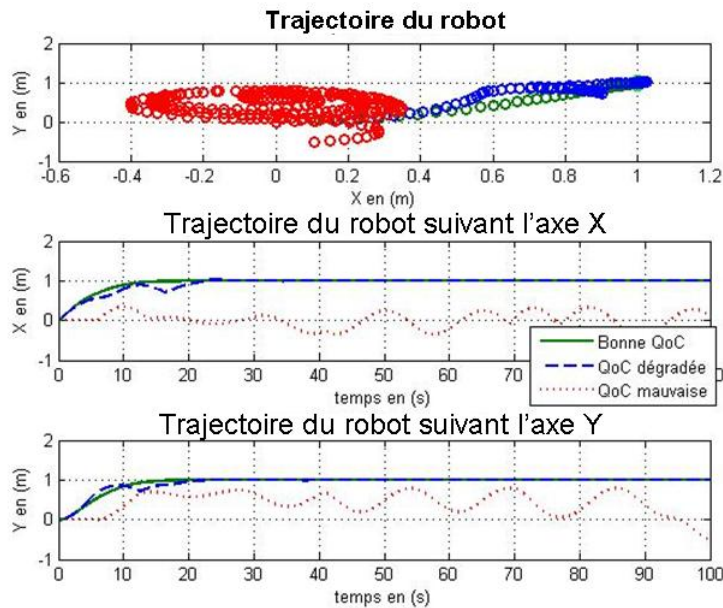


Figure 4.24 – Différentes trajectoires du robot en simulation en fonction de taux de pertes de paquets fixé dans TrueTime

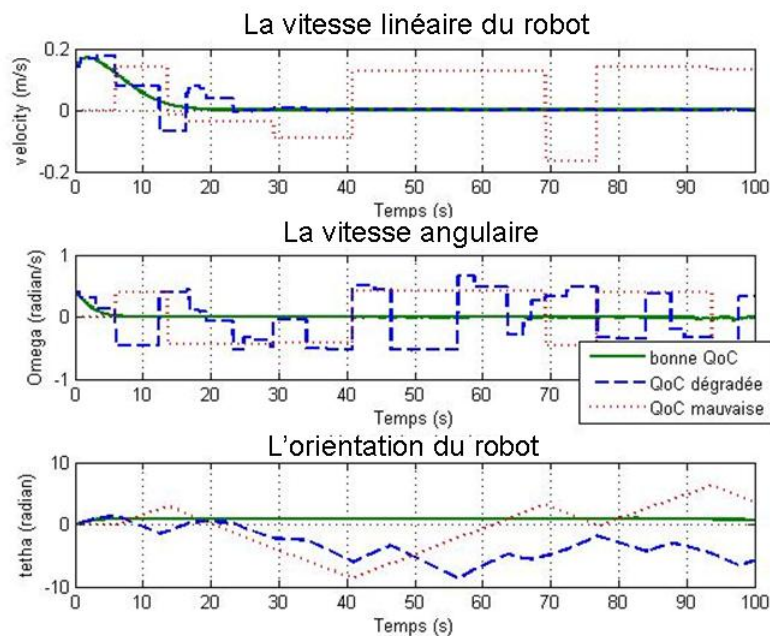


Figure 4.25 – Évolution des vitesses linéaire, angulaire et de l'orientation du robot avec réseau et une station

4.8 Environnement multi-robots communicants

L'objectif de cette section est de définir une architecture de communication dans un environnement multi-robots communicants. Le scénario voulu est que les robots soient distribués aléatoirement et ils doivent après se déplacer pour faire une formation en communiquant entre eux d'une part et avec la station de contrôle d'autre part.

Dans un premier temps, on va considérer deux robots similaires qui partagent des informations avec la station de contrôle.

4.8.1 Architecture de communication

Dans cette section on va traiter une architecture de communication infrastructure. On considère qu'il y a deux robots qui communiquent avec une station (mode infrastructure) (voir Fig.4.26). Comme il y a pas de priorité sur Wi-Fi, les deux robots envoient

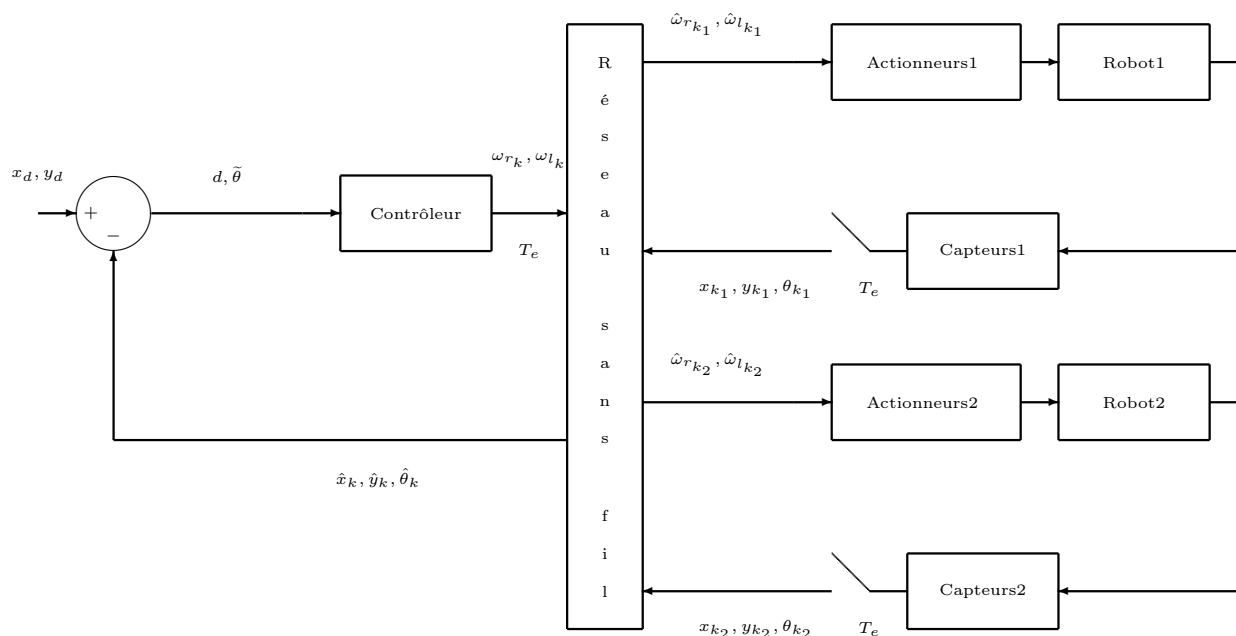


Figure 4.26 – Architecture de commande de deux robots dans un mode infrastructure

leurs mesures en même temps, donc il y a toujours des collisions ; la station ne reçoit rien ; comme on l'a précisé avant, la tâche contrôleur est guidée par les événements, donc elle n'est pas activée et la station n'envoie pas de commande (Fig.4.27). Les deux robots ne bougent pas.

Pour résoudre ce problème, on doit attribuer des priorités à chaque robot sans passer par la configuration réseau parce qu'elle ne le permet pas. L'idée est d'introduire des

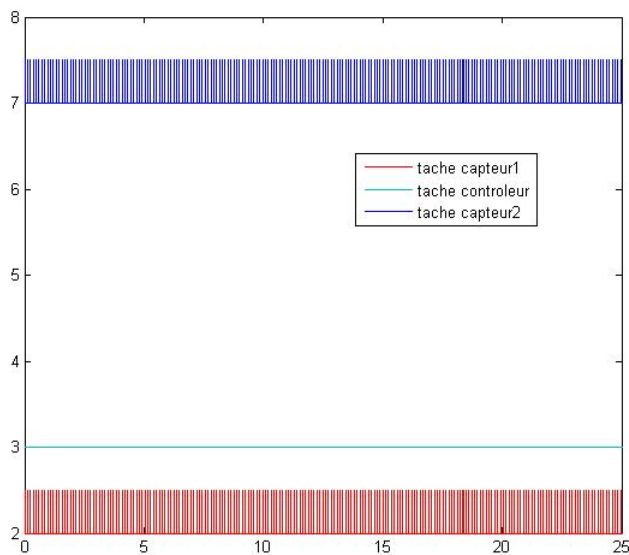


Figure 4.27 – Ordonnancement des différentes tâches avec deux robots

offsets (décalage en temps d'une tâche par rapport à une autre suivant les priorités) différents dans le fichier d'initialisation de chaque tâche capteur. Dans notre cas, nous avons laissé le premier robot avec un offset de 0s et le deuxième robot avec un offset de 0.01s. Donc, le premier robot commence la communication avec la station en premier et ensuite dès qu'il termine la communication, le deuxième robot envoie ses mesures et ainsi de suite. La Fig.4.28 montre l'ordonnancement des tâches avec des offsets. On constate que les deux robots envoient leurs mesures, le contrôleur les reçoit, calcule la commande et envoie les valeurs des commandes à chaque robot. La Fig.4.29 montre les trajectoires respectives des deux robots pour un suivi de trajectoires.

4.9 Conclusion

Le principal objectif de ce chapitre, dans l'approche co-conception, est de définir une architecture de communication entre un robot et une station ou plusieurs. Un simulateur de robot a été élaboré en utilisant Matlab/Simulink. Un simulateur réseau a été aussi utilisé pour simuler la communication entre le robot et la station. Le simulateur utilisé est TrueTime qui est compatible avec Simulink.

Dans le but d'assurer une bonne qualité de service (QoS) et une bonne qualité de contrôle (QoC), une stratégie de reconfiguration en utilisant le Handoff a été proposée pour aider le robot à décider du moment de changement de station s'il y en a plusieurs. Cette stratégie prend en compte la QoS du réseau. Le robot peut aussi décider de changer de contrôleur s'il n'est en regard d'aucune station. Il s'agit d'une reconfiguration du

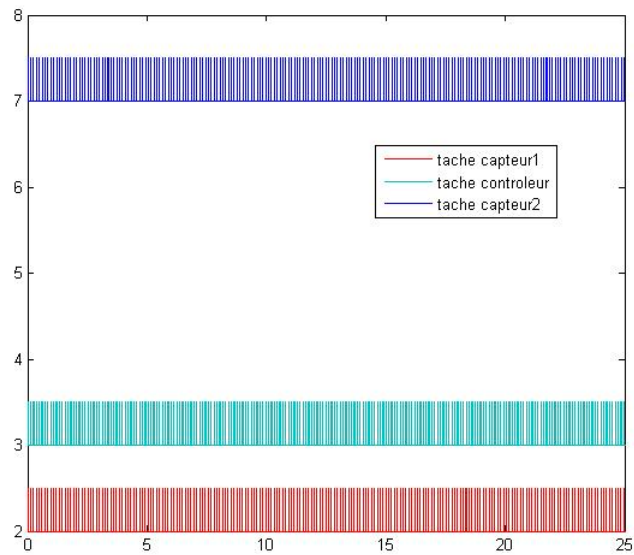


Figure 4.28 – Ordonnancement des différentes tâches avec deux robots et des priorités

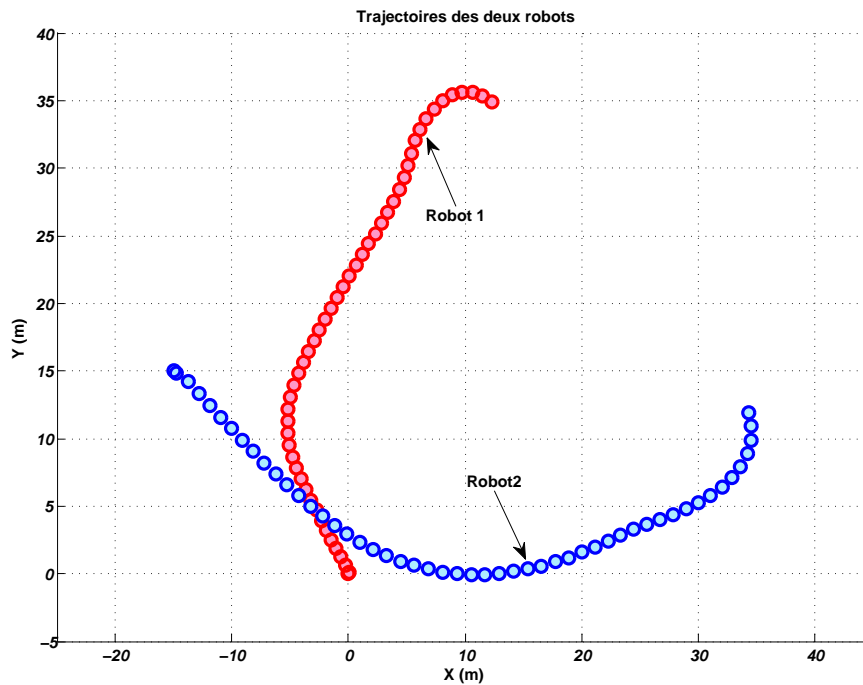


Figure 4.29 – Trajectoires de deux robots (en simulation)

contrôleur pour que le robot soit autonome. Ces deux stratégies ont été vérifiées en utilisant Matlab/TrueTime.

Chapitre 4. Influence de la QoS sur la QoC et stratégies d'évitement de la dégradation de QoC

Une étude de l'effet de la charge du réseau sur la QoC a été aussi traitée ainsi que l'influence des pertes de paquets sur la QoC du robot.

Enfin, une architecture de communication infrastructure dans un environnement multi-robots a été introduite et une solution applicative pour attribuer des priorités a été proposée pour éviter les collisions.

Dans ce chapitre nous avons réussi à adapter le réseau aux contraintes de la commande en changeant la station de contrôle et à adapter la commande aux contraintes dynamiques du réseau sans fil en commutant vers un contrôleur interne dégradé. Cette reconfiguration de l'architecture de communication et aussi de l'architecture de commande n'a pas pris en compte l'aspect co-conception. Le défi maintenant est de décider quand le robot a vraiment besoin de changer de station en fonction de l'état de la QoC et aussi de changer de contrôleur en fonction de l'état de la QoS.

Chapitre 5

Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

Sommaire

| | | |
|------------|---|------------|
| 5.1 | Introduction | 118 |
| 5.2 | Pourquoi les réseaux bayésiens ? | 118 |
| 5.3 | Les réseaux bayésiens (RBs) | 119 |
| 5.3.1 | Approche qualitative | 120 |
| 5.3.2 | Approche quantitative | 122 |
| 5.4 | Expertise et étude statistique | 123 |
| 5.4.1 | Influence des pertes de paquets sur la QoC du robot : | 124 |
| 5.4.2 | Influence de la distance et de la présence d'obstacles sur la puissance du signal reçu (RSSI) : | 129 |
| 5.4.3 | Influence du retard et du RSSI sur le pourcentage des pertes de paquets : | 129 |
| 5.4.4 | Étude du délai de transmission : | 130 |
| 5.4.5 | Détermination des probabilités marginales des nœuds sans parent : | 131 |
| 5.5 | Aide à la décision et diagnostic | 131 |
| 5.5.1 | Aide à la décision | 132 |
| 5.5.2 | Diagnostic | 133 |
| 5.6 | Apprentissage de paramètres | 133 |
| 5.6.1 | Critère de la QoS | 133 |
| 5.6.2 | Critère de la QoC | 135 |
| 5.6.3 | Validation du réseau bayésien trouvé par apprentissage | 137 |
| 5.7 | Réseaux bayésiens distribués | 139 |

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

| | | |
|------------|--|------------|
| 5.7.1 | Prise de décisions et diagnostic distribué | 140 |
| 5.7.2 | Priorités dynamiques | 141 |
| 5.7.3 | Résultats et simulation | 142 |
| 5.7.4 | Scénario 1 | 143 |
| 5.7.5 | Scénario 2 | 144 |
| 5.8 | Conclusion | 145 |

5.1 Introduction

Dans ce chapitre, le principal objectif est d'adapter le réseau aux contraintes de la commande en changeant la station de contrôle et d'adapter la commande aux contraintes dynamiques du réseau sans fil en commutant vers un contrôleur interne dégradé. Le défi est de décider quand le robot a vraiment besoin de changer de station et aussi de contrôler. Pour cela, avant de déclencher la commutation de contrôleur ou le Handoff, on a besoin d'un outil d'aide à la décision qui, en fonction des différents paramètres qui influencent le réseau, peut quantifier la QoS du réseau et aussi la QoC du robot en estimant leurs états considérés globalement comme bon, dégradé ou mauvais. Nous avons besoin aussi d'identifier et localiser en-ligne les causes de la dégradation de ces deux paramètres pour les éviter ou les corriger. Nous envisageons aussi de prévoir cette dégradation pour l'éviter avant qu'elle ne se produise. Nous proposons donc d'utiliser les réseaux bayésiens pour l'aide à la décision et le diagnostic en-ligne. Dans ce chapitre, un modèle basé sur les réseaux bayésiens distribués a été proposé pour prendre des décisions et aussi pour diagnostiquer le système. En fonction des résultats donnés par le réseau bayésien, le système peut prendre des décisions à la fois pour la reconfiguration de l'architecture de communication et aussi de l'architecture de commande et/ou de diagnostic dont le principal but est d'assurer un certain niveau de QoC. Ce chapitre reprendra certaines formules ou figures du chapitre précédent pour rendre la lecture plus fluide.

5.2 Pourquoi les réseaux bayésiens ?

Le choix d'utiliser les réseaux bayésiens pour prendre des décisions et aussi diagnostiquer le système (robot et réseau sans fil) est justifié par :

1. L'aspect stochastique des réseaux sans fil ;
2. La difficulté de modéliser le comportement du réseau sans fil et son influence sur la QoC du véhicule mobile ;

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

3. La possibilité de rassembler et de fusionner des données de diverses natures dans un même modèle [Naïm et al., 2004] : retour d'expérience, expertise (règles logiques, équations, statistiques ou probabilités), et observations.

Dans notre cas, par exemple, chacune de ces sources d'informations, quoique présente, est insuffisante à elle seule pour fournir une modélisation et une représentation précise et réaliste du système.

La représentation graphique d'un réseau bayésien est explicite, intuitive et compréhensible par un non spécialiste [Naïm et al., 2004], ce qui facilite l'utilisation du modèle ainsi que sa validation. En général, il est plus facile de s'appuyer sur un modèle graphique que d'utiliser une boîte noire dont on ignore le contenu.

Le point fort d'un réseau bayésien est sa polyvalence : on peut utiliser le même modèle pour évaluer, prévoir, diagnostiquer ou optimiser des décisions, ce qui contribue à rentabiliser l'effort de sa construction.

Tout ces points nous ont poussé à utiliser les réseaux bayésiens. Nous allons à présent étudier plus en détail les réseaux bayésiens, l'acquisition des connaissances pour modéliser notre système avec un réseau bayésien et son utilisation pour prendre des décisions, diagnostiquer ainsi qu'optimiser l'utilisation des ressources du réseau sans fil. L'objectif de ce chapitre n'est pas de faire une étude approfondie et détaillée sur les réseaux bayésiens mais de montrer comment les utiliser dans notre cas (pour plus de détails sur les réseaux bayésiens voir l'annexe A). Le tableau 5.1 présente une comparaison des réseaux bayésiens avec d'autres techniques. Les réseaux bayésiens montrent leurs efficacité par rapports aux autres méthodes de classifications dans le cas des données mixtes et/ou incomplètes, des données ou des informations tachées d'incertitudes. De plus, vu que les réseaux bayésiens sont des modèles graphiques, ils sont les plus lisibles et compréhensibles par rapport aux autres méthodes.

5.3 Les réseaux bayésiens (RBs)

Un réseau bayésien est défini par [Naïm et al., 2004] :

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

$$P(V_1, V_2, \dots, V_n) = \prod_{i=1}^n P(V_i | C(V_i)) \quad (5.3.1)$$

$C(V_i)$ est l'ensemble des causes (parents) de V_i dans le graphe G .

Les RB sont à la fois des modèles de représentation des connaissances et aussi des machines à calculer les probabilités conditionnelles (CPT : Conditional Probability Table).

Tableau 5.1 – Comparaison des réseaux bayésiens avec d’autres méthodes

| | SE | AD | RN | RàPC | RB |
|---------------------------|----|----|----|------|----|
| Expertise | ++ | - | - | - | + |
| Données | - | + | ++ | + | + |
| Mixte (Expertise+données) | - | + | + | + | ++ |
| Généralisation | - | + | ++ | + | + |
| Données incomplètes | - | - | + | - | ++ |
| Incertitude | + | - | - | - | ++ |
| Lisibilité | + | + | - | + | ++ |
| Facilité | - | ++ | + | - | - |
| Performances | + | + | ++ | + | + |

- : inconvénient, + : avantages, ++ : meilleure technique

SE : Systèmes Experts, AD : Arbre de Décision, RN : Réseaux de Neurones, RàPC : Raisonnement à Partir de Cas.

Il est donc nécessaire de distinguer deux approches différentes pour modéliser les connaissances par RB : l’approche qualitative et l’approche quantitative.

5.3.1 Approche qualitative

Dans cette section, nous allons présenter l’aspect qualitatif qui est basé sur la théorie des graphes et représente la notion de causalité, c’est-à-dire, le lien de causes à effets. Pour cela, nous devons passer par deux étapes différentes :

- L’identification des variables et de leurs espaces d’états
- La détermination de la structure du réseau bayésien.

Identification des variables et de leurs espaces d’états

Dans cette étape, l’intervention humaine d’un expert est indispensable pour déterminer l’ensemble des variables X_i qui caractérisent le système. Dès que ces variables sont identifiées et validées, il est ensuite nécessaire de déterminer l’espace d’état de chaque variable X_i , c’est-à-dire l’ensemble de ses valeurs possibles. Ces valeurs peuvent être discrètes ou continues. Dans le cas d’une variable continue, la discrétisation est indispensable parce que la majorité des logiciels destinés à ce fait ne traite que des variables discrètes. Toutefois il faut naturellement faire attention de ne pas faire des discrétisations trop fines conduisant à des tables de probabilités de grande taille et aussi à une utilisation excessive de la mémoire du système.

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

Revenons à notre système. Pour l'approche envisagée, nous avons étudié tous les paramètres qui influencent la QoS du réseau et la QoC du robot. La QoC est classiquement influencée par trois paramètres : la QoS du réseau, la mission du robot M (dans notre cas, l'atteinte d'une cible, l'évitement des obstacles ou bien le suivi de trajectoire) et l'état du robot R . La mission du robot influence sa dynamique et ainsi la QoC du robot. En se référant à [Hespanha et al., 2007] et [Li & Wang, 2008], la QoS est influencée essentiellement par deux paramètres, à savoir, le retard Δ_t entre le moment de transmission et le moment de réception, et le pourcentage de pertes de paquets PL (voir Fig.5.1). La puissance du signal reçu (Request Signal Strength Indicator (RSSI)) ainsi que le Bit Error Rate (BER) et la taille des buffers jouent un rôle sur le nombre de paquets perdus. Le BER a une valeur moyenne fixe suivant le réseau sans fil utilisé, donc on peut l'enlever de notre RB car on n'utilise qu'un seul réseau. On considère, dans notre cas, que le retard influence le nombre de pertes de paquets car si le retard dépasse une période d'échantillonnage, on considère que le paquet est perdu. Le retard considéré ici est le temps de propagation d'un paquet, le retard dû à l'attente de paquet dans la file d'attente n'est pas traité. Il est bien évidemment influencé par le nombre de collisions qui dépend à leurs tour du nombre de nœuds qui veulent communiquer en même temps. Le nœud PL du réseau bayésien est influencé aussi par le changement du signal reçu (RSSI) qui dépend de la distance entre le robot et la station et aussi la présence ou non des obstacles. Dans un RB, chaque nœud possède un ou plusieurs modes (ou états). La Table 5.2 présente la signification des différents nœuds du RB proposé ainsi que leurs modes ou états. Le choix de ces modes sera expliqué au fur et à mesure dans ce chapitre.

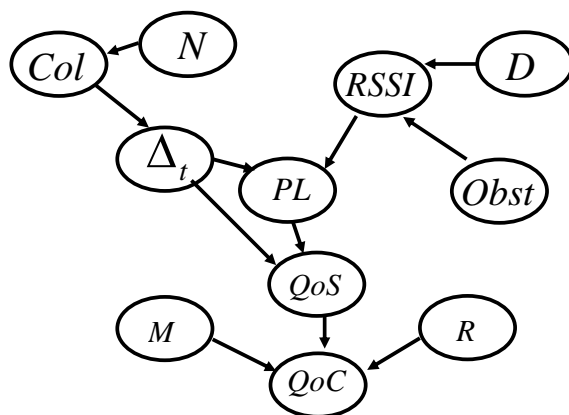


Figure 5.1 – Réseau bayésien du système

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

Tableau 5.2 – Signification des différents nœuds du RB et leurs modes

| Nœuds | Signification | Modes |
|------------|---------------------------------------|--|
| QoC | Qualité de Contrôle | Bonne, dégradée, mauvaise |
| QoS | Qualité de Service | Bonne, dégradée, mauvaise |
| R | Etat du robot | Bon, dégradé, défaillant |
| M | Mission | Cible, Obstacles, Trajectoire |
| PL | Perte de Paquets | $0\% \leq PL < 40\%$, $40\% \leq PL < 60\%$, $60\% \leq PL < 100\%$ |
| RSSI | Request Signal Strength Indicator | $RSSI > -48dBm$, $RSSI < -48dBm$ |
| Δ_t | Retard agissant sur la QoS | $\Delta_t < T_e$, $\Delta_t \geq T_e$ |
| Obst | Obstacle | Oui, Non |
| D | Distance entre la station et le robot | |
| Col | Collision | ≤ 0.8 , > 0.8 |
| N | Nombre de nœuds | ≤ 14 , > 14 |

Détermination de la structure du réseau bayésien

Cette deuxième étape a pour objectif de déterminer les liens entre les différentes variables du système. Pour cela, il faut répondre à la question : pour quels couples (i, j) la variable X_i influence-t-elle la variable X_j . Cette étape est extrêmement importante, et tout le reste se base sur elle. Dans la plupart des cas, cette étape est déterminée par l'interrogation d'un expert. Dans notre cas, une recherche bibliographique a été faite dans le domaine de la commande pour déterminer les paramètres qui influencent la QoC d'un système physique. Ainsi que dans le domaine des réseaux pour déterminer les variables qui influencent la QoS d'un réseau sans fil.

Lorsqu'on dispose d'une quantité suffisante d'informations sur le système concernant les variable X_i , la structure du réseau bayésien peut être apprise automatiquement par apprentissage de structure. Mais dans notre situation, l'étude théorique du système est suffisante pour déterminer la structure du RB.

5.3.2 Approche quantitative

L'aspect quantitatif des RBs consiste à associer à chaque nœud une table de probabilité qui le caractérise donc les lois de probabilité conjointe des variable X_i . Dans un premier temps, la connaissance d'un expert concernant les lois de probabilités sera intégrée au modèle. Ensuite, l'apprentissage des paramètres sera utilisé pour apporter une meilleure connaissance des probabilités.

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

Concrètement, deux cas se présentent en fonction de la position de la variable X_i dans le graphe :

1. La variable X_i n'a pas de variable parente : dans ce cas, il faut déterminer la loi de probabilité marginale de X_i ;
2. La variable X_i possède des variables parentes : dans ce cas, il faut exprimer la dépendance de X_i en fonction des variables parentes, soit par des probabilités conditionnelles soit par une équation déterministe que le logiciel convertira ensuite en probabilités conditionnelles. Ces deux méthodes seront utilisées dans notre situation.

5.4 Expertise et étude statistique

Trouver les probabilités conditionnelles a priori en utilisant l'expertise est délicate pour certaines variables. Pour cela, on peut utiliser ce qu'on appelle les tables de conversion d'appréciations qualitatives en probabilités, comme l'échelle de Lichtenstein et Newman proposé par [Ayyub, 2001; Lannoy & Procaccia, 2001]. Dans cette section, une étude statistique a été faite pour déterminer toutes les CPT *a priori* nécessaires pour le RB proposé en se basant sur les observations du système.

Avant cette première étude, une hypothèse a été adoptée. Nous avons besoin de déterminer les différents modes ou états de la QoS en fonction du pourcentage de pertes de paquets. Ce choix est arbitraire et peut être changé. La Table 5.3 présente la CPT *a priori* de la QoS en fonction des pertes de paquets. Cette hypothèse peut poser quelques questions par rapport à l'utilité du nœud QoS du RB proposé, mais la difficulté de quantifier la QoS en fonction des pertes de paquets nous oblige pour le moment à adopter cette hypothèse. Nous essayerons bien évidemment d'enlever cette forte hypothèse lors de l'apprentissage de paramètres.

Tableau 5.3 – Probabilités *a priori* de la QoS en fonction des pertes de paquets

| $PL(\%)$ | $0 \leq PL < 40$ | $40 \leq PL < 60$ | $60 \leq PL < 100$ |
|-----------------|------------------|-------------------|--------------------|
| $P(QoS = b PL)$ | 1 | 0 | 0 |
| $P(QoS = d PL)$ | 0 | 1 | 0 |
| $P(QoS = m PL)$ | 0 | 0 | 1 |

b =bonne, d =dégradée et m =mauvaise

5.4.1 Influence des pertes de paquets sur la QoC du robot :

On va étudier *a priori* de façon statistique l'influence des pertes de paquets sur la QoC du robot de façon à discriminer trois états {Bonne QoC, QoC dégradée, Mauvaise QoC}. Pour cela, on a simulé les pertes sur TrueTime suivant l'Algorithme 5 expliqué à la section 4.7. En ligne, le paramètre PL évoluera en fonction d'autres paramètres du réseau (Fig.5.1).

Le robot peut remplir trois missions : atteinte d'une cible, évitement d'obstacles et suivi de trajectoire. L'influence des pertes change en fonction de la mission envisagée ce qui justifie l'ajout de la variable ' M ' (Mission du système) qui influence la QoC du système. Cette variable a comme états les différentes missions du robot, à savoir atteinte d'une cible, suivi de trajectoire ou évitement d'obstacles.

1. **Atteinte d'une cible :** Nous avons choisi un critère quadratique (Eq. 5.4.1) pour déterminer les intervalles des différents états de la QoC. On a tracé l'évolution de ce critère en fonction du taux de pertes de paquets sur la Fig. 5.2. Cette courbe nous a aidé pour la définition des trois états.

$$QoC_{crit\u00e9rion} = \sum_{i=0}^k \epsilon_x(i)^2 + \epsilon_y(i)^2 \quad (5.4.1)$$

ϵ_x et ϵ_y sont les erreurs entre les points de la courbe de référence (à 0% de pertes de paquets) et les points des autres courbes (en fonction du pourcentage des pertes de paquets) à l'instant t suivant les axes X et Y respectivement. k est le nombre de points représentant chaque courbe. La Fig. 5.3 montre les différentes trajectoires

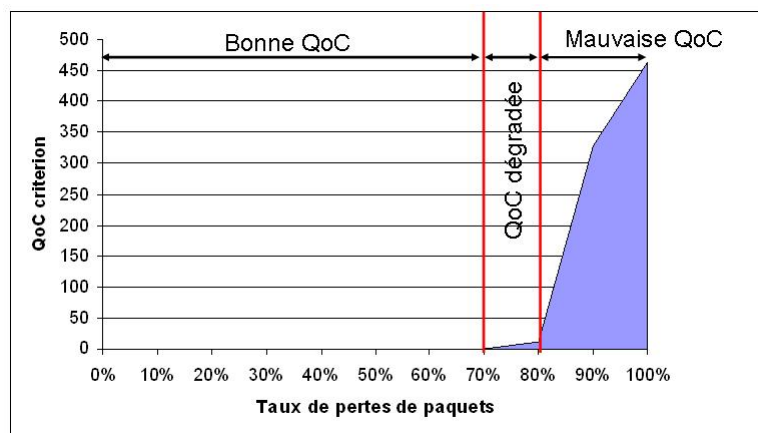


Figure 5.2 – D\u00e9gradation de la QoC en fonction des pertes de paquets dans le cas d'atteinte d'une cible

du robot en fonction de l'\u00e9tat de la QoC, \u00e0 savoir : bonne, d\u00e9grad\u00e9e ou mauvaise, et

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

l'évolution de ces trajectoires suivant l'axe X et Y et aussi l'évolution des vitesses linéaire, angulaire et de l'orientation du robot avec réseau et une station en fonction des trois états possibles de la QoC.

En se basant sur les résultats de la Fig.4.23 et l'hypothèse de la Table 5.3, la CPT

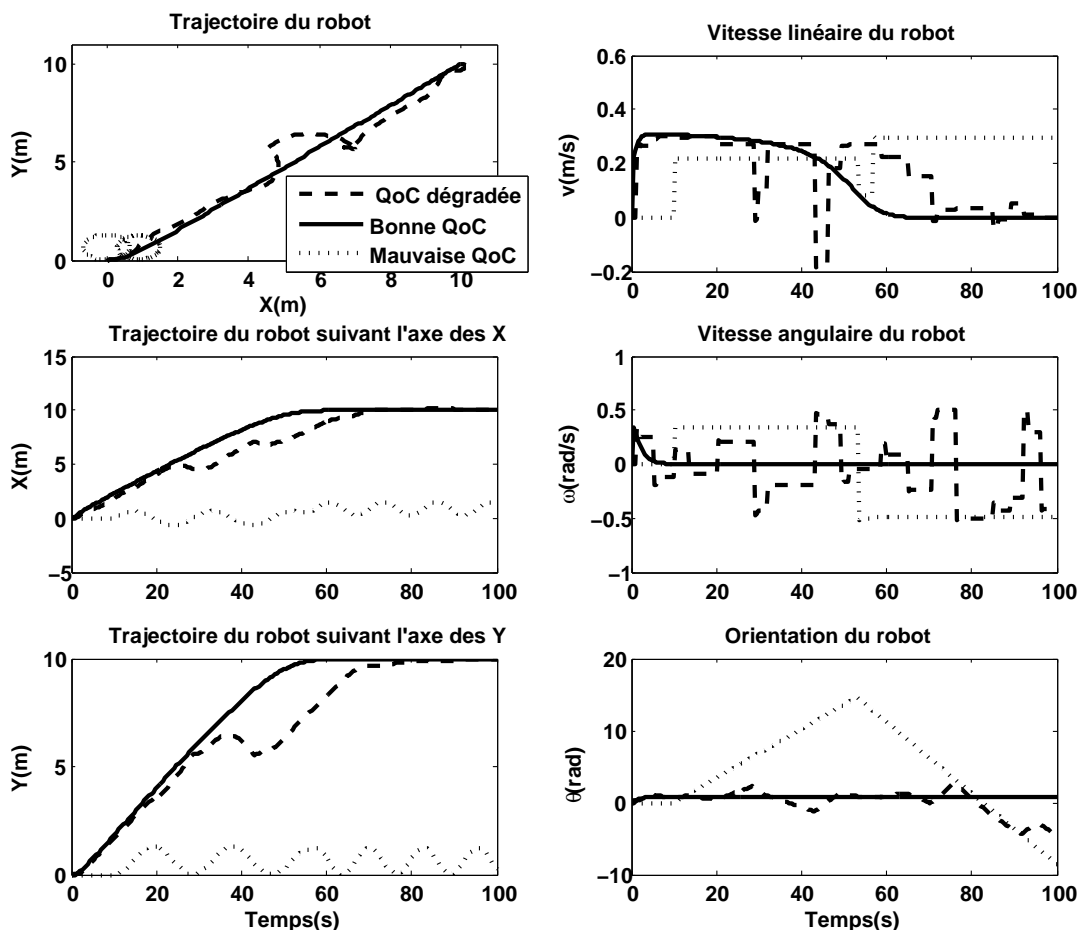


Figure 5.3 – Différentes trajectoires du robot, évolution des vitesses linéaire, angulaire et de l'orientation en simulation en fonction du taux de pertes de paquets fixé dans TrueTime

a priori de la QoC en fonction de la QoS a été définie (Table 5.4).

- Suivi de trajectoire :** Pour l'influence des pertes de paquets en cas de suivi de trajectoire, les pertes de paquets ont moins d'influence. L'étude statistique a montré que la QoC commence à se dégrader à partir de 80% de pertes et devient mauvaise à partir de 90% de pertes. La Fig. 5.5 montre trois trajectoires différentes du robot en fonction des pertes de paquets et dans trois états différents de la QoC.

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

Tableau 5.4 – CPT *a priori* de la QoC en fonction de l'état du robot et de la QoS du réseau sans fil dans le cas de l'atteinte d'une cible

| R | Normal | | | dégradé | | | défaillant | | |
|------------------------|--------|---|------|---------|------|------|------------|---|---|
| QoS | b | d | m | b | d | m | b | d | m |
| $P(QoC = b QoS, R, M)$ | 1 | 1 | 0.25 | 0 | 0 | 0 | 0 | 0 | 0 |
| $P(QoC = d QoS, R, M)$ | 0 | 0 | 0.25 | 1 | 0.75 | 0.25 | 0 | 0 | 0 |
| $P(QoC = m QoS, R, M)$ | 0 | 0 | 0.5 | 0 | 0.25 | 0.75 | 1 | 1 | 1 |

b =bonne, d =dégradée, m =mauvaise et R =Etat du Robot.

Dans le cas d'une bonne QoC, le robot suit parfaitement la trajectoire. Dans le cas d'une QoC dégradée, le robot essaie de la suivre mais il s'éloigne le plus souvent de cette trajectoire surtout lorsqu'il veut tourner. Enfin, dans le cas d'une QoC mauvaise, le robot n'arrive plus à suivre la trajectoire (voir Fig. 5.5).

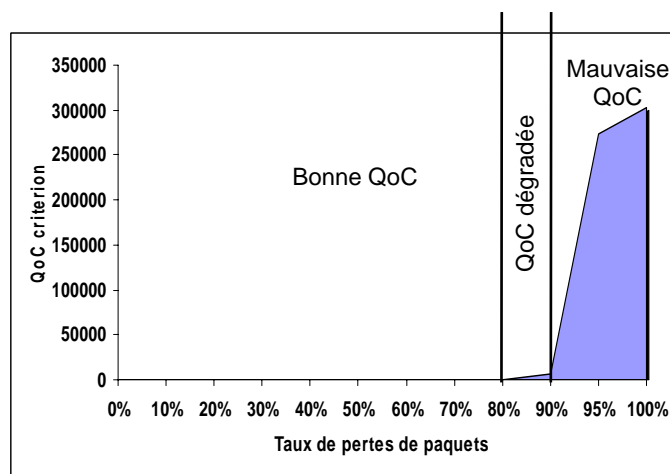


Figure 5.4 – Dégradation de la QoC en fonction des pertes de paquets dans le cas de suivi de trajectoire

Tableau 5.5 – CPT *a priori* de la QoC en fonction de l'état du robot et de la QoS du réseau sans fil dans le cas du suivi de trajectoire

| R | Normal | | | dégradé | | | défaillant | | |
|------------------------|--------|---|------|---------|-----|-----|------------|---|---|
| QoS | b | d | m | b | d | m | b | d | m |
| $P(QoC = b QoS, R, M)$ | 1 | 1 | 0.5 | 0 | 0 | 0 | 0 | 0 | 0 |
| $P(QoC = d QoS, R, M)$ | 0 | 0 | 0.25 | 1 | 0.8 | 0.4 | 0 | 0 | 0 |
| $P(QoC = m QoS, R, M)$ | 0 | 0 | 0.25 | 0 | 0.2 | 0.6 | 1 | 1 | 1 |

b =bonne, d =dégradée, m =mauvaise et R = Etat du Robot.

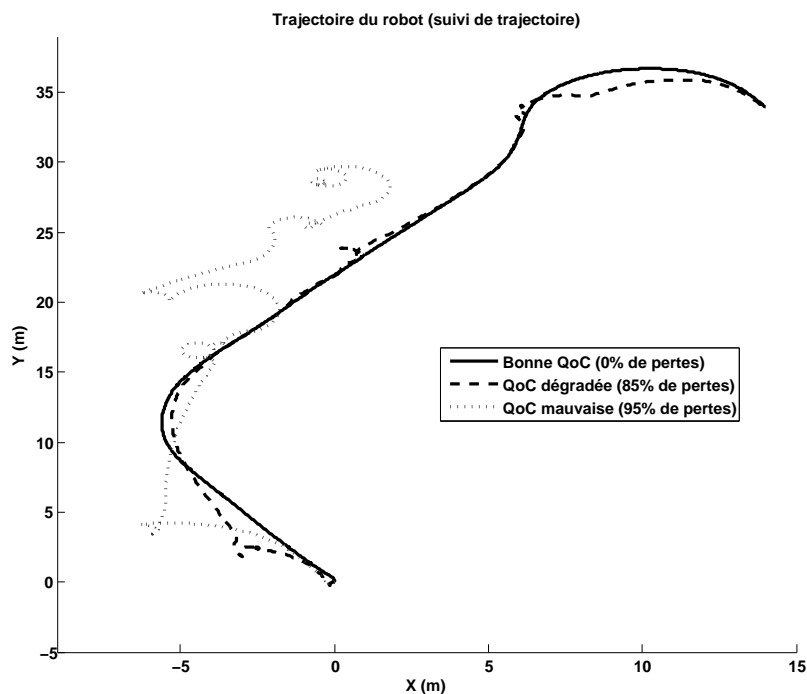


Figure 5.5 – Trajectoire du robot en fonction des pertes de paquets dans le cas de suivi de trajectoire

3. **Evitement d'obstacles** : Pour l'évitement d'obstacles, le critère utilisé pour connaître l'état de la QoC en fonction des pertes de paquets est la distance entre le robot et l'obstacle. Si la distance entre les deux est supérieure à la somme du rayon du robot et le rayon de l'obstacle, le robot a réussi à éviter l'obstacle et donc la QoC est bonne. En revanche, si la distance est inférieure ou égale à la somme des deux rayons, le robot a touché l'obstacle et donc la QoC est mauvaise.

Après une étude statistique et exhaustive de l'influence des pertes de paquets sur la QoC du robot au moment où il détecte un obstacle, la Fig. 5.6 montre la distance entre le robot et l'obstacle dans le cas où la QoC est bonne et dans le cas où la QoC est mauvaise. La Fig. 5.7 montre la trajectoire dans les deux états de la QoC qui est influencée par les pertes de paquets. On remarque donc qu'à partir de 60% de pertes, le robot touche l'obstacle.

D'après l'étude faite sur la QoC en fonction des pertes, nous avons pu remplir la CPT *a priori* de la QoC en fonction de l'état du robot et de la QoS du réseau sans fil (Table 5.6).

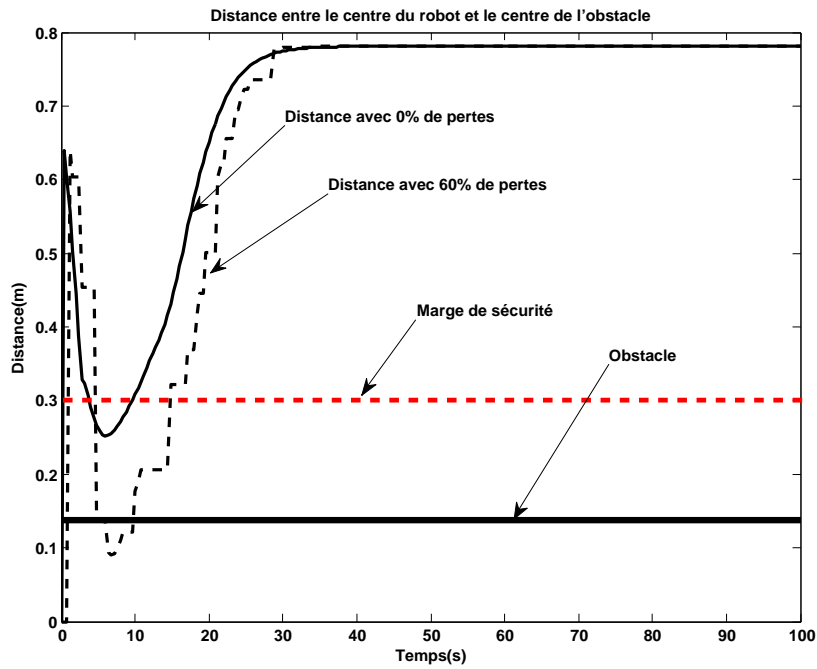


Figure 5.6 – Distance entre le robot et l'obstacle en fonction des pertes de paquets

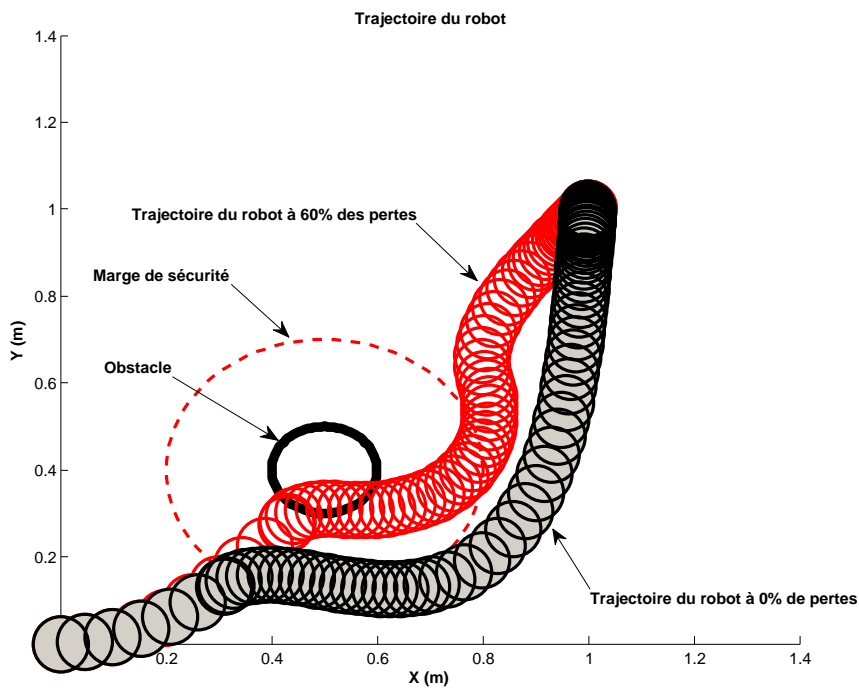


Figure 5.7 – Trajectoire du robot avec obstacle en fonction des pertes de paquets

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

Tableau 5.6 – CPT *a priori* de la QoS en fonction de l'état du robot et de la QoS du réseau sans fil dans le cas d'évitement d'obstacles

| R | Normal | | | dégradé | | | défaillant | | |
|------------------------|--------|---|---|---------|-----|---|------------|---|---|
| | b | d | m | b | d | m | b | d | m |
| $P(QoS = b QoS, R, M)$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $P(QoS = d QoS, R, M)$ | 0 | 0 | 0 | 1 | 0.5 | 0 | 0 | 0 | 0 |
| $P(QoS = m QoS, R, M)$ | 0 | 0 | 1 | 0 | 0.5 | 1 | 1 | 1 | 1 |

b =bonne, *d* =dégradée, *m* =mauvaise et *R* = Etat du Robot.

5.4.2 Influence de la distance et de la présence d'obstacles sur la puissance du signal reçu (RSSI) :

Dans cette partie, une étude de l'influence de la distance et de la présence d'obstacle sur la RSSI a été faite sur TrueTime pour définir la CPT *a priori* du nœud RSSI. Pour cela, on a simulé un robot communiquant avec une station se déplaçant à partir d'un point initial $(x_0, y_0) = (0, 0)$ vers une cible qui est choisie d'une façon à ce que le robot quitte à un moment la zone de couverture de la station. Nous avons pu remarquer que le robot perd complètement la communication après $18m$. Donc, le rayon de courbure de la zone de couverture de la station est de $18m$. $D = 18m$ sera pris comme un seuil de distance pour que le robot perde la communication. Concernant les obstacles, pour simplifier notre étude, on admet que si un obstacle se trouve entre la station et le robot, la communication sera interrompue dans 70% des cas. En conséquence, on a pu extraire des informations sur la distance et les obstacles sur la Table 5.7.

Tableau 5.7 – CPT *a priori* de la RSSI en fonction de la distance entre le robot et la station et la présence d'obstacles

| Obs | Non | | Oui | |
|---------------------------|----------------|----------------|----------------|----------------|
| | < <i>seuil</i> | > <i>seuil</i> | < <i>seuil</i> | > <i>seuil</i> |
| $P(RSSI \geq -48 Obs, D)$ | 1 | 0 | 0.3 | 0 |
| $P(RSSI < -48 Obs, D)$ | 0 | 1 | 0.7 | 1 |

5.4.3 Influence du retard et du RSSI sur le pourcentage des pertes de paquets :

Dans cette partie, nous considérons que si le RSSI est inférieur à un seuil (*seuil* = $-48dBm$) les paquets transmis sont perdus. De plus, si le retard est supérieur à une

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

période d'échantillonnage, le paquet est considéré comme ancien et il sera jeté automatiquement. En conséquence, nous pouvons déterminer les probabilités *a priori* des différents états des pertes (PL) en fonction du retard (Δ_t) et du $RSSI$. La Table 5.8 présente toutes ces probabilités.

Tableau 5.8 – CPT *a priori* des pertes de paquet en fonction du retard et du RSSI

| Δ_t | $\Delta_t < T_e$ | | $\Delta_t > T_e$ | |
|-------------------------|------------------|---------|------------------|---------|
| $RSSI(dBm)$ | ≥ -48 | < -48 | ≥ -48 | < -48 |
| $P(0\% \leq PL < 40)$ | 1 | 0 | 0 | 0 |
| $P(40\% \leq PL < 60)$ | 0 | 0 | 0 | 0 |
| $P(60\% \leq PL < 100)$ | 0 | 1 | 1 | 1 |

5.4.4 Étude du délai de transmission :

Dans cette partie, nous nous intéressons à l'étude du délai de transmission afin de trouver la CPT *a priori* du nœud retard (Δ_t) en fonction des paramètres ayant une influence directe sur ce dernier. Le terme retard ou délai de transmission désigne le temps que met un paquet pour être acheminé d'une source vers une destination. Ce temps prend en compte principalement deux sous-délais : le délai dans la file d'attente (Buffer) des nœuds et le délai de transmission (τ) du paquet sur le médium physique. Dans notre cas, on traite seulement le délai de propagation du paquet sur le médium physique. Le problème de la synchronisation des horloges a été traité dans de nombreux travaux [Wehbi et al., 2008] mais, pour ne pas compliquer notre étude, nous supposons que toutes les horloges des nœuds sont synchronisées. En nous basant sur l'étude de [Bianchi et al., 1996], qui a proposé une méthode d'estimation de retard dans 802.11 en utilisant le mode DCF, nous pouvons estimer le retard de transmission entre deux nœuds à partir des données de la charge du réseau, la longueur des trames et aussi la taille de la fenêtre de contention (CW) qui est un temps aléatoire entre deux transmissions successives. La probabilité de collision est calculée en utilisant l'approximation proposée par [Bianchi et al., 1996] (voir chapitre 4). En nous basant sur les équations du chapitre précédent 4.3.17, 4.3.9 et 4.3.10, la courbe de " p " en fonction de " N " et la courbe du " Δ_t " estimé en fonction de " p " ont été tracées pour définir les seuils de " N " et de " Col " pour que le retard dépasse une période d'échantillonnage " $T_e = 400ms$ ". On a remarqué que la probabilité de collision dépasse 0.8 si le nombre de nœuds dépasse 14 nœuds, le seuil de probabilité de collision qui fait que le retard dépasse $T_e = 400ms$ est 0.8. On peut donc extraire les deux tableaux de CPT *a priori* des nœuds Δ_t et Col . La Table 5.9 présente la CPT *a priori* du retard en fonction de la probabilité de collision. La Table

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

5.10 présente la CPT *a priori* de la collision en fonction du nombre de nœuds voulant communiquer.

Tableau 5.9 – CPT *a priori* du retard en fonction de la probabilité de collision

| Col | ≤ 0.8 | > 0.8 |
|------------------------|------------|---------|
| $P(\Delta_t \leq T_e)$ | 1 | 0 |
| $P(\Delta_t > T_e)$ | 0 | 1 |

Tableau 5.10 – CPT *a priori* de la collision en fonction du nombre de nœuds voulant communiquer

| N | ≤ 14 | > 14 |
|-------------------|-----------|--------|
| $P(Col \leq 0.8)$ | 1 | 0 |
| $P(Col > 0.8)$ | 0 | 1 |

5.4.5 Détermination des probabilités marginales des nœuds sans parent :

Dans cette partie, les probabilités marginales des nœuds sans parent seront déterminées. Pour le nœud "Etat du Robot", on considère qu'un diagnostiqueur interne utilisant les méthodes de diagnostic classiques, peut détecter et localiser les défaillances liées au robot (moteurs, capteurs ...). Pour le RB que nous avons proposé, ce qui nous intéresse est l'état du robot en général (normal, dégradé ou défaillant). En ce qui concerne les deux nœuds D et N , ce sont des nœuds mesurés, mais on doit déterminer des probabilités *a priori* initiales. Et comme à l'état initial, on ne peut pas connaître la distance ni le nombre de nœuds communicants, nous avons mis une probabilité d'initialisation pour chaque état. Cette probabilité sera changée dès que l'on mesure les valeurs de chaque variable ou lors d'une observation.

5.5 Aide à la décision et diagnostic

Après avoir déterminé toutes les probabilités conditionnelles et marginales *a priori*, nous pouvons passer maintenant à comment prendre une décision et comment diagnostiquer le système en calculant les probabilités *a posteriori* en fonction des évidences ou des mesures.

Tableau 5.11 – Observations

| Noeud | Observations |
|-------|--------------------|
| D | $P(D < Seuil) = 1$ |
| N | $P(N < 14) = 1$ |

5.5.1 Aide à la décision

Prendre une décision dans notre application est très important, que ce soit au niveau du changement de station ou bien de la commutation de contrôleur. Pour cela, on a besoin d'estimer ou de déterminer l'état de la QoC et de la QoS pour pouvoir prendre une décision adéquate à la situation. En fonction des observations sur le système, les tables de probabilités *a posteriori* $P(QoS|Observations)$ et $P(QoC|Observations)$ seront calculées par inférence en utilisant l'algorithme d'inférence JLO [Naïm et al., 2004]. Supposons qu'on ait des informations sur la distance entre le robot et la station et aussi sur le nombre de nœuds qui veulent communiquer (robots et stations) (Table 5.11). En actualisant les tables de probabilités de ces deux nœuds, les probabilités *a posteriori* de chaque nœud du réseau seront calculées et le résultat est présenté sur la Fig.5.8.

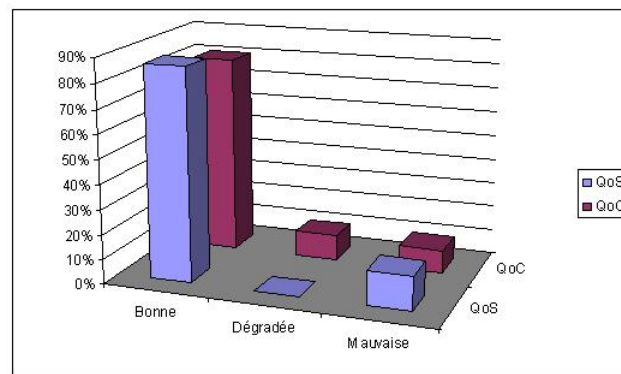


Figure 5.8 – Etat du RB après convergence avec une connaissance sur la distance et le nombre de nœuds communicants

Nous remarquons qu'avec ces évidences, la QoC et la QoS sont dans un état "bon". Changeons maintenant nos évidences sur le RSSI. Le robot sort de la zone de couverture de la station donc la valeur du RSSI est inférieure au seuil. La Fig.5.9 montre qu'avec cette connaissance, la QoC est mauvaise avec une probabilité de 49% et la QoS est mauvaise à 100%. En connaissant des informations sur le système, nous avons pu qualifier la QoC du robot et la QoS du réseau. En conséquence, on peut prendre une décision en fonction de l'état de la QoC et de la QoS pour changer la station ou bien commuter le contrôleur vers un contrôleur embarqué.

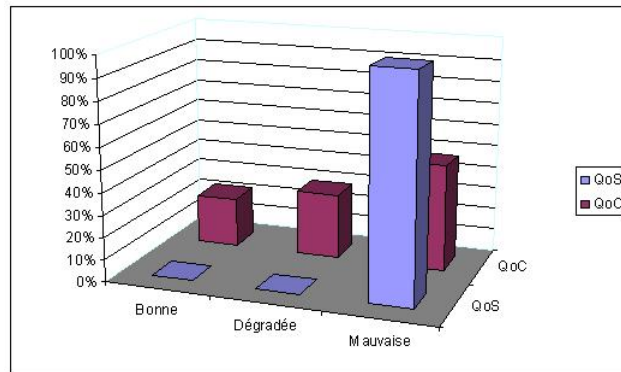


Figure 5.9 – Etat du RB après convergence avec une connaissance sur la distance et le nombre de nœud communicants et le RSSI

5.5.2 Diagnostic

Si nous observons une dégradation de la QoC, nous pouvons en utilisant le RB proposé détecter et localiser la cause de la dégradation de la QoC. Avec les connaissances ou les évidences adoptées, à savoir, la distance est inférieure au seuil, le nombre de nœuds est inférieur au seuil aussi et il n’y a pas d’obstacle, le RB peut localiser la cause. La seule raison qui peut causer cette dégradation est la défaillance du robot.

5.6 Apprentissage de paramètres

Après avoir déterminé les probabilités du réseau bayésien proposé en utilisant une étude statistique et en se basant sur l’expertise sur le système, nous essayons maintenant d’affiner les tables de probabilités a priori grâce à une base de données du système réel et aussi du système simulé. Cette base de données a été obtenue en se basant sur quelques critères définis pour ce type de systèmes (robot unicycle et un réseau sans fil). En particulier, nous proposons dans cette section un critère d’évaluation de la QoS qui permet d’enlever l’hypothèse forte représentée par le tableau 5.3. De même, un critère de QoS qui prend en considération la mission du robot, la QoC et l’état du robot. Des critères de QoS, QoC seront proposés dans ce chapitre. Ces critères sont spécifiques aux véhicules mobile et les missions envisagées, et ne peuvent pas être appliqués sur un autre système sans changement.

5.6.1 Critère de la QoS

Comme précisé précédemment, la QoS est influencée particulièrement par les pertes de paquets et aussi le retard. Pour construire la base de données, nous avons besoin de

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

quantifier la QoS pour ce système, et de la qualifier ensuite par des plages de valeurs qui définissent les modes (ou les états) de la QoS. Pour cela, nous proposons le critère suivant :

$$QoS = \max\left(\frac{(1 - PL) * (nT_e - \Delta_t) * \rho_i}{nT_e}, 0\right) \quad (5.6.1)$$

PL étant la probabilité de pertes de paquets, elle est définie par des tests sur le système réel. Δ_t est le retard entre le capteur et le contrôleur. ρ_i est une variable qui dépend de la mission i du système ($i \in \{obstacle, cible\}$) tel que $\rho_{obstacle} < \rho_{cible}$ ($\rho_{cible} = 1$ et $\rho_{obstacle} = 0.5$). T_e est la période d'échantillonnage. n est une constante qui caractérise le nombre maximal d'échantillons perdus que l'on peut tolérer pour que le système fonctionne parfaitement en débarqué.

La constante n est définie pour que le véhicule mobile ne heurte pas un obstacle. Puisque la vitesse moyenne du robot est de $0.1m/s$ et $T_e = 400ms$ le robot se déplace donc de $40mm$ à chaque période d'échantillonnage, et comme la zone de sécurité d'un obstacle est fixée à $100mm$, le robot heurte l'obstacle en deux périodes d'échantillonnage et demi. Par conséquent et pour des raisons de sécurité on prend $n = 2$. La Fig. 5.10 montre la courbe de la dégradation de la QoS en fonction du nombre de noeuds communicants.

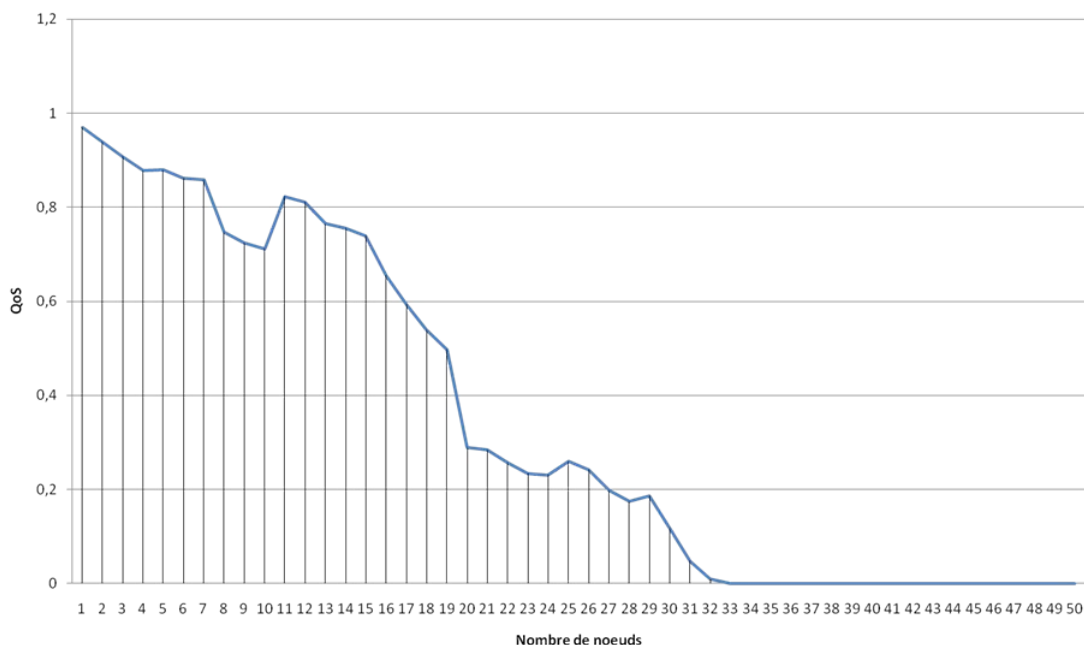


Figure 5.10 – QoS en fonction du nombre de noeuds

5.6.2 Critère de la QoC

Comme précisé dans la section précédente, la QoC est influencée principalement par trois paramètres, à savoir la QoS, l'état du robot ainsi que la mission du système. Ceci nous amène à proposer le critère suivant :

$$QoC = QoS * \kappa_j \tag{5.6.2}$$

κ_j étant une variable qui dépend de l'état j du robot tel que $j \in \{\text{Bon, dégradé ou mauvais}\}$ et $\kappa_{bon} > \kappa_{deg} > \kappa_{mauvais}$; dans notre cas on a pris $\kappa_{bon} = 1, \kappa_{deg} = 0.5$ et $\kappa_{mauvais} = 0$. La courbe 5.11 montre la variation de la QoC en fonction de la QoS dans deux cas différents (Atteinte d'une cible et évitement d'obstacle). On remarque que le robot a besoin de plus de QoS dans le cas de l'évitement d'obstacle.

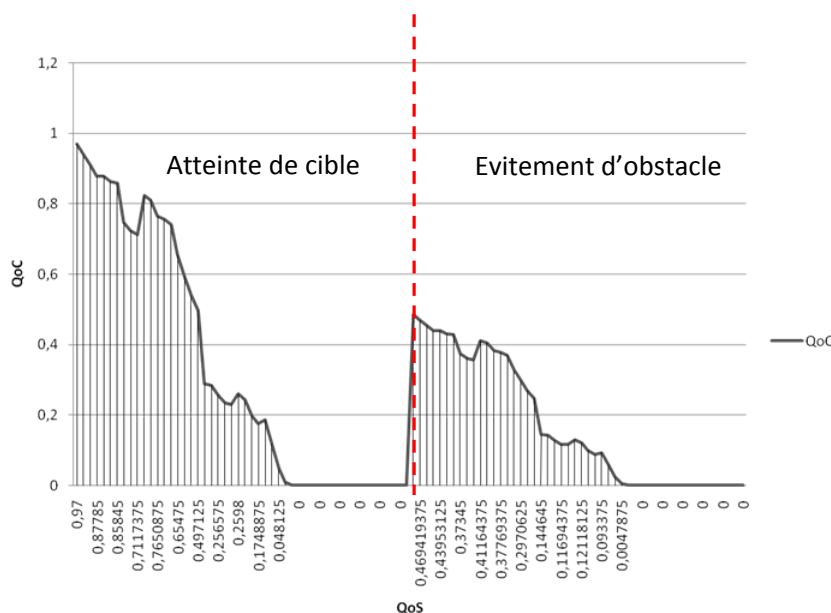


Figure 5.11 – QoC en fonction de la QoS avec la mission atteinte de cible et aussi évitement d'obstacle

La formule 4.3.9 a été utilisée pour déterminer la probabilité de collision en fonction de nombre de noeuds voulant communiquer.

Maintenant, en fonction de ces critères, on a pu construire une base de données basée sur différents types de données : relations mathématiques déterministes (les équations 4.3.17, 5.6.1 et 5.6.2), historique du système réel ainsi que simulations sur Matlab/Simulink et la toolBox TrueTime. Ces données seront fusionnées ensuite en utilisant le logiciel BayesiaLab pour déterminer les différents modes de chaque variable et aussi les tables de

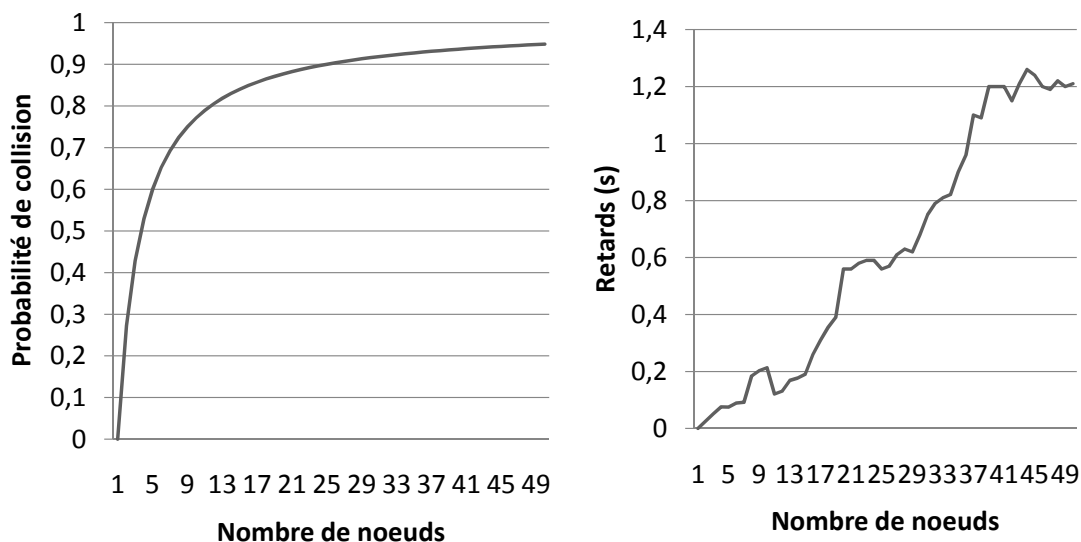


Figure 5.12 – Probabilité de collision et retard en fonction des nombre de noeuds voulant communiquer

probabilités associées par l’estimation statistique. Comme toutes les variables sont observables, la méthode la plus simple est l’estimation statistique, qui consiste à estimer les probabilités d’un événement par la fréquence d’apparition de cet événement. L’estimateur est alors donnée par [Naïm et al., 2004] :

$$\hat{P}(X_i = x_k | pa(X_i) = x_j) = \hat{\theta}_{i,j,k}^{LH} = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}} \quad (5.6.3a)$$

où $N_{i,j,k}$ est le nombre d’événements dans la base de données pour lesquels la variable X_i est dans l’état x_k et ses parents sont dans l’état x_j .

L’utilisation du logiciel BayesiaLab a été d’une grande utilité pour l’apprentissage statistique des paramètres. Toutefois, plusieurs étapes sont nécessaires pour construire le réseau bayésien associé au système étudié.

1. Importation des données (sous format .txt)
2. Sélection et typage des variables
3. Discrétisation des variables continues
4. Génération du réseau bayésien.

Après avoir suivi les étapes nécessaires pour construire le graphe sur le logiciel, l’étape de validation du réseau est indispensable pour vérifier sa conformité à la réalité.

5.6.3 Validation du réseau bayésien trouvé par apprentissage

La validation du réseau bayésien fourni par le logiciel BayesiaLab en utilisant la méthode par apprentissage est indispensable. Nous commençons d’abord par définir les observation dans le cas idéal du réseau et du robot (aucun problème n’est détecté), c’est-à-dire, les noeuds observés $\{Col, D, Obst, M, R\}$ sont dans leurs états respectifs $Col \leq Col_{seuil}$, la distance D est suffisante pour pouvoir communiquer, il n’y a pas d’obstacles entre la station et le robot, la mission du robot est l’atteinte d’une cible et le robot est dans un état de fonctionnement normal. La fig.5.13 montre les probabilités (en bleu) de chaque noeud en fonction des observations (en vert). On remarque que

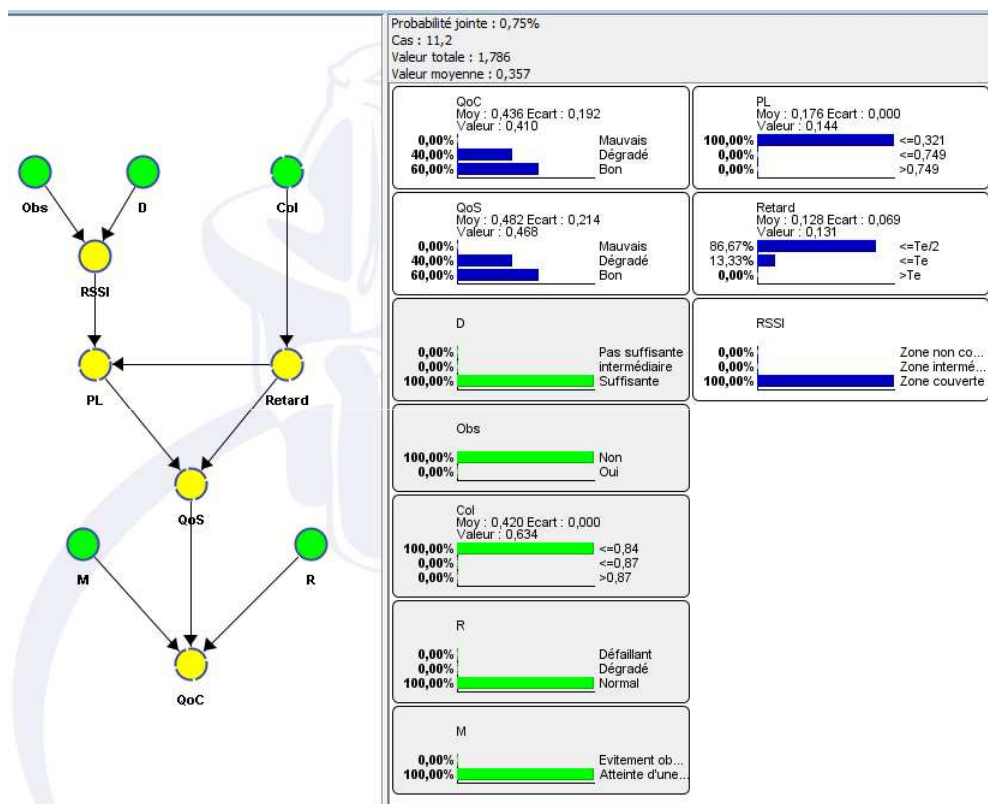


Figure 5.13 – Evolution des probabilités de chaque noeud en fonction des observations dans ce cas et avec les observations choisies auparavant la QoS et la QoC sont dans la majorité des cas bonnes. Cela est dû aux bonnes conditions d’utilisation du réseau sans fil et aussi au fonctionnement normal du robot.

La Fig. 5.14 montre deux scenarios d’observations et les probabilités de la QoS et la QoC (les observations en vert et les probabilités conditionnelles en bleu). Nous avons proposé une approche basée sur les réseaux bayésiens dans le but de qualifier à la fois la QoS et la QoC, et de permettre l’aide à la décision et le diagnostic d’un système. Le

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

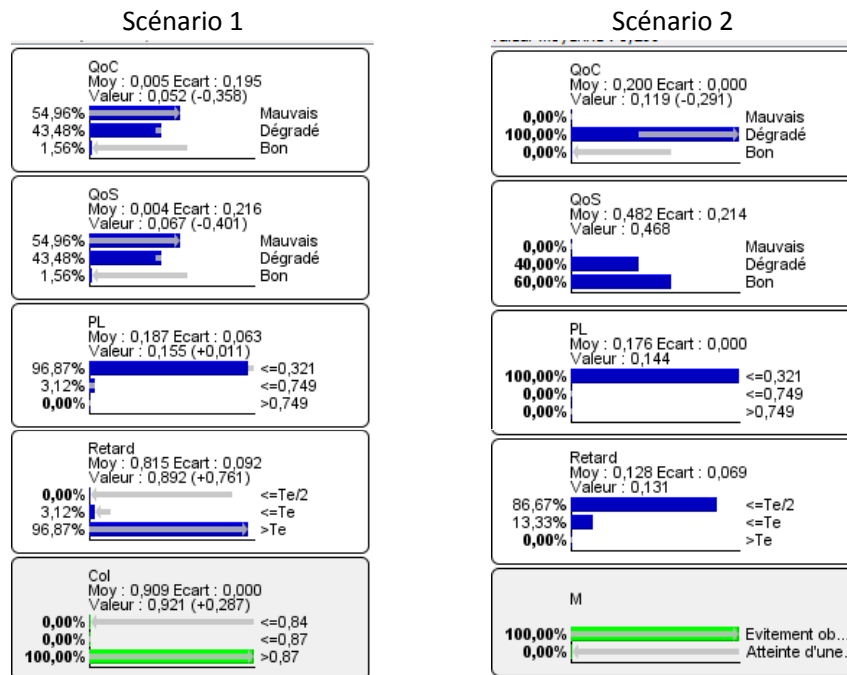


Figure 5.14 – Evolution des probabilités de chaque noeud en fonction des observations

choix de cette approche est justifié par la possibilité d'utiliser le même modèle pour l'aide à la décision et le diagnostic du système et aussi la disponibilité de différentes sources d'informations.

Pour déterminer les probabilités *a priori* de chaque noeud, nous avons traité dans cette section l'influence des différents paramètres sur chacun des noeuds du RB proposé. Nous avons présenté ensuite quelques scénarios en ajoutant à chaque fois des observations. Le RB proposé a été utilisé pour le diagnostic et l'aide à la décision.

L'objectif le plus intéressant de cette étude est d'utiliser le même modèle du réseau bayésien proposé pour pronostiquer ou anticiper la dégradation de la QoS du réseau et la QoC du robot pour éviter qu'elle ne se produise. L'implémentation de ce RB dans notre simulateur du robot commandé en réseau est en cours.

Vu la complexité de calculer les probabilités *a posteriori* de chaque noeud du RB, la distribution du RB est envisageable pour la suite et la communication entre sous RB sera assurée par le réseau, ce qui nous amène à étudier l'influence de l'intégration du réseau sur la performance du réseau bayésien distribué.

5.7 Réseaux bayésiens distribués

Les systèmes embarqués autonomes sont limités en terme de mémoire, temps de calcul et aussi d'énergie. Pour cela, l'intégration des réseaux bayésiens dans ce genre de systèmes n'est pas souhaitable vu la complexité en terme de calcul et de mémoire de ces derniers, ce qui entraîne une consommation d'énergie importante et ensuite diminue leur autonomie. Pour remédier à ce problème, une stratégie de distribution du réseau bayésien proposé dans la section précédente a été étudiée. La Fig. 5.15 montre l'architecture de distribution du réseau bayésien en question pour un environnement multi-robots. Seuls les nœuds qui sont observés par le robot (à savoir la QoC, QoS, l'état du robot et la mission) sont embarqués sinon le reste sera débarqué (dans la station de contrôle). pour l'échange d'information QoS entre le robot et la station de contrôle, on a utilisé l'algorithme de Pearl (voir annexe A). Comme notre graphe présente une structure en arbre, il est par définition connexe et sans circuit, autrement dit, il existe une unique chaîne entre deux noeuds du RB. Cette propriété permet de partitionner le graphe relativement à un noeud qui est dans notre cas le noeud QoS ce qui nous permet après d'envoyer à travers le réseau sans fil les π - message de la station de contrôle au robot et les λ - message du robot à la station de contrôle (voir algorithme de Pearl dans l'annexe A pour le calcul des π - message et λ - message). Ces messages sont envoyés sous forme de trame de contrôle (la trame la plus petite possible).

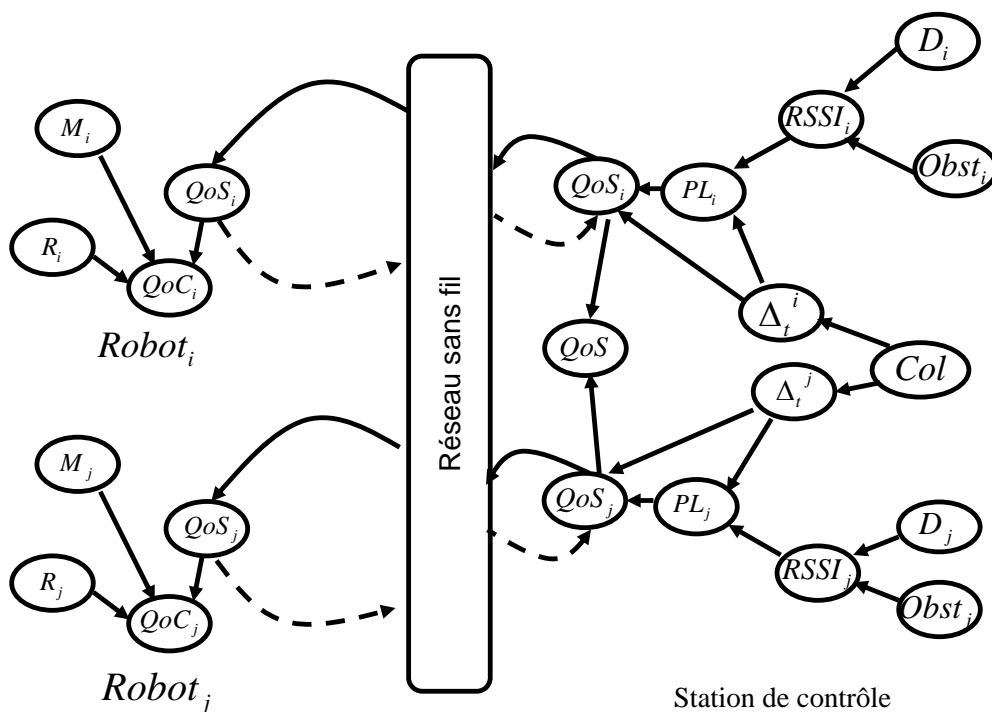


Figure 5.15 – Réseau bayésien distribué

5.7.1 Prise de décisions et diagnostic distribué

Après avoir proposé un modèle distribué du RB représentant le système et définir toutes les probabilités *a priori*, le graphe acyclique sera utilisé pour diagnostiquer en-ligne la dégradation de la QoC et de la QoS et aussi pour prendre les décisions en-ligne pour éviter la dégradation et garantir un niveau acceptable de QoC et/ou QoS. Par contre, pour diagnostiquer la cause de la dégradation de la QoC, le réseau bayésien a besoin d'informations et de connaissance sur la QoC du robot. En effet, pour connaître en-ligne l'état de la QoC (bonne, dégradée ou mauvaise), un critère de la QoC est proposé. Ce critère prend en considération la vitesse du robot et son orientation par rapport à la cible à atteindre (Voir Algorithme 4). Cet algorithme donne au RB des informations sur la QoC du robot. Si la distance entre le robot et la station de contrôle augmente, la QoC est considérée comme mauvaise. Et si la distance diminue, l'erreur d'orientation sera vérifiée. Si l'erreur d'orientation diminue, la QoC est bonne sinon elle est dégradée. En utilisant cet algorithme, on peut calculer la probabilité pour chaque état durant toute la trajectoire. La Fig. 5.16 montre la probabilité de chaque état de la QoC (bon, dégradé ou mauvais) dans le meilleur cas (bonne QoS, pas de défauts sur le robot). Sur cette figure, on remarque qu'il y a deux périodes à savoir la période transitoire et la période permanente. La période transitoire correspond à la période pendant laquelle on n'a pas beaucoup d'information sur la QoC du robot. Mais, au fur et à mesure le robot peut quantifier sa QoC avec précision ce qui est représenté par la période permanente (Fig .5.16).

Algorithm 4 QoC criterion

```
if  $d = 0$  and  $\dot{d} = 0$  then  
     $QoC = \text{bonne}$   
else if  $\dot{d} < 0$  then  
  
    if  $\dot{\theta}(k) \leq 0$  and  $\dot{\theta}(k-1) \leq 0$  then  
         $QoC = \text{good}$   
    else  
         $QoC = \text{dégradée}$   
    end if  
else  
     $QoC = \text{mauvaise}$   
end if
```

Suivant les informations sur la QoC, le résultat donné par le réseau bayésien concernant le diagnostic de la cause de la dégradation de la QoC, le robot décide de changer de station de contrôle avec laquelle il communique si il y en a une autre qui assure un niveau suffisant de QoS pour laquelle la QoC s'améliore [Mechraoui et al., 2009]. Le robot

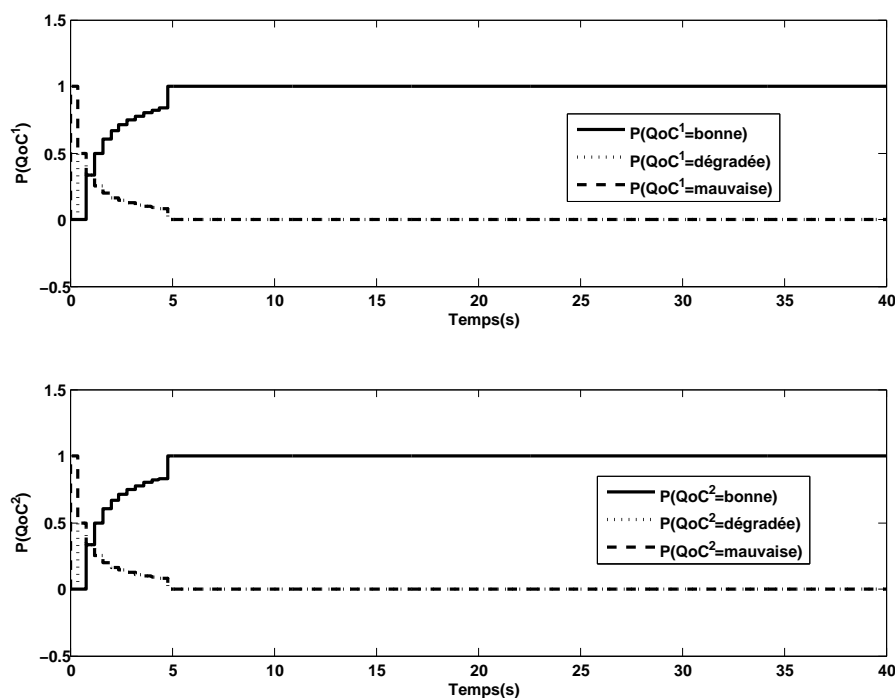


Figure 5.16 – $P(QoC)$ des deux robots dans le meilleur cas

peut aussi envisager de changer de contrôleur s'il n'y a pas d'autres stations qui peuvent assurer un niveau suffisant de QoS. Le contrôleur sera donc embarqué. L'algorithme 5 illustre toutes les situations pour garantir toujours au minimum une QoC dégradée en reconfigurant soit l'architecture de commande soit de communication.

Dans l'algorithme 5, S_c représente l'ensemble des stations candidates.

Algorithm 5 Decision algorithm

```

if  $QoC = \text{bad}$  and  $P(Col = \text{yes}) \geq Col_{Threshold}$  and  $D < D_{Threshold}$  and  $S_c = \Phi$  then
    Switch to embedded controller
else if  $QoC = \text{mauvaise}$  and  $S_c = S_j$  then
    Execute Horizontal Handoff
else
    Continue the communication with  $S_i$ 
end if

```

5.7.2 Priorités dynamiques

On peut aussi utiliser le réseau bayésien distribué pour déterminer la priorité de chaque robot, s'il est en situation critique, ou bien s'il n'a pas besoin de communication. Pour cela, en rendant le noeud QoC observable en utilisant l'algorithme 5, nous pouvons

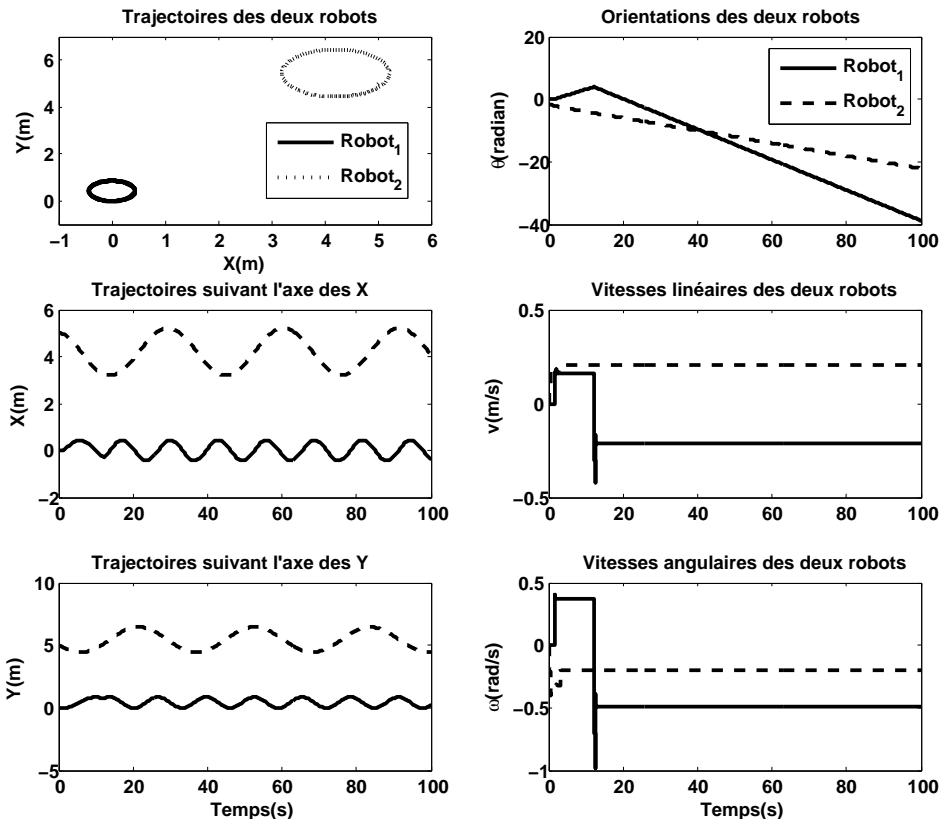


Figure 5.17 – Trajectoires des deux robots suivant l’axe des X et des Y avec positions initiales des deux robots $(x_0^1, y_0^1, \theta_0^1) = (0, 0, 0)$ et $(x_0^2, y_0^2, \theta_0^2) = (5, 5, -\frac{\pi}{2})$ sous les limites du réseau

inférer en fonction de l’état de la QoC si on a besoin de plus de ressources réseau ou non du côté robot. Si on a besoin de plus de ressources, le robot demande la priorité. La station de son côté et avec la partie embarquée du réseau bayésien étudie en fonction des observations sur les noeuds observables la possibilité d’allouer une bande passante plus large au robot demandeur. Si la station peut libérer quelques robots en raison de leurs dynamiques ou missions qui n’ont pas besoin de mise à jour, elle peut donc allouer plus de ressources aux demandeurs sinon, elle ordonne au robot demandeur de switcher vers un contrôleur embarqué.

5.7.3 Résultats et simulation

Pour vérifier les algorithmes proposés et l’efficacité du réseau bayésien pour le diagnostic et la prise de décision, deux scénarios ont été étudiés.

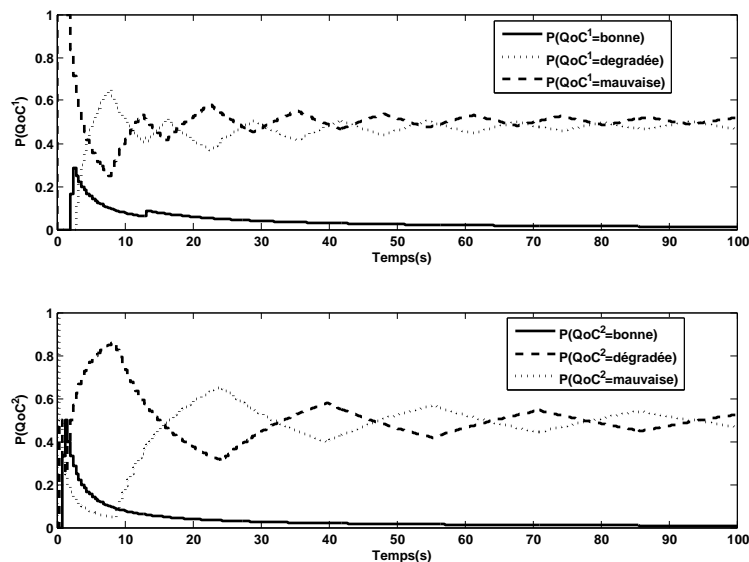


Figure 5.18 – $P(QoC)$ des deux robots dans le cas du réseau chargé et collisions

5.7.4 Scénario 1

Dans le premier scénario, deux robots communiquent avec une station de contrôle. Les robots sont toujours dans la zone de couverture de la station, ce qui n'entraîne pas la perte de communication. Donc, la probabilité $P(D < D_{Threshold}) = 1$ pour chaque robot. Pour limiter les ressources du réseau sans fil et ainsi étudier le comportement des robots sous les contraintes du réseau, le débit du réseau sans fil a été forcé à 400bits/s . Avec ce débit, et dans le meilleur des cas, les trames capteurs de chaque robot et celles du contrôleur seront transmises. Mais, s'il y a la moindre collision, les paquets seront perdus et donc la QoC se dégrade progressivement jusqu'à ce qu'elle devient "mauvaise". En utilisant l'algorithme 4, le robot peut qualifier la QoC et propage l'information via le RB. En effet, en fonction des observations, le RB peut localiser les causes possibles de la dégradation de la QoC du robot et la QoS du réseau sans fil.

Dans ce scénario, les deux robots ne peuvent pas communiquer avec la station de contrôle (voir tâche contrôleur de la Fig.5.20) ce qui donne une évidence que la QoC est mauvaise (les robots tournent avec une vitesse linéaire et angulaire constantes Fig.5.17), les deux robots ne peuvent pas atteindre la cible. On remarque aussi, sur la Fig. 5.18, que l'état de la QoC oscille entre dégradé et mauvais.

Durant ce scénario, le RB calcule la probabilité *a posteriori* $P(V_i | QoC = bad, D < D_{Threshold})$ pour chaque nœud du réseau en utilisant le théorème de Bayes, la loi marginale et la loi jointe (V_i sont les nœuds du RB). En analysant les probabilités *a posteriori* calculées par le RB, la cause la plus probable de la mauvaise QoC est les collisions avec $P(Col = yes | QoC = bad, D < D_{Threshold}) \approx 0.9$. La probabilité *a posteriori*

Chapitre 5. Prise de décision et diagnostic en-ligne de plusieurs robots mobiles commandés en réseau en utilisant les RB distribués

$P(Col|QoC = bad, D < D_{Threshold})$ est calculée comme suit :

$$P(Col|QoC = bad, D < D_{Threshold}) = \frac{P(QoC = bad, D < D_{Threshold}|Col)P(Col)}{P(QoC = bad, D < D_{Threshold})}$$

$$P(QoC = bad, D < D_{Threshold}|Col) = \frac{\sum_{QoS} \sum_R \sum_{PL} \sum_{RSSI} \sum_{Obst} \sum_{\Delta_t} P(QoC, QoS, R, PL, RSSI, Obst, D, \Delta_t)}{\sum_{QoS} \sum_R \sum_{PL} \sum_{RSSI} \sum_{Obst} \sum_{\Delta_t} \sum_{Col} P(QoC, QoS, R, PL, RSSI, Obst, D, \Delta_t, Col)}$$

$$P(QoC = bad, D < D_{Threshold}) = P(QoC = bad|QoS, R, PL, RSSI, Obst, D, \Delta_t, Col) \times P(D < D_{Threshold})$$

Après exécution de l'algorithme 5, $Robot_1$ décide à l'instant $t = 5s$ d'interrompre la communication avec la station de contrôle, de reconfigurer la stratégie de contrôle et commuter vers un contrôleur embarqué. Avec cette stratégie, $Robot_1$ garde au minimum une QoC dégradée et ainsi il arrive à rejoindre la cible (Fig.5.19). En conséquence, $Robot_2$ continue la communication avec la station de contrôle avec une bonne QoS et aussi une meilleure QoC. La Fig.5.20 montre l'évolution de communication des deux robots avant et après la reconfiguration de l'architecture de commande. Sur cette figure, après avoir reconfiguré l'architecture de commande et de communication du $Robot_1$, la probabilité de collision sera nulle et la station de contrôle reçoit la position du $Robot_2$ et envoie à son tour les vitesses linéaire et angulaire désirées.

Dans ce scénario et avec les mêmes conditions du réseau sans fil, les deux robots ne peuvent pas atteindre leurs cibles respectives (Fig. 5.17) par contre avec la reconfiguration proposée, les deux robots peuvent atteindre leurs cibles sans problème (Fig. 5.19).

5.7.5 Scénario 2

Durant ce scénario, le débit du réseau sans fil a été remis à (800kbit/s). On suppose aussi qu'à partir de $t = 5s$ de simulation, le capteur lié au moteur droit du $Robot_1$ est défaillant. Le défaut est simulé en ajoutant un step sur $\omega_{right} = 5rad/s$. En conséquence, la QoC sera mauvaise (voir Fig.5.21). Après avoir exécuté l'inférence bayésienne, le RB calcule la probabilité conditionnelle *a posteriori* $P(R = faulty|QoC = bad, QoS = good) = 1$. Donc, le RB a localisé et isolé la cause de la dégradation de la QoC à savoir l'état défaillant du robot.

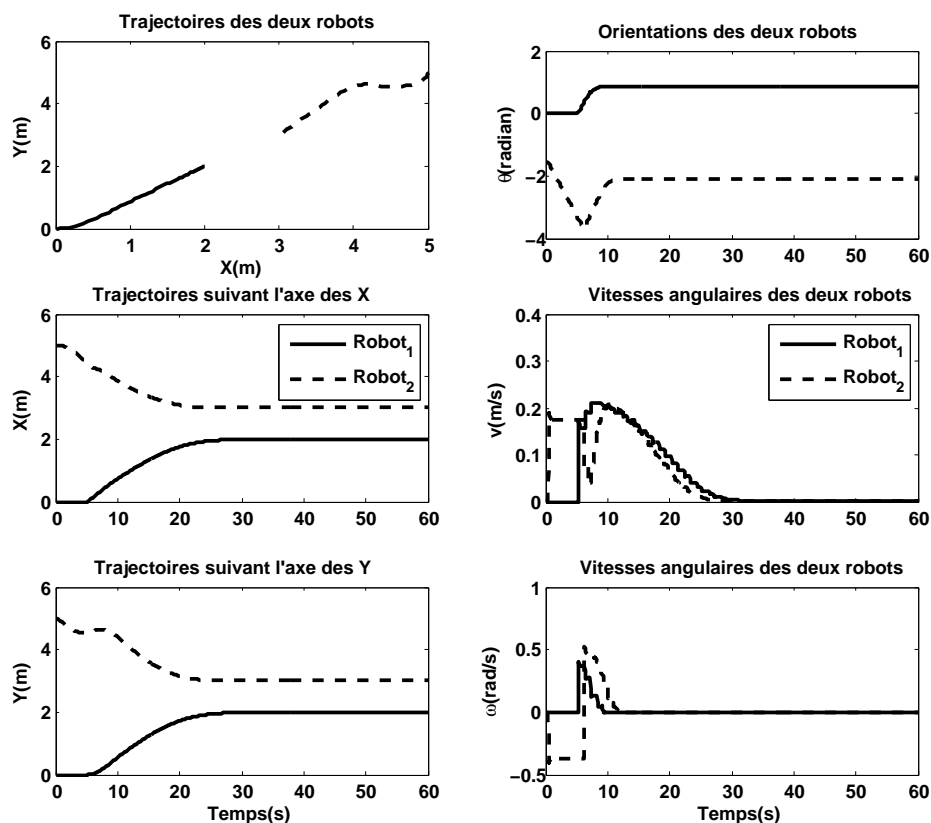


Figure 5.19 – Trajectoires des deux robots pour atteindre leurs cibles sous les limites du réseau et avec la reconfiguration de la commande et l’architecture de communication

5.8 Conclusion

Dans ce chapitre, une nouvelle approche basée sur les réseaux bayésiens distribués a été proposée pour modéliser la Qualité de Service et la Qualité de Contrôle des WNCS. Ce modèle prend en compte tous les paramètres et variables influençant la QoS du réseau et la QoC du robot. L’utilisation du réseau bayésien est justifiée par le comportement stochastique du réseau sans fil. Les principaux objectifs de l’utilisation des réseaux bayésiens sont le diagnostic, et la prise de décision en-ligne pour la reconfiguration de l’architecture de communication et/ou de l’architecture de commande. Le point fort des réseaux bayésiens est que le même modèle peut être utilisé pour le diagnostic et l’aide à la décision. Le modèle proposé accompli le but souhaité, à savoir l’identification de la dégradation de la QoC et de la QoS.

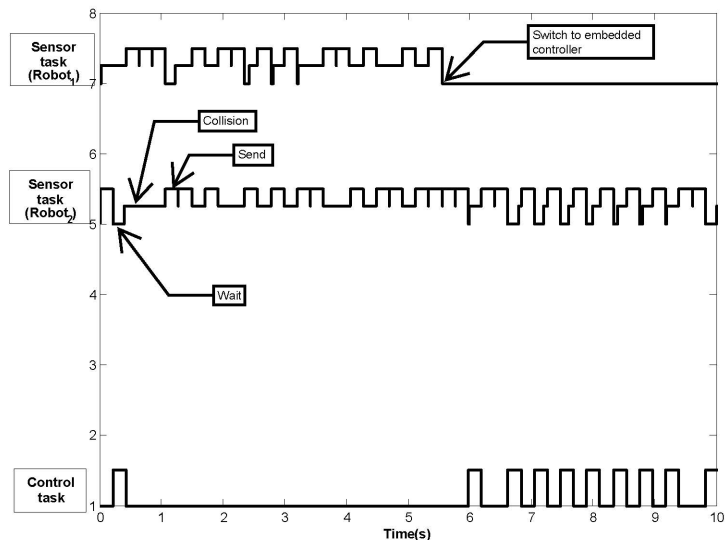


Figure 5.20 – Ordonnancement des deux tâches capteur de chaque robot et la tâche contrôleur

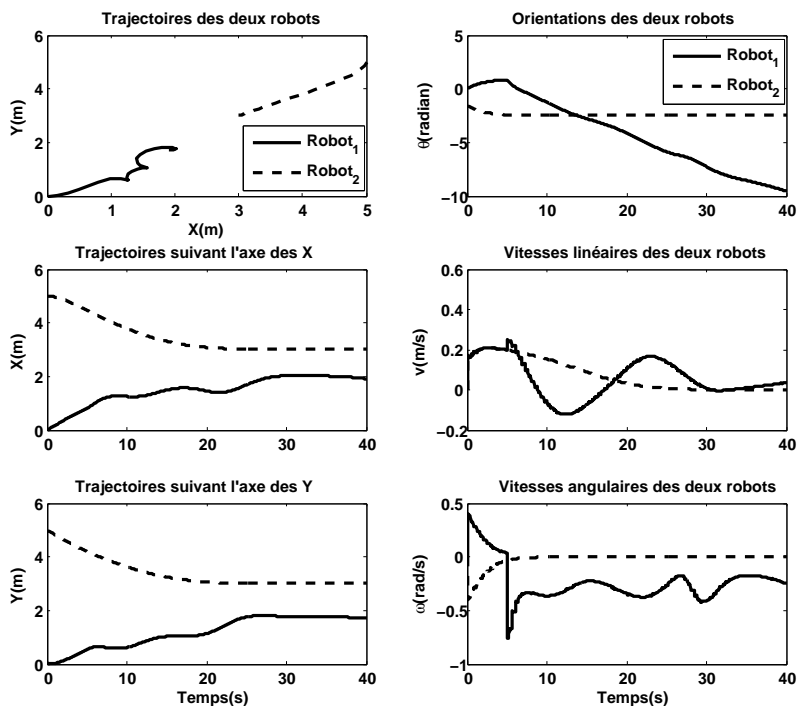


Figure 5.21 – Trajectoires des deux robots avec un défaut capteur du $Robot_1$ à l'instant $t = 5s$

Chapitre 6

Conclusion générale

Dans le contexte technologique et économique actuel, l'utilisation des réseaux sans fil est devenue primordiale dans le domaine de la surveillance, la productique, le trafic routier, la robotique mobile etc. Toutefois, dans tous ces domaines et surtout pour la commande des systèmes temps réel et à sûreté critique, la performance et la sûreté de fonctionnement sont cruciales pour satisfaire les besoins et bien accomplir les missions. Cependant, et avec les technologies disponibles actuellement en termes de réseaux sans fil, l'utilisation des réseaux sans fil pour la commande des systèmes temps réel critiques peut influencer le bon fonctionnement de ceux-ci et nécessite d'être étudiées avec rigueur. Les pertes de paquets et les retards sont les principaux paramètres qui influencent la QoS du réseau et ainsi la performance du système commandé à travers un réseau et donc la QoC du système.

La QoS du réseau est un paramètre important pour le choix et l'utilisation du réseau sans fil. En revanche, la QoC est un paramètre à prendre en compte lors de l'intégration du réseau dans la boucle de commande. Il est toutefois difficile de faire le lien entre la QoS du réseau et la QoC du système étudié à cause du comportement stochastique du réseau sans fil. De plus, l'influence de la QoS sur la QoC d'une part et l'influence de la QoC sur la QoS complexifie l'étude de ces deux paramètres parce que chacune est dépendante de l'autre. Dans ce but, l'utilisation des réseaux bayésiens s'est avérée une bonne solution pour ce type de problèmes parce qu'ils permettent d'une part de connaître la cause de la dégradation de la QoS et/ou la QoC (diagnostic) et aussi de prendre une décision adéquate pour améliorer soit la QoC soit la QoS (prise de décisions). Nous avons donc proposé un modèle de réseau bayésien qui prend en considération les différents paramètres qui influencent la QoS du réseau sans fil et aussi la QoC du système et qui permet de prendre une décision que ce soit pour la reconfiguration de l'architecture de commande (commutation de contrôleur) ou pour la reconfiguration de l'architecture de communication (Ordonnancement, Handoff).

Chapitre 6. Conclusion générale

Ces deux reconfigurations permettent de garder une bonne QoC du système et donc d'assurer son bon fonctionnement.

Le système étudié dans ce mémoire est le robot mobile unicycle Khepera III. Ce Benchmark est utilisé pour tester nos approches dans le cadre de la co-conception des systèmes commandés à travers un réseau sans fil.

Dans le deuxième chapitre de ce rapport, nous avons pu survoler d'une façon globale les principales approches possibles pour traiter les NCSs (Analyse de stabilité, estimation pour les NCSs, synthèse de commande et co-conception). Nous avons situé notre travail dans l'état de l'art. Nous avons également présenté les différentes méthodes de diagnostic des systèmes (méthodes à base de modèles analytiques et méthodes à base de connaissances). En nous concentrant sur les méthodes à base de connaissances, nous avons présenté les différentes architectures possibles de diagnostic (centralisé, décentralisé et distribué).

Suite à l'application choisie dans le cadre de cette thèse (Robot mobile unicycle), le chapitre trois a pour but de présenter le robot Khepera III (Composition, modèle cinématique et dynamique) et de l'architecture de commande proposée. Nous avons donc détaillé cette architecture dans le cas de trois missions envisageables à savoir l'atteinte d'une cible, le suivi de trajectoire et l'évitement d'un ou plusieurs obstacles. Nous nous sommes intéressés aussi à l'architecture de commande temps réel (Programmation sous TrueTime pour la simulation et la programmation sous langage C pour l'expérimentation). Nous avons présenté aussi dans ce chapitre l'architecture de communication en spécifiant le protocole, la topologie et le réseau utilisé. Suite à cette présentation, nous avons montré quelques résultats expérimentaux et de simulation. Quelques difficultés ont été constatées pour l'implémentation de tout ces algorithmes sur le robot réel, telles que les erreurs liées à l'utilisation d'une seule source de mesures de position (odométrie). Cela est dû au manque de matériel et de temps pour utiliser un système de vision pour la position du robot et des obstacles. L'utilisation de la toolBox TrueTime a posé quelques problèmes et présente des limitations de stratégie d'ordonnancement, par exemple l'absence de la stratégie de GTS (expliquée dans l'annexe B) et qui permet de donner des priorités aux différentes tâches du système et ainsi améliorer la QoS du réseau par rapport à chaque tâche et sa criticité.

Dans le chapitre quatre, nous nous sommes intéressés à l'influence de l'intégration du réseau sans fil dans la boucle de commande. L'influence de la QoS du réseau sur la QoC du robot a été étudiée. Pour cela, nous avons fait une étude bibliographique sur l'évaluation de performance du réseau utilisé en termes de débit, probabilité de

transmission de paquet et retard d'un paquet, sous l'hypothèse que le réseau est saturé. Ensuite, nous avons proposé une méthode pour adapter le réseau aux besoins dynamiques de la commande (Handoff) et aussi pour adapter la commande aux besoins dynamiques du réseau (commutation de contrôleur) en étudiant tous les paramètres qui influencent la QoS et la QoC. Cette méthode était efficace mais présente quelques problèmes au niveau de la non prise en compte de la QoS et la QoC simultanément et de la non détection des défauts liés au réseau sans fil et au système, d'où l'intérêt du chapitre 5.

Le dernier chapitre a permis de prendre en considération toutes les contraintes pour bien diagnostiquer les dégradations simultanées de la QoS et de la QoC et puis prendre des décisions adéquates. L'utilisation des réseaux bayésiens a été présentée. Nous avons dans ce chapitre pu construire d'une façon qualitative et quantitative le réseau bayésien qui modélise le comportement du robot contrôlé en réseau. Une base de données provenant du robot réel, de la simulation et de l'expertise a été utilisée pour trouver les probabilités associées à chaque nœud du RB. L'utilisation des réseaux sans fil pour la communication, nous a permis de distribuer le RB pour alléger les calculs au niveau de chaque robot. Nous avons ensuite présenté les résultats de simulation de la prise de décision (pour reconfigurer le réseau ou la commande) et de diagnostic (pour détecter et localiser les défauts liés au robot et aussi au réseau sans fil). Les résultats sont satisfaisants en simulation. Le test de cette approche sur le robot réel n'a pas été fait par manque de temps.

Pour résumer, dans ce mémoire, l'architecture de communication de deux robots mobiles se déplaçant dans un espace 2D avec une station de contrôle fixe a été considérée comme un exemple pertinent de véhicules autonomes dialogant via une infrastructure de communication. Suivant les informations délivrées par le réseau bayésien distribué proposé, la reconfiguration de l'architecture de communication (Horizontal Handoff) et/ou de la stratégie de contrôle sont envisageables. Cette décision a pour objectif de garantir au minimum une QoC dégradée. La stratégie de contrôle peut être débarquée dans le cas d'une bonne QoS du réseau ou embarquée (dégradée) dans le cas de mauvaise QoS de la station courante et de la non-disponibilité d'une autre station qui assure un niveau meilleur de la QoS pour un bon niveau de QoC. Dans ce dernier cas, la loi de commande doit être dégradée en prenant en compte la mémoire et la vitesse du processeur du robot.

Les perspectives de ces travaux sont nombreuses. L'utilisation des réseaux bayésiens dynamiques est envisageable pour intégrer, en plus de l'aspect diagnostic et prise de décision, un aspect pronostic (prévoir ou anticiper la dégradation de la QoS et de la QoC) pour éviter la dégradation de la QoC et la QoS avant qu'elle ne se produise. Le même réseau bayésien pourrait être utilisé en ajoutant quelques nœuds dynamiques dans deux ou plusieurs laps de temps. La probabilité des différents états de la QoC et

Chapitre 6. Conclusion générale

de la QoS à $t + 1$ sachant les mêmes nœuds à t permet d'anticiper la dégradation de la QoS et de la QoC.

Une autre perspective évidente est de programmer tous ces algorithmes proposés dans le robot Khepera III et la station de contrôle pour vérifier leur efficacité sur le robot réel.

Annexe A

Réseaux bayésiens

Sommaire

| | |
|--|------------|
| A.1 Introduction | 151 |
| A.2 Définition des réseaux bayésiens | 152 |
| A.3 Approche qualitative | 153 |
| A.3.1 La représentation graphique de la causalité | 153 |
| A.3.2 D-séparation (bloquage) | 155 |
| A.4 Approche quantitative | 156 |
| A.4.1 Probabilités | 159 |
| A.4.2 Circulation d'information et indépendance conditionnelle | 163 |
| A.4.3 Conclusion | 163 |
| A.5 Inférence | 164 |
| A.5.1 Introduction | 164 |
| A.5.2 Algorithme de Pearl | 165 |
| A.5.3 Algorithme de l'arbre de jonction dit JLO | 173 |

A.1 Introduction

Les réseaux bayésiens, parfois appelés réseaux de croyance ou réseaux causaux probabilistes [Pearl, 1988], constituent l'un des formalismes les plus complets et les plus cohérents pour l'acquisition, la représentation et l'utilisation de connaissances par des ordinateurs [Naïm et al., 2004]. Cette technologie connaît de plus en plus d'applications, depuis le contrôle de véhicules autonomes jusqu'à la modélisation des risques opérationnels, en passant par le data mining ou la localisation des gènes, sans oublier le diagnostic et le pronostic qui constituent les principaux domaines d'applications dans nos travaux. Les réseaux bayésiens, qui doivent leur nom aux travaux de Thomas Bayes au XVIIIe

siècle sur la théorie des probabilités, sont le résultat de recherches effectués dans les années 80, dues à J. Pearl [Pearl, 1988]. L'objectif initial de ces travaux était d'intégrer la notion d'incertitude dans les systèmes experts [Naïm et al., 2004]. En général, un expert est capable de porter un jugement sur une situation, même en l'absence de toutes les données nécessaires. En diagnostic des systèmes complexes, par exemple, on peut trouver les mêmes combinaisons d'observations qui correspondent aux mêmes diagnostics mais avec différents niveaux d'incertitude, il n'y a donc pas des règles strictes qui permettent de passer systématiquement d'un ensemble d'observations à un diagnostic. Donc pour qu'un système expert puisse être utilisé dans de tels domaines, il faut qu'il soit capable de raisonner sur des faits et des règles incertaines. Les réseaux bayésiens constituent donc une approche possible pour intégrer l'incertitude dans le raisonnement [Pearl, 1988] et [Naïm et al., 2004]. Nous allons nous intéresser donc au diagnostic de défaillances par réseaux bayésiens.

A.2 Définition des réseaux bayésiens :

Un réseau bayésien est défini par [Naïm et al., 2004] :

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

$$P(V_1, V_2, \dots, V_n) = \prod_{i=1}^n P(V_i|C(V_i)) \quad (\text{A.2.1})$$

Où $C(V_i)$ est l'ensemble des causes (parents) de V_i dans le graphe G .

En d'autres termes les réseaux bayésiens fournissent un formalisme pour représenter une distribution jointe de probabilités sur un ensemble de variables aléatoires. Ils apportent des outils naturels permettant de traiter deux grands problèmes couramment rencontrés en intelligence artificielle, en mathématiques appliquées ou en ingénierie : l'incertitude et la complexité. Ils sont le mariage entre la théorie des probabilités et la théorie des graphes. Donc, ils sont à la fois :

- Des modèles de représentations des connaissances,
- Et des machines à calculer les probabilités conditionnelles.

Il est donc nécessaire de distinguer deux approches différentes pour modéliser les connaissances par réseaux bayésiens : l'approche qualitative et l'approche quantitative.

A.3 Approche qualitative

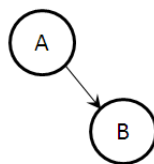
Comme J. Pearl le montre dans son livre [Pearl, 1988], l'intérêt des réseaux bayésiens est de permettre aux experts de se concentrer sur la construction d'un modèle qualitatif avant même de penser aux spécifications numériques.

Dans cette partie, nous allons présenter donc l'aspect intuitif qualitatif qui est basé sur la théorie des graphes. La notion la plus importante dans les réseaux bayésiens est bien la causalité directe, c'est-à-dire, relier les causes à leurs effets.

A.3.1 La représentation graphique de la causalité :

Intuitivement, pour relier des causes à leurs effets dans un graphe, il faut mettre une flèche orientée entre les deux. Dorénavant, nous appellerons les causes "parents" et les effets "enfants".

La figure ci-dessous représente deux événements A et B qui sont les nœuds et une relation



de causalité entre ces nœuds sous forme d'arc qui relie la cause à son effet. On peut dire d'après cette représentation que la connaissance que l'on a de A détermine la connaissance que l'on a de B. La relation entre les deux événements peut être stricte, c'est-à-dire que, sachant avec certitude que A est vrai, je peux en déduire B avec certitude. Ou bien une influence simple, c'est-à-dire, si l'on connaît A avec certitude, l'opinion sur B est modifiée, sans qu'on puisse toutefois affirmer si B est vrai ou faux. Avant d'aller plus loin, il est important de comprendre que, bien que la flèche soit orientée de A vers B, elle peut fonctionner dans les deux sens, même si la relation causale est stricte. Supposons, par exemple, que la relation causale soit l'implication logique $A \Rightarrow B$. Cette relation signifie que si A est vrai, B l'est également. Si A est faux, B peut être vrai ou faux.

| A | B |
|-----|-----|
| V | V |
| F | V |
| F | F |

Cette table nous permet d'affirmer que si B est faux, A l'est également. Donc, comme conclusion on peut dire que *s'il existe une relation causale de A vers B , toute information sur A peut modifier la connaissance que l'on a de B , et, réciproquement, toute information sur B peut modifier la connaissance que l'on a sur A .* [Naïm et al., 2004]

- **Circulation de l'information dans un graphe causal :** Nous allons maintenant étudier comment l'information circule au sein d'un graphe causal. Pour bien illustrer la notion de circulation d'information, nous allons étudier un exemple simple sur le problème de démarrage d'une voiture.

Exemple : Ceci est un exemple de raisonnement inspiré de la vie quotidienne d'un individu qui a une voiture (cet exemple est tiré d'une thèse [Jansson, 2004]). Ce matin-là, il sort de sa maison pour aller au travail avec sa voiture. Malheureusement, lorsqu'il a essayé de la démarrer, il entend le démarreur tourner, mais rien ne se produit. Il peut y avoir plusieurs raisons à ce problème. On peut entendre le starter tourner, donc le problème ne provient sûrement pas de la puissance de la batterie. Par conséquent, les causes les plus probables sont :

1. Le réservoir est vide,
2. Les bougies sont sales.

Une manière de structurer une situation pour raisonner sous l'incertain est de construire un graphe (ou réseau) représentant des relations causales entre les événements. Pour simplifier la situation, supposons que nous ayons les événements {oui, non} pour le carburant, {oui, non} pour les bougies sales, {plein, à moitié, vide} pour la position de l'indicateur de niveau de carburant, et {vrai, faux} pour le démarreur. Ceux-ci s'appellent également les états (ou les modes).

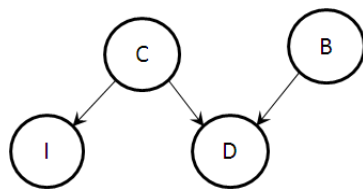


Figure A.1 – Problème de démarrage d'une voiture

| | | |
|---|--|-----------------------|
| B | Les bougies d'allumage sont sales | Oui, non |
| D | Le voiture ne démarre pas | Vrai, faux |
| C | Le réservoir est vide | Oui, non |
| I | La position de l'indicateur de niveau de carburant | Plein, à moitié, vide |

Le graphe de la Fig.A.3.1 illustre donc la première étape consistant à modéliser qualitativement le problème et à déterminer les influences existant entre les variables. Maintenant, comment l'information D (la voiture ne démarre pas), dont on sait qu'elle est vraie, est-elle utilisée ?

Tout d'abord, le modèle nous indique que D a pu être causé soit par C , soit par B , et que I est influencé par C seulement. Sans informations complémentaires, les deux causes (C et B) sont a priori également plausibles, l'information a circulé de D à C et B (*effet* \rightarrow *cause*).

Pour trouver la cause de ce dysfonctionnement, l'individu jette alors un coup d'œil à l'indicateur du niveau de carburant, il s'aperçoit qu'il est à moitié. Il en déduit alors que les bougies sont sales. Donc le fait que I est à moitié renforce la croyance en B .

Dans cet exemple, on voit bien que l'information a circulé de I à C (*effet* \rightarrow *cause*) et de C à D (*cause* \rightarrow *effet*), Ce qui montre bien que l'information dans un réseau bayésien circule dans les deux sens.

Du point de vue général, l'information certaine (ou l'observation) se propage dans un réseau en modifiant les croyances que nous avons dans les autres faits. Mais quels chemins cette information peut prendre à l'intérieur d'un réseau ? Pour cela, la notion de D-séparation (Direction Separation) est très utile pour trouver les chemins bloqués dans un graphe.

A.3.2 D-séparation (bloquage)

Nous allons étudier quels chemins une information (observation) peut prendre à l'intérieur d'un graphe. Pour cela, nous allons utiliser l'exemple précédent en ajoutant un autre nœud pour représenter toutes les situations possibles.

| | | |
|---|--|-----------------------|
| B | Les bougies d'allumage sont sales | Oui, non |
| D | La voiture ne démarre pas | Vrai, faux |
| C | Le reservoir est vide | Oui, non |
| I | La position de l'indicateur de niveau de carburant | Plein, à moitié, vide |
| A | Alarme | Oui, non |

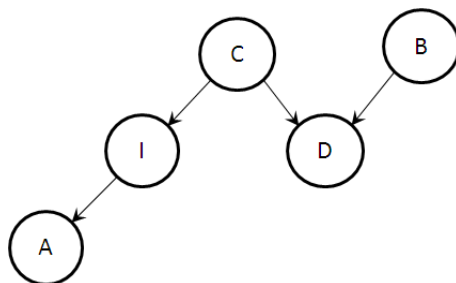


Figure A.2 – Problème de démarrage d’une voiture

Nous allons considérer dans le tableau suivant trois cas possibles, qui décrivent l’ensemble des situations possibles faisant intervenir trois événements. Nous illustrons les trois cas avec l’exemple proposé précédemment. Les trois cas possibles :

- connexion convergente,
- connexion divergente,
- connexion en série.

On voit bien que la circulation de l’information suit des règles bien précises dans les deux sens de propagation.

Résumé : On dira que X et Y sont d-séparés par Z si pour tous les chemins entre X et Y , l’une au moins des deux conditions suivantes est vérifiée :

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

Notation : " X est d-séparé de Y par Z " est noté $(X|Z|Y)$

A.4 Approche quantitative

L’aspect quantitatif des réseaux bayésiens consiste à associer à chaque nœud une table de probabilité qui le caractérise (une table, c’est la spécification de l’ensemble des probabilités de la variable pour chacune de ses valeurs possibles sachant chacune des valeurs de ses parents).

Nous reprenons l’exemple du problème de démarrage d’une voiture. Nous allons assimiler la croyance ou la plausibilité d’un fait, à une probabilité mathématique et nous montrerons par la suite la similarité des résultats qualitatifs trouvés ci-dessus et des résultats quantitatifs trouvés ci-dessous. Pour cela, on va associer à chaque nœud une table de probabilité en fonction des modes.

Les probabilités a priori marginales $P(B = \text{oui ou non})$ et $P(C = \text{oui ou non})$ sont déterminées par expérience sur le système (dans le cas général ces probabilités sont souvent données par un expert du domaine modélisé, ou bien apprises à partir d’un

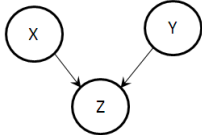
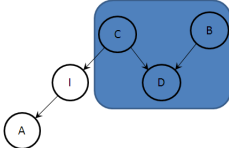
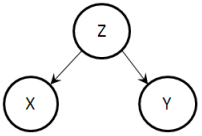
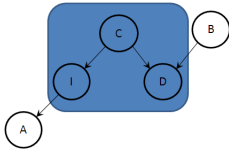
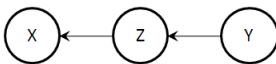
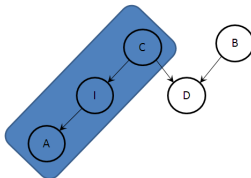
| Réseau | Exemple | Description de l'exemple |
|--|---|--|
|  <p>connexion convergente</p> <p>L'information ne peut circuler de X à Y que si Z est connu</p> |  | <p>Le fait qu'il y ait du carburant dans le réservoir de la voiture ou non n'a aucun lien a priori avec le fait que les bougies sont sales ou non. En revanche, si la voiture ne démarre pas, on a tendance à croire que le réservoir est vide. Si maintenant, après vérification du réservoir, on s'aperçoit que le réservoir de la voiture n'est pas vide, on est assuré que le problème vient des bougies d'allumage.</p> |
|  <p>connexion divergente</p> <p>L'information ne peut circuler de X à Y que si Z n'est pas connu</p> |  | <p>Si l'aiguille de l'indicateur du niveau de carburant est dans la position " 0 ", on a tendance à croire que le réservoir est vide, et donc la voiture ne démarrera pas. Si en revanche, on sait que le réservoir est vide, on peut affirmer que la voiture ne démarrera pas, et l'information que l'on peut avoir sur la position de l'indicateur n'y change rien.</p> |
|  <p>connexion en série</p> <p>L'information ne peut circuler de X à Y que si Z n'est pas connu</p> |  | <p>S'il n'y a pas de carburant dans le réservoir, l'indicateur du niveau de carburant indique qu'il est vide. Maintenant si l'indicateur indique qu'il est vide l'alarme se déclenche. Si on sait déjà que l'indicateur indique que le réservoir est vide, le fait d'aller voir le réservoir ne nous apprend plus rien sur le fait que l'alarme se déclenche.</p> |

Figure A.3 – Circulation de l'information dans un graphe causal

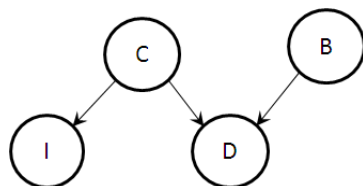


Figure A.4 – problème de démarrage d’une voiture

corpus d’exemples). Si nous savons que les bougies d’allumage deviennent sales dans 5% des cas, nous aurons une probabilité de 0,05 pour ce cas. Et si nous savons que l’individu oublie dans 10% des cas de remplir son réservoir avant qu’il soit vide, la probabilité qu’il soit vide est égale à 0,1.

| | | | | | |
|----------|------|------|----------|-----|-----|
| <i>B</i> | oui | non | <i>C</i> | oui | non |
| $P(B)$ | 0,05 | 0,95 | $P(C)$ | 0,9 | 0,1 |

Maintenant, pour les probabilités a priori conditionnelles $P(D|B, C)$ et $P(I|C)$, nous pouvons les déterminer d’une façon simple et très intuitive.

| | | | | |
|---------------|-----|-----|------|-----|
| <i>B</i> | oui | | non | |
| <i>C</i> | oui | non | oui | non |
| $P(D = vrai)$ | 0,9 | 1 | 0,05 | 1 |
| $P(D = faux)$ | 0,1 | 0 | 0,95 | 0 |

Si les bougies sont sales ($B=oui$) et le réservoir n’est pas vide ($C=oui$), nous aurions une forte probabilité pour que la voiture ne démarre pas, nous l’estimons donc à 90% de manière arbitraire ¹. Maintenant, si le réservoir est vide, quelle que soit la situation des bougies, nous serions sûrs que la voiture ne démarrera pas (une certitude) donc la probabilité associée est à 100%. Il reste que le cas où les bougies ne sont pas sales ($B=non$) et le réservoir n’est pas vide ($C=oui$), dans ce cas il est fort probable que la voiture démarre normalement dans 95% des cas (5% qui restent sont causés par d’autres événement qu’on néglige pour la suite).

| | | |
|----------------------|------|-----|
| <i>C</i> | oui | non |
| $P(I = plein)$ | 0,25 | 0 |
| $P(I = \frac{1}{2})$ | 0,75 | 0 |
| $P(I = vide)$ | 0 | 1 |

¹les expériences pourraient aider à fixer ces résultats

Pour la probabilité $P(I|C)$, nous admettons que le chauffeur de la voiture ne remplit pas souvent son réservoir totalement, donc on associe une probabilité de 25% pour que le réservoir soit plein sachant qu'il y a du carburant dans le réservoir, et une probabilité de 75% pour que le réservoir soit à moitié dans le même cas. Et enfin, on a une certitude pour que le réservoir soit vide s'il n'y a pas de carburant.

Maintenant, si la voiture ne démarre pas, laquelle des deux causes conditionnelles est la plus probable? Pour répondre à cette question mathématiquement, il faut connaître certaines notions et lois des probabilités conditionnelles. Résumons les lois et les notions nécessaires pour résoudre ce problème.

A.4.1 Probabilités

Dans l'exemple précédent, nous voulons connaître la probabilité $(C, B|D = oui)$, nous obtiendrons cette probabilité en faisant appel aux règles de probabilités suivantes : La principale règle qui nous fournit une relation entre la probabilité conditionnelle et les événements joints est :

$$P(a, b) = P(a|b)P(b) = P(b|a)P(a) \quad (\text{A.4.1})$$

Cela ramène au fameux théorème de Bayes

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)} \quad (\text{A.4.2})$$

et

$$P(a|b, c) = \frac{P(b|a, c)P(a|c)}{P(b|c)} \quad (\text{A.4.3})$$

Nous aurons aussi besoin de la loi marginale utilisée pour calculer une probabilité marginale dans un réseau bayésien.

$$P(A) = \sum_B P(A, B) \quad (\text{A.4.4})$$

Notons aussi que la loi marginale est commutative c'est-à-dire

$$\sum_a \sum_b = \sum_b \sum_a \quad (\text{A.4.5})$$

Chaque réseau bayésien a un univers qui est représenté par la distribution de la probabilité jointe $P(U)$. C'est la multiplication de toutes les probabilités a priori marginales (pour les nœuds sans parents) et conditionnelles (pour les nœud avec parents) (équation A.2.1). Maintenant, nous pouvons calculer $P(C, B|D = oui)$. Nous expliquons la procédure dans ce qui suit.

D'après l'équation (A.2.1)

$$P(B, D, C, I) = P(B)P(C)P(D|B, C)P(I|C) \quad (\text{A.4.6})$$

Annexe A. Réseaux bayésiens

Après calculs, nous obtenons le tableau A.1 :

| | | <i>C = oui</i> | | | <i>C = non</i> | | |
|----------------|-----------------|------------------|-------------------------------------|-----------------|------------------|-------------------------------------|-----------------|
| | | <i>I = plein</i> | <i>I = $\frac{1}{2}$</i> | <i>I = vide</i> | <i>I = plein</i> | <i>I = $\frac{1}{2}$</i> | <i>I = vide</i> |
| <i>B = oui</i> | <i>D = vrai</i> | 0,010125 | 0,030375 | 0 | 0 | 0 | 0,005 |
| | <i>D = faux</i> | 0,001125 | 0,003375 | 0 | 0 | 0 | 0 |
| <i>B = non</i> | <i>D = vrai</i> | 0,00106875 | 0,00320625 | 0 | 0 | 0 | 0,095 |
| | <i>D = faux</i> | 0,2030625 | 0,6091875 | 0 | 0 | 0 | 0 |

Tableau A.1 – La table de probabilités jointes des différents modes

A partir de cette table (ou matrice), nous pouvons calculer n'importe quelle probabilité. L'univers $P(B, D, C, I)$ contient la probabilité voulue ($P(C, B|D)$), et pour l'extraire, on utilise la loi marginale (équation(A.4.4)) et le théorème de Bayes (équations (A.4.2) et (A.4.3)).

$$P(C, B|D) = \frac{P(C, D, B)}{P(D)} \quad (\text{A.4.7})$$

tel que :

$$P(D) = \sum_{C, B, I} P(B, D, C, I) \quad (\text{A.4.8})$$

Donc on obtient après calculs le tableau suivant :

| <i>D</i> | vrai | faux |
|-------------|---------|---------|
| <i>P(D)</i> | 0,18325 | 0,81675 |

Nous pouvons calculer aussi la probabilité marginale de I comme suit :

$$P(I) = \sum_{C, B, D} P(B, D, C, I) \quad (\text{A.4.9})$$

Nous obtenons

| <i>I</i> | plein | $\frac{1}{2}$ | vide |
|-------------|-------|---------------|------|
| <i>P(I)</i> | 0,225 | 0,675 | 0,1 |

Annexe A. Réseaux bayésiens

En utilisant le théorème de Bayes (équation (A.4.2) et la loi jointe (équation (A.4.1)), nous pouvons calculer la probabilité conditionnelle a posteriori $P(C, B|D)$.

Nous pouvons donc écrire :

$$P(C, B|D) = P(C|B, D)P(B|D) \quad (\text{A.4.10})$$

$$P(C|B, D) = \frac{P(C, B, D)}{P(B, D)} \quad (\text{A.4.11})$$

$$P(B|D) = \frac{P(B, D)}{P(D)} \quad (\text{A.4.12})$$

Avec ces trois équations, nous pouvons extraire la probabilité voulue

$$\Rightarrow P(C, B|D) = \frac{P(C, B, D)}{P(B, D)} \frac{P(B, D)}{P(D)} = \frac{P(C, B, D)}{P(D)} \quad (\text{A.4.13})$$

Donc, nous aurons besoin de calculer aussi la probabilité jointe $P(C, B, D)$. En utilisant la loi marginale sur $P(C, B, D, I)$:

$$P(C, B, D) = \sum_I P(C, B, D, I) \quad (\text{A.4.14})$$

nous trouvons :

| B | oui | | non | |
|-------------------|--------|-------|---------|-------|
| C | oui | non | oui | non |
| $D = \text{vrai}$ | 0,0405 | 0,005 | 0,04275 | 0,095 |
| $D = \text{faux}$ | 0,0045 | 0 | 0,81225 | 0 |

Les résultats donnés par l'équation (A.4.13) s'écrivent :

| B | oui | | non | |
|---------------------------|---------|----------|----------|----------|
| C | oui | non | oui | non |
| $P(B, C D = \text{vrai})$ | 0,2210 | 0,027285 | 0,233287 | 0,518417 |
| $P(B, C D = \text{faux})$ | 0,00551 | 0 | 0,99449 | 0 |

A partir du tableau ci-dessus, nous pouvons aussi calculer les probabilités $P(B|D)$ et $P(C|D)$ en utilisant la loi marginale :

$$P(B|D) = \sum_C P(C, B|D) \quad (\text{A.4.15})$$

Et

$$P(C|D) = \sum_B P(C, B|D) \quad (\text{A.4.16})$$

Nous obtenons les deux tables de probabilités suivantes :

| B | Oui | Non |
|-----------------|----------|----------|
| $P(B D = vrai)$ | 0,248285 | 0,751704 |
| $P(B D = faux)$ | 0,00551 | 0,99449 |

| C | Oui | Non |
|-----------------|----------|----------|
| $P(C D = vrai)$ | 0,454287 | 0,545702 |
| $P(C D = faux)$ | 1 | 0 |

D'après les deux tableaux précédents, si la voiture ne démarre pas ($D = vrai$), nous avons tendance à croire que le réservoir est vide ($P(C = non) = 0,5457$). Maintenant, si nous vérifions la position de l'indicateur de niveau du carburant et nous nous apercevons qu'il est à moitié, cette observation va changer sûrement notre croyance sur le fait que le carburant est le responsable du dysfonctionnement de la voiture. Avec cette nouvelle donnée, nous essayons de calculer la probabilité $P(B, C|D = vrai, I = \frac{1}{2})$.

$$P(B, C|D = vrai, I = \frac{1}{2}) = \frac{P(B, C, D, I)}{\sum_{B,C} P(B, C, D = vrai, I = \frac{1}{2})} \quad (A.4.17)$$

Les résultats obtenus sont dans la table suivante :

| B | Oui | | Non | |
|-------------------------------------|----------|-------|----------|-------|
| | Oui | Non | Oui | Non |
| $P(B, C D = vrai, I = \frac{1}{2})$ | 0,904337 | 0 | 0,095477 | 0 |

Nous pouvons calculer aussi les deux probabilités $P(B|D = vrai, I = \frac{1}{2})$ et $P(C|D = vrai, I = \frac{1}{2})$, en appliquant la loi marginale :

| B | Oui | Non |
|----------------------------------|----------|----------|
| $P(B D = vrai, I = \frac{1}{2})$ | 0,904337 | 0,095477 |

| C | Oui | Non |
|----------------------------------|-------|-------|
| $P(C D = vrai, I = \frac{1}{2})$ | 1 | 0 |

On a donc : $P(C = non|D = vrai, I = \frac{1}{2}) = 0$, En d'autres terme, compte tenu du modèle, si l'indicateur de niveau de carburant indique que le réservoir est à moitié, le problème est dû très probablement à la saleté des bougies avec une probabilité qui est plus de 0,9. A ce moment le chauffeur de la voiture est certain que le problème vient des

bougies, ce qui correspond exactement au raisonnement intuitif mené avant.

Comme conclusion sur cette partie, nous pouvons dire que la transposition d'un graphe causal en espace de probabilité conduit à des résultats conformes au raisonnement intuitif que l'on peut mener directement sur ce graphe.

A.4.2 Circulation d'information et indépendance conditionnelle

Sur le même exemple, nous savons que B et C sont indépendants. Nous pouvons donner un sens quantitatif à la notion de circulation d'information. Comme D est connu, l'information peut circuler suivant le circuit $B \rightarrow D \leftarrow C$. Qu'est ce que cela signifie en termes de probabilités ?

Nous avons, d'après les tables de probabilités précédentes, les probabilités suivantes :

$$P(B = \text{oui}, C = \text{non} | D = \text{vrai}) = 0,005$$

Or

$$P(B = \text{oui} | D = \text{vrai}) = 0,2482$$

Et

$$P(C = \text{non} | D = \text{vrai}) = 0,5457$$

Et donc :

$$P(B = \text{oui}, C = \text{non} | D = \text{vrai}) \neq P(B = \text{oui} | D = \text{vrai})P(C = \text{non} | D = \text{vrai})$$

B et C ne sont donc pas indépendants conditionnellement à D . Qu'es-ce que cela signifie intuitivement ?

Simplement que si deux facteurs indépendants peuvent être la cause du même effet, et que celui-ci est observé, c'est nécessairement l'une ou l'autre des deux causes qui l'a produit. Donc les valeurs des deux causes sont liées : elles ne sont plus indépendantes, a posteriori.

Qualitativement, que remarquons nous ? B et C sont d-séparés par D quand D n'est pas connu, l'information ne circule donc pas de B à C et l'inverse (ils sont donc indépendants), mais si D est connu (B et C ne sont pas d-séparés par D), l'information peut circuler de B à C et de C à B (ils sont dépendants).

A.4.3 Conclusion

L'exemple précédent nous a permis de constater les faits suivants :

1. La transposition d'un graphe causal en espace de probabilités conduit à des résultats conformes au raisonnement intuitif que l'on peut mener directement sur ce graphe.

2. Les propriétés graphiques (d-séparation) peuvent être mises en correspondance avec les propriétés probabilistes (indépendance conditionnelle).
3. Les calculs mis en œuvre, même sur des cas très simples, sont lourds (NP-difficile). Pour cela, nous étudions dans la section suivante la notion d'inférence qui a pour but de faciliter les calculs et la propagation de l'information dans un réseau bayésien.

A.5 Inférence

A.5.1 Introduction

Les réseaux bayésiens ont été développés pour tenter de résoudre certains problèmes de prédiction et d'abduction, courants en intelligence artificielle. Dans ce type de tâches, il est nécessaire de trouver une interprétation cohérente des observations avec les données connues a priori en limitant les calculs. L'inférence probabiliste signifie donc le calcul de $P(X|Y)$ où X est un ensemble d'observations et Y un ensemble de variables décrivant le problème et qui sont jugées importantes pour la prédiction ou le diagnostic, où l'on doit reconsidérer son appréciation de la situation en fonction d'une ou plusieurs nouvelles observations ; en d'autres termes, l'inférence, appelée aussi mise à jour des probabilités, représente tout calcul portant sur les distributions de probabilités associées à un réseau bayésien [Naïm et al., 2004].

Le problème général pour calculer les probabilités a posteriori dans un réseau bayésien est NP-difficile [Cooper, 1988]. Cependant, il y a plusieurs algorithmes qui ont été proposés pour résoudre ce problème.

Les premiers algorithmes exacts d'inférence pour les réseaux bayésiens ont été proposés dans [Pearl, 1988] : il s'agissait d'une architecture à passage de messages et ils étaient limités aux arbres. Dans cette technique, à chaque nœud est associé un processeur qui peut envoyer des messages de façon asynchrone à ses voisins jusqu'à ce qu'un équilibre soit atteint, en un nombre fini d'étapes. Cette méthode a été depuis étendue aux réseaux quelconques pour donner l'algorithme JLO. Cette méthode est aussi appelée algorithme de l'arbre de jonction et a été développée dans [Jensen et al., 1990]. Bien que l'inférence dans des réseaux quelconques soit NP-difficile [Pearl, 1988], la complexité en temps pour chacune des méthodes citées précédemment est calculable à l'avance. Quand le résultat dépasse une limite raisonnable, on préfère alors utiliser une méthode d'approximation [Pearl, 1988]. Ces méthodes exploitent la topologie du réseau et effectuent un échantillonnage sur des sous-ensembles locaux de variables de façon séquentielle et concurrente [Jordan et al., 1999]. Nous allons, dans cette étude, ne nous intéresser qu'aux méthodes d'inférences exactes.

Remarque : d'autres méthodes ont été proposées telles que l'algorithme de Shafer She-

noy [Shafer & Shenoy, 1988], l'inférence probabilistique symbolique proposée par D'Ambrosio [Shachter et al., 1990] et l'algorithme de Zhang et Poole [Zhang & Poole, 1992]. Toutes ces méthodes ont été traitées après par Zhang et Poole, en faisant une comparaison des avantages et des inconvénients de chacune [Zhang & Poole, 1994].

A.5.2 Algorithme de Pearl

Dans cette partie, nous ferons un résumé de l'algorithme de Pearl qui ne concerne que les arbres, appelé aussi l'algorithme de Message passing (plus de détails dans [Pearl, 1988]).

Admettons que nous avons un nœud X qui a m enfants, Y_1, Y_2, \dots, Y_m , et n parents $U = \{U_1, \dots, U_n\}$ (Fig. A.5)

Comme le graphe que l'on considère ici est un arbre, il est par définition connexe et

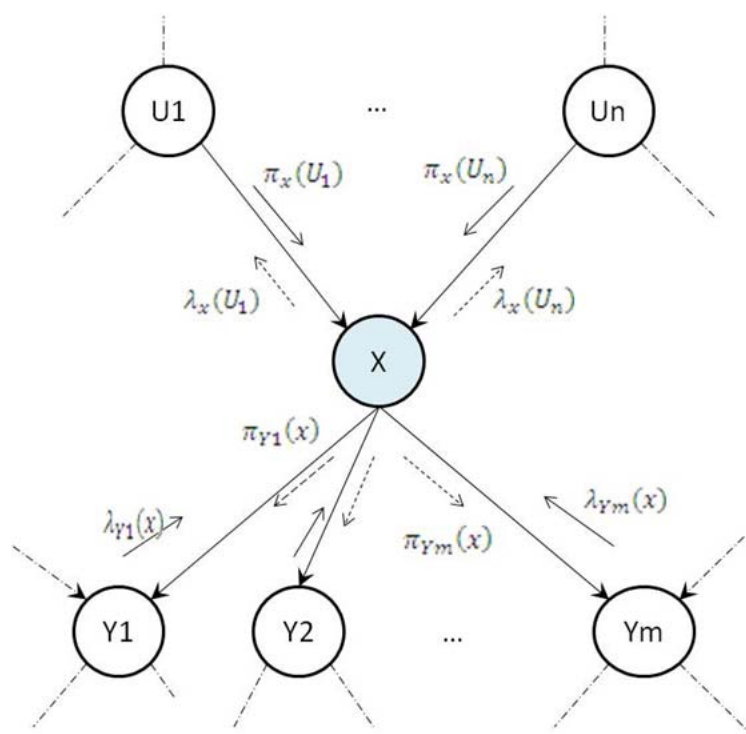


Figure A.5 – Algorithme de Pearl

sans circuit. Autrement dit, il existe un chemin unique entre deux nœud de ce graphe. Cette propriété permet de partitionner le graphe relativement à un nœud X :

- (+) les nœuds dont les chemins vers X passent par un parent de X ,
- (-) les nœuds dont les chemins vers X passe par un enfant de X ,

Considérons une information e sur l'ensemble du réseau, cette observation peut se partitionner en deux sous-ensembles différents relativement au nœud X : e_x^+, e_x^- qui correspondent aux informations élémentaires sur des nœuds respectivement de types (+) et (-)

Annexe A. Réseaux bayésiens

Nous voulons calculer la probabilité $P(X|e)$ en utilisant cet algorithme.

En appliquant le théorème de Bayes à $P(X|e)$:

$$P(X|e) = P(x|e_X^+, e_X^-) = \frac{P(e_X^-|X, e_X^+)P(X|e_X^+)}{P(e_X^-|e_X^+)} = \alpha P(e_X^-|x, e_X^+)P(X|e_X^+) \quad (\text{A.5.1})$$

tel que :

$$\alpha = \frac{1}{P(e_X^-|e_X^+)} \quad (\text{A.5.2})$$

Comme e_X^- et e_X^+ sont d-séparés (indépendants conditionnellement) par X nous aurons :

$$P(e_X^-|X, e_X^+) = P(e_X^-|X) \quad (\text{A.5.3})$$

Et

$$P(e_X^-|e_X^+) = P(e_X^-) \quad (\text{A.5.4})$$

D'où

$$P(X|e) = \alpha P(e_X^-|X)P(X|e_X^+) \quad (\text{A.5.5})$$

tel que

$$\alpha = \frac{1}{P(e_X^-)} \quad (\text{A.5.6})$$

Pour simplifier les écriture, nous faisons ce changement de variable :

$$P(e_X^-|X) = \lambda(X) \quad (\text{A.5.7})$$

Et

$$P(e_X^+|X) = \pi(X) \quad (\text{A.5.8})$$

Donc

$$P(X|e) = \alpha \lambda(X) \pi(X) \quad (\text{A.5.9})$$

Maintenant, comment calculer $\lambda(X)$ et $\pi(X)$. Pour les calculer, X doit recevoir des informations de tous le réseau. Pour cela, les étapes suivantes sont nécessaires pour la propagation de l'information.

1. Le noeud X demande à ses voisins de lui envoyer leurs messages pour qu'il puisse calculer $P(X|e)$,
2. X a besoin des λ - messages de tous ses enfants pour pouvoir calculer $\lambda(X)$
3. Il a aussi besoin des π - messages de tous ses parents pour pouvoir calculer $\pi(X)$
4. Les enfants de X Y_1, \dots, Y_m , qui veulent envoyer les λ - messages ($\lambda_{Y_j}(X)$) vers leur parent X ont besoin des λ - messages de tous leurs enfants et de π - messages de tous leurs parents sauf X
5. Le noeud U qui veut envoyer un π - message ($\pi_x(U)$) vers son enfant X a besoin des π - messages de tous ses parents et des λ - messages de tous ses enfants sauf X .

6. Lorsque X reçoit tous les messages qu'il avait demandés, il les multiplie avec sa table de probabilité conditionnelle (loi jointe). Puis, il effectue des sommes (loi marginale) pour obtenir la probabilité voulue ($P(x|e)$).

Généralisation de l'algorithme

Soit un réseau bayésien de graphe $G = (U, E)$, l'algorithme suivant permet de calculer $P(X|e)$ pour tout nœud X du réseau bayésien.

Pour tout nœud $V_i = e_i$ (observé)

$\lambda(X_i) = 1$ sinon 0

$\pi(X_i) = 1$ sinon 0

Pour les nœuds sans parents $\pi(X_i) = P(X_i) \rightarrow$ probabilité a priori

Pour les nœuds sans enfants $\lambda(X_i) = 1$

1. (Pour tout nœud X) si X a reçu tous les messages π – *messages* de ses parents \rightarrow calculer $\pi(X)$

le calcul de $\pi(X) = P(X|e^+)$:

On a :

$U_1, \dots, U_n \rightarrow$ les parents de X

T_k sont les enfants d'un U (parent de X) sauf X $\pi_X(u) \rightarrow \pi$ – *message* entre X et U

$$\pi_X(u) = \pi(u) \prod_k \lambda_{T_k}(u) \quad (\text{A.5.10})$$

$$\pi(X) = \sum_{u_1, \dots, u_n} P(X|u_1, \dots, u_n) \prod_{j=1}^n \pi_{X_i}(u_j) \quad (\text{A.5.11})$$

2. (Pour tout nœud X) si X a reçu tous les λ – *messages* de ses enfants \rightarrow calculer $\lambda(X)$

le calcul de $\lambda(X) = P(e^-|X)$:

soit Y_1, \dots, Y_m les enfants de X

$\lambda_{Y_j}(X) \rightarrow \lambda$ – *message* envoyé entre X et Y

$$\lambda_{Y_j}(X) = \sum_{y_j} \lambda_{Y_j}(y_j) \sum_{z_1, \dots, z_p} P(y_j|X, z_1, \dots, z_p) \prod_{k=1}^p \pi_{Y_j}(z_k) \quad (\text{A.5.12})$$

$$\lambda(X) = \prod_{j=1}^m \lambda_{Y_j}(X) \quad (\text{A.5.13})$$

3. (Pour tout nœud X) si $\pi(x)$ a été calculé et X a reçu tous les λ – *messages* de tous les enfants sauf $Y \rightarrow$ calculer $\pi_Y(x)$ et l'envoyer à Y

le calcul de $\pi_Y(X)$:

Pour calculer $\pi_Y(x)$, nous avons besoin de calculer d'abord $\lambda_{Y_j}(X)$:

Soit $Z = Z_1, \dots, Z_p$ les autres parents de Y

$$\pi_{Y_j}(x) = \pi(x) \prod_{k \neq j} \lambda_{Y_k}(x) \quad (\text{A.5.14})$$

4. (Pour tout nœud X) si $\lambda(X)$ a été calculé et X a reçu tous les π – messages de ses parents sauf $U \leftarrow$ calculer $\lambda_U(x)$ et l'envoyer à U .

$$\lambda_X(u) = \sum_X \lambda(X) P(X|u) \quad (\text{A.5.15})$$

L'algorithme de Pearl ne s'applique que sur les réseaux bayésiens dont le graphe est un arbre. Le problème se pose de l'étendre à tout réseau bayésien. L'une des méthodes proposée également par [Pearl, 1988] consiste à regrouper les nœuds qui ont les mêmes enfants et les considérer comme un seul nœud ce qui permet d'obtenir un graphe qui soit un arbre (plus de détails sur l'exemple ci-dessous).

Pour bien illustrer nos propos, nous allons étudier un exemple de trois vannes.

Exemple

Considérons que nous avons trois vannes connectées comme indique la figure (A.6) (cet exemple a été tiré des travaux de [Weber & Simon, 2008]). L'objectif de cet exemple est de savoir si le système reste contrôlable ou pas (c'est à dire, si on peut toujours contrôler le passage du fluide dans le système ou pas).

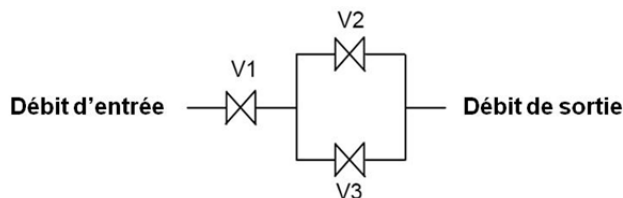


Figure A.6 – Exemple des trois vannes

Chaque vanne a trois états possibles :

- Ok,
- RO : la vanne reste toujours ouverte (défaillance),
- RC : la vanne reste toujours fermée (défaillance).

Le réseau bayésien correspondant est donné ci-dessous.

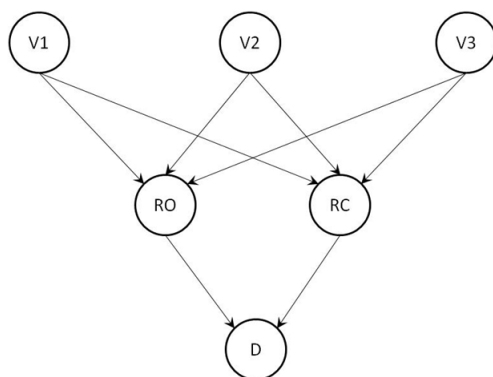


Figure A.7 – Réseau bayésien du système trois vannes

| V_i | Ok | RO | RF |
|----------|------|-------|-------|
| $P(V_i)$ | 0,99 | 0,007 | 0,003 |

Tableau A.2 – Distribution de probabilités des 3 vannes

| BO | BF | $Valeur$ |
|--------|--------|----------|
| $Vrai$ | $Vrai$ | $Faux$ |
| | $Faux$ | $Faux$ |
| $Faux$ | $Vrai$ | $Faux$ |
| | $Faux$ | $Vrai$ |

Tableau A.3 – Distribution de probabilités de la disponibilité du fluide

Par expérience sur le système, nous allons définir toutes les probabilités a priori associées à chaque nœud du graphe dans les tableaux ci-dessous.

Après avoir attribué à chaque nœud une table de probabilités a priori, nous pouvons maintenant appliquer l'algorithme de Pearl soit pour calculer les probabilités marginales (dans le cas où il n'y a pas d'observations), soit pour calculer les probabilités a posteriori (en présence d'observations).

Calculs des probabilités marginales

Tous les λ à "1" car $P(\phi|X) = 1$ tel que ϕ représente la non présence d'observations.
 $\pi(V_1) = \pi(V_2) = \pi(V_3) = P(V_{1,2,3})$.

1. Les nœuds sans parents (V_1, V_2 et V_3) transmettent leurs impacts (π – messages) à leurs enfants (RO et RF). Après avoir reçu tous les messages de leurs parents, RO et RF peuvent calculer respectivement $\pi(RO)$ et $\pi(RF)$ en appliquant les relations A.5.11 et A.5.10.

| <i>Vanne</i> ₁ | <i>Vanne</i> ₂ | <i>Vanne</i> ₃ | <i>Valeur</i> | <i>Vanne</i> ₁ | <i>Vanne</i> ₂ | <i>Vanne</i> ₃ | <i>Valeur</i> |
|---------------------------|---------------------------|---------------------------|---------------|---------------------------|---------------------------|---------------------------|---------------|
| <i>Ok</i> | <i>Ok</i> | <i>Ok</i> | <i>Faux</i> | <i>Ok</i> | <i>Ok</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Faux</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Faux</i> |
| | <i>RO</i> | <i>Ok</i> | <i>Faux</i> | | <i>RO</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Faux</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Faux</i> |
| | <i>RF</i> | <i>Ok</i> | <i>Faux</i> | | <i>RF</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Faux</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Faux</i> |
| <i>RO</i> | <i>Ok</i> | <i>Ok</i> | <i>Faux</i> | <i>RO</i> | <i>Ok</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Vrai</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Faux</i> |
| | <i>RO</i> | <i>Ok</i> | <i>Faux</i> | | <i>RO</i> | <i>Ok</i> | <i>Vrai</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Vrai</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Vrai</i> |
| | <i>RF</i> | <i>Ok</i> | <i>Faux</i> | | <i>RF</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Vrai</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Faux</i> |
| <i>RF</i> | <i>Ok</i> | <i>Ok</i> | <i>Faux</i> | <i>RF</i> | <i>Ok</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Faux</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Faux</i> |
| | <i>RO</i> | <i>Ok</i> | <i>Faux</i> | | <i>RO</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Faux</i> |
| | | RF | <i>Faux</i> | | | RF | <i>Faux</i> |
| | <i>RF</i> | <i>Ok</i> | <i>Vrai</i> | | <i>RF</i> | <i>Ok</i> | <i>Faux</i> |
| | | RO | <i>Vrai</i> | | | RO | <i>Faux</i> |
| | | RF | <i>Vrai</i> | | | RF | <i>Faux</i> |

Tableau A.4 – Distribution de probabilités de RO et RF

Nous trouvons pour RO :

$$\pi(RO = Vrai) = 0,0001 \text{ et } \pi(RO = Faux) = 0,9999$$

Et pour RF :

$$\pi(RF = Vrai) = 0,003 \text{ et } \pi(RF = Faux) = 0,997$$

Maintenant, RO et RF peuvent envoyer leurs π – messages à leurs enfants (*D*).
D'où :

$$\pi(D = Vrai) = 0,9969 \text{ et } \pi(D = Faux) = 0,0031$$

Annexe A. Réseaux bayésiens

Comme il n'y a pas d'observations, λ de chaque nœud ne change pas (restent à "1") donc les étapes 2,3 et 4 vues dans l'algorithme A.5.2 ne servent à rien dans ce cas.

Nous pouvons maintenant calculer toutes les probabilités marginales de chaque nœud en appliquant la relation A.5.9, d'où les résultats suivants :

$$P(RO = Vrai) = 0,0001 \text{ et } P(RO = Faux) = 0,9999$$

$$P(RF = Vrai) = 0,0030 \text{ et } P(RF = Faux) = 0,9970$$

$$P(D = Vrai) = 0,9969 \text{ et } P(D = Faux) = 0,0031$$

Nous remarquons que si le système est contrôlable dans 99,69% des cas, les trois vannes fonctionnent correctement. Maintenant, nous considérons qu'on a une observation ou une panne (le fluide ne passe plus), c'est à dire, $P(D = Faux) = 1$. L'objectif est de trouver la vanne responsable de ce dysfonctionnement en appliquant l'algorithme de Pearl, et comme le réseau n'est pas un arbre, il faut d'abord faire un regroupement de nœuds (figure A.8).

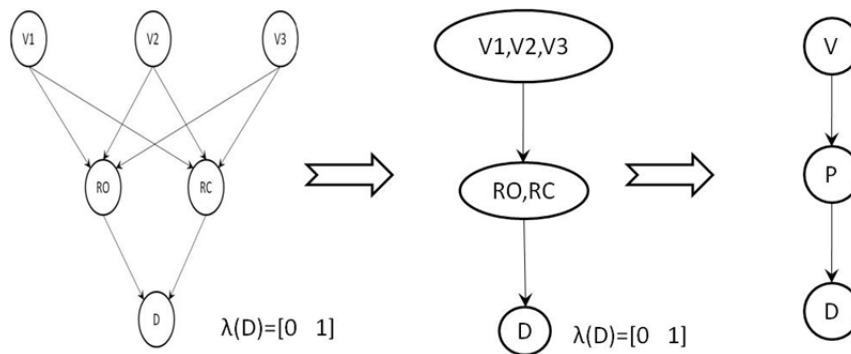


Figure A.8 – Regroupement de nœuds

- V représente les trois vannes.
- P représente la panne (RO et RF), P prend les états suivants (FF, FV, VF, VV).

Calculs des probabilités a posteriori

Nous avons maintenant une chaîne de nœuds auxquels nous pouvons appliquer facilement l'algorithme de Pearl.

D est observé donc $\lambda(D = Faux) = 1$

1. Comme il n'y a pas d'observations au niveau des parents (V_1, V_2 et V_3) cette étape est déjà faite dans le calcul des probabilités marginales.

Annexe A. Réseaux bayésiens

2. Le nœud sans enfants (D) transmet son λ – *message* (calculé en utilisant la relation A.5.12) à son parent (P).

$$\lambda_D(P) = P(D = F|P)$$

D'où le tableau suivant :

| P | VV | VF | FV | FF |
|----------------|------|------|------|------|
| $\lambda_D(P)$ | 1 | 1 | 1 | 0 |

Tableau A.5 – λ – *message* entre D et P

Après avoir reçu le message de son enfant, P peut calculer $\lambda(P)$ en appliquant la relation A.5.13

| P | VV | VF | FV | FF |
|--------------|------|------|------|------|
| $\lambda(P)$ | 1 | 1 | 1 | 0 |

Tableau A.6 – $\lambda(P)$

Maintenant, P peut envoyer son λ – *message* à son parent V .

Calculons maintenant $\lambda(V)$ en multipliant la table de probabilités (A.7) par les différentes probabilités du tableau (A.5), d'où le tableau suivant :

L'inférence bayésienne se termine à cette étape (les étapes 3 et 4 ne sont pas nécessaires car il y a pas d'observations au niveau des parents). Il ne reste maintenant qu'à revenir à l'état précédent du réseau bayésien, c'est à dire, décomposer les nœuds P et V . Pour cela, il suffit d'appliquer la loi marginale (équation A.4.4) afin de calculer les probabilités conditionnelles a posteriori de chaque nœud en fonction de l'observation.

Nous obtenons donc les trois tables de probabilités suivantes :

| V_1 | Ok | RO | RC |
|-------------------|---------|--------|--------|
| $P(V_1 D = faux)$ | 0,00286 | 0,0314 | 0,9656 |

| V_2 | Ok | RO | RC |
|-------------------|--------|--------|--------|
| $P(V_2 D = faux)$ | 0,9716 | 0,0224 | 0,0057 |

| V_3 | Ok | RO | RC |
|-------------------|--------|--------|--------|
| $P(V_3 D = faux)$ | 0,9716 | 0,0224 | 0,0057 |

- **Conclusion** : on peut dire que l'état anormal du système est causé par le fait que la vanne 1 est restée coincée.

| $V \downarrow, P \longrightarrow$ | VV | VF | FV | FF |
|-----------------------------------|------|-----------------|-----------------|-----------------|
| Ok, Ok, Ok | 0 | 0 | 0 | 0,970299 |
| Ok, Ok, RO | 0 | 0 | 0 | 0,0068607 |
| Ok, Ok, RC | 0 | 0 | 0 | 0,00294 |
| Ok, RO, Ok | 0 | 0 | 0 | 0,0068607 |
| Ok, RO, RO | 0 | 0 | 0 | $4,851.10^{-5}$ |
| Ok, RO, RC | 0 | 0 | 0 | $2,079.10^{-5}$ |
| Ok, RC, Ok | 0 | 0 | 0 | 0,00294 |
| Ok, RC, RO | 0 | 0 | 0 | $2,079.10^{-5}$ |
| Ok, RC, RC | 0 | 0 | $8,91.10^{-6}$ | 0 |
| RO, Ok, Ok | 0 | 0 | 0 | 0 |
| RO, Ok, RO | 0 | $4,851.10^{-5}$ | 0 | 0 |
| RO, Ok, RC | 0 | 0 | 0 | 0 |
| RO, RO, Ok | 0 | $4,851.10^{-5}$ | 0 | 0 |
| RO, RO, RO | 0 | $3,43.10^{-7}$ | 0 | 0 |
| RO, RO, RC | 0 | $1,47.10^{-7}$ | 0 | 0 |
| RO, RC, Ok | 0 | 0 | 0 | 0 |
| RO, RC, RO | 0 | $1,47.10^{-7}$ | 0 | 0 |
| RO, RC, RC | 0 | 0 | $6,3.10^{-8}$ | 0 |
| RC, Ok, Ok | 0 | 0 | 0,00294 | 0 |
| RC, Ok, RO | 0 | 0 | $2,079.10^{-5}$ | 0 |
| RC, Ok, RC | 0 | 0 | $8,91.10^{-6}$ | 0 |
| RC, RO, Ok | 0 | 0 | $2,079.10^{-5}$ | 0 |
| RC, RO, RO | 0 | 0 | $1,47.10^{-7}$ | 0 |
| RC, RO, RC | 0 | 0 | $6,3.10^{-8}$ | 0 |
| RC, RC, Ok | 0 | 0 | $8,91.10^{-6}$ | 0 |
| RC, RC, RO | 0 | 0 | $6,3.10^{-8}$ | 0 |
| RC, RC, RC | 0 | 0 | $2,7.10^{-8}$ | 0 |

Tableau A.7 – Table de probabilités de V sachant P

A.5.3 Algorithme de l'arbre de jonction dit JLO

L'algorithme JLO, du nom de ses auteurs : F.V. Jensen, S.L. Lauritzen et K.G. Olesen s'applique à des réseaux ne comprenant que des variables à valeurs discrètes [Jensen et al., 1990] et [Lauritzen & Wermuth, 1989]. Des extensions pour des distributions gaussiennes et des mixtures de gaussiennes ont été proposées dans [Lauritzen & Wermuth, 1989].

L'algorithme de l'arbre de jonction se comporte de la façon suivante :

1. **La phase de construction** : elle nécessite un ensemble de sous-étapes permettant de transformer le graphe initial en un arbre de jonction, dont les nœuds sont

| $V \downarrow, P \rightarrow$ | VV | VF | FV | FF |
|-------------------------------|------|-----------------|-----------------|------|
| Ok, Ok, Ok | 0 | 0 | 0 | 0 |
| Ok, Ok, RO | 0 | 0 | 0 | 0 |
| Ok, Ok, RC | 0 | 0 | 0 | 0 |
| Ok, RO, Ok | 0 | 0 | 0 | 0 |
| Ok, RO, RO | 0 | 0 | 0 | 0 |
| Ok, RO, RC | 0 | 0 | 0 | 0 |
| Ok, RC, Ok | 0 | 0 | 0 | 0 |
| Ok, RC, RO | 0 | 0 | 0 | 0 |
| Ok, RC, RC | 0 | 0 | $8,91.10^{-6}$ | 0 |
| RO, Ok, Ok | 0 | 0 | 0 | 0 |
| RO, Ok, RO | 0 | $4,851.10^{-5}$ | 0 | 0 |
| RO, Ok, RC | 0 | 0 | 0 | 0 |
| RO, RO, Ok | 0 | $4,851.10^{-5}$ | 0 | 0 |
| RO, RO, RO | 0 | $3,43.10^{-7}$ | 0 | 0 |
| RO, RO, RC | 0 | $1,47.10^{-7}$ | 0 | 0 |
| RO, RC, Ok | 0 | 0 | 0 | 0 |
| RO, RC, RO | 0 | $1,47.10^{-7}$ | 0 | 0 |
| RO, RC, RC | 0 | 0 | $6,3.10^{-8}$ | 0 |
| RC, Ok, Ok | 0 | 0 | 0,00294 | 0 |
| RC, Ok, RO | 0 | 0 | $2,079.10^{-5}$ | 0 |
| RC, Ok, RC | 0 | 0 | $8,91.10^{-6}$ | 0 |
| RC, RO, Ok | 0 | 0 | $2,079.10^{-5}$ | 0 |
| RC, RO, RO | 0 | 0 | $1,47.10^{-7}$ | 0 |
| RC, RO, RC | 0 | 0 | $6,3.10^{-8}$ | 0 |
| RC, RC, Ok | 0 | 0 | $8,91.10^{-6}$ | 0 |
| RC, RC, RO | 0 | 0 | $6,3.10^{-8}$ | 0 |
| RC, RC, RC | 0 | 0 | $2,7.10^{-8}$ | 0 |

Tableau A.8 – Table de probabilités de $\lambda_P(V)$

des regroupements de nœuds du graphe initial. Cette transformation est nécessaire, d'une part pour éliminer les boucles du graphe, et d'autre part, pour obtenir un graphe plus efficace quant au temps de calcul nécessaire à l'inférence, mais qui reste équivalent au niveau de la distribution de probabilité représentée. Cette transformation se fait en trois étapes :

- la moralisation du graphe,
- la triangulation du graphe et l'extraction des cliques qui formeront les nœuds du futur arbre,
- la création d'un arbre dit arbre de jonction.

2. **La phase de propagation** : il s'agit de la phase de calcul probabiliste où les nouvelles informations concernant une ou plusieurs variables sont propagées à l'ensemble du réseau, de manière à mettre à jour l'ensemble des distributions de probabilités du réseau. Ceci se fait en passant des messages contenant une information de mise à jour entre les nœuds de l'arbre de jonction précédemment construit. A la fin de cette phase, chaque nœud peut calculer sa probabilité conditionnelle en fonction de l'observation.

Le déroulement de cet algorithme sera illustré sur l'exemple de la Fig. A.7.

La moralisation du graphe

La première étape de transformation du graphe est la moralisation. Elle consiste à marier deux à deux les parents de chaque variable, c'est-à-dire à les relier par un arc non-dirigé. Le graphe moral a donc la propriété de créer une clique pour chaque nœud et ses parents.

Après avoir moralisé le graphe, on finit par transformer complètement le graphe en un graphe non-orienté en enlevant les directions de chaque arc (figure ci-dessous).

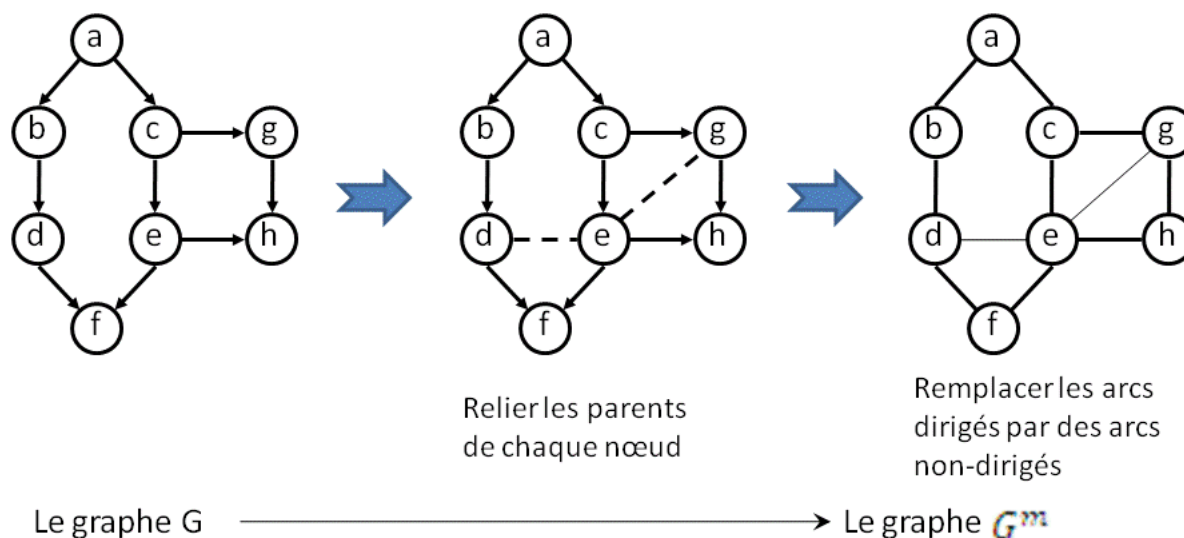


Figure A.9 – Moralisation du graphe

Les arcs en pointillés représentent les arcs qui ont été rajoutés. La moralisation nécessite que tous les nœuds parents d'un même nœud soient reliés deux à deux.

La triangulation

La deuxième étape consiste à trianguler le graphe moral et à en extraire des cliques de nœuds, qui sont des sous-graphes complets du réseau initial. Ces cliques formeront les nœuds de l'arbre de jonction utilisé pour l'inférence.

On dit qu'un graphe est triangulé si et seulement si l'ensemble de ses nœuds peuvent être éliminés. Un nœud peut être éliminé si tous ses voisins sont connectés deux à deux. Donc un nœud peut être éliminé si il appartient à une clique dans le graphe. Une telle clique forme un noeud pour le futur arbre de jonction qui est en train d'être construit. Ainsi, il est possible de trianguler le graphe et de construire les nœuds de l'arbre de jonction en même temps en éliminant les nœuds dans un certain ordre. Si aucun nœud n'est éliminable, il faut en choisir un parmi les nœuds restants et rajouter les arcs nécessaires entre ses voisins pour qu'il devienne éliminable. Le noeud choisi sera celui pour lequel l'espace d'état de la clique formée sera le plus petit possible. En effet, plus les cliques sont petites, plus l'espace de stockage, et le temps de calcul, sont réduits.

Dans le cas général, un graphe non orienté est triangulé si et seulement si tout cycle de longueur supérieur à 3 possède une corde (c'est à dire une arête reliant deux nœuds non adjacents dans le cycle) [Naïm et al., 2004].

Nous illustrons nos propos en appliquant cette étape au graphe moralisé de l'exemple de l'étape précédente.

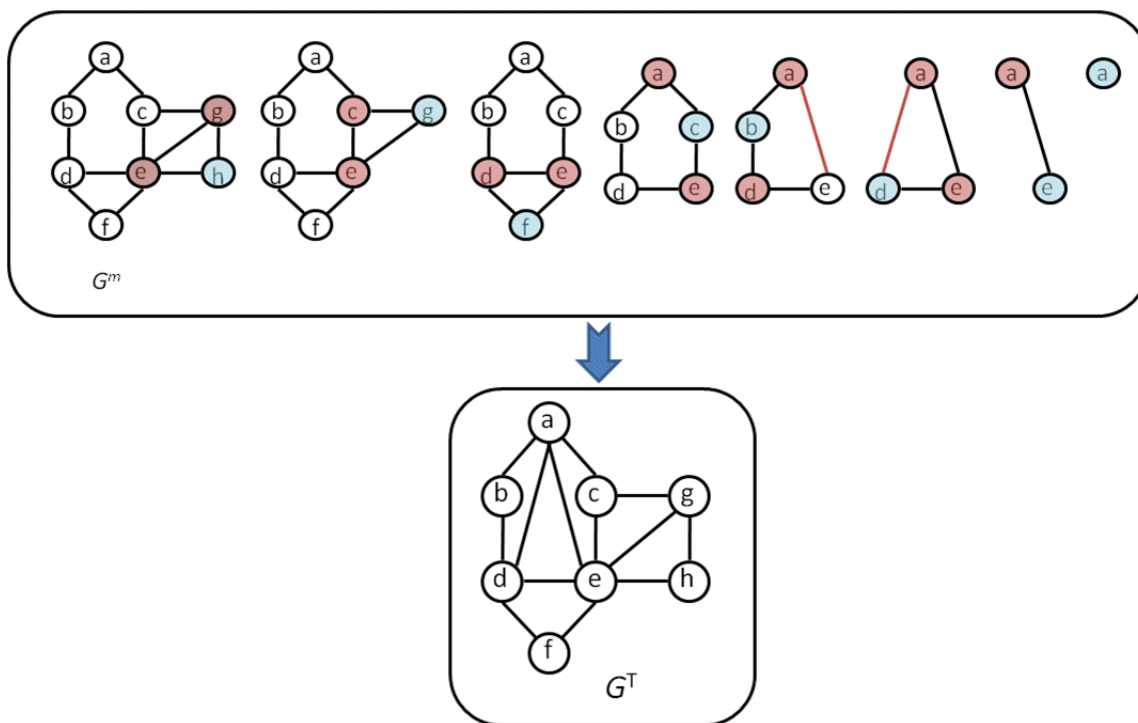


Figure A.10 – Triangulation du graphe

| Num | le nœud éliminé | cliques | l'arc de triangulation ajouté |
|-----|-----------------|---------|-------------------------------|
| 1 | h | egh | - |
| 2 | g | ceg | - |
| 3 | f | def | - |
| 4 | c | ace | a-e |
| 5 | b | abd | a-d |
| 6 | d | ade | - |
| 7 | e | ae | - |
| 8 | a | a | - |

Tableau A.9 – Différentes étapes de triangulation

Remarque : La numérotation faite dans le tableau (A.9) n'est pas unique et donc la triangulation n'est pas unique, et cela en fonction de l'algorithme utilisé. Après avoir moralisé et triangulé notre réseau bayésien, nous pouvons maintenant construire l'arbre de jonction.

L'arbre de jonction

La construction de l'arbre de jonction est la dernière étape avant de procéder à l'inférence proprement dite. Nous rappelons que pour un réseau bayésien donné, l'arbre de jonction est construit une et une seule fois. Les calculs probabilistes auront lieu dans l'arbre de jonction autant de fois que nécessaire. Cependant, pour un réseau bayésien donné, il existe plusieurs arbres de jonction possibles : il sont fonction de l'algorithme de triangulation et de l'algorithme de construction utilisé.

Annexe B

Réseaux sans fil

Sommaire

| | |
|---|------------|
| B.1 Les technologies | 180 |
| B.1.1 Les WWAN (IEEE 802.20) | 181 |
| B.1.2 Les WRAN (IEEE 802.22) | 182 |
| B.1.3 Les WMAN (IEEE 802.16) | 182 |
| B.1.4 Les WLAN (IEEE 802.11) | 182 |
| B.1.5 Les WPAN(IEEE 802.15) | 183 |
| B.1.6 Autres types de réseaux sans fil | 183 |
| B.2 Fonctionnement | 183 |
| B.2.1 Fonctionnement en mode Ad-Hoc | 184 |
| B.2.2 Fonctionnement en mode infrastructure | 184 |
| B.2.3 L'accès au médium | 185 |
| B.3 IEEE 802.11 Wi-Fi | 186 |
| B.3.1 Le protocole Wi-Fi | 186 |
| B.3.2 Les modes d'accès au médium | 187 |
| B.4 802.11e | 189 |
| B.4.1 EDCF (Enhanced DCF) ou EDCA (Enhanced Distribution Channel Access) | 190 |
| B.4.2 EPCF (Enhanced PCF) ou HCF (Hybrid Coordination Function) | 191 |
| B.5 IEEE 802.15.4 | 191 |
| B.6 Le protocole Zigbee | 192 |
| B.6.1 La couche physique | 192 |
| B.6.2 La sous couche d'accès au médium (MAC) | 193 |
| B.7 Le mode d'accès au médium | 194 |

B.1 Les technologies

Les réseaux sans fil ont pris une place importante dans le monde industriel ces dernières années pour apporter des solutions aux problèmes de mobilité, flexibilité et extensibilité limitée ou inexistante dans les réseaux filaires. Cependant, ces réseaux présentent d'autres problèmes de communication qui viennent s'ajouter aux difficultés rencontrées dans les réseaux filaires (retards, gigue, pertes de messages, désynchronisation...). Les réseaux sans fil ajoutent les difficultés suivantes :

- topologie du réseau évoluant au cours du temps (stations mobiles, communication entre un mobile et plusieurs stations au sol),
- sensibilité aux perturbations extérieures (diminution du débit, perte de la communication, sensibilité au piratage),
- possibilité d'utiliser plusieurs canaux de communications,
- principe des terminaux cachés (deux stations dans le même réseau ne communiquent pas forcément).

Chaque catégorie de réseau sans fil correspond à un usage différent, en fonction de ses caractéristiques (vitesse de transmission, débit maximum, coût de l'infrastructure, coût de l'équipement connecté, sécurité, souplesse d'installation et d'usage, consommation et autonomie,...). La Fig. B.1 décrit les différentes catégories de réseaux sans fil suivant leur étendue et la figure B.2 les principales normes de réseaux sans fil.

Les normes IEEE décrivent les deux premières couches (la couche physique et la couche

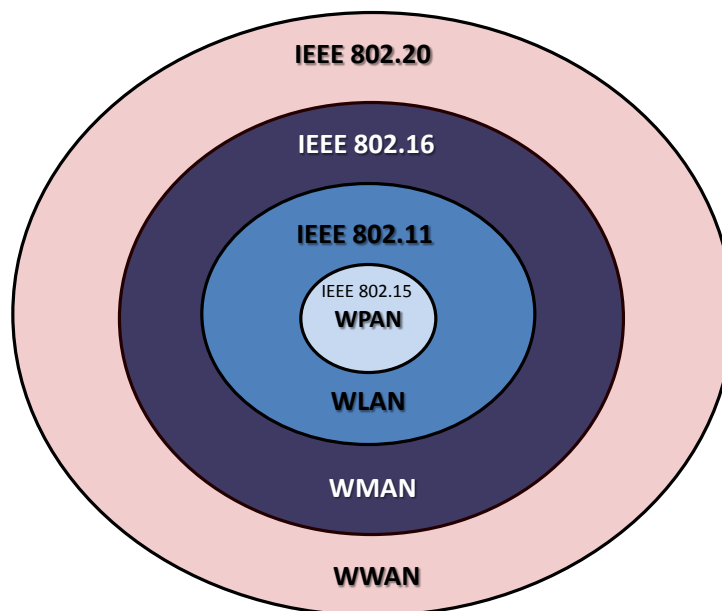


Figure B.1 – Différentes catégories de réseaux sans fil

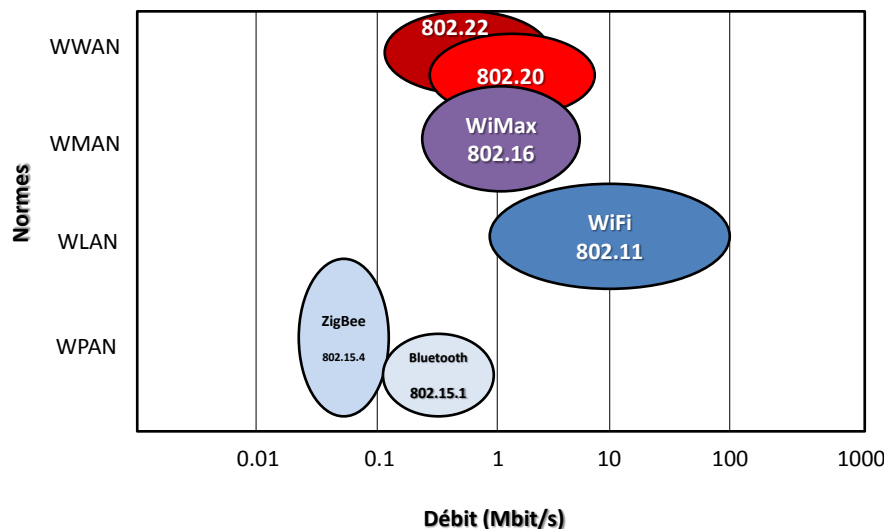


Figure B.2 – Différents réseaux sans fil selon leur vitesse et leur portée

liaison) du modèle OSI (Open Systems Interconnection) qui est un modèle de communications entre ordinateurs proposé par l'ISO (Organisation internationale de normalisation). Les couches supérieures (la couche réseau et la couche transport) définissant les protocoles doivent être indépendantes de l'architecture du réseau. Les sections suivantes décrivent chaque norme.

B.1.1 Les WWAN (IEEE 802.20)

La norme IEEE 802.20, connue sous le nom de MBWA (Mobile Broadband Wireless Access) a été développée en 2002. Elle permet de créer les réseaux métropolitains mobiles qui ont pour but de permettre le déploiement mondial de réseaux sans fil haut débit à un coût accessible et disponible avec une connexion permanente.

Cette norme utilise des bandes de fréquences en dessous des 3,5 GHz. Elle permet des débits maximaux par utilisateur de 1 Mbits/s en descente et 300 Kbit/s en montée avec des cellules d'un rayon de 15km maximum. Elle autorise les terminaux à se déplacer à plus de 250 km/h pour pouvoir être utilisée dans les TGV. D'autres versions sont prévues, utilisant un canal plus large de 5 MHz permettant des débits de 4 Mbits/s en descente et 1,2 Mbit/s en montée pour chaque utilisateur. Les principales technologies utilisées sont les suivantes :

- GSM (Global System for Mobile Communication ou en français Groupe Spécial Mobile)
- GPRS (General Packet Radio Service)
- UMTS (Universal Mobile Telecommunication System).

B.1.2 Les WRAN (IEEE 802.22)

La norme IEEE 802.22 a pour but de créer des réseaux sans fil dont la taille d'une cellule est comprise entre 40 et 100km, en utilisant la bande de fréquence VHF/UHF (Very High Frequency/Ultra High Frequency) entre 54 et 862 MHz (fréquences de télévision) sans interférer avec les canaux utilisés par la télévision. Les transmissions hertziennes des chaînes de télé sont séparées par des bandes de fréquences non utilisées. L'espace libre du spectre peut donc permettre à d'autres réseaux d'échanger des données. Pour cela, les protocoles qui implémentent cette norme ne doivent pas interférer avec les transmissions hertziennes. De même que pour la télévision, la transmission est de type point à multipoint. Cette solution vise à couvrir au maximum le territoire avec des débits et des services de types xDSL. Cette norme permet donc de fournir un accès sans fil aux régions peu peuplées avec un haut débit.

B.1.3 Les WMAN (IEEE 802.16)

La norme IEEE 802.16 est appelée aussi BWA (Broadband Wireless Access). Elle a pour but de créer des réseaux locaux sans fil de la taille d'une ville. Elle offre une alternative aux réseaux câblés entre différents bâtiments.

Il existe plusieurs versions de cette norme. La norme IEEE 802.16 fonctionne dans la bande de fréquence 10 à 66 GHz. La norme IEEE 802.16a fonctionne quant à elle dans la bande de fréquence de 2 à 11 GHz, cette dernière devrait permettre une portée sur un rayon de 50 kilomètres et atteindre une bande de fréquence 10 à 66 GHz comme le transport de flux audio/vidéo, la téléphonie numérique. La norme IEEE 802.16e ajoute la mobilité à ces réseaux (Handoff).

Enfin, la norme IEEE 802.16.2 permet l'interopatibilité entre toutes les normes 802.16. Toutes les normes IEEE 802.16 incluent de base la notion de QoS permettant par exemple le transport de la voix ou la vidéo.

B.1.4 Les WLAN (IEEE 802.11)

La norme IEEE 802.11 sert à créer des réseaux sans fil, d'une taille d'une cinquantaine de mètres. Cette norme est prévue pour transférer de gros débits.

Il existe de nombreuses normes dérivées de celle-ci. Les trois plus connues sont la norme 802.11b qui offre un débit de 11Mbit/s dans la bande de fréquence des 2,4 GHz, la norme IEEE 802.11a qui offre un débit de 54 Mbit/s dans la bande de fréquence des 5,3 GHz et enfin la norme IEEE 802.11g qui est un mariage des deux normes précédentes en offrant un débit de 54 Mbit/s dans la bande de fréquence des 2,4 GHz.

La norme IEEE 802.11i concerne la sécurité d'authentification et le chiffrement de données. La norme IEEE 802.11n peut atteindre des débits de 100 Mbit/s sur la bande de fréquence des 2,4 GHz. La norme IEEE 802.11e veut améliorer la QoS (Qualité de Service), elle est utilisée surtout pour la transmission de voix et de vidéos.

B.1.5 Les WPAN(IEEE 802.15)

La norme IEEE 802.15 sert à créer des petits réseaux sans fil, appelés WPAN. Ces réseaux sont de l'ordre d'une dizaine de mètres et sont prévus pour connecter différents périphériques autonomes entre eux (réseaux de capteurs). Cette norme est appelé Bluetooth, mais ce n'est qu'un seul cas de cette norme. La norme IEEE 802.15.1 a été adoptée à partir des spécifications Bluetooth déjà existantes. Mais la norme IEEE 802.15 est divisée en quatre parties :

- IEEE 802.15.1 : définit le standard Bluetooth 1.X permettant un débit d'environ 1 Mbit/s. Ce débit a été multiplié par 10 avec la norme Bluetooth 2.X.
- IEEE 802.15.2 : définit des recommandations pour l'utilisation de la bande de fréquence des 2.4 GHz (fréquence utilisée par d'autres réseaux sans fil).
- IEEE 802.15.3 : définit la norme UWB (Ultra Wide Band), standard connu sous le nom de Wimedia, géré par la Wimedia Alliance.
- IEEE 802.15.4 : définit la norme ZigBee qui possède un débit faible mais consomme très peu d'énergie.

B.1.6 Autres types de réseaux sans fil

Il existe d'autres types de réseaux sans fil qui sont en cours de développement, on cite : RFID, SP100.11, wirelessHART, 6lowPAN.

B.2 Fonctionnement

Un réseau sans fil est un réseau qui couvre une partie d'une zone d'une dizaine de mètres à plusieurs kilomètres en fonction du réseau utilisé ou des technologies utilisées (voir section B.1). Le fait de diviser une zone en cellules permet de réutiliser la bande de fréquence de transmission pour créer plusieurs réseaux. Une cellule peut être assimilée à une partie de réseau, et peut comme les réseaux filaires, implémenter plusieurs topologies différentes. Les topologies les plus utilisées sont les topologies point à point (Fig. B.3b), point à multi-point (Fig. B.3a) et la topologie en mailles (Fig. B.3c).

Il y a deux modes de fonctionnement d'un réseau sans fil :

- le mode ad-Hoc (Fig. B.4) ;

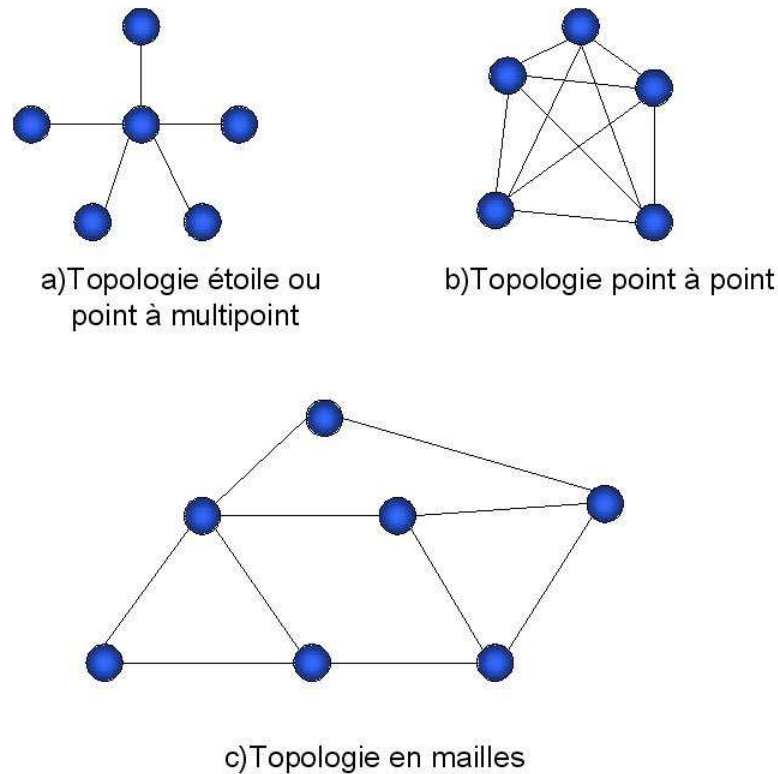


Figure B.3 – Différentes topologies

– le mode infrastructure (Fig. B.5).

B.2.1 Fonctionnement en mode Ad-Hoc

Dans ce type d'architecture les stations jouent chacune le même rôle. En effet, elles doivent opérer de manière autonome et être capables d'effectuer toutes les opérations nécessaires à l'établissement du réseau. En particulier, les stations doivent être capables d'effectuer les procédures d'authentification et d'association.

B.2.2 Fonctionnement en mode infrastructure

Ce mode désigne un réseau composé d'une infrastructure qui permet l'échange de l'information entre les différentes stations du réseau. Cette infrastructure est basée sur un matériel appelé un point d'accès (AP).

Dans ce mode, une cellule d'un réseau est composée d'un point d'accès (AP) et d'un ensemble de stations appelées Basic Server Set (BSS). Il est illustré sur la Fig. B.5.

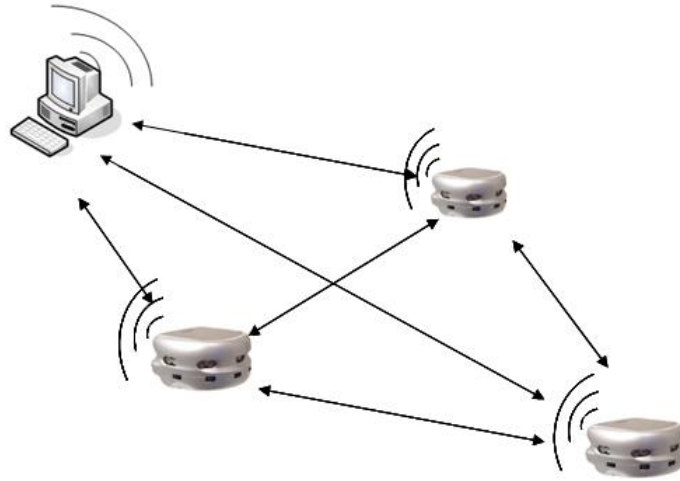


Figure B.4 – Mode de fonctionnement Ad-Hoc



Figure B.5 – Mode de fonctionnement infrastructure

B.2.3 L'accès au médium

L'accès au médium peut se faire de deux manières suivant la topologie du réseau. Les réseaux ad-Hoc utilisent uniquement la méthode d'accès DCF (Distributed Coordination Function) qui s'appuie sur le protocole CSMA/CA. Les autres réseaux utilisent soit la méthode d'accès DCF, soit PCF (Point Coordination Function).

Le mécanisme d'accès de base au médium est le DCF qui se base essentiellement sur "Carrier Sense Multiple Access with Collision Avoidance" (CSMA/CA). Le CSMA est connu notamment dans les réseaux filaires Ethernet qui utilisent principalement le mode d'accès CSMA/CD. Le principe du CSMA/CA est simple. Avant qu'une station ne décide d'émettre, elle vérifie que le canal de transmission n'est pas occupé. La vérification de la disponibilité du médium s'effectue par la couche physique. Si la détection a montré

que le médium est libre, la station peut envisager de transmettre l'information. Sinon, elle doit attendre jusqu'à ce qu'elle reçoive que le canal est libre. Puis, un mécanisme aléatoire génère des délais aléatoires (*CW*) avant transmission (random backoff). Chaque station exécute cet algorithme localement afin de déterminer à quel moment elle peut commencer sa transmission. Cette méthode nécessite l'emploi d'un espace de temps, d'une durée minimale spécifiée, entre les transmissions. Cet espace est appelé IFS (Inter Frame Space) ou espace intertrame. Quatre IFS ont été définis par le standard. Ils sont

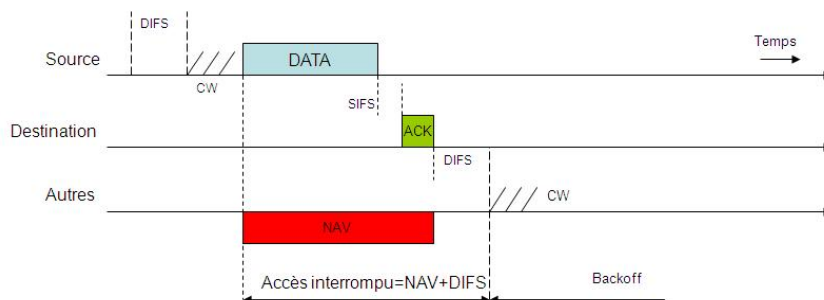


Figure B.6 – Méthode d'accès CSMA/CA

définis du plus court au plus long. Le premier SIFS (Short IFS) est le plus court de tous. Il est utilisé avant la transmission des trames d'acquiescement (Aknowlegment ACK) et entre les trames issues d'une même station. Le DIFS (DCF IFS) est le plus utilisé (avec le SIFS). Il est utilisé en mode DCF comme temps minimal d'attente avant transmission. Enfin, le plus long EIFS (Extended IFS) est utilisé lorsqu'il y a détection de collision pour éviter les collisions en série. Il existe aussi dans le cas de transmission en mode PCF le PIFS (PCF IFS). Il permet aux transmissions PCF d'avoir l'accès au médium par l'utilisation d'un IFS plus petit que celui utilisé pour la transmission des trames en DCF.

B.3 IEEE 802.11 Wi-Fi

B.3.1 Le protocole Wi-Fi

La norme IEEE 802.11 WLAN (Wi-Fi) possède de grandes similitudes avec Ethernet :

- les trames peuvent être envoyés à une, à plusieurs et/ou à toutes les stations,
- les stations partagent le même média, l'air pour le Wi-Fi, le fil pour Ethernet,
- les dialogues peuvent être écoutés.

Là où Ethernet utilise le CSMA/CD, Wi-Fi met en oeuvre différentes stratégies (CSMA/CA,DCF,PCF...) pour éviter les collisions qui obligent de refaire l'émission des messages perdus. Le Wi-Fi utilise principalement le mode d'accès au médium CSMA/CA

décrit dans le paragraphe précédent mais avec le problème des terminaux cachés : les nœuds ne peuvent pas éviter les collisions si ces nœuds sont cachés les uns par rapport aux autres et qu'ils veulent émettre vers le même destinataire. En effet, ils ne détectent pas leurs activités réciproques, c'est pour cela que Wi-Fi emploie une solution alternative qui s'appelle le DCF.

B.3.2 Les modes d'accès au médium

1. **Le cycle DCF** : Le DCF se base principalement sur le mécanisme d'échange de paquets de contrôle RTS-CTS (Fig. B.7). Avant de transmettre ses données, un émetteur envoie un paquet RTS (Request to Send) à son destinataire. Tous les nœuds à portée de communication de l'émetteur qui ont reçu ce RTS savent qu'une communication va commencer. Et comme la durée de la communication est précisée dans le paquet RTS, ces nœuds ne peuvent pas émettre pendant toute cette période. Cette opération est réalisée grâce au NAV (Network Allocation Vector) (voir Fig. B.7) qui stocke la valeur de cette durée et qui joue le rôle d'une horloge. Le récepteur qui reçoit le RTS renvoie un paquet CTS (Clear to Send) s'il n'est pas lui-même bloqué par son NAV. Le CTS a le même effet que le RTS pour les nœuds à portée du récepteur. Lors de la réception du CTS, l'émetteur sait que le médium a été réservé et qu'il peut donc émettre ses données. Ce mécanisme permet de réduire la plupart des collisions dans les configurations de nœuds cachées, mais comme nous le verrons par la suite, il ne permet pas de résoudre tous les problèmes. Avec le mécanisme RTS/CTS, les collisions sont évitées en amont. Mais on perd de la bande passante avec les paquets RTS, CTS, ACK et les temps d'attente. Comme précisé ci-dessus, le DCF ne peut pas résoudre tous les problèmes et il a ses limites :

- DCF est optimal quand il y a peu de stations qui communiquent en même temps. Si elles sont nombreuses, le nombre de collisions augmente et le débit peut chuter d'une façon vertigineuse.
- Une station lente influence tout le réseau, car les temps d'attente sont fixés sur les temps d'émission maximaux des messages sur tout le réseau.
- De plus, vu que chaque station doit attendre le silence pour communiquer, une interférence peut interrompre totalement le trafic.
- DCF ne permet pas de garantir le temps minimal de transmission puisqu'il repose sur un mécanisme aléatoire. Ce n'est pas gênant pour des applications asynchrones (navigation internet, e-mails, etc.). En revanche, ça l'est pour des applications synchrones (systèmes temps réels, systèmes commandés en réseau...) qui nécessitent des transmissions en temps réel. DCF atteint alors sa limite et devient insuffisant.

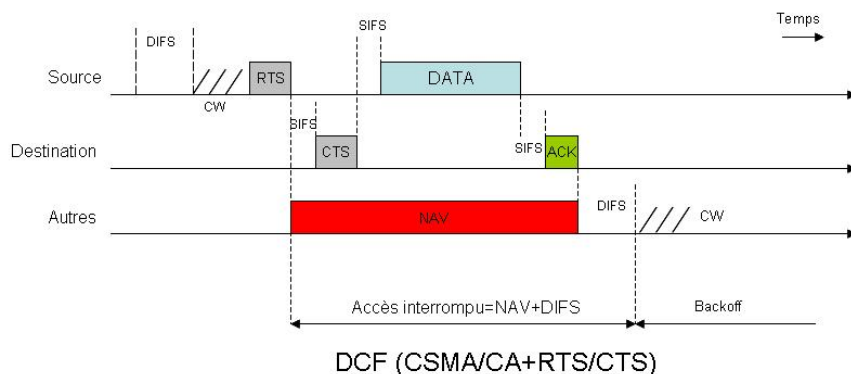


Figure B.7 – Mode RTS-CTS

C'est pour cela qu'un autre mécanisme d'accès au médium a été proposé à savoir le PCF.

2. **Le cycle PCF** Dans le mode PCF, c'est l'AP (point d'accès) qui est le directeur, donc PCF n'est utilisable qu'en mode Infrastructure. Les collisions sont quasi impossibles et les temps de latence garantis : on dit que le système est CF (Contention Free ou en français Libre de Toute Dispute). Le fonctionnement de ce mode est basé sur l'AP qui interroge successivement chacune des stations par une requête CF-Poll (Fig. B.8). La station qui veut transmettre répond immédiatement par un CF-ACK et l'AP lui répond en lui donnant l'autorisation de transmettre et un temps pendant lequel elle peut transmettre. Ensuite, la station émet un ou plusieurs paquets en prenant en compte le temps accordé. En revanche, pour éviter qu'une station, qui a la main et qui n'émet pas, bloque tout le réseau, l'AP attend pendant un intervalle nommé PIFS (PCF Inter Frame Space). Si la station n'a rien émis, alors l'AP passe à la station suivante.

Donc avec cette stratégie, le PCF divise le temps de parole d'une façon équitable. Il est performant pour transférer des données synchrones. Mais, on peut perdre de la bande passante si de nombreuses stations n'ont rien à émettre. Les autres attendent pour leur tour. Pour limiter cela et permettre aussi aux stations incompatibles avec le PCF de communiquer, le standard 802.11 impose au PCF d'être accompagné de DCF. Pendant un instant, toutes les stations sont en PCF, puis elles se mettent en DCF, puis reviennent en PCF, etc (voir Fig. B.9).

3. **Le cycle PCF-DCF** Pour que les stations sachent quand elles peuvent parler librement (DCF) et quand elles doivent attendre que l'AP leur donne la parole (PCF), elles doivent être synchronisées. C'est le rôle des trames balises, envoyées régulièrement par l'AP (Fig. B.9), qui indiquent le début d'une séquence PCF-DCF, la durée de la séquence totale et la durée maximale de la phase PCF. À tout moment pendant la phase PCF, l'AP peut décider de passer à la phase DCF en envoyant une balise appelée CF-End. Une station en mode DCF uniquement

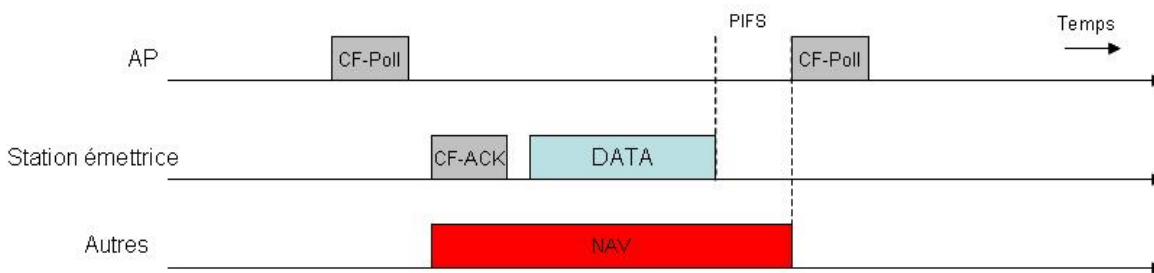


Figure B.8 – Mode PCF



Figure B.9 – Cycle PCF-DCF

pourra donc se connecter à un AP configuré en PCF, mais elle disposera d'une bande passante réduite car elle ne pourra pas communiquer pendant la phase DCF (le PIFS est plus court que le DIFS de sorte que si une station ne connaît pas le mode PCF, elle ne pourra pas prendre la parole pendant la phase PCF, car le silence ne sera pas assez long). En revanche, une station PCF qui se connecte à un AP et qui ne gère que le DCF passera automatiquement dans ce mode. En revanche, avec l'arrivée des applications temps réel exigeantes, (VoIP, vidéoconférence, NCS, etc.), la QoS assurée par le 802.11e, est indispensable.

B.4 802.11e

Bien que le PCF puisse permettre d'améliorer la qualité de service (QoS) pour les applications temps réel, le 802.11e apporte des solutions encore plus complètes :

- On associe chaque paquet Wi-Fi à une classe de trafic TC (Trafic Classe ou AC, Access Priority) particulière. Un numéro lui est ajouté indiquant son niveau de priorité. Le standard 802.11e propose jusqu'à huit niveaux de TC et un AP doit en mettre en oeuvre au minimum quatre.
- Deux nouvelles fonctions de coordination sont proposées. Elles traitent les paquets différemment selon la TC. Les fonctions sont Enhanced DCF ou EDCA (Enhanced Distribution Channel Access) et Enhanced PCF ou HCF (Hybrid Coordination Function).

B.4.1 EDCF (Enhanced DCF) ou EDCA (Enhanced Distribution Channel Access)

EDCF améliore DCF selon trois principes :

- des délais d’attente différents selon la priorité du message (IAFS),
- la gestion des files d’attente selon les priorités par les stations,
- la possibilité pour une station d’émettre plusieurs messages (TXoP).

1. IAFS :

Le mécanisme est le même que DCF. Cependant les messages à haute priorité (TC élevé) ont plus de chances d’être émis rapidement que ceux à basse priorité. Pour cela, EDCF règle les délais DIFS et CW selon les classes. Plus la classe est prioritaire, plus les délais d’attente sont courts. On n’utilise plus de DIFS mais des AIFS (la durée est supérieure au DIFS).

2. Gestion des files d’attente par les stations :

Chaque station gère une file d’attente par classe de trafic et applique des règles probabilistes pour déterminer de quelle file d’attente le prochain paquet à émettre fera partie. Un paquet à transmettre entre donc dans la file d’attente selon sa classe de trafic, puis lorsque son tour arrive, il doit remplir deux conditions avant d’être transmis. La première condition est une condition interne, le paquet transmis doit être contre les paquets des autres files d’attente du même adaptateur. La deuxième est une condition externe, le paquet transmis doit être contre les paquets des autres stations.

3. TxOP (opportunité de transmission), émission en rafale :

EDCF permet aussi aux stations d’émettre plusieurs messages en rafale. La station a une opportunité de transmission (TxOP). La durée maximale d’une TxOP peut être précisée dans les trames balises du point d’accès. Pendant une TXoP la station émet autant de messages qu’elle le désire en rafale en les espaçant par des SIFS. Le SIFS est le délai le plus court, donc personne ne peut l’interrompre.

L’EDFC est simple à mettre en place, cependant, il repose sur le hasard (il peut arriver que certains messages prioritaires soit retardés un peu trop longtemps) et un message peu prioritaire peut être émis avec beaucoup de retard s’il y a un trafic régulier et plus prioritaire sur le réseau, ce qui peut faire perdre de l’efficacité au réseau. C’est pourquoi on utilise EPCF.

B.4.2 EPCF (Enhanced PCF) ou HCF (Hybrid Coordination Function)

EPCF est similaire à PCF (l'AP contrôle les temps de parole, puis il y a une alternance avec une phase EDCF), mais apporte plusieurs améliorations :

- l'AP peut reprendre la main pendant les phases EDCF,
- la possibilité pour une station d'émettre plusieurs messages(TXoP),
- l'AP peut choisir l'ordre dans lequel il interroge les stations,
- le point d'accès délègue la gestion des files d'attente aux stations.

Détaillons un peu chaque amélioration :

1. Le point d'accès peut reprendre la main pendant les phases EDCF :

En EPCF, il y a alternance entre les phases EPCF et EDCF. Cependant, lors d'une phase EDCF, l'AP peut reprendre la main quand il le désire. Pour cela l'AP attend le premier silence d'une longueur PIFS, puisque PIFS est inférieur au DIFS et à l'AIFS.

2. La possibilité pour une station d'émettre plusieurs messages (TxOP) :

Lorsqu'une station obtient la parole elle dispose d'un TxOP comme EPCF, elle peut envoyer des paquets en rafale.

3. l'AP peut choisir l'ordre dans lequel il interroge les stations :

Pour donner la parole à celui qui a le plus besoin, l'AP va connaître les files d'attente de chaque station pour chaque classe de trafic (les stations l'indiquent au début de chaque message dans l'en-tête MAC modifié par le 802.11e). Le point d'accès donne donc la parole en fonction des priorités TC, de la longueur des files d'attente, du temps de parole cumulé pour chaque station et des paramètres de QoS.

4. L'AP délègue les files d'attente :

Le rôle de l'AP ici est de distribuer le temps de parole. Lorsqu'il donne la parole à une station, il n'impose pas une file d'attente à utiliser. C'est la station qui la gère seule. Cela donne une partie du travail et des responsabilités à chaque station afin de réduire la charge sur l'AP.

Comme pour le DCF et le PCF, l'EPCF et l'EDCF peuvent cohabiter en faisant une alternance entre les deux pour donner plus de fluidité au réseau. La Fig. B.10 présente le fonctionnement de cette alternance entre ces deux modes d'accès.

B.5 IEEE 802.15.4

ZigBee est un réseau WPAN de faible débit, faible consommation énergétique, de courte portée et d'une bonne souplesse d'installation. Il est utilisé principalement pour interconnecter des unités embarquées contraintes énergétiquement (réseau de capteurs,

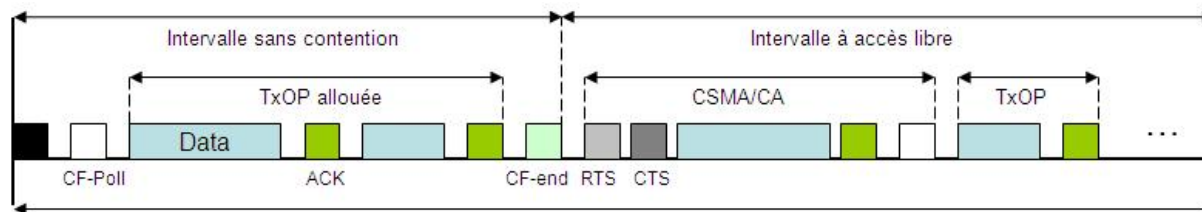


Figure B.10 – Cycle EPCF-EDCF

unités de contrôle ou de commande). Le standard 802.15.4 est un protocole des basses couches de OSI (physique et liaison) et l'Alliance ZigBee définit les hautes couches comme le montre la Fig. B.12.

Cette technologie a pour but la communication de courte distance telle que le propose déjà la technologie Bluetooth, tout en étant moins chère et avec une souplesse et une simplicité d'installation. La couche physique IEEE 802.15.4 supporte trois bandes de fréquence : 2450MHz, 915MHz et 866MHz.

Dans un réseau ZigBee, on peut distinguer deux types de nœuds : FFD (Full Function Device) et RFD (Reduced Function Device)(Fig. B.11).

Le FFD peut assurer trois rôles dans un réseau : coordinateur, routeur ou dispositif relié à un capteur (plus petite fonction possible, appelé terminal). En revanche, le RFD est prévu pour des applications simples (signaler l'état d'un capteur, contrôler l'activation d'un actionneur). Il est considéré comme un terminal (end device), dans le sens où il n'est pas indispensable ou nécessaire au réseau.

Pour communiquer, au moins un FFD et des RFD doivent utiliser le même canal physique. Le FFD peut dialoguer avec des RFD et des FFD, tandis que le RFD dialogue avec un FFD uniquement.

B.6 Le protocole Zigbee

Zigbee est un standard basé sur IEEE 802.15.4, ce qui veut dire que Zigbee utilise le standard IEEE pour construire un modèle protocolaire illustré sur la Fig. B.12.

B.6.1 La couche physique

La couche physique s'occupe principalement du signal, qui est transmis à travers un médium (dans Zigbee c'est l'air). La modulation, DSSS (Direct Sequence Spread Spectrum, étalement de spectre à séquence directe)[Castaño, 2006], le choix du canal,

l'estimation de qualité et les mesures de détection d'énergie [Kim et al., 2007] sont aussi assurés par la couche physique. La technologie ZigBee est aussi basée sur une technologie sans fil à 2,4 Ghz par étalement de spectre (par séquence directe).

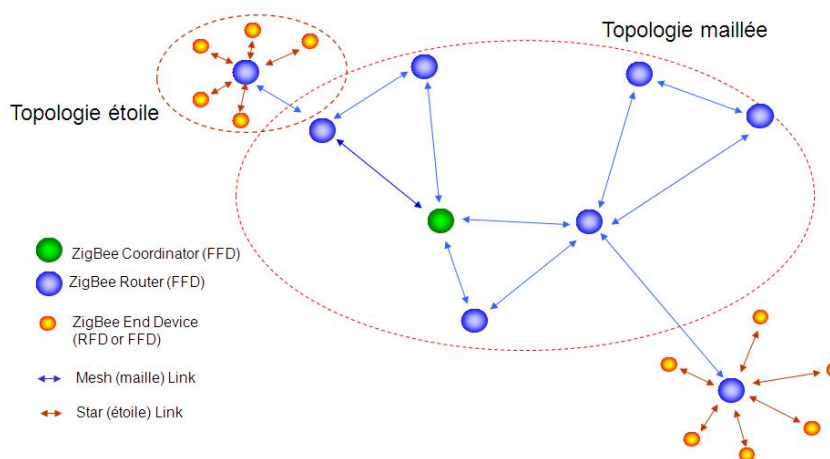


Figure B.11 – Réseau ZigBee

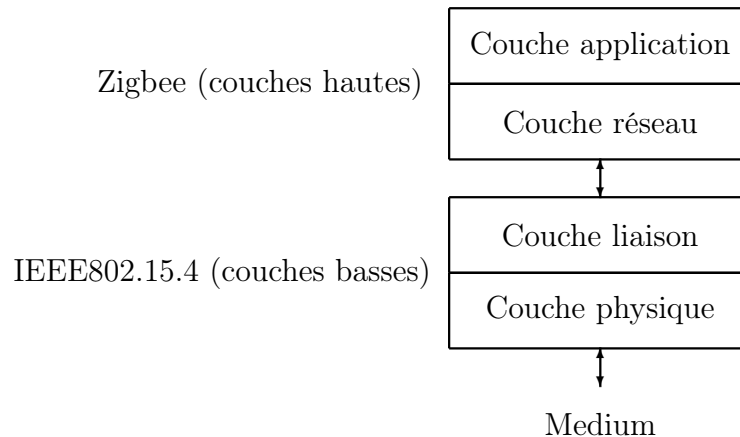


Figure B.12 – Modèle protocolaire

B.6.2 La sous couche d'accès au médium (MAC)

Comme précisé dans la section B.5, il existe deux types de nœuds : les FFD et les RFD (Fig. B.11). Les nœuds FFD supportent la totalité des fonctions de la couche MAC telles que le routage et elles peuvent aussi jouer le rôle d'un coordinateur (Personal Area Network (PAN) coordinator), qui est capable d'envoyer des messages appelés *Beacon* pour assurer la synchronisation entre les différents PAN (Personnel Area Networks). Les

nœuds RFDs sont utilisés seulement comme un terminal ou End-Device et ils sont bien adaptés aux capteurs et actionneurs [Kim et al., 2007].

B.7 Le mode d'accès au médium

Dans le standard IEEE 802.15.4 l'évitement des collisions se fait en utilisant le mode d'accès CSMA/CA. Les caractéristiques de la couche MAC sont la gestion des balises, l'accès au canal, la gestion des GTS (Guaranteed Time Slot)[Howitt & Gutierrez, 2003]. Il existe cependant deux modes de fonctionnement de la couche MAC selon le type de topologie utilisé et le besoin en débit garanti, à savoir :

- le mode sans balise (non-beacon), c'est le mode CSMA/CA,
- le mode avec balise (beacon), avec l'envoi à période régulière d'une balise pour synchroniser les dispositifs, garantissant un débit au nœud ayant un GTS.

1. Le mode sans balise (non-beacon)

Ce mode signifie que le coordinateur reste par défaut dans l'état d'attente de données.

Le dispositif qui veut transmettre écoute si le canal est libre ou non. Si c'est le cas, alors il transmet, sinon il attend un temps aléatoire.

Lorsque le coordinateur a des données à transmettre à un autre FFD ou RFD, il attend que le nœud rentre en communication et lui demande les données. Le coordinateur envoie alors un accusé de réception. Si des données sont en attente, le coordinateur transmet les données en utilisant le même principe (CSMA/CA). S'il n'y a pas de données en attente, le coordinateur envoie une trame de données vide (longueur 0). Le nœud accuse réception des données.

Le mode sans balise est généralement utilisé pour les capteurs qui dorment la majorité du temps. Quand un événement se passe, les capteurs se réveillent instantanément et envoient une trame d'alerte. Le coordinateur, dans ce type de réseau, doit être alimenté sur secteur, car il ne dort jamais : il est par défaut dans l'état d'écoute d'une trame d'événement ou de présence. Dans ce type de réseau, le coordinateur n'émet pas de balise et empêche toute synchronisation des dispositifs.

Cette solution a pour avantage d'optimiser l'autonomie des batteries des capteurs et d'utiliser le canal uniquement lorsqu'il est nécessaire de transmettre des données utiles. Par contre, du fait de CSMA/CA, l'accès au canal n'est pas garanti dans une période donnée (tout dépend de la densité du trafic dans le réseau et du nombre de dispositifs voulant transmettre en même temps).

2. Le mode avec balise (beacon)

L'objectif du mode fonctionnant avec l'envoi périodique d'une balise est de synchroniser les nœuds avec le coordinateur. Dans un réseau ZigBee, tous les nœuds

(FFD et RFD) fonctionnent de façon indépendante. En revanche, pour communiquer sur le réseau, ils doivent savoir le moment où se réveiller et transmettre. Pour cela, il est nécessaire qu'ils se synchronisent par rapport au coordinateur (ou à un routeur) responsable du routage des données à travers le réseau.

Lors de la réception d'une balise (beacon), tous les nœuds sont informés de la durée de la supertrame (période d'activité du coordinateur) et à quel moment ils peuvent transmettre leurs données. Ils reçoivent aussi les informations sur le coordinateur (à partir de quel moment il dort et pour quelle durée). Les nœuds savent alors quand ils peuvent dormir ou transmettre. L'envoi régulier d'une balise permet de synchroniser l'ensemble des dispositifs et d'informer des changements dans la supertrame. De plus, la balise d'un coordinateur ou d'un routeur est transmise sans utiliser CSMA/CA. En effet, tous les nœuds doivent se réveiller quelques instants avant l'émission de la balise et sont mis en attente de cette trame pour se synchroniser. Le nœud écoute la balise du réseau, se synchronise sur la balise et transmet des trames de données en utilisant les slots de la supertrame [Howitt & Gutierrez, 2003], [Streton & Stanfield, 2005](Fig. B.13).

La supertrame est en fait la période d'activité du coordinateur qui a 16 slots de temps. Elle est divisée en deux parties (Fig. B.13) :

- le CAP (Contention Access Period) similaire au mode balise. Dans cette partie tous les dispositifs peuvent transmettre de façon aléatoire, mais en respectant la durée d'un slot,
- le CFP (Contention Free Period) permettant de garantir l'accès au canal à un nœud pendant une durée déterminée en nombre de slots, appelée GTS (Guaranteed Time Slot).

Tous les nœuds qui veulent communiquer pendant la CAP (Contention Access Period) entre deux balises sont mis en concurrence avec les autres en utilisant CSMA/CA. Comme les nœuds peuvent émettre uniquement à partir d'un nouveau slot, l'intervalle de temps entre deux tentatives d'accès au canal sera aligné sur les slots. La transmission est cependant limitée par la taille de la CAP. Si le dispositif n'a pas pu émettre pendant la CAP, il doit attendre la prochaine supertrame. De même, si le nombre de slots nécessaire à la transmission de données est plus important que le nombre de slots restant dans la CAP, alors le dispositif devra différer l'envoi à la prochaine supertrame. Toutes les transactions doivent être terminées avant la prochaine balise du réseau. Le coordinateur peut dédier des portions de la supertrame à des applications. Ces portions sont appelées Guaranteed Time Slots (GTS). Ces slots sont réservés en partant du dernier slot de la supertrame et en revenant vers la balise. Le coordinateur peut allouer jusqu'à 7 GTS. Un GTS peut occuper plus d'un slot. Mais une portion suffisante du CAP doit être réservée pour prévoir l'arrivée de nouveaux nœuds dans le réseau.

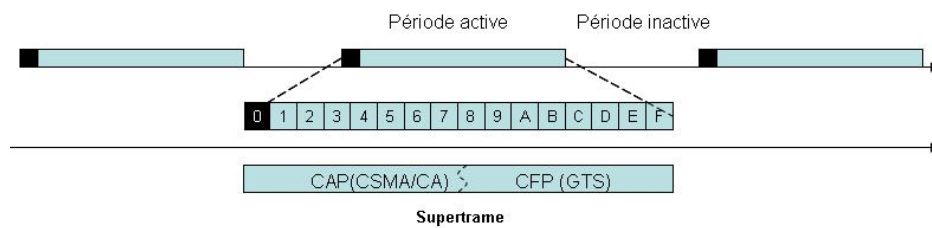


Figure B.13 – Structure de la super trame

Bibliographie

- Aicardi, M., Casalino, G., Bicchi, A. & Balestrino, A. [1995], Closed loop steering of unicycle like vehicles via Lyapunov techniques, *in* 'IEEE Robotics and Automation Magazine', Vol. 2(1), pp. 27–35.
- Anderson, M., Henriksson, D. & Cervin, A. [2007], Truetime 1.5 reference manual, Technical report.
- Angermann, M. & Kammann, J. [2002], Cost metrics for decision problems in wireless ad-hoc networking, *in* 'IEEE CAS Workshop on Wireless Communications and Networking, Pasadena, California, USA'.
- Arzen, K.-E., Bicchi, A., Dini, G., Hailes, S., Johansson, K. H., Lygeros, J. & Tzes, A. [2007], A component-based approach to the design of networked control systems, *in* 'European Control Conference, Kos , Greece'.
- Aubrun, C., Simon, D. & Song, Y.-Q. [2010], *Co-design Approaches for Dependable Networked Control Systems*.
- Ayyub, B. M. [2001], Elicitation of expert opinions for uncertainty and risks, *in* 'CRC press'.
- Baillieul, J. & Antsaklis, P. J. [2007], Control and communication challenges in networked real-time systems, *in* 'Proceedings of the IEEE', Vol. 95, pp. 9–28.
- Benzerrouk, A., Adouane, L., Martinet, P. & Andreff, N. [2008], Toward an hybrid control architecture for a mobile multi-robots system, *in* 'Proceeding of the 3rd National Conference on Control Architectures of Robots, Bourges, France'.
- Berbra, C. [2009], Diagnostic des systèmes embarqués en réseau. Application à un mini-drone hélicoptère, PhD thesis, Institut Polytechnique de Grenoble, Gipsa-Lab.
- Berbra, C., Gentil, S. & Lescecq, S. [2009], Hybrid priority scheme for networked control quadrotor, *in* '17th Mediterranean Conference on Control and Automation, MED'09, Grèce'.
- Berbra, C., Gentil, S., Lescecq, S. & Thiriet, J. [2008], Co-design for a safe network control DC motor, *in* 'Proceeding of 17th IFAC World Congress, Seoul, South Korea'.
- Bertsekas, D. & Gallager, R. [1987], Data networks, *in* 'Englewood Cliffs, NJ : Prentice-Hall'.

Bibliographie

- Bianchi, G. [2000], Performance analysis of the IEEE 802.11 distributed coordination function, *in* 'IEEE Journal on Selected Areas in Communications', Vol. 18, pp. 535–547.
- Bianchi, G., Fratta, L. & Oliveri, M. [1996], Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol, *in* 'IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Taipei, Taiwan', pp. 407–411.
- Branicky, M. S., Liberatore, V. & Phillips, S. M. [2003], Networked control system co-simulation for co-design, *in* 'American Control Conference, Denver, USA'.
- Branicky, M. S., Phillips, S. M. & Zhang, W. [2002], Scheduling and feedback co-design for networked control systems, *in* 'Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA'.
- Carnevale, D., Teel, A. R. & Nesic, D. [2007], A Lyapunov proof of an improved maximum allowable transfer interval for networked control system, *in* 'IEEE Transactions on Automatic Control', Vol. 52, pp. 892–897.
- Carvalho, M. M. & Garcia-Luna-Aceves, J. J. [2003], Delay analysis of IEEE 802.11 in single-hop networks, *in* 'Proceedings of the 11th IEEE International Conference on Network Protocols, Atlanta, GA, USA'.
- Castaño, J. G. [2006], Algorithms and protocols enhancing mobility support for wireless sensor networks based on Bluetooth and Zigbee, Technical report, Department of Computer Science and Electronics, Malardalen University, Sweden.
- Cervin, A. & Eker, J. [2005], Control-scheduling codesign of real-time systems : The control server approach, *in* 'Journal of Embedded Computing', Vol. 1, pp. 209–224.
- Chatzimisios, P., Boucouvalas, A. & Vitsas, V. [2003], Packet delay analysis of IEEE 802.11 MAC protocol, *in* 'Electronics Letters', Vol. 39, pp. 1358–1359.
- Chen, L.-J., Sun, T., Chen, B., Rajendran, V. & Gerla, M. [2004], A smart decision model for vertical handoff, *in* 'Proceedings of the 4th International Workshop on Wireless Internet and Reconfigurability, Athens, Greece'.
- Chhaya, H. S. & Gupta, S. [1997], Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol, *in* 'Wireless Networks', Vol. 3, pp. 217–234.
- Chunmao, L. & Jian, X. [2006], Adaptive delay estimation and control of networked control systems, *in* 'IEEE International Symposium on Communications and Information Technology, Bangkok, Thailand'.
- Cooper, G. [1988], Probabilistic inference using bayesian networks is np-difficile, *in* 'Technical Report KSL-87-27, Medical Computer Science, Stanford University'.
- Cremean, L. B., Foote, T. B., Gillula, J. H., Kogan, G. H. H. D., Kriechbaum, K. L., Lamb, J. C., Lindzey, J. L. L., Rasmussen, C. E., Stewart, A. D., Burdick, J. W. & Murray, R. M. [2006], Alice : An information-rich autonomous vehicle for high-speed desert navigation, *in* 'Journal of Robotic Systems', Vol. 23, pp. 777–810.

Bibliographie

- de Wit, C. C., Jaglin, J. & Siclet, C. [2007], Energy-aware 3-level coding and control co-design for sensor network systems, *in* '16th IEEE International Conference on Control Applications, Singapour, Singapore'.
- de Wit, C. C. & Sordalen, O. [1992], Exponential stabilization of mobile robots with non-holonomic constraints, *in* 'IEEE Trans. Automatic Control', Vol. 37, pp. 1792–1797.
- Elia, N. & Eisebeis, J. [2004], Limitation of linear control over packet drop networks, *in* 'Proceedings of 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas', Vol. 5, pp. 5152–5157.
- Fierro, R. & Lewis, F. [1996], Practical point stabilization of a nonholonomic mobile robot using neural networks, *in* 'Proceeding Conference on Decision and Control, Kobe, Japan'.
- Fu, J. & Dai, Y. [2009], Mixed H_2/H_∞ control for networked control systems (NCSs) with markovian packet-loss, *in* 'Proceedings of the International Conference on Web Information Systems and Mining', pp. 563–575.
- Gao, H. & Chen, T. [2008], Network-based H_∞ output tracking control, *in* 'IEEE Transactions on Automatic Control', Vol. 53, pp. 655–667.
- Georgiev, D. & Tilbury, D. [2006], Packet-based control : The H_2 -optimal solution, *in* 'Automatica', Vol. 42, pp. 137–144.
- Guo, Y. & Li, S. [2010], A new networked predictive control approach for systems with random network delay in the forward channel, *in* 'IEEE Transactions on Industrial Electronics', Vol. 41, pp. 511–520.
- Gupta, V., Spanos, D., Hassibi, B. & Murray, R. M. [2005], Optimal LQG control across packet-dropping links, *in* 'Systems and Control Letters', pp. 360–365.
- Hadjicostis, C. N. & Touri, R. [2002], Feedback control utilizing packet dropping network links, *in* 'Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, USA', Vol. 2, pp. 1205–1210.
- Hassan, K. [1996], *Nonlinear Systems*, 2nd edn, Prentice Hall.
- Hassibi, A., Boyd, S. & How, J. [1999], Control of asynchronous dynamical systems with rate constraints on events, *in* 'Proceedings of 38th IEEE Conference on Decision and Control, Phoenix, Arizona, USA'.
- Hespanha, J., Naghshtabrizi, P. & Xu, Y. [2007], A survey of recent results in networked control systems, *in* 'Proceeding of IEEE special issue on Technology of Networked Control Systems', Vol. 95, pp. 138–162.
- Ho, T. S. & Chen, K. C. [1996], Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LAN, *in* 'Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Taipei, Taiwan', pp. 392–396.

Bibliographie

- Howitt, I. & Gutierrez, J. A. [2003], IEEE 802.15.4 low rate-wireless personal area network coexistence issues, Technical report.
- Huang, L. & Lai, T.-H. [2002], On the scalability of IEEE 802.11 ad-hoc networks, *in* ‘ACM MobiHoc, EPFL Lausanne, Suisse’.
- Huo, Z., Fang, H. & Yan, G. [2005], Co-design for NCS robust fault-tolerant control, *in* ‘IEEE International Conference on Industrial Technology (ICIT05)’.
- Imer, O. C., Yuksel, S. & Basar, T. [2006], Optimal control of LTI systems over unreliable communication links, *in* ‘Automatica’, Vol. 42, pp. 1429–1439.
- Isermann, R. [1997], Supervision : fault-detection and fault-diagnosis methods. an introduction, *in* ‘Control engineering practice’, Vol. 5, pp. 639–652.
- Isermann, R. [2005], Model-based fault-detection and diagnosis-status and applications, *in* ‘Annual Reviews in Control’, Vol. 29, pp. 71–85.
- Jansson, M. [2004], Fault Isolation Utilizing Bayesian Networks, *in* ‘Master’s Thésis, Royal Institute of technology, Sweden’.
- Jensen, F., Lauritzen, S., & Olesen, K. [1990], Bayesian updating in recursive graphical models by local computations, *in* ‘Computational Statistical Quaterly’, Vol. 4, pp. 269–282.
- Jia, N., Song, Y.-Q. & Simonot-Lion, F. [2007], Graceful degradation of the quality of control through data drop policy, *in* ‘European Control Conference - ECC 2007, Kos, Greece’.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. & Saul, L. K. [1999], An introduction to variational methods for graphical models, *in* ‘Machine Learning’, Vol. 37, pp. 183–233.
- Juanole, G., Mouney, G., Sauter, D., Aubrun, C. & Calmettes, C. [2010], Decision making improvement for diagnosis in networked control systems based on dynamic message scheduling, *in* ‘18th Mediterranean Conference on Control and Automation, Marrakech, Maroc’.
- Khan, Z. H., Genon-Catalot, D. & Thiriet, J.-M. [2009], Co-design in heterogeneous wireless networked control systems, *in* ‘International Multiconference on Computer Science and Information Technology, Mragowo, Pologne’.
- Kim, D. & Kim, J. [2003], A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer, *in* ‘Robotics and Autonomous Systems’, Vol. 42, pp. 17–30.
- Kim, D., Park, S., Kang, K. & Lee, D. [2007], A deterministic wireless network for feedback control based on IEEE 802.15.4, *in* ‘7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems, Toulouse, France’.
- Lambercy, F. & Caprari, G. [2007], Khepera3 user manual version 2.2, Technical report.

Bibliographie

- Lannoy, A. & Procaccia, H. [2001], L'utilisation du jugement d'expert en sûreté de fonctionnement, *in* 'Lavoisier'.
- Lauritzen, S. & Wermuth, N. [1989], Graphical models for associations between variables, some of which are qualitative and some quantitative, *in* 'Annals of Statistics', Vol. 17, pp. 31–57.
- Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M. & Davis, R. E. [2007], Collective motion, sensor networks, and ocean sampling, *in* 'Proceedings of the IEEE', Vol. 95, pp. 48–74.
- Li, C., Xiao, J. & Zhang, Y. [2006], Stochastic delay estimation and adaptive control of networked control systems, *in* 'Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization, Lisbon, Portugal', pp. 72–76.
- Li, L. & Wang, F.-Y. [2008], Control and communication synthesis in networked control systems, *in* 'International Journal of Intelligent Control and Systems', Vol. 13, pp. 81–88.
- Ling, Q. & Lemmon, M. [2003], Optimal dropout compensation in networked control systems, *in* 'IEEE conference on decision and control, Maui, Hawaii'.
- Liu, F., Yao, Y., He, F. & Chen, S. [2008], Stability analysis of networked control systems with time-varying sampling periods, *in* 'Journal of Control Theory and Application', Vol. 6, pp. 22–25.
- Liu, G.-P., Xia, Y., Chen, J., Rees, D. & Hu, W. [2007], Networked predictive control of systems with random network delays in both forward and feedback channels, *in* 'Proceedings of the 44th Conference on Decision and Control, and the European Control Conference, Seville, Spain', Vol. 54, pp. 1282–1297.
- Ma, C. & Fang, H. [2005], Stability of networked control systems with multi-step delay based on time-division algorithm, *in* 'Journal of Control Theory and Application', Vol. 4, pp. 404–408.
- Martinez, S., Cortés, J. & Bullo, F. [2007], Motion coordination with distributed information, *in* 'IEEE Control Systems Magazine', Vol. 27(4), pp. 75–88.
- Mechraoui, A., Khan, Z. H., Thiriet, J.-M. & Gentil, S. [2009], Co-design for wireless networked control of an intelligent mobile robot, *in* 'International Conference on Informatics in Control, Automation and Robotics (ICINCO), Milan, Italy'.
- Mechraoui, A., Thiriet, J.-M. & Gentil, S. [2008], Control of an intelligent robot system over a wireless network, *in* 'Proceeding of the 23rd IAR Workshop on Advanced Control and Diagnosis, Coventry, United Kingdom'.
- Mechraoui, A., Thiriet, J.-M. & Gentil, S. [2010a], Aide à la décision et diagnostic par réseaux bayésiens d'un robot mobile commandé en réseau, *in* 'Sixième Conférence Internationale Francophone d'Automatique Nancy, France'.

Bibliographie

- Mechraoui, A., Thiriet, J.-M. & Gentil, S. [2010*b*], On-line distributed bayesian decision and diagnosis of wireless networked mobile robots, *in* ‘18th Mediterranean Conference on Control and Automation, Marrakech, Maroc’.
- Montestruque, L. A. & Antsaklis, P. J. [2002], Model-based networked control systems, necessary and sufficient conditions for stability, *in* ‘Proceedings of the 10th Mediterranean Conference on Control and Automation, Lisbon, Portugal’.
- Montestruque, L. A. & Antsaklis, P. J. [2003], Stochastic stability for model-based networked control systems, *in* ‘Proceedings of the American Control Conference, Denver, Colorado’.
- Naghstibrizi, P. & Hespanha, J. P. [2005], Designing an observer-based controller for a network control system, *in* ‘Proceedings of the 44th Conference on Decision and Control, and the European Control Conference, Seville, Spain’, pp. 848–853.
- Naïm, P., Willemin, P., Leray, P., Pourret, O. & Becker, A. [2004], *Réseaux bayésiens*, 3rd edn, Eyrolles.
- Nkansa-Gyekye, Y. & Agbinya, J. [2007], Vertical handoff decision algorithm for UMTS-WLAN, *in* ‘Proceeding of the 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, Sydney, Australia’.
- Pearl, J. [1988], *Probabilistic reasoning in intelligent systems : Networks of plausible inference*, Morgan Kaufman Publishers.
- Richard, J.-P. & Divoux, T. [2007], *Systèmes commandés en réseau*.
- Schwager, M., Slotine, J. & Rus, D. [2007], Decentralized adaptive control for coverage with networked robots, *in* ‘Proceeding of IEEE International Conference on Robotics and Automation, Roma, Italy’.
- Seiler, S. C. S. P. [2003], Estimation with lossy measurements : Jump estimators for jump systems, *in* ‘IEEE Transaction on Automatic Control’, Vol. 48, pp. 2163–2171.
- Shachter, R., D’Ambrosio, B. & Favero, B. A. D. [1990], Symbolic probabilistic inference in belief networks, *in* ‘AAAI-90’, pp. 126–131.
- Shafer, G. & Shenoy, P. [1988], Clocal computation in hypertrees, *in* ‘Working Paper No 201, Business School, University of Kansas’.
- Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M. & Sastry, S. [2004], Kalman filtering with intermittent observations, *in* ‘IEEE Transactions on Automatic Control’, Vol. 49, pp. 1453–1464.
- Stevens-Navarro, E. & Wong, V. [2006], Comparison between vertical handoff decision algorithms for heterogeneous wireless networks, *in* ‘Proceeding of the 63rd IEEE Vehicular Technology Conference’, Vol. 2, pp. 947–951.
- Streeton, M. & Stanfield, C. [2005], Zigbee : the telemetry solution ?, *in* ‘IEE Telemetry and Telematics Seminar, London, UK’.

Bibliographie

- Su, R., Wonham, W., Kurien, J. & Koutsoukos, X. [2002], Distributed diagnosis for qualitative systems, *in* '6th International Workshop on Discrete Event Systems, Zaragoza', pp. 169–174.
- Subbiah, A. & Blough, D. M. [2002], Distributed diagnosis in dynamic fault environments, *in* 'IEEE Transactions on Parallel and Distributed Systems', Vol. 15, pp. 169–174.
- Toibero, J. M., Carelli, R. & Kuchen, B. [2007], Switching control of mobile robots for autonomous navigation in unknown environments, *in* 'Proceeding of IEEE International Conference on Robotics and Automation, Roma, Italy'.
- Tripathi, N. D., Reed, N. J. H. & Vanlandingham, H. F. [1998], Handoff in cellular systems, *in* 'IEEE Personal communications'.
- Val, T., Fraise, P. & Andreu, D. [2003], Vers l'utilisation de bluetooth pour la commande à distance de robots mobiles, *in* 'Journal Européen des Systèmes Automatisés', Vol. 37, pp. 859–892.
- Varsakelis, D. H. & Zhang, L. [2008], LQG control of networked control systems with access constraints and delays, *in* 'International Journal of Control', Vol. 81, pp. 1266–1280.
- Verron, S., Tiplica, T. & Kobi, A. [2008], Détection de fautes par réseaux bayésiens dans les procédés multivariés, *in* 'Diagnostic des systèmes complexes D. Lefebvre, H. Chafouk, A. El Hami, M. Bennouna'.
- Verron, S., Weber, P., Theilliol, D., Tiplica, T., Kobi, A. & Aubrun, C. [2009], Decision with bayesian network in the concurrent faults event, *in* '7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelone, Espagne'.
- Vu, H. L. [2006], Collision probability in saturated IEEE 802.11 networks, *in* 'Australian Telecommunication Networks and Applications Conference, Melbourne, Victoria, Australia'.
- Walsh, G. C., Beldiman, O. & Bushnell, L. G. [2001], Asymptotic behavior of nonlinear networked control systems, *in* 'IEEE Transaction on Automatic Control', Vol. 46, pp. 1093–1097.
- Wang, H., Katz, R. & Giese, J. [1999], Policy-enabled handoffs across heterogeneous wireless networks, *in* 'Proceeding of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana, USA'.
- Wang, L., Chen, A. & Chen, H. [2007], Network selection with joint vertical and horizontal handoff in heterogeneous Wlan and mobile Wimax systems, *in* 'Proceedings of the 63rd IEEE Vehicular Technology Conference, Melbourne, Australia'.
- Weber, P. & Simon, C. [2008], Dynamic evidential networks in system reliability analysis : A dempster shafer approach, *in* '16th Mediterranean Conference on Control and Automation, Ajaccio, France'.

Bibliographie

- Wehbi, B., Laouiti, A. & Cavalli, A. R. [2008], Efficient time synchronization mechanism for wireless multi hop networks, *in* '19th annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Cannes, France'.
- Wu, J. & Chen, T. [2007], Design of networked control systems with packet dropouts, *in* 'IEEE Transactions on Automatic Control', Vol. 52, pp. 1314–1319.
- Xiaomei, Z., Yufan, Z. & Guoping, L. [2008], Stochastic stability of networked control systems with network-induced delay and data dropout, *in* 'Journal of control theory and applications', Vol. 6, pp. 405–409.
- Zacharie, M. [2009], Intelligent OkiKoSenPBX1 security patrol robot via network and map-based route planning, *in* 'Journal of Computer Science', Vol. 1, pp. 79–85.
- Zhang, L. & Poole, D. [1992], Sidestepping the triangulation problem in bayesian net computations, *in* 'Proc. of 8th Conference on Uncertainty in Artificial Intelligence, Stanford, California, USA', pp. 360–367.
- Zhang, L. & Poole, D. [1994], A simple approach to bayesian networks computations, *in* 'Proceedings of the 10th Canadian conference on artificial intelligence, Alberta, Canada'.
- Zhang, L., Shi, Y., Chen, T. & Huang, B. [2005], A new method for stabilization of networked control systems with random delays, *in* 'IEEE Transactions on Automatic Control', Vol. 50, pp. 1177–1181.
- Zhang, L. & Varsakelis, D. H. [2006], Communication and control co-design for networked control systems, *in* 'Automatica', Vol. 42, pp. 953–958.
- Zhang, W., Branicky, M. & Phillips, S. [2001], Stability of networked control systems, *in* 'IEEE Control Systems Magazine', Vol. 21, pp. 84–99.
- Zhou, D., Huang, L. & Lai, T.-H. [2008], On the scalability of IEEE 802.11 ad-hoc-mode timing synchronization function, *in* 'ACM Wireless Networks', Vol. 14, pp. 479–499.
- Zhu, H., Yang, Q. & Kwak, K. [2005], Performance analysis of fast handoff with mobility prediction, *in* 'Proceeding of IEEE International Symposium on Communications and Information Technology, Beijing, China'.

