



**HAL**  
open science

## Réseaux de Communication et Applications de Contrôle-Commande

Xuan Hung Nguyen

► **To cite this version:**

Xuan Hung Nguyen. Réseaux de Communication et Applications de Contrôle-Commande. Systèmes embarqués. INSA de Toulouse, 2011. Français. NNT: . tel-00663316v1

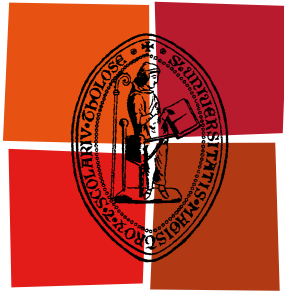
**HAL Id: tel-00663316**

**<https://theses.hal.science/tel-00663316v1>**

Submitted on 26 Jan 2012 (v1), last revised 31 Jan 2012 (v2)

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

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



Université  
de Toulouse

# THÈSE

En vue de l'obtention du  
**DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**

**Délivré par :**  
Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)

**Discipline ou spécialité :**

Systemes Embarqués

---

**Présentée et soutenue par :**

NGUYEN Xuan-Hung

**le :** jeudi 15 décembre 2011

**Titre :**

Réseaux de Communication et Applications de Contrôle-Commande

---

**Ecole doctorale :**

Systemes (EDSYS)

**Unité de recherche :**

Laboratoire d'Analyse et d'Architecture des Systemes (LAAS)

**Directeur(s) de Thèse :**

M. Guy JUANOLE

M. Gérard MOUNEY

**Rapporteurs :**

M. Ye-Qiong SONG

M. Jean-Marc THIRIET

**Membre(s) du jury :**

M. Guy JUANOLE

M. Ye-Qiong SONG

M. Jean-Marc THIRIET

M. Zoubir MAMMERI

M. Germain GARCIA

M. Yvon TRINQUET





Université  
de Toulouse

# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)

**Discipline :** Systèmes Embarqués

**Présentée et soutenue par :** NGUYEN Xuan Hung

Le 15 décembre 2011

# Réseaux de Communication et Applications de Contrôle-Commande

### Directeurs de thèse :

M. Guy JUANOLE      Professeur Emérite, Université Paul Sabatier, Toulouse  
M. Gérard MOUNEY      Maître de Conférences, Université Paul Sabatier, Toulouse

### Rapporteurs :

M. Ye-Qiong SONG      Professeur, Institut National Polytechnique de Lorraine, Nancy  
M. Jean-Marc THIRIET      Professeur, Université Joseph Fourier, Grenoble

### Autres membres du jury :

M. Yvon TRINQUET      Professeur, Université de Nantes, Nantes  
M. Zoubir MAMMERI      Professeur, Université Paul Sabatier, Toulouse  
M. Germain GARCIA      Professeur, Institut National des Sciences Appliquées, Toulouse

**École Doctorale :** Systèmes (EDSYS)

**Unité de Recherche :** Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS)

LAAS-CNRS, 7 avenue du Colonel Roche 31077 Toulouse



Réseaux de Communication et  
Applications de Contrôle-Commande

Nguyen Xuan Hung

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)  
Institut National des Sciences Appliquées de Toulouse (INSA)

Toulouse, Décembre 2011

© Nguyen Xuan Hung 2011  
All Rights Reserved ...

*à mes parents*

*à mon frère*





# Remerciements

Le travail présenté dans ce manuscrit de thèse, a été effectué au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche scientifique (LAAS-CNRS). Je tiens tout d'abord à remercier Monsieur Raja CHATILA, Monsieur Jean-Louis SANCHEZ et Monsieur Jean ARLAT, directeurs successifs du LAAS-CNRS pour m'avoir accueilli au sein du laboratoire. Je tiens à remercier également Monsieur François VERNADAT, responsable du groupe OLC (Outils Logiciels pour la Communication) et Monsieur Olivier BRUN, responsable du groupe MRS (Modélisation et contrôle des Réseaux et Signaux), de m'avoir permis d'effectuer mes travaux de thèse dans ces groupes.

Je remercie vivement Monsieur le Professeur Emérite Guy JUANOLE, directeur de ma thèse. Je tiens à le remercier pour ses conseils scientifiques, ses critiques constructives, ses propositions sur lesquels la thèse s'est construite. Je lui suis particulièrement reconnaissant pour sa disponibilité, des fois tard du soir jusqu'à minuit, des fois tôt à partir de 7h, des fois en weekend. Sa disponibilité était exceptionnelle. Je le remercie également pour sa gentillesse durant ces trois années.

Je remercie de tout mon cœur Monsieur Gérard MOUNEY, co-directeur de ma thèse. Je tiens à le remercier pour m'avoir encadré, pour son suivi, ses conseils utiles, son aide technique et sa gentillesse. Mes remerciements particulièrement pour le premier temps de la thèse où sa présence et son aide m'ont permis d'intégrer rapidement le travail de thèse.

Je tiens à exprimer ma reconnaissance envers tous les membres du jury :

- Monsieur Yvon TRINQUET, Professeur à l'Université de Nantes, je le remercie pour son intervention en tant que président de jury et pour ses remarques utiles sur cette thèse ;
- Monsieur Ye-Qiong SONG, Professeur à l'Institut National Polytechnique de Lorraine (Nancy), et Monsieur Jean-Marc THIRIET, Professeur à l'Université Joseph Fourier (Grenoble), à qui je témoigne ma gratitude pour leur lecture très attentive et détaillée de ce manuscrit ainsi que pour leurs commentaires constructifs ;
- Monsieur Germain GARCIA, Professeur à l'Institut National des Sciences Appliquées de Toulouse et Monsieur Zoubir MAMMERI, Professeur à l'Université Paul Sabatier de Toulouse, que je remercie pour avoir accepté d'être examinateurs lors de la soutenance et pour leur évaluation de cette thèse.

Je tiens à remercier également Monsieur Christophe CALMETTES, Maître de conférences au centre universitaire d'Albi pour son aide en Automatique.

De plus, je remercie Monsieur Slim ABDELLATIF, Maître de conférences à l'Institut National des Sciences Appliquées de Toulouse, pour son aide au plan administratif et ses propositions de travail sur les réseaux maillés, travail qui n'a pas pu être abordé faute de temps.

Je tiens également à remercier Monsieur Khalil DRIRA, Directeur de recherche au LAAS-CNRS, responsable du projet OSEO AMIC-TCP (Architecture de multiplexage Informatique Communication Transport en Commun), pour le financement de mon travail pendant les 2 derniers mois de la thèse.

Je remercie tous mes collègues pour avoir passé de bons moments studieux et joyeux au sein du laboratoire, tout spécialement Thanh, The Tung, Van, Nam, Hung Cao, Quynh Anh, Ty Mai, Rodrigo, Nouha, Luiz.

Je remercie mes copains de Toulouse, d'avoir passé de bons moments de la vie quotidienne (les fêtes, les voyages, les sorties), d'avoir partagé avec moi les moments de stress, les projets personnels et professionnels, tout spécialement Cat Tan, Quang Hai, Thanh Lam, Phan Hang, Hoang Huong, Nam Phuong.

Je remercie chaleureusement mon amie Nhung Kenbe pour ses encouragements, en particulier dans la période de fin de thèse.

Je remercie profondément enfin mes parents et mon frère auxquels je dédie ce travail, pour leur soutien moral et leurs encouragements.

# Table des matières

|  |           |
|--|-----------|
| <b>Introduction</b>  | <b>1</b>  |
| <b>1 Sur les systèmes commandés en réseau</b>                                | <b>5</b>  |
| 1.1 Introduction   | 6         |
| 1.2 De la commande continue à la commande à travers les réseaux              | 7         |
| 1.2.1 Une chronologie  | 7         |
| 1.2.2 Sur la Qualité de Contrôle (QdC)                                       | 9         |
| 1.2.3 Sur les retards  | 10        |
| 1.3 Nécessité d'une approche de co-conception                                | 13        |
| 1.3.1 Ajustement de la période d'échantillonnage                             | 16        |
| 1.3.2 Mise en œuvre de la relation QdC→QdS                                   | 19        |
| 1.4 Sur l'ordonnancement de messages (couche MAC)                            | 21        |
| 1.4.1 Réseau filaire : CAN   | 21        |
| 1.4.2 Réseaux non filaires   | 23        |
| 1.5 Notre travail dans le contexte des recherches actuelles sur les SCR      | 27        |
| 1.5.1 Co-conception  | 27        |
| 1.5.2 Réseaux sans fil et protocoles MAC à priorités et sans collision       | 28        |
| 1.5.3 Un outil de simulation : TrueTime                                      | 29        |
| <b>2 Réseau CAN : concept de priorité hybride (relation QdC→QdS)</b>         | <b>31</b> |
| 2.1 Introduction   | 32        |
| 2.2 Contexte d'étude   | 32        |
| 2.2.1 Application de contrôle-commande distribuée considérée                 | 32        |
| 2.2.2 Réseau CAN avec priorités statiques                                    | 34        |
| 2.2.3 Idées de la priorité hybride   | 36        |
| 2.3 Trois schémas de priorité hybride  | 38        |
| 2.3.1 Schéma ph  | 39        |
| 2.3.2 Schéma ph + sts  | 40        |
| 2.3.3 Schéma ph+std  | 41        |
| 2.4 Implantation d'une application de contrôle-commande et d'un flux externe | 44        |
| 2.4.1 Introduction   | 44        |
| 2.4.2 Références à considérer : QdS et QdC avec les priorités statiques      | 44        |
| 2.4.3 QdS fournie par les trois schémas de priorité hybride                  | 46        |
| 2.4.4 QdC fournie par les différents schémas de priorité hybride             | 50        |
| 2.5 Implantation de plusieurs applications de contrôle-commande              | 52        |
| 2.6 Commande d'un drone à quatre rotors                                      | 54        |
| 2.6.1 Le système étudié  | 54        |
| 2.6.2 Implantation sur CAN   | 55        |
| 2.7 Conclusion   | 58        |

|          |   |            |
|----------|---|------------|
| <b>3</b> | <b>Réseau CAN : Mise en œuvre de la relation <math>QdS \Rightarrow QdC</math></b>   | <b>61</b>  |
| 3.1      | Introduction . . . . .  | 62         |
| 3.2      | Contexte . . . . .  | 62         |
| 3.2.1    | Le type d'application de contrôle-commande considéré . . . . .  | 62         |
| 3.2.2    | Implantations considérées sur un réseau CAN . . . . .   | 64         |
| 3.2.3    | Modèle d'une application de contrôle-commande implantée sur un réseau . . . . .   | 64         |
| 3.2.4    | Critère d'évaluation de performance d'une implantation . . . . .  | 65         |
| 3.3      | Mise en œuvre de la relation $QdS \rightarrow QdC$ . . . . .  | 65         |
| 3.3.1    | Idées principales . . . . .   | 65         |
| 3.3.2    | Méthode des pôles dominants . . . . .   | 66         |
| 3.3.3    | Calculs inhérents à la méthode des pôles dominants . . . . .  | 66         |
| 3.3.4    | Calcul de $u_k$ . . . . .   | 69         |
| 3.3.5    | Validation de la méthode des pôles dominants . . . . .  | 69         |
| 3.3.6    | Considération des différentes implantations sur le réseau CAN . . . . .   | 71         |
| 3.4      | Mise en œuvre de la relation $QdS \Rightarrow QdC$ . . . . .  | 74         |
| 3.4.1    | Idée de base . . . . .  | 74         |
| 3.4.2    | Implantation de la relation $QdS \Rightarrow QdC$ . . . . .   | 75         |
| 3.4.3    | Evaluation de performance . . . . .   | 76         |
| 3.5      | Récapitulation des résultats obtenus . . . . .  | 76         |
| 3.6      | Conclusion . . . . .  | 79         |
| <b>4</b> | <b>Réseaux locaux sans fil : Protocoles MAC de type CSMA sans collision</b>   | <b>81</b>  |
| 4.1      | Introduction . . . . .  | 82         |
| 4.2      | Préliminaires . . . . .   | 83         |
| 4.2.1    | La couche physique . . . . .  | 83         |
| 4.2.2    | La sous-couche MAC . . . . .  | 86         |
| 4.3      | Spécification des protocoles MAC CSMA sans collision basés sur des messages BlackBurst . . . . .                          | 87         |
| 4.3.1    | Considération de priorités statiques . . . . .  | 88         |
| 4.3.2    | Considération de priorités hybrides . . . . .   | 92         |
| 4.4      | Spécification des protocoles MAC CSMA sans collision du type « CANlike » . . . . .  | 97         |
| 4.4.1    | Idées de base : Adaptation du protocole CAN au contexte sans fil . . . . .  | 97         |
| 4.4.2    | Notion de signal de synchronisation . . . . .   | 97         |
| 4.4.3    | Evaluation des bits de l'ID . . . . .   | 99         |
| 4.4.4    | Les protocoles . . . . .  | 100        |
| 4.4.5    | Bilan des paramètres du protocole CANlike . . . . .   | 101        |
| 4.5      | Conclusion . . . . .  | 101        |
| <b>5</b> | <b>Réseaux locaux sans fil et applications de contrôle-commande : sur l'utilisation des protocoles MAC sans collision</b> | <b>103</b> |
| 5.1      | Introduction . . . . .  | 104        |
| 5.2      | Contexte de l'étude . . . . .   | 104        |
| 5.2.1    | L'application de contrôle-commande considérée . . . . .   | 104        |

---

|          |  |            |
|----------|--|------------|
| 5.2.2    | Les protocoles considérés et leur paramètres . . . . .   | 105        |
| 5.3      | Protocoles MAC sans collision basés sur des priorités statiques . . . . .                                  | 106        |
| 5.3.1    | Considération d'une seule application de contrôle-commande : étude de<br>l'influence du champ ID . . . . . | 106        |
| 5.3.2    | Considération d'une application de contrôle-commande et d'un flux externe                                  | 109        |
| 5.3.3    | Considération de quatre applications de contrôle-commande . . . . .  | 110        |
| 5.3.4    | Mise en œuvre de la relation $QdS \rightarrow QdC$ . . . . .   | 112        |
| 5.4      | Protocole CANlike avec la priorité hybride $ph^+$ (CANlike-hyb) . . . . .                                  | 113        |
| 5.4.1    | Mise en œuvre de la relation $QdC \rightarrow QdS$ . . . . .   | 113        |
| 5.4.2    | Mise en œuvre de la relation $QdS \rightleftharpoons QdC$ . . . . .  | 113        |
| 5.5      | Récapitulation . . . . .   | 114        |
| 5.6      | Conclusion . . . . .   | 117        |
| <b>6</b> | <b>Conclusion et perspectives</b>  | <b>119</b> |
| <b>A</b> | <b>L'effet des zéros supplémentaires sur la performance du système second ordre</b>                        | <b>123</b> |
| <b>B</b> | <b>Résultats <math>\Delta J/J_0\%</math> pour DCF-WiFi</b>   | <b>125</b> |
|          | <b>Bibliographie</b>   | <b>127</b> |
|          | <b>Liste des publications</b>  | <b>135</b> |



# Introduction

L'évolution, suivie durant ces dernières années par les technologies de l'information et de la communication, permet aujourd'hui de mettre en œuvre, soit sur un ordinateur (appelons le système informatique centralisé) soit sur un système informatique distribué c'est-à-dire un ensemble de calculateurs géographiquement distribués et inter-connectés au moyen d'un réseau de communication (local ou longue distance ; filaire et/ou sans fil), une grande diversité d'applications et en particulier, des applications temps critique.

Parmi ces applications temps critique, on constate actuellement le développement croissant, dans des secteurs très divers de notre société (systèmes embarqués, robotique, domotique, télé-chirurgie, ...) d'applications de contrôle-commande de procédés physiques. Ce sont précisément ces applications qui concernent notre travail.

Ces applications sont constituées de trois tâches (la tâche capteur et la tâche actionneur qui sont associées au procédé physique ; la tâche contrôleur qui élabore la loi de commande) et ont une structure en boucle (c.-à-d. « feedback » en anglais).

La criticité de ces applications est liée d'une part, aux contraintes temporelles, qui résultent de la dynamique des procédés physiques à observer et à commander et, d'autre part, à la structuration en boucle qui fait de la stabilité un critère de base fondamental.

L'implantation de ce type d'application, qu'elle soit mise en œuvre dans un système informatique centralisé (les tâches capteur, actionneur et contrôleur sont situées dans un même ordinateur) ou dans un système informatique distribué (les tâches capteur et actionneur et la tâche contrôleur sont dans des calculateurs différents et donc elles échangent des messages à travers le réseau) requiert de ce système une Qualité de Service (QoS) réactive, c'est-à-dire compatible avec l'exigence de la Qualité de Contrôle (QdC) ce qui pose d'autant plus de difficultés que le système informatique (centralisé ou distribué) peut être partagé avec d'autres types d'applications.

La QoS dépend des mécanismes opératoires des systèmes informatiques utilisés et donc principalement :

- dans le cas d'un système informatique centralisé, de l'exécutif qui gère l'ordonnancement des tâches (tâches des applications de contrôle-commande et tâches des autres types d'application) ;
- dans le cas d'un système informatique distribué, des exécutifs des calculateurs et des mécanismes de communication entre les tâches distantes (ordonnancement des messages et transfert des messages dans la ressource de transmission).

Bien que le développement d'applications de contrôle-commande sur des systèmes informatiques (centralisés ou distribués) nécessite une bonne compréhension et l'intégration des domaines du contrôle-commande et de l'informatique temps réel (centralisée ou distribuée), on n'a pas eu jusqu'aux années 1990 de vraies relations entre les communautés scientifiques concernées (Automatique, Informatique Temps Réel, Réseaux de communication). Le développement de telles applications résultait de deux étapes séparées et en isolation : d'une part la communauté automatique, faisant abstraction des aspects d'implantation et se situant dans le cadre des modèles des procédés à commander (linéaire, non linéaire) et du déterminisme temporel de la théorie des systèmes échantillonnés (or, pratiquement, les contraintes d'implantations, du fait du partage des



ressources, ne permettent pas d'assurer le déterminisme temporel ; le déterminisme temporel ne pourrait être obtenu que si on utilisait des ressources dédiées), d'autre part, la communauté Informatique temps réel (centralisée ou distribuée) qui n'avait jamais eu comme objectif prioritaire la compréhension de la spécificité des applications de contrôle-commande, en particulier au niveau des besoins en termes d'attributs des tâches des applications de contrôle-commande et des messages échangés par ces tâches.

En conséquence, on peut dire que ces développements isolés traduisaient un conservatisme qui induisait des solutions non optimales. D'où la nécessité d'orienter des recherches de type pluridisciplinaire considérant conjointement les problématiques Automatique - Informatique temps réel - Réseau de communication.

A partir des années 1990-1995, on a commencé à avoir des activités de recherche de type pluridisciplinaire. Comme exemples, citons des études sur la prise en compte de délais introduits par des réseaux [Halevi 1988], leur modélisation et la conception des contrôleurs permettant de compenser ces délais [Luck 1990]. D'autres études ont concerné plus particulièrement le développement d'application de contrôle-commande dans un contexte centralisé (sur un processeur) avec la démonstration de l'inadéquation des techniques standards d'ordonnancement temps réel de tâches pour les tâches des applications de contrôle-commande et de l'importance de définir des techniques basées sur des critères de performances du contrôle (notion de conception conjointe contrôle-ordonnancement de tâches [Marti 2002], [Cervin 2003], [Simon 2003]).

Au début des années 2000, sont apparues dans différents pays, des communautés de personnes ayant des problématiques de recherche touchant à ces trois domaines que sont les applications de contrôle-commande (Automatique), les systèmes informatiques temps réel et les Réseaux de communication et qui ont œuvré pour la reconnaissance d'un thème fédérateur appelé *les Systèmes Commandés en Réseau* (« Networked Control Systems »). Un « Technical Committee » IFAC nommé « Networked Systems » a d'ailleurs été créé en 2005. C'est dans ce contexte que se situe le travail présenté dans ce mémoire de thèse qui concerne la mise en œuvre d'applications de contrôle-commande dans un contexte réseau de communication local (filaire, sans fil). L'objectif final est de réaliser une conception conjointe (contrôle-ordonnancement des messages) c.-à-d. de mettre en œuvre une relation bidirectionnelle entre la QdC fournie par le contrôleur et la QdS fournie par l'ordonnancement des messages dans le réseau (relation notée  $QdS \rightleftharpoons QdC$ ).

Une grande partie du travail est consacrée à la définition et l'étude de techniques d'ordonnancement des messages sur la base de priorités représentatives de la connaissance que le réseau a de la QdC. On utilise, pour caractériser la priorité d'un message d'un flux, la notion de priorité hybride qui consiste en deux composantes : une composante représentant la priorité du flux (c'est une priorité statique) et une composante représentant l'urgence de la transmission du message (c'est une priorité dynamique qui dépend de la QdC). Cette notion permet de mettre en œuvre la relation ( $QdC \rightarrow QdS$ ) (« Application aware dynamic QoS adaptation » [Song 2009]). Plusieurs modalités d'implantation de la notion de priorité hybride ont été spécifiées en considérant la couche MAC du réseau CAN. Toujours dans le cadre de cette partie de travail, et compte tenu du grand développement actuel des réseaux sans fil ainsi que de leur utilisation grandissante dans le contexte industriel, nous avons développé une étude de spécification des principaux protocoles MAC de type CSMA (Carrier Sense Multiple Access) sans collision pour les réseaux sans fil. Ces protocoles

sans collision permettent d'assurer un déterminisme que n'ont pas les protocoles CSMA/CA les plus souvent utilisés (IEEE 802.11 - mode DCF, IEEE 802.15.4 - mode non-beacon). La technique de priorité hybride pour l'ordonnement des messages a été également testée avec ces protocoles sans collision. Les mêmes types d'analyse effectués avec le réseau CAN ont été également mis en œuvre sur un réseau sans fil basé sur le protocole MAC CSMA sans collision.

Ce mémoire de thèse comprend cinq chapitres :

- le premier chapitre présente un état de l'art sur le domaine des systèmes commandés en réseau (« Networked Control Systems ») et les éléments fondamentaux de la couche MAC des réseaux CAN, WiFi et ZigBee qui seront utilisés dans le cadre du travail de thèse ;
- le deuxième chapitre présente la notion de priorité hybride (qui caractérise une relation  $QdC \rightarrow QdS$ ) pour l'ordonnement des messages ainsi que les différentes modalités de sa mise en œuvre ; cette étude est faite dans le cadre de la couche MAC du réseau CAN ;
- le troisième chapitre présente les résultats des mises en œuvre de la relation  $QdS \rightarrow QdC$  et de la relation  $QdS \rightleftharpoons QdC$  sur le réseau CAN ;
- le quatrième chapitre est relatif à une proposition de protocoles MAC CSMA sans collision pour les réseaux locaux sans fil ;
- le cinquième chapitre présente l'utilisation, dans les réseaux sans fil, de protocoles MAC CSMA sans collision pour l'implantation d'applications de contrôle-commande (mise en œuvre des relations  $QdS \rightarrow QdC$ ,  $QdC \rightarrow QdS$  et  $QdS \rightleftharpoons QdC$ ).



# Sur les systèmes commandés en réseau

---

## Sommaire

---

|            |  |           |
|------------|--|-----------|
| <b>1.1</b> | <b>Introduction</b>  | <b>6</b>  |
| <b>1.2</b> | <b>De la commande continue à la commande à travers les réseaux</b>         | <b>7</b>  |
| 1.2.1      | Une chronologie  | 7         |
| 1.2.2      | Sur la Qualité de Contrôle (QdC)   | 9         |
| 1.2.3      | Sur les retards  | 10        |
| <b>1.3</b> | <b>Nécessité d'une approche de co-conception</b>                           | <b>13</b> |
| 1.3.1      | Ajustement de la période d'échantillonnage                                 | 16        |
| 1.3.2      | Mise en œuvre de la relation QdC→QdS                                       | 19        |
| <b>1.4</b> | <b>Sur l'ordonnement de messages (couche MAC)</b>                          | <b>21</b> |
| 1.4.1      | Réseau filaire : CAN   | 21        |
| 1.4.2      | Réseaux non filaires   | 23        |
| <b>1.5</b> | <b>Notre travail dans le contexte des recherches actuelles sur les SCR</b> | <b>27</b> |
| 1.5.1      | Co-conception  | 27        |
| 1.5.2      | Réseaux sans fil et protocoles MAC à priorités et sans collision           | 28        |
| 1.5.3      | Un outil de simulation : TrueTime  | 29        |

---

## 1.1 Introduction

Les systèmes commandés en réseau, notés SCR (Networked Control Systems) sont des systèmes où l'ensemble des composants actionneurs, capteurs, contrôleurs est spatialement distribué et les informations (référence d'entrée, sortie du procédé, signal de commande, ...) sont échangées à travers un réseau [Richard 2007], [Zhang 2001]; ce réseau est, en général, partagé par d'autres applications.

Les activités de recherche dans le domaine des SCR se partagent en deux thèmes

- commande des réseaux (control of networks) : l'objectif est que le réseau fournisse un certain niveau de performance (Qualité de Service, QoS) aux applications par une utilisation optimale des ressources du réseau. Les méthodes de modélisation, analyse, commande et optimisation de l'Automatique sont appliquées à la résolution de la congestion dans les réseaux tout en maximisant l'utilisation des ressources. Les problèmes réseaux de type ordonnancement, routage, contrôle de flux, allocation de ressources, ... font partie de ce champ de recherche. Des éléments bibliographiques de ce domaine peuvent être trouvés dans [Zampieri 2008]. La thèse [Ariba 2009] se situe dans ce thème ;
- commande à travers des réseaux (control over networks) : ce thème s'intéresse à la mise en œuvre d'applications de contrôle-commande à travers des réseaux et donc à l'adaptation de l'application aux performances du réseau (QoS). Le problème principal à résoudre est d'atteindre un objectif de commande (Qualité de Contrôle, QoC) malgré les insuffisances de la communication entre les différentes tâches distribuées. Les perturbations sont liées au partage des ressources de calcul (calculateurs) et des ressources de communication (réseau), entre plusieurs applications, qui induisent des retards aléatoires ou des pertes. Les théories conventionnelles de la commande, avec des hypothèses idéales (périodes fixes), ne prenant pas en compte les retards doivent être adaptées à ce nouveau contexte.

Ces deux approches sont complémentaires, la problématique étant de contrôler ou d'adapter le système de communication (QoS) pour garantir des performances (QoC) aux applications et d'adapter la demande des applications et donc la QoC à ce que le réseau peut fournir (QoS). Notre travail se situe fondamentalement dans le deuxième thème et porte sur la commande à travers les réseaux.

Ce chapitre commence par un historique de la commande (de la commande continue à la commande à travers des réseaux) et introduit les problématiques des SCR : QoC, retards (origines et compensation). Les performances des applications sont conditionnées par la QoS fournie par le système informatique distribué ; cette QoS dépend, en particulier, de l'ordonnancement des tâches (calculateur) et des messages (réseau) et il est nécessaire de concevoir des méthodes de co-conception contrôle et ordonnancement : une section est consacrée à cette nécessité d'une approche de co-conception. Notre travail étant plus particulièrement focalisé sur l'ordonnancement des messages, nous présentons ensuite des réseaux filaires (CAN) et non filaires qui nous servent de référence et nous faisons une revue des mécanismes MAC pour la QoS des réseaux (sans fil) orientés temps réel. Enfin nous situons notre travail dans le contexte des recherches actuelles sur les SCR.

## 1.2 De la commande continue à la commande à travers les réseaux

### 1.2.1 Une chronologie

Dans [Baillieul 2007] les auteurs proposent une chronologie de l'évolution de la commande, c.-à.d. depuis la commande continue jusqu'à la commande échantillonnée, puis de cette dernière aux systèmes commandés en réseau. Les dates retenues pour ces trois grandes périodes sont :

- 1932 : naissance de l'automatique classique, Nyquist fixe des principes s'appliquant à toute boucle de commande. Ces travaux succèdent à des travaux sur la régulation tout au long du XIX<sup>e</sup> siècle (Maxwell, Huygens, ...). Des années 1930 à 1950, les travaux de Bode, Nichols, Evans établissent les méthodes d'analyse et de synthèse des systèmes continus bouclés ;
- après 1950 : l'intérêt de la communauté automatique se porte sur les systèmes à commande échantillonnée (notion de temps discret) et par la suite (1970) à leur mise en œuvre à partir de calculateurs numériques. Le passage du temps continu au temps discret induit des changements de modèles et de méthodes de conception. Un précurseur en France a été Y. Sevely [Sevely 1973]. Ces méthodes sont maintenant bien établies et on peut citer, parmi d'autres, des livres de référence sur ce sujet [Åström 1997], [Franklin 2002]. Parallèlement à partir des années 1960 les travaux de Kalman donnent naissance à l'automatique moderne ;
- 1983 : projet de réseau pour l'automobile de la société Bosch GmbH qui aboutit en 1986 au protocole CAN (Controller Area Network), et en 1987 à des composants CAN. Depuis les progrès réalisés sur les calculateurs et les réseaux industriels (filaires et non filaires) font que les performances en bande passante, garanties de Qualité de Service (QdS) ont augmenté, que les coûts de calcul et de communication ont baissé. Ce faisant, les SCR ont pénétré tous les secteurs industriels : automobile, robotique, avionique, domotique, les grands systèmes de production, de distribution et de gestion de l'énergie, de l'eau, du gaz et les transports.

La figure 1.1 illustre cette évolution de la commande continue à la commande à travers les réseaux et les problématiques sous-jacentes :

- sous-schéma (1) : la notion de retour (« feedback ») ; théorie de la commande et nécessité d'une Qualité de Contrôle (QdC) pour les systèmes continus, basée sur la notion de système bouclé ;
- sous-schéma (2) : la boucle fermée à travers un calculateur (dédié) ; théorie de la commande échantillonnée et problématique de QdC ;
- sous-schéma (3) : plusieurs systèmes de commande à travers un calculateur partagé ; partage des ressources de calcul et problématique d'ordonnancement des tâches de commande : la QdS de l'ordonnancement (retards, gigue) influençant la QdC ;
- sous-schéma (4) : systèmes commandés en réseau (SCR) donc problématique de partage de réseau. Avant d'arriver à la configuration proposée on peut préciser que des systèmes distribués géographiquement existaient depuis plusieurs années, voire décades, pour la conduite de procédés chimiques, des raffineries, des réseaux d'énergie électrique, ... Dans de tels

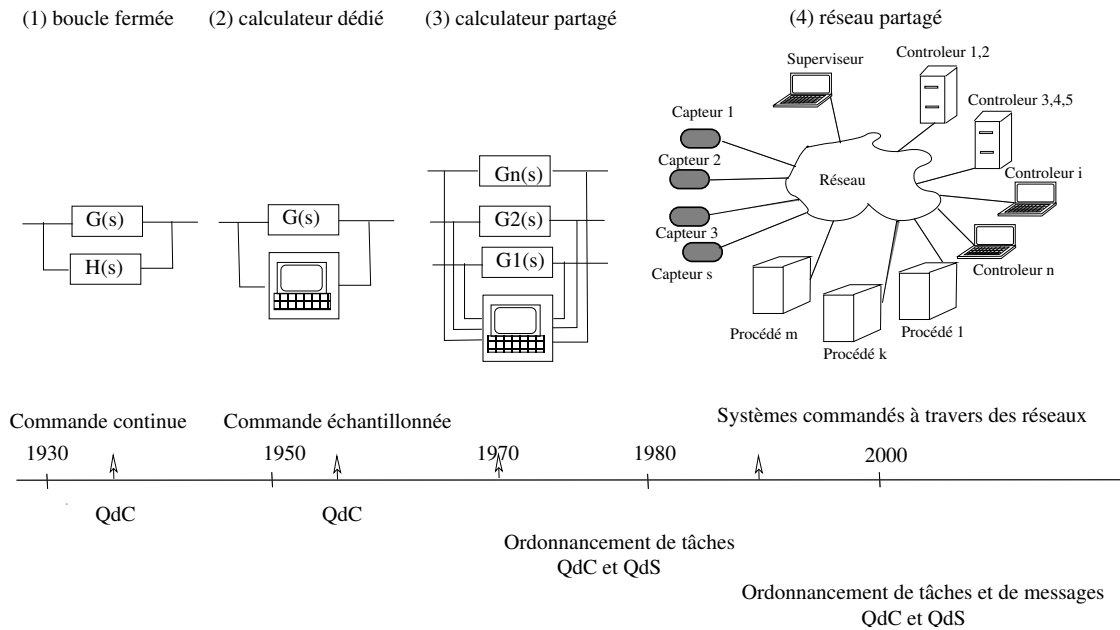


FIGURE 1.1 – Chronologie.

Les systèmes les composants étaient connectés point à point et le système était conçu pour remonter l'information des capteurs vers un centre de décisions où les lois de commande étaient élaborées avant d'être communiquées aux actionneurs. Comparés avec ces systèmes de commande point à point, les avantages des systèmes commandés en réseau sont un câblage plus léger ou plus du tout de câblage dans le cas de réseaux sans fil, une utilisation répartie de l'informatique où les informations sont transmises à travers le réseau, une plus grande flexibilité (possibilité d'ajouter des unités de calcul ou des composants), un moindre coût d'utilisation. Les réseaux autorisent la réalisation de systèmes distribués complexes à la fois horizontalement (coordination entre pairs : capteurs, contrôleurs, actionneurs) et verticalement (niveau supervision, reconfiguration), ils sont donc un élément important des systèmes manufacturiers.

Les systèmes commandés en réseau qui connaissent un grand développement dans le contexte technologique actuel, sont des systèmes dont la caractéristique fondamentale est le partage de ressources (ressource de calcul, ressource de transmission) entre applications. Un partage de ressources par plusieurs applications pose deux problèmes fondamentaux : d'une part, la fréquence de la demande du partage par les applications (et donc la fréquence d'échantillonnage imposée) et, d'autre part la gestion du partage du service entre les différentes applications ; le partage du service dépend à la fois du partage de la ressource calcul (problème d'ordonnancement des tâches) et du partage de la ressource de transmission (problème d'ordonnancement des messages plus particulièrement). Ce partage induit évidemment des retards (constants ou aléatoires) qui caractérisent la Qualité de Service (QdS) fournie aux applications. La fréquence d'échantillonnage permise et le retard du service fourni conditionne fortement la Qualité de Contrôle (QdC).

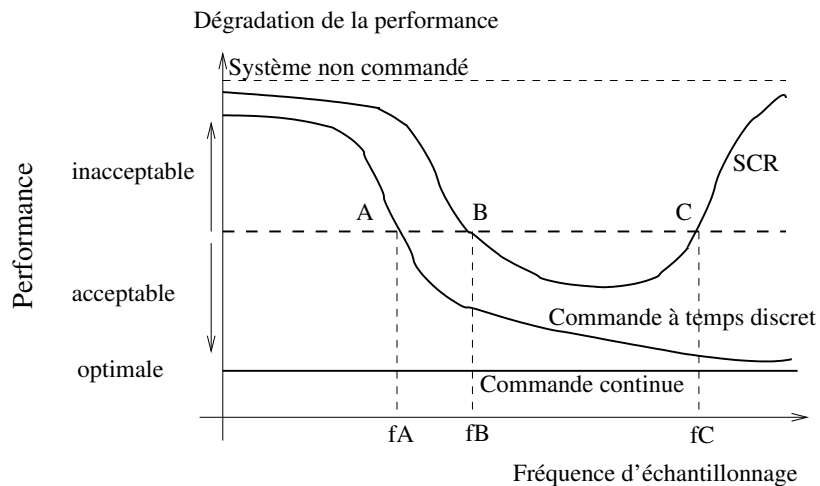


FIGURE 1.2 – Comparaison des performances, des commandes continues, échantillonnées et en réseau, en fonction de la fréquence d'échantillonnage.

### 1.2.2 Sur la Qualité de Contrôle (QdC)

La Qualité de Contrôle peut être jugée à partir du dépassement dans le cas d'une réponse à un échelon, de la précision, de la marge de phase, du temps de montée, du temps de réponse. Un critère de QdC pourra être une combinaison de ces critères.

Considérons une loi de commande établie en temps continu et discrétisée ou établie directement à temps discret ; la fréquence d'échantillonnage a une influence importante sur la Qualité de Contrôle. Une limite inférieure de cette fréquence est donnée par le théorème de Shannon, mais des règles empiriques permettent la détermination de la période d'échantillonnage  $h$ , par exemple dans notre travail nous utiliserons une règle donnant la période d'échantillonnage  $h$  en fonction de la pulsation propre du système bouclé  $\omega_n$  [Åström 1997] :

$$0.1 \leq \omega_n h \leq 0.6 \quad (1.1)$$

Une idée généralement admise est que plus petite est la période d'échantillonnage et plus grande est la Qualité de Contrôle. Dans [Lian 2002], [Moyné 2007] les auteurs montrent comment les performances varient en fonction de la période d'échantillonnage en comparant les performances obtenues à temps continu, à temps discret et à travers un réseau.

La figure 1.2 qu'ils proposent, présente la dégradation des performances en fonction de la fréquence d'échantillonnage. La performance obtenue pour le système à temps continu (constante car indépendante de la fréquence d'échantillonnage) sert de référence, pour une loi de commande donnée.

Les performances du système à temps discret approchent les performances du système à temps continu quand la fréquence d'échantillonnage augmente. Ceci suppose qu'il n'y a pas d'autres variations que la période et suppose donc un système de commande dédié (soit matériel, soit logiciel mais dans ce cas la tâche, qui réalise échantillonnage, calcul de la commande et actuation, ne partage pas le processeur avec d'autres tâches ; dans le cas de ressources limitées et partagées on retrouverait la problématique des SCR). La dégradation de performance (point A) dépend de



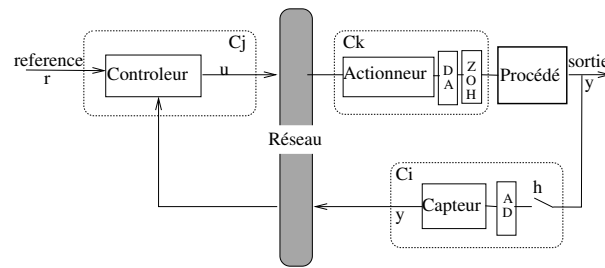


FIGURE 1.3 – Une application de contrôle-commande implantée sur un réseau.

la bande passante du procédé et traduit une période  $h$  trop grande qui ne remplit pas la contrainte (1.1) (par exemple  $\omega_n h \gg 0.6$ ).

Pour la commande à travers un réseau, la performance augmente avec la fréquence d'échantillonnage mais la performance devient acceptable (point  $B$ ) pour une plus petite période que dans le cas d'une commande à temps discret ( $fB > fA$ ) car il faut tenir compte des retards introduits par le réseau et par l'ordonnement des tâches dans les divers composants distribués. La performance est acceptable sur un intervalle  $[B, C]$  puis se dégrade, à cause du réseau, dans le cas d'une diminution de période. En effet, si la fréquence augmente le trafic réseau augmente ainsi que la contention pour l'accès au réseau et donc les retards et éventuellement les pertes. Entre les deux extrêmes il existe une zone où la période d'échantillonnage est optimale par rapport à la commande et aux capacités du réseau.

Le réseau et les unités de calcul sont des ressources partagées et limitées qui ne peuvent pas supporter des fréquences d'utilisation élevées et le résultat obtenu marque le compromis à trouver entre la demande de QdC et la variabilité de la QdS.

### 1.2.3 Sur les retards

#### 1.2.3.1 Origines des retards

Considérons un système informatique distribué composé de plusieurs calculateurs connectés à travers un réseau. Les calculateurs  $C_i$ ,  $C_j$ ,  $C_k$  sont concernés par une application de contrôle-commande : la sortie du système est captée dans le site  $C_i$ , le contrôleur est implanté dans le site  $C_j$  où la commande est calculée et envoyée au site  $C_k$  qui pilote l'actionneur. La figure 1.3 représente l'implantation d'une boucle de commande sur un réseau où dans les calculateurs  $C_i$ ,  $C_j$  et  $C_k$  sont respectivement implantées les tâches Capteur, Contrôleur et Actionneur.

Le système informatique distribué assure le transfert qui débute à l'instant où l'échantillon numérique est produit, dans le site  $C_i$ , par le convertisseur analogique numérique (noté AD) pour la tâche Capteur, et finit à l'instant où cet échantillon numérique est délivré, dans le site  $C_k$ , par la tâche Actionneur au convertisseur numérique analogique (noté DA). Ce service de transfert est caractérisé par un délai  $\tau_D$  et une probabilité de perte  $P_p$ .

Si nous considérons une architecture minimale avec trois couches (architecture d'un réseau local) le délai  $\tau_D$  comprend trois composantes : la composante « couche application » qui dépend de l'ordonnement des tâches ; la composante « couche liaison » qui dépend du type du protocole de transfert des messages et de l'ordonnement des trames séries compte tenu de

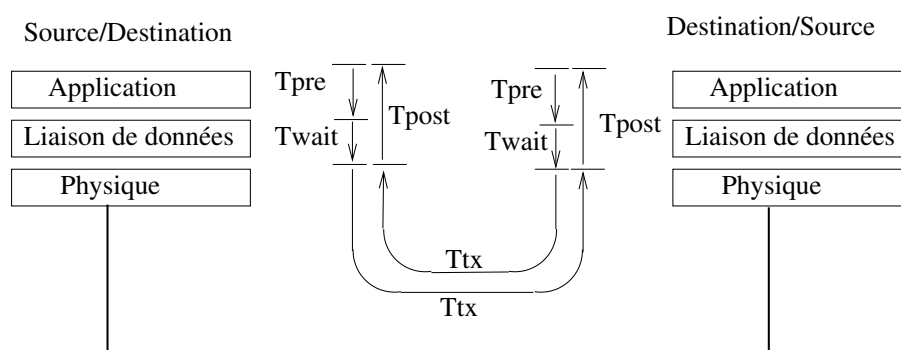


FIGURE 1.4 – Communication d'un nœud source à un nœud destination.

l'environnement externe ; la composante « couche physique » qui dépend du débit et du temps de propagation sur le bus. Ce délai  $\tau_D$  peut être caractérisé par une gigue *-jitter-*, c'est-à-dire l'écart entre la borne supérieure et la borne inférieure du délai.

Ces différentes composantes sont visualisées sur la figure 1.4, extraite de [Lian 2002], où on considère seulement une communication entre un nœud source et un nœud destination. Au niveau de la couche application, le temps noté  $T_{pre}$  tient compte du retard variable lié à l'ordonnancement de la tâche producteur, et des retards constants liés à l'encapsulation et au codage des données ; puis si le réseau est occupé le temps  $T_{wait}$  représente l'attente de sa libération. Ce temps d'attente est essentiellement lié au protocole MAC. Puis le message est émis. La durée de l'émission  $T_{tx}$  dépend de la longueur de la trame et de la vitesse de transmission, auquel s'ajoute le temps de propagation ( $T_{frame} + \tau_{PT}$ ). A la réception le message est décodé, transmis à la couche application, et traité par la tâche consommateur ; un nouveau retard variable est introduit ici au niveau du partage de la ressource de calcul, ce retard est noté  $T_{post}$ . Finalement la durée de communication d'un nœud source à un nœud destination est exprimée par  $\tau_{sd}$  (s : source, d : destination) :

$$\tau_{sd} = T_{pre} + T_{wait} + T_{tx} + T_{post} \quad \text{avec} \quad T_{tx} = T_{frame} + \tau_{PT}$$

### 1.2.3.2 Effets et compensation des retards de calcul et de communication

De nombreux auteurs ont montré l'influence des retards sur la dégradation des performances du système commandé. Considérons la figure 1.3. Nous nous plaçons dans un contexte généralement admis et fixé par [Nilsson 1998] (que nous rappelons ici) et nous utilisons des notations majoritairement utilisées :

- retard entre capteur et contrôleur, noté  $\tau_{cc}^k$ . Ces retards sont constitués des différentes composantes vues précédemment (expression de  $\tau_{sd}$ ) ;
- temps de calcul de la loi de commande au niveau du contrôleur, noté  $\tau_c^k$  ;
- retard entre contrôleur et actionneur, noté  $\tau_{ca}^k$ .

l'exposant  $k$  indique la possible variation dans le temps. Le retard pendant la période  $k$  est :

$$\tau^k = \tau_{cc}^k + \tau_c^k + \tau_{ca}^k$$

Dans une perspective de commande échantillonnée à période constante  $h$  on considère un échantillonnage périodique donc un capteur guidé par le temps. Pour que le retard entre l'échantillon-

nage (capteur) et l'application de la commande (actionneur) soit le plus court possible, le contrôleur et l'actionneur sont guidés par les événements ; ils démarrent leur activité dès qu'ils reçoivent une information du capteur (au niveau du contrôleur) ou du contrôleur (au niveau de l'actionneur). Une hypothèse supplémentaire est que le retard  $\tau$  est toujours inférieur à une période :  $\tau < h$ . Les retards étant variant dans le temps, les systèmes contrôlés à travers des réseaux perdent la propriété d'invariance temporelle.

Nilsson [Nilsson 1998] modélise les retards introduits par le réseau (principalement niveau MAC) et propose des techniques de compensation de ces retards dans le cas de retards constants, aléatoires et indépendants, aléatoires avec une distribution de probabilité pilotée par une chaîne de Markov.

Une première technique de compensation est de rendre le système invariant dans le temps [Luck 1990] en introduisant des buffers temporisés, le buffer résultant étant plus long que le retard dans le pire cas, le temps de transfert est vu comme constant. Un observateur estime l'état du procédé contrôlé et un prédictor calcule la commande. Le retard échantillonnage-actuation devient constant et l'actuation est périodique, cependant ce retard basé sur le pire cas est plus grand que celui qui serait nécessaire en temps réel, et conduit donc à une dégradation des performances. Ce retard peut être ici supérieur à une période d'échantillonnage.

Dans le cas de retards aléatoires l'hypothèse de retards indépendants permet d'obtenir un modèle simple et l'auteur considère des distributions de probabilité différentes pour le retard capteur-contrôleur  $\tau_{cc}$  et le retard contrôleur-actionneur  $\tau_{ca}$ . Comme noté par l'auteur, cette hypothèse est simplificatrice, car dans un système de communication, la corrélation des retards avec les précédents existe (cas d'un réseau chargé et problèmes de contention, dans l'accès au médium). Dans une phase d'analyse l'auteur montre qu'il est des exemples (cas d'un pendule inverse) où le système peut être stable pour des retards constants (ici  $h/2$ ) et devenir instable quand les retards sont variables (uniformément répartis entre 0 et  $h$ ).

Nilsson propose donc une méthode de commande optimale stochastique pour prendre en compte les retards aléatoires et il résout un problème Linéaire Quadratique Gaussien (LQG) dans le cas de retards aléatoires et indépendants et dans le cas de retards modélisés à partir de chaînes de Markov. Les résultats obtenus montrent de meilleures performances pour cette commande optimale que pour d'autres méthodes (contrôleur calculé pour un retard moyen, compensation en rendant le retard déterministe).

Ces méthodes et d'autres méthodes de compensation des retards sont aussi présentées dans [Tipsuwan 2003]. Dans ce papier, une analyse des retards dans la boucle de commande est proposée à partir d'un système du second ordre (modèle d'un moteur à courant continu) corrigé par un contrôleur proportionnel intégral : le dépassement et le temps de réponse augmentent avec le retard ; de plus les auteurs montrent, à partir d'une étude dans le lieu des racines, l'effet de retards (constants) sur la marge de stabilité et la réduction de celle-ci lorsque les retards augmentent.

Un autre travail sur les relations entre la QdS (exprimée par un retard) et la QdC est présenté dans [Marti 2004]. Les auteurs s'intéressent à l'effet des retards sur les performances d'une application de contrôle commande (pendule inverse) et considèrent dans un premier temps des retards constants puis dans un deuxième temps des retards variables.

Les effets des retards se traduisent par :

- une réponse retardée : le retard échantillonneur-actionneur retarde l'application de la commande au procédé ;
- une erreur augmentée : plus grand est le retard et plus grande est l'erreur en boucle fermée.

Après analyse des retards variables les auteurs proposent une technique de commande qui s'adapte à cette variabilité en supposant que le retard  $\tau_{cc} + \tau_{ca}$  est connu à chaque cycle. Le problème est formulé dans l'espace d'état et le retour d'état suppose un retard constant mais ici il varie à chaque exécution de la boucle de commande ; l'algorithme de commande proposé suppose que la période et le retard soient connus à chaque itération. La période est bien constante (échantillonnage guidé par le temps) mais le retard  $\tau_{cc} + \tau_{ca}$  ne peut pas être connu du contrôleur car  $\tau_{ca}$  est inconnu. En vue de dépasser ce problème un changement d'architecture est proposé et le contrôleur et l'actionneur sont regroupés sur un même site. Dans ce cas les délais sont variables mais connus. La stratégie de commande doit prendre en compte cette variation et les auteurs supposent que cette variation est limitée et que le retard  $\tau$  prend des valeurs discrètes sur un intervalle, valeurs pouvant être déterminées hors-ligne en fonction des caractéristiques du réseau. Ainsi, à chaque exécution  $i$ , le retard  $\tau_i$  prend une valeur parmi un nombre fini de valeurs entre 0 et le retard dans le pire cas. Le modèle à temps discret du système n'est plus invariant dans le temps, mais le retour d'état à chaque itération est obtenu pour le retard spécifique  $\tau_i$ . Dans le cas de la commande d'un pendule inverse une comparaison entre une commande à retard constant (actuation périodique) et un correcteur adaptatif aux retards (actuation asynchrone) est faite à partir d'un critère de performance basé sur l'intégrale de l'erreur. La QdC obtenue pour le correcteur proposé est meilleure que dans le cas d'un retard constant. Une analyse de stabilité permet aux auteurs de s'assurer de la stabilité du système en boucle fermée.

### 1.3 Nécessité d'une approche de co-conception

A son origine dans les années 50 la théorie de la commande à temps discret ne s'est pas intéressée au problème des ressources partagées et limitées du système de commande. Ainsi il est communément supposé que le contrôleur pilote un seul procédé sur un ordinateur dédié. Depuis les années 70 le milieu académique s'est intéressé à l'ordonnancement temps réel, mais avec peu de travaux sur les tâches de contrôle. Ordonnancer c'est choisir à un instant donné parmi un ensemble de tâches concurrentes la tâche qui sera exécutée et garantir pour cette tâche que la contrainte d'échéance soit remplie.

Une tâche périodique  $T_i$  est modélisée par sa période  $P_i$ , le temps d'exécution dans le pire cas (WCET)  $C_i$ , une échéance relative  $D_i$  et une priorité  $\pi_i$ . Le problème de l'ordonnancement d'un ensemble de tâches indépendantes sur un processeur est d'assurer que l'échéance de chaque instance est satisfaite, c'est-à-dire que la tâche est exécutée avant l'échéance. Ces garanties sont des garanties temps réel dures.

Ce modèle peut être utilisé pour des tâches de contrôle-commande et leur exécution sur un processeur, l'échéance étant déduite de la période d'échantillonnage. La conception est alors faite en deux étapes, conception de la tâche de commande et ordonnancement temps réel. Ces deux étapes sont séparées et la première est de la compétence de l'automaticien (qui considère le système

temps réel comme une plate-forme où le contrôleur sera implémenté trivialement), la deuxième relève de l'informatique (un contrôleur est un bout de code caractérisé par  $T_i(C_i, D_i, P_i)$ ). Si les politiques d'ordonnancement utilisées garantissent les échéances, la QdC est aussi généralement garantie (aux perturbations près introduites par la gigue). Cependant la garantie de l'échéance pour toutes les tâches dont les tâches de contrôle (considérées comme temps-réel dures), demande généralement des ressources très importantes conduisant à des sur-réservations et à une solution non optimale. Dans [Gupta 2010] les auteurs illustrent la différence entre des tâches temps-réel dures et souples à partir d'une fonction d'utilité présentée figure 1.5 (où l'utilité est notée  $U$ ). Pour des tâches temps-réel dur cette fonction tend vers zéro dès que l'échéance est dépassée, et la tâche doit donc être exécutée avant cette échéance ; pour des tâches temps réel souples la fonction d'utilité décroît graduellement.

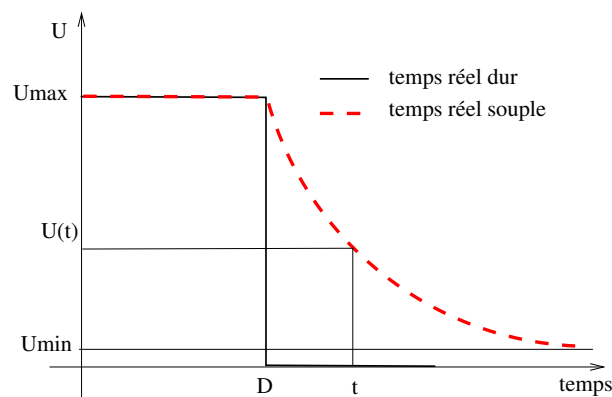


FIGURE 1.5 – Fonction d'utilité pour des applications temps réel dures et souples.

Il n'est pas nécessaire de considérer une tâche de contrôle-commande comme devant respecter strictement une échéance. En fait une boucle de commande est conçue pour obtenir une marge de stabilité et de la robustesse par rapport aux incertitudes des paramètres du procédé, et aussi par rapport aux incertitudes temporelles [Song 2009]. Les boucles de commande sont capables de supporter des variations de période d'échantillonnage et une variation de 10% de la période nominale peut-être en règle générale acceptée sans être compensée. De même des pertes de données occasionnelles peuvent se produire sans perte de stabilité.

Le développement isolé des correcteurs et de l'ordonnancement temps réel introduisant du surdimensionnement et conduisant à des solutions non optimales il est nécessaire de concevoir conjointement le correcteur et l'ordonnancement temps-réel. Le problème de co-conception commande et ordonnancement est informellement posé dans [Cervin 2003] : « Given a set of processes to be controlled and a computer with limited computational ressources, design a set of controllers and schedule them as real time tasks such that the overall control performance is optimized ». L'objectif de la thèse [Cervin 2003] est de réduire le fossé entre les deux domaines.

La problématique est la même lorsque la commande est implantée à travers un réseau et on retrouve deux grandes approches privilégiant soit la conception de contrôleurs, soit le réseau :

- on définit le contrôleur en considérant seulement l'application, c'est-à-dire sans considérer le problème de l'implantation sur un système temps réel distribué. La stabilité du système contrôlé dépend de la conception du système de communication où un effort doit être fait

pour réduire les retards (différentiation de types de données, chacun avec sa priorité et son assignation des ressources réseau en fonction des priorités) et les pertes. Dans ce contexte il est nécessaire d'évaluer la QdS (retards, pertes) pour caractériser la robustesse de l'application par rapport à la QdS.

Ce type de recherche demande de bien identifier tous les mécanismes de QdS afin d'être capable de faire une analyse exhaustive de son impact sur les performances des applications ;

- l'alternative est d'ignorer le réseau de communication ; le réseau est vu comme un système à retard. La conception du contrôleur doit compenser ce retard. La synthèse de la loi de commande peut être améliorée en considérant le type de retard (constant, variant dans le temps) et sa distribution de probabilité.

Cette deuxième approche est celle de la communauté automatique, la première approche est celle de la communauté communication/réseau.

Une approche de co-conception doit permettre de définir le contrôleur en intégrant le comportement du système informatique temps réel distribué. Cette intégration peut être faite soit hors ligne soit en ligne.

- si on a la connaissance de la QdS, on calcule le contrôleur, hors ligne et, si les retards sont bornés et connus, une loi de commande peut être calculée pour le pire cas. Par exemple, dans [Vatanski 2006] le retard de bout en bout est estimé à partir du calcul réseau ; ces retards sont traduits en incertitudes et conduisent à une synthèse de commande robuste. Cependant ces méthodes basées sur le pire cas ne sont pas optimales ;
- en ligne, en mesurant la QdS par les moyens d'un service de métrologie et en agissant sur les paramètres de QdS. Cette technique adaptative qui utilise les performances QdS mesurées est de plus en plus utilisée et ces méthodes de co-conception conduisent à des contrôleurs et à des consommations de ressources plus optimaux.

Un projet ANR, le projet Safe-NECS pour Safe Networked Control Systems s'est déroulé pendant ma thèse et s'est intéressé à développer des techniques de co-conception contrôle et ordonnancement de tâches, contrôle et ordonnancement de messages et au problème du diagnostic distribué. Ce projet s'intéressait aux différentes boucles de régulation d'un SCR :

- boucle(s) du système à contrôler basée(s) sur une informatique distribuée où les ressources partagées sont les calculateurs et le réseau ;
- boucle d'ordonnancement régulé. Il s'agit de résoudre le problème du partage de ressources (ordonnancement de tâches sur les calculateurs et ordonnancement de messages pour l'accès au réseau) en garantissant un niveau de performances pour les applications. Cet ordonnancement régulé des tâches et des messages est piloté par les performances des applications (QdC) ;
- boucle de supervision. Les mécanismes de supervision et de reconfiguration sont également distribués et communiquent à travers le réseau.

Les résultats de ce projet ont conduit à la publication de l'ouvrage [Aubrun 2010], très riche en références dans le domaine de la co-conception et sur le domaine du diagnostic et du diagnostic dans les SCR. Notre participation a porté sur la mise en œuvre de priorités dynamiques fonction des performances de l'application et donc sur l'allocation de ressources réseau à partir de priorités allouées en fonction de la QdC.

Des approches nouvelles de conception sont à définir. Les systèmes à imaginer doivent être adaptatifs [Song 2009]. Non seulement le réseau doit être capable d'allouer les ressources nécessaires à l'application quand c'est nécessaire mais l'application doit accepter une dégradation de la QoS, qui provoque une dégradation (acceptable) de la QdC. De façon générale deux voies pour obtenir une conception d'un SCR efficace peuvent être distinguées :

- Adaptation du contrôleur aux contraintes de mise en œuvre ou encore « implementation aware control law design » [Aubrun 2010]. L'idée est d'adapter en ligne les paramètres de la loi de commande en ajustant, par exemple, la période d'échantillonnage ;
- Mise en œuvre de la relation QdC→QoS ou encore « control aware QoS adaptation » [Aubrun 2010]. L'idée est de réallouer les ressources d'implémentation en ligne pour maintenir ou augmenter la QoS demandée par l'application.

### 1.3.1 Ajustement de la période d'échantillonnage

Une méthode généralement utilisée pour résoudre le problème de surcharge dans le cas de systèmes périodiques est de réduire la charge en augmentant la période d'échantillonnage. Cette variation de période influe sur les paramètres de la loi de commande qui doit être recalculée dynamiquement pour cette nouvelle fréquence.

Nous allons présenter différentes techniques représentatives de cet ajustement périodique en distinguant comme dans [Song 2009] deux types de solutions : ajustement explicite de période, et ajustement implicite de période basé sur la sélection de tâches/messages en accord avec un modèle dit  $m, k$ -firm.

#### 1.3.1.1 Ajustement explicite de la période

##### Variation régulée de la période : Ordonnancement régulé ou Feedback scheduling

Un premier travail fondateur qui intègre la conception du contrôleur et l'ordonnancement de tâches de contrôle est proposé dans [Seto 1996]. Dans leur travail les auteurs posent le problème du choix de la période d'échantillonnage d'un ensemble de tâches de contrôle sous contrainte d'ordonnancement. La performance d'une tâche est établie à partir d'un critère de coût, fonction de la période d'échantillonnage, et un problème d'optimisation est résolu afin de déterminer la période optimale. Le critère de QdC est un critère quadratique et le problème s'écrit :

$$\min_{h_1 \dots h_n} \sum_{i=1}^n w_i J_i(h_i)$$

sous contrainte d'ordonnancement

$$\sum \frac{C_i}{h_i} < U_{sp}$$

$C_i$  est le temps d'exécution de la tâche  $i$ ,  $w_i$  un coefficient de pondération,  $U_{sp}$  est le niveau d'utilisation CPU demandé garantissant l'ordonnancement des tâches (dépendant de la politique d'ordonnancement). La résolution de ce problème d'optimisation non linéaire est faite hors ligne, vu sa complexité.

Un ordonnanceur qui ajuste en ligne ses paramètres pour optimiser la QdC ou la QoS peut être vu comme un système bouclé. L'idée d'ordonnancement en boucle fermée de tâches temps-réel

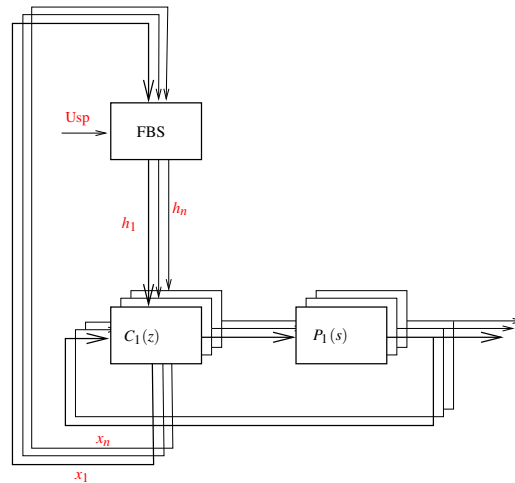


FIGURE 1.6 – L'ordonnanceur FBS (feedback scheduler) et les boucles de commande multiples.

a fait l'objet de peu de travaux de la communauté temps-réel à l'exception de [Stankovic 1999]. Le cadre est celui de l'ordonnement de tâches temps-réel (et non de tâches de contrôle) et la problématique est ici de minimiser le taux d'échéances dépassées (miss ratio  $M$ ) tout en régulant le taux d'utilisation du processeur  $U$ . L'algorithme proposé FC-EDF (Feedback Control- Earliest Deadline First) est réalisé à partir d'un correcteur Proportionnel-Intégral-Dérivé (PID) sur  $M$  et  $U$  qui en contrôlant l'admission des tâches garantit un niveau de QoS pour les tâches admises.

Dans le domaine des tâches de contrôle, l'ordonnement régulé (ou feedback scheduling) s'appuie sur le fait que les tâches pilotent des procédés physiques et que les performances de l'application peuvent être utilisées par l'ordonnanceur. L'objectif est d'optimiser la performance globale de toutes les tâches de contrôle sous contrainte d'ordonnement. Un ordonnement régulé pour une commande LQ est proposé par [Eker 2000] et [Cervin 2002]. Pour résoudre en ligne le problème d'optimisation la fonction coût est approchée par une fonction quadratique de  $h$  ( $J_i = \alpha_i + \beta_i h^2$ ) et une approximation linéaire ( $J_i = \alpha_i + \gamma_i h$ ). Ces travaux sont prolongés par [Henriksson 2005] où les auteurs utilisent des contrôleurs LQ et expriment le critère de coût en fonction de la période d'échantillonnage et de l'état des procédés. La figure 1.6 représente les boucles d'asservissement des différents procédés  $P_1, \dots, P_n$  et l'ordonnement régulé en fonction

- de l'état des procédés commandés ( $x_1, \dots, x_n$ )
- du taux d'utilisation des ressources.

Pour maintenir le taux d'utilisation de CPU au taux de référence  $U_{sp}$  l'ordonnanceur assigne des périodes  $h_1, \dots, h_n$  aux différents correcteurs  $C_1, \dots, C_n$ . Le problème d'assignation de la période est formulé ainsi :

$$\min_{h_1, \dots, h_n} \sum_{i=1}^n J_i(x_i(t), h_i, T_{fbs})$$

sous

$$\sum_{i=1}^n \frac{C_i}{h_i} \leq U_{sp}$$

Comme dans le cas des travaux déjà cités, les valeurs des périodes sont obtenues en ligne par des approximations.  $T_{fbs}$  est la période d'activation de l'ordonnanceur, sa valeur résulte d'un



compromis : trop petite elle consomme des ressources, trop grande la réaffectation des ressources suit mal la dynamique des procédés.

Ces travaux concernant l'ordonnancement régulé de tâches de contrôle et donc le calcul en ligne d'une période optimale d'échantillonnage se continuent et donnent lieu à des publications, dont certaines très récentes [Castane 2006], [Marti 2009], [Cervin 2011] (intégration des temps de calcul, du bruit agissant sur le procédé).

#### **Table de correcteurs et commutation vers ces correcteurs en ligne selon le retard**

La thèse [Marti 2002] présente une approche de conception qui prend en compte les contraintes d'ordonnancement des tâches et relâche les contraintes temporelles d'équidistance des instants d'échantillonnage et instants d'actuation traditionnellement faits. Au lieu de préciser, à l'étape de conception du contrôleur, une valeur unique pour la période et le retard entre échantillonnage et actuation, il spécifie un ensemble de valeurs à la fois pour la période et le retard  $\tau$ . Cette nouvelle approche conduit à définir hors ligne une banque de correcteurs, chacun d'eux étant conçu et adapté à une période d'échantillonnage spécifique, et à commuter entre ces correcteurs en fonction de décisions liées à la gigue de l'ordonnancement. Le système temps réel résultant est basé sur un échantillonnage irrégulier avec des retards échantillonnage-actuation variables, les correcteurs mis en œuvre garantissent un critère de QdC (par exemple garantie d'un temps de réponse). Une formulation du problème de synthèse du correcteur dans l'espace d'état permet de conclure à la stabilité du modèle.

Pour illustrer la méthode l'auteur considère un ensemble de tâches à implémenter dont une tâche de contrôle pilotant un pendule inverse (périodicité de 80 ms, durée de la tâche de 20 ms). L'algorithme d'ordonnancement mis en œuvre est l'algorithme EDF (Earliest Deadline First) et son application conduit à de la gigue sur la période d'échantillonnage et des retards échantillonneur-actionneur variables. La période  $h \in \{60, 70, 80, 90, 100\}$  ms et le retard  $\tau^k \in \{20, 30\}$  ms. La réponse du système est dans ces conditions instable pour un correcteur calculé pour une période fixe de 80 ms. Différents correcteurs seront calculés hors ligne, pour toutes les paires possibles de période et de retard avec la même contrainte de QdC (temps de réponse  $< 2$  s). La mise en œuvre en ligne de ce correcteur qui ajuste ses paramètres à chaque exécution pour prendre en compte la compensation de la gigue lié à l'ordonnancement EDF conduit à une solution stable satisfaisant la QdC.

Ce travail est prolongé par les travaux de [Buttazzo 2004] et [Buttazzo 2007] où les auteurs se donnent un critère de QdC et cherchent à réduire le nombre de correcteurs à définir hors ligne qui garantissent une QdC donnée. Un correcteur est défini pour une période nominale et la QdC se dégrade de manière significative lorsque la période augmente. Lorsque la période calculée par l'ordonnanceur n'est plus adaptée au correcteur il faut changer de correcteur. Les correcteurs sont calculés hors ligne pour des valeurs discrètes de période, c'est ainsi que pour l'exemple traité la période peut varier de 0.25 s à 0.54 s et que 4 correcteurs sont précalculés pour des périodes nominales de 0.31, 0.38, 0.46, 0.54 s. En ligne le gestionnaire de QdC doit réagir aux conditions de surcharge et sélectionner une période d'échantillonnage qui satisfait les contraintes d'ordonnancement et la QdC.

**Conception de correcteurs robustes à la variation de période d'échantillonnage** [Robert 2006, Ben Gaïd 2010]. L'application de la méthode précédente, banque de correcteurs et

commutation entre ces correcteurs, individuellement stables, en fonction des décisions d'un feedback scheduler, peut conduire à des instabilités lors des commutations. L'approche LPV (Linear Parameter Varying) est proposée pour concevoir un correcteur linéaire à temps discret avec une période d'échantillonnage variable et des garanties de QdC. Le schéma LPV permet de garantir la stabilité quels que soient les instants de variation des périodes à l'intérieur d'un intervalle pré-défini. Cependant cette approche garantissant la stabilité ne s'applique qu'aux systèmes linéaires.

### 1.3.1.2 Ajustement implicite de la période : modèle $(m, k)$ -firm

Pour tenir compte d'une surcharge les approches précédentes conduisent à une variation explicite de la période d'échantillonnage. Une sélection de tâches (de messages) éliminés en accord avec un modèle dit  $(m, k)$ -firm conduit à un ajustement implicite de la période ; la période est augmentée suite à une élimination et cet ajustement est un multiple de la période nominale.

Un système satisfaisant la contrainte du modèle  $(m, k)$ -firm suppose la satisfaction d'au moins  $m$  échéances parmi  $k$  ; le maintien de seulement  $m$  instances est le pire cas.  $m$  et  $k$  sont tels que  $0 < m \leq k$  ; le cas  $m = k$  est équivalent au cas idéal et correspond à une tâche temps réel dure, toutes les instances doivent satisfaire l'échéance. Dans le cas général plus de  $m$  échéances sont remplies durant une fenêtre de  $k$  instances consécutives et le système n'est pas dans les conditions du pire cas. Une tâche peut-être vue comme une tâche périodique à exécuter ou un message à transmettre et cette méthode peut aussi s'appliquer à l'ordonnement de messages.

Si un système de commande peut accepter des dégradations de performances à partir de la situation  $(k, k)$ -firm (idéal) jusqu'à  $(m, k)$ -firm (critique) c'est qu'il a été conçu pour donner des performances acceptables quand  $k - m$  instances sont éliminées.  $k$  correspond au nombre maximum d'instances consécutives non exécutées que le système accepte avant instabilité ;  $m$  permet d'obtenir la QdC souhaitée.

Des références à ces méthodes et à leur application dans le domaine de l'ordonnement de tâches et à l'ordonnement de messages sont données dans [Song 2009] et [Aubrun 2010].

### 1.3.2 Mise en œuvre de la relation QdC $\rightarrow$ QdS

L'objectif est ici de maintenir ou d'améliorer les performances offertes par le réseau (QdS) en fonction des demandes des applications afin de maintenir la QdC. Pour cela, il faut d'une part pouvoir agir sur les ressources réseau pour ajuster la QdS et d'autre part établir la relation entre la QdC et les besoins de QdS pour pouvoir prendre en compte les besoins venant des applications. Cette allocation de ressources est dynamique et répond à des besoins sollicités par les applications. Les avantages de cette dynamicité sont la libération des ressources quand elles sont moins nécessaires alors qu'elles pourraient être maintenues dans le cas d'attribution statique ou fonction d'une évaluation au pire cas.

Parmi les possibilités d'action sur la QdS dans le cas le mise en œuvre de la relation QdC  $\rightarrow$  QdS nous citerons

- allocation de priorités hybrides (dynamique + statique) aux messages de CAN à partir des besoins des applications, la partie dynamique traduit l'urgence de la communication, cette urgence est déduite du signal d'erreur  $e$  ou du signal de commande  $u$ , elle est donc contrôlée

par les performances de l'application. A une priorité importante doit correspondre un retard du capteur à l'actionneur le plus petit possible. Le chapitre 2 de notre thèse présente ce mécanisme et donne des références sur ce domaine pour le réseau CAN ; par la suite de notre travail nous proposons d'étendre cette technique à des réseaux sans fil ;

- allocation dynamique de bande passante pour Switched Ethernet [Diouri 2007]. Dans un commutateur une politique d'ordonnement FIFO est utilisée pour sélectionner les trames à émettre. En utilisant la classification de service il est possible de remplacer cette politique par une politique WFQ (Weighted Fair Queing). Un poids est associé à chaque file d'attente et l'ordonneur donne à chaque file une bande passante fonction de son poids. Ce partage de la bande passante est défini à partir du retard capteur-actionneur admissible défini à partir du niveau de QdC exigé ;
- allocation dynamique de priorité, dans IEEE 802.15.4/ZigBee en fonction des besoins des applications évalués en fonction de l'écart de la réponse à la consigne (erreur  $|e|$ ) [Boughanmi 2009]. Les auteurs proposent de modifier un paramètre du protocole MAC et ainsi de modifier des intervalles de tirage aléatoire de temps d'attente avant émission. Deux niveaux de priorité sont ainsi définis alors que le protocole initial (voir Sect. 1.4.2.1) place tous les messages au même niveau. Lorsque l'erreur dépasse un seuil, l'application est considérée comme critique et les messages de contrôle changent de niveau de priorité (afin que le retard soit moindre). Cette solution rejoint la problématique de priorisation des messages comme pour CAN.

Ces deux visions, mise en œuvre de la relation  $QdC \rightarrow QdS$  et de la relation  $QdS \rightarrow QdC$  ne sont pas exclusives et comme déjà évoqué lorsque les mécanismes de QdS ne permettent pas de garantir la QdC il faut pouvoir adapter la QdC à la QdS proposée. Dans cet esprit [Colandairaj 2007] propose dans le cadre non filaire (réseau IEEE 802.11, WiFi) d'adapter la vitesse de communication en fonction des erreurs de communication (adaptation de la communication) et conjointement d'adapter la période d'échantillonnage en fonction des paramètres réseau (adaptation du contrôle). La transmission à des vitesses plus faibles dans le cas de taux d'erreur trop important est moins susceptible d'erreurs, donc de retransmissions, ce qui conduit à des retards moins importants et à une amélioration de la QdC. Cependant transmettre à faible vitesse diminue la bande passante et peut conduire à des congestions. Ce problème est alors résolu en augmentant la période d'échantillonnage. Cette période est calculée en fonction de paramètres mesurés de QdC et de QdS (retard, débit, perte). Ces changements de fréquence conduisent à des changements de correcteurs qui peuvent entraîner des problèmes d'instabilité à la commutation ainsi que nous l'avons déjà noté.

Les mécanismes à partir desquels il est possible d'agir sur la QdS dépendent des protocoles et sont spécifiques à chaque réseau, nous avons ici présenté deux exemples du monde filaire (CAN, Switched Ethernet) et deux exemples du monde non filaire (ZigBee, WiFi). Notre travail portant majoritairement sur l'ordonnement des messages, nous allons nous intéresser aux protocoles de la sous-couche MAC et présenter des réseaux sur lesquels nous avons travaillé (CAN) ou qui nous ont servi de référence (IEEE 802.11 et IEEE 802.15.4) ; puis dans le cas non filaire nous nous intéresserons aux mécanismes de QdS permettant de donner un caractère temps-réel à ces réseaux.

## 1.4 Sur l'ordonnement de messages (couche MAC)

La sous-couche MAC de la couche liaison de données décrit le protocole d'accès au réseau ; le MAC résout les problèmes de contention d'accès au réseau et est responsable du caractère temps-réel du réseau. Les mécanismes d'accès au réseau se répartissent en 3 grandes classes :

- TDMA : Time Division Multiplexing Access. Le temps d'accès au médium est alloué de manière circulaire (Round Robin), soit par la circulation d'un jeton, soit avec une station maître et des stations esclaves. L'allocation des tranches de temps étant bien ordonnée il ne peut pas y avoir de collisions. Ces réseaux sont déterministes, le temps d'attente maximum est le temps de cycle.
- CSMA : accès aléatoire avec retransmissions en cas de collision, Ethernet (Carrier Sense Multiple Access/Collision Detection, CSMA/CD) et les mécanismes des réseaux sans fil (Carrier Sense Multiple Access/Collision Avoidance, CSMA/CA). Du fait de son coût faible, Ethernet est très répandu même dans le monde industriel. Ce réseau n'est pas déterministe et la QoS ne peut pas être garantie. Des collisions peuvent se produire et dans ce cas les messages sont retransmis après une attente aléatoire.

Switched Ethernet combine TDMA et accès aléatoire. Les stations sont reliées à un commutateur et ce commutateur retransmet les paquets vers leur destination. Il ne peut pas y avoir de collision sur le réseau mais la congestion au commutateur peut amener des pertes. Les mécanismes d'évitement de collisions (CSMA/CA) des réseaux sans fil utilisent des temps d'attente aléatoires pour minimiser la probabilité de transmission de plusieurs stations au même instant, mais les collisions peuvent tout de même se produire.

- Accès aléatoire avec priorités pour l'arbitrage des collisions (mécanisme du réseau CAN). Des collisions peuvent se produire mais elles ne sont pas destructives (des situations de collision induisent des compétitions qui donnent un vainqueur).

Notre travail a porté sur le réseau CAN et son mécanisme de priorité et nous avons proposé dans le cadre non filaire des protocoles sans collision que nous avons comparé à WiFi et ZigBee. Nous nous intéressons donc plus particulièrement à ces réseaux.

### 1.4.1 Réseau filaire : CAN

Le réseau CAN [CiA 2002], [Bosch 1991], est basé sur un bus série partagé par les stations au moyen du mécanisme CSMA et de l'arbitrage par priorités de la sous-couche MAC. La résolution des collisions est déterministe et s'appuie sur des priorités associées aux identificateurs (adresses) des trames. Le conflit de bus est résolu par le mécanisme du « ET câblé », c'est-à-dire qu'un bit dominant (bit 0) écrase un bit récessif (bit 1). Plus petite est l'adresse, plus grande est la priorité. Une station qui ne reconnaît pas l'adresse de la trame qu'elle est en train d'émettre, détecte une collision et s'arrête jusqu'à ce que le bus redevienne libre. La station qui ne détecte pas de collision continue. Une station qui émet tous ses bits sans détecter de collision est le vainqueur unique de la compétition pour l'accès au bus compte-tenu de l'unicité des identificateurs. Les priorités associées aux identificateurs reflètent les priorités des données incluses dans les trames.

La longueur d'une trame est donnée par la relation suivante [Navet 1999] :

$$L = 47 + 8d + \lfloor \frac{34 + 8d - 1}{4} \rfloor \quad (1.2)$$

avec  $d$  est le nombre d'octets du champ de données ( $d \leq 8$ ).

Plusieurs vitesses, qui dépendent de la longueur des bus CAN, sont utilisables : la vitesse maximum est 1 Mbits/s.

### Quelques résultats pour implanter un SCR à travers CAN

Beaucoup de travaux ont été conduits dans le domaine des SCR sur le réseau CAN. Parmi ceux-ci nous citerons [Walsh 2001]. Les auteurs montrent qu'il vaut mieux utiliser une technique d'écrasement lorsque le réseau n'est pas disponible plutôt qu'une technique de file d'attente FIFO. Pour cela, ils comparent au nœud capteur le comportement d'une file FIFO (qui stocke les données à émettre), avec un tampon avec écrasement. Dans ce dernier cas, si une donnée n'est pas émise par le capteur alors qu'un nouvel échantillonnage se produit, c'est cette dernière valeur qui sera émise vers le contrôleur (cette technique qui donne de meilleurs résultats est nommée TOD pour Try-Once-Discard).

Dans [Juanole 2005a], les auteurs arrivent à la même conclusion : dans le cas d'un réseau chargé ou surchargé il vaut mieux avoir des pertes provoquées par des écrasements plutôt que des retards devenant prohibitifs, liés au mécanisme de file d'attente FIFO. Ils proposent une modification du simulateur TrueTime (voir Sect. 1.5.3) pour mettre en œuvre ce mécanisme d'écrasement à tous les nœuds producteurs de données.

Ces mêmes auteurs montrent dans le cadre d'une étude sur les priorités statiques associées aux flux de données, [Juanole 2005b], que dans le cas de réseaux fortement chargés la priorité du flux contrôleur-actionneur ( $Pca$ ) doit être supérieure à la priorité du flux capteur-trôleur ( $Pcc$ ), la priorité est ainsi donnée à la fermeture de la boucle de commande. Dans le cas contraire les flux capteurs sont émis sur le réseau et les flux contrôleurs de moindre priorité peuvent en cas de surcharge ne pas avoir accès au réseau et donc ne pas piloter les actionneurs. Dans le cadre de simulations les auteurs montrent que la parité flux capteur, flux contrôleur (même nombre de flux) est obtenue en donnant au capteur une priorité moindre que celle du contrôleur. Lorsque le contrôleur reçoit une donnée, il est prioritaire pour réémettre la commande vers l'actionneur.

Dans notre travail nous considérerons toujours  $Pca > Pcc$ .

Dans le travail déjà cité, concernant l'ordonnement de messages à partir de priorités (réseau CAN) [Walsh 2001], les auteurs proposent un algorithme d'affectation des priorités en fonction de l'erreur en boucle fermée. Cette allocation dynamique des priorités donne la priorité au système dont la valeur absolue de l'erreur est la plus grande et porte le nom de MEF-TOD pour « Maximum Error First with Try On Discard ». La structuration de l'identificateur en deux niveaux, l'un dynamique, l'autre statique (pour assurer l'unicité des identificateurs) est présente dans l'ordonnement Mixed Traffic Scheduler, [Zuberi 1997] et [Zuberi 2000], où les auteurs combinent EDF (champ dynamique) et FP (champ statique). Cette idée de l'ordonnement basée sur les performances de l'application en boucle fermée à travers l'erreur est aussi proposée dans [Yepez 2003] et [Marti 2004]. Leurs travaux sont prolongés par [Velasco 2006] où des classes de messages sont définies (messages contrôleur-actionneur : les plus prioritaires, messages

capteur-contrôleur : priorités inférieures aux précédentes, messages d'usage général : les moins prioritaires). Cette répartition en classes obtenue par structuration de l'identificateur garantit un retard moindre dans la boucle de commande.

Une partie de notre travail de thèse synthétise et prolonge les travaux [Juanole 2007b], [Juanole 2007a], [Juanole 2008] sur la priorité hybride où un champ de l'identificateur caractérise l'urgence de la transmission (constante ou variable), l'autre champ statique résout les collisions en cas d'égalité des champs dynamiques. Ce travail fait l'objet du chapitre 2 de notre thèse.

## 1.4.2 Réseaux non filaires

### 1.4.2.1 Réseaux standards

Nous allons présenter les principaux éléments des fonctions de base (fonction MAC et couche physique) de deux réseaux sans fil très répandus et qui par la suite (Chap. 4) nous serviront de référence : IEEE 802.11, WiFi et IEEE 802.15.4, ZigBee.

#### IEEE 802.11, WiFi [Crow 1997], [Mangold 2003]

Deux modes d'accès au médium sont spécifiés dans le standard : le mode centralisé où une station peut autoriser l'accès au médium à d'autres stations à travers des trames de commande (polling frames). Cette station est dite « Point Coordination Function (PCF) station » et ce mode est nommé PCF ; le mode distribué ou « Distributed Coordination Function (DCF) » où toutes les stations peuvent accéder au médium à travers une technique CSMA/CA.

Deux versions du mode DCF sont définies : un accès basé sur une double poignée de main et un accès à travers Request-To-Send/Clear-To-Send (RTS/CTS) basé sur une quadruple poignée de main. Nous considérerons ici uniquement le mode d'accès distribué et la version de base du DCF.

Une station qui veut émettre un paquet de données doit trouver le médium libre pendant DIFS (DCF Inter Frame Space) avant la transmission. Si le médium n'est pas libre un temporisateur dit backoff timer est lancé. La valeur initiale du Backoff  $BV$  est choisie aléatoirement,  $BV \in [0, CW_{\min}]$  où  $CW_{\min}$  est la largeur initiale de la fenêtre de contention en nombre de tranches de temps élémentaires (slots). Ce nombre est le nombre de slots que la station doit attendre avant d'être autorisée à transmettre. Si le médium passe de libre à occupé le décompte est gelé jusqu'au prochain DIFS à partir duquel il reprendra. Si une collision a lieu,  $CW$  est doublé jusqu'à qu'il atteigne une valeur maximum,  $CW_{\max}$ . Ici contrairement à un réseau câblé, comme une station ne peut pas écouter alors qu'elle émet, la détection de collision n'est pas possible (à l'émetteur) et après chaque émission l'émetteur attend un acquittement (ACK) du récepteur. Le temps d'attente pour une trame d'acquiescement est SIFS (Short Inter Frame Space) où  $SIFS < DIFS$ . Après un succès,  $CW$  est réinitialisé à  $CW_{\min}$ . Après une transmission réussie l'émetteur lance une nouvelle attente de durée aléatoire (post-backoff), même s'il n'y a pas d'autre données à émettre. Ce post-backoff assure qu'il y a toujours un temps aléatoire entre deux émissions consécutives à un nœud donné.

La figure 1.7 montre une transmission réussie de la source S vers le receveur R et les divers temps d'attente.

La longueur en bits d'une trame de données MAC est de  $L_D = 272 + n$ ,  $n$  étant la longueur du champ des données avec  $n \leq 18496$ . La longueur, en bits de la trame d'acquiescement est  $L_A = 112$ .

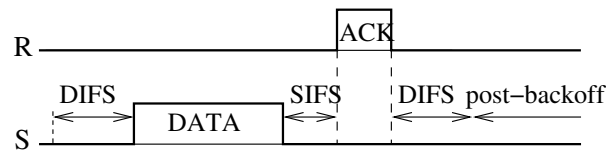


FIGURE 1.7 – Transmission réussie (IEEE 802.11).

Pour la couche physique nous considérons le protocole 802.11b et la technique de modulation DSSS (Direct Sequence Spread Spectrum). Cette technique utilise la bande de fréquence de 2.4 GHz (Industrial Scientific and Medical (ISM) band). Les vitesses de transmission sont de 54, 11, 5.5, 2 et 1 Mbits/s. La couche physique introduit 144 bits de préambule et un en-tête de 48 bits soit un total de  $L_p = 192$  bits. Ces bits sont émis à 1 Mbits/s quelle que soit la vitesse de transmission des trames MAC.

Les paramètres d'une transmission DSSS sont donnés par le tableau 1.1.

|             |                |             |                  |
|-------------|----------------|-------------|------------------|
| slot time   | $20 \mu s$     | DIFS        | $50 \mu s$       |
| SIFS        | $10 \mu s$     | $CW_{\max}$ | 1023 (slot time) |
| $CW_{\min}$ | 31 (slot time) |             | $20460 \mu s$    |
|             | $620 \mu s$    |             |                  |

TABLE 1.1 – Paramètres d'une transmission DSSS.

### IEEE 802.15.4, ZigBee [Callaway 2002], [IEEE 802.15 2003]

Le protocole MAC supporte deux modes d'accès au médium :

- mode avec balises dit slotted CSMA/CA qui suppose une topologie structurée avec un nœud coordinateur, une supertrame est utilisée pour piloter la communication entre les stations ;
- mode sans balise ou unslotted CSMA/CA, technique utilisée pour les réseaux *ad hoc*, où une station envoie ses données en utilisant le mécanisme CSMA/CA.

Nous intéressent aux réseaux *ad hoc*, nous présenterons succinctement ce dernier mode. Quand une station veut émettre une donnée, elle attend d'abord, sans tester le médium pour une durée aléatoire. Si le médium est libre à la fin de ce temps de backoff, la station émet son paquet. Sinon elle démarre une nouvelle attente de durée aléatoire (le nombre d'essais est limité). Le temps de backoff est basé sur le concept de backoff periode (20 symboles de 4 bits, durée  $320 \mu s$ ) et de backoff exposant ( $BE$ ) ; ce temps est calculé aléatoirement en nombre de backoff periodes dans l'intervalle  $[0, 2^{BE} - 1]$  (la valeur par défaut de  $BE$  est 3)<sup>1</sup>. Une trame d'acquiescement est utilisée pour contrôler l'occurrence de collisions, cette trame est émise par le récepteur après un temps de retournement (noté TT) dont la valeur est de 12 symboles de 4 bits.

La longueur en bits de la trame de données est  $L_D = 72 + n$ ,  $n$  étant la longueur de la donnée ( $n \leq 944$ ). La longueur en bits de la trame d'acquiescement est  $L_A = 40$ . Le temporisateur SIFS est également utilisé (durée de 48 bits) :  $SIFS = 192 \mu s$ .

La figure 1.8 montre une transmission réussie de la source S vers le receveur R et les divers temps d'attente.

1. C'est en changeant les bornes de l'intervalle que des niveaux de priorité différents ont été obtenus dans [Boughanmi 2009], technique d'allocation de priorités dynamiques présentée dans la sous-section 1.3.2.

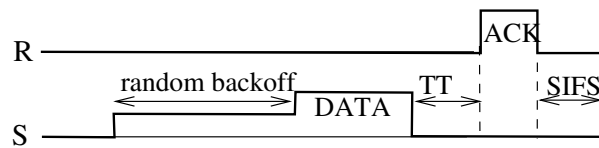


FIGURE 1.8 – Transmission réussie (IEEE 802.15.4).

Concernant la couche physique, nous considérons la bande de fréquence de 2.4 Ghz (ISM band), qui conduit à une vitesse de transmission de 250 kbits/s. La couche physique ajoute 48 bits supplémentaires aux trames MAC.

#### 1.4.2.2 Réseaux orientés temps réel : les mécanismes MAC pour la QoS dans les réseaux sans fil *ad hoc*

Les technologies sans fil sont maintenant utilisées à grande échelle dans le domaine industriel, entre autre pour conduire des procédés. Dans ces applications, les communications sans fil doivent satisfaire des contraintes temps-réel qui sont plus difficiles à obtenir que dans un contexte filaire. Un des grands inconvénients de la communication sans fil est que les émetteur-récepteur (transceivers) ne sont pas capables d'émettre et d'écouter simultanément sur le même canal.

Pour fournir des garanties de QoS il est nécessaire d'adapter des technologies sans fil existantes aux contraintes temps-réel ou d'en développer de nouvelles. Le MAC qui implémente l'ordonnancement des messages de données (venant des applications) à travers la couche physique est responsable de la résolution des conflits entre les différentes stations pour l'accès au canal de communication. Il a donc une influence directe sur la sécurité et l'efficacité des transmissions et est un élément important dans la QoS d'un réseau.

Un état de l'art très complet sur les communications temps-réel et les technologies sans fil incluant Bluetooth, IEEE 802.11, IEEE 802.15.4 est proposé dans [Willig 2005] et une très riche bibliographie du domaine y est fournie.

Nous considérons ici des réseaux *ad hoc* (association auto-organisée et adaptative de composants sans fil) qui n'ont pas de contrôle centralisé. Pour ce type de réseaux, une classification des protocoles MAC en fonction de leurs mécanismes et des objectifs de QoS est proposée dans [Kumar 2006]. Dans cette étude les auteurs distinguent

1. DCF avec des classes de priorités et « enhanced DCF », qui peuvent être implantées sur IEEE 802.11
  - DCF avec des classes de priorités (PC-DCF). Deng et Chang [Deng 1999] utilisent les deux possibilités de IEEE 802.11 pour créer de la différenciation : l'espace IFS utilisé entre deux trames de données (si deux stations utilisent des IFS différents la station qui a l'IFS le plus court gagne la compétition) et le mécanisme de backoff (des valeurs différentes des paramètres sont utilisés en fonction de la classe de priorité). Ces éléments combinés fournissent quatre niveaux de priorités ;
  - Enhanced DCF (EDCF). EDCF [Mangold 2003] introduit des priorités et huit catégories de trafic. Chaque trame de données est assignée à une classe de trafic (TC) basée sur la priorité assignée par les couches de niveau supérieur. Par rapport à la classe de trafic, des durées d'arbitrage entre trames, AIFS, et des fenêtres de contention différentes,



CW, sont définies. Durant la phase d'arbitrage EDCF utilise AIFS(TC), CWmin(TC) et CWmax(TC) au lieu de DIFS, CWmn, CWmax pour le standard.

2. Résolution de la contention par signaux Black-Burst, par des attentes différenciées (elimination by sieving ES-DCF) et Black-Burst fonction de l'échéance (deadline bursting DB-DCF) afin de fournir des retards bornés.
  - Black-Burst Contention (BB-DCF) [Sobrinho 1999], [Sobrinho 1996]. Un nœud qui a un besoin d'émettre occupe le médium avec des impulsions d'énergie (Black-Burst). La durée de l'impulsion est proportionnelle à la priorité du message. Si plusieurs nœuds tentent de transmettre simultanément, la station de plus haute priorité sera la seule à trouver le médium libre quand elle cessera sa transmission. De cette façon la station de plus haute priorité sait qu'elle a gagné la compétition ;
  - Elimination par « Sieving and Deadline Bursting » [Pal 2002]. Les auteurs proposent deux variantes de DCF IEEE 802.11 en utilisant le mécanisme de différenciation par IFS et les signaux de Black-Burst.
    - Elimination par tamis ou attentes différenciées (ES-DCF). Cette procédure a trois phases : élimination, acquisition du canal et résolution des collisions. Dans la phase d'élimination, chaque nœud attend sur une durée basée sur la priorité, PIFS, (la valeur DIFS pour les messages non temps-réel est supérieure à la valeur de PIFS). Deux ou plusieurs paquets temps réel, en différents nœuds, peuvent avoir le même IFS. Après attente relative à cet IFS, si le canal est libre, un nœud envoie un signal RTS et attend CTS. S'il obtient un CTS la transmission peut commencer, sinon il rentre dans une phase de résolution de collision par envoi d'un Black-Burst. La longueur du BB correspond à l'identificateur (unique) du nœud ;
    - Deadline Bursting (DB-DCF). Les trois phases sont la résolution de la contention par Black-Burst, l'acquisition du canal et la résolution de la collision. La différence avec le protocole précédent porte sur la première phase. Les longueurs des BB sont proportionnels aux échéances relatives (proportionnels à l'urgence des paquets temps-réel). Dans ce protocole deux signaux BB sont émis, le premier est basé sur le deadline, et si il y a une collision le second BB correspond à l'identificateur (unique) du nœud.
3. Distributed Fair Scheduling (DFS) [Vaidya 2000], [Vaidya 2005] : garantie d'une proportion équitable du canal aux différents flux. Ce schéma, noté DFS, utilise le mécanisme de backoff de IEEE 802.11 afin de déterminer le vainqueur de la compétition : le temps de backoff calculé est proportionnel à la taille du paquet et inversement proportionnel à la pondération du flux. Ainsi un paquet de grande priorité se verra assigné un backoff plus court et par conséquent ce flux aura plus grande sortance. Si une collision se produit un nouvel intervalle de backoff est calculé en accord avec le standard IEEE 802.11.

Parmi ces différents mécanismes permettant de donner un caractère temps réel au protocole MAC nous avons travaillé sur les signaux d'annonce (BlackBurst) et les attentes différenciées (WhiteBurst) qui nous permettent de donner des priorités statiques ou dynamiques aux messages et nous proposons par la suite de notre travail des protocoles à priorités et sans collision.

## 1.5 Notre travail dans le contexte des recherches actuelles sur les SCR

Des articles récents ont porté sur les tendances de la recherche dans le domaine des SCR et sur les défis des SCR [Baillieul 2007], [Zampieri 2008], [Gupta 2010]. Les recherches actuelles portent sur la modélisation des retards, la prédiction de ces retards et de leur variabilité et sur la conception de systèmes robustes aux variations de délais. Des efforts sont faits dans le domaine des développements de la co-conception où la conception de l'ordonnancement, des protocoles de transfert et du contrôleur doit être traitée conjointement avec pour objectif de garantir la QdC des applications.

### 1.5.1 Co-conception

Pour une application de contrôle-commande, l'allocation dynamique de ressource peut être faite par l'adaptation de mécanismes de QdS en fonction de la QdC de l'application (écart à la sortie, signal de commande ou fonction de coût caractérisant la QdC). C'est dans ce domaine que porte principalement notre travail (cf. Chap. 2) : ordonnancement des messages en fonction de la QdC (relation QdC→QdS). L'ordonnancement proposé est un ordonnancement hybride : un champ de l'identificateur caractérise l'urgence de transmission (dynamique), l'autre champ statique caractérise le flux.

Quand l'adaptation de la QdS atteint ses limites (charge trop importante du réseau) les applications de commande doivent s'adapter en ajustant par exemple leur période d'échantillonnage. Nous avons dans ce cadre déjà cité (voir Sect. 1.3.1.1) les travaux sur la régulation de tâches entraînant une variation contrôlée de la période. Le trafic est réduit en adaptant la charge du réseau (relation QdS→QdC). Cette variation de la fréquence d'échantillonnage en fonction de la QdC peut aussi avoir pour objectif de relâcher des ressources pour des applications « best effort » quand l'application a moins besoin de ces ressources. C'est ce que nous faisons remarquer dans le cas de la commande d'un drone (Sect. 2.6).

Il est aussi possible de changer de correcteur en fonction de la QdS proposée (voir le travail [Marti 2002] relatif à l'adaptation des correcteurs aux retards liés soit à l'ordonnancement des tâches, soit à la charge du réseau). C'est également ce que nous proposons dans les chapitres 3 et 5 où nous travaillons à la mise en œuvre à la fois de la relation QdS→QdC (correcteur adapté au retard mesuré) et de la relation QdC→QdS (priorités pilotées par la QdC) c.-à-d. la mise en œuvre de la relation bidirectionnelle  $QdS \rightleftharpoons QdC$ .

Toutes ces approches sont des approches co-conception car elles considèrent au même moment non seulement la QdS réseau mais aussi la QdC.

### 1.5.2 Réseaux sans fil et protocoles MAC à priorités et sans collision

La grande tendance est l'utilisation des réseaux sans fil à tous les niveaux [Moyne 2007]. Cette migration du réseau câblé au réseau sans fil pose des problèmes d'interférence, de sécurité, de fiabilité et de déterminisme dans les transmissions. Cette tendance appelle à définir de nouveaux protocoles MAC ayant des caractéristiques temps réel.

Beaucoup de recherches ont été conduites sur les protocoles MAC dans les dernières années et une sélection des protocoles, pour la communication sans fil dans le monde industriel, est proposée dans [Willig 2008]. L'auteur souligne le besoin de définir des protocoles déterministes basés sur des priorités et considère nécessaire de développer des recherches sur les protocoles de type CAN (pour le sans fil). Pour l'auteur les mécanismes à considérer pour obtenir ou tendre vers des protocoles sans fil déterministes sont :

- envoi d'un signal d'annonce (BlackBurst) ou temps d'attente (WhiteBurst) : les nœuds voulant participer à une compétition pour l'accès au médium envoient un signal d'annonce proportionnel à la priorité du paquet à émettre (technique BB) ou attendent une durée inversement proportionnelle à la priorité du paquet à émettre (technique du tamis ou technique WhiteBurst). Dans les deux cas la longueur maximale de l'annonce ou de l'attente est linéaire avec le nombre de priorités ( $2^n$  pour un identificateur de  $n$  bits) ;
- adaptation du protocole CAN. La technique d'arbitrage bit à bit de CAN ne peut pas être directement appliquée dans le cas sans fil : cette technique suppose que la station puisse à la fois émettre et écouter. Une adaptation de ce protocole est proposée dans [You 2003a] et [Pereira 2007a] : un intervalle de temps est utilisé pour chaque bit de priorité, sur ce slot une station qui a un bit dominant (0) transmet, une station qui a un bit récessif écoute ; quand une station qui a un bit récessif reçoit un signal elle a perdu la compétition et se retire. La durée de la phase d'arbitrage est linéaire en  $n$  (nombre de bits de l'identificateur).

Ces deux possibilités sont étudiées dans notre travail. Dans un premier temps nous considérons l'utilisation d'impulsions BB, dans un deuxième temps nous nous intéresserons aux protocoles « dominance protocols » ou « binary countdown protocols », soit encore les protocoles dits CANlike.

Comme déjà mentionné nous considérons des réseaux *ad hoc* où les stations communiquent sans sollicitation. De plus toutes les stations sont chacune dans le domaine de transmission de toutes les autres, ainsi la communication entre deux stations se fait en un seul saut et comme toutes les stations sont visibles deux à deux nous ne considérerons pas le problème du terminal caché et du terminal exposé. Notre projet est d'utiliser les priorités des flux pour éviter des collisions et obtenir la QoS. La QoS obtenue à partir des priorités suppose que les applications fixent les priorités pour l'accès au médium. Ces priorités peuvent être statiques (associées aux flux) ou hybrides (dynamique + statique) ; la partie statique assure l'unicité de l'identificateur et la partie dynamique varie avec l'urgence de la communication ; cette urgence est estimée à partir d'un critère de QoS. Ces deux cas de priorités sont considérés dans notre étude (Chap. 4).

### 1.5.3 Un outil de simulation : TrueTime

L'insertion d'un réseau dans une boucle de commande est un défi pour la modélisation, l'analyse et la conception des systèmes commandés en réseau. Les techniques pour la modélisation du procédé, de l'échantillonneur, du contrôleur et de l'actionneur sont bien connues mais les techniques pour modéliser les principales caractéristiques d'un réseau de communication et les interactions entre le réseau et le système à commander sont très complexes et ne permettent pas de traiter le problème dans sa globalité. Il est donc nécessaire de faire appel à des outils de simulation. Parmi ces outils de simulation, TrueTime [Ohlin 2007] [Henriksson 2003], qui est orienté co-conception (commande, ordonnancement de tâches, réseau), est particulièrement bien adapté à nos objectifs et c'est cet outil qui a été utilisé tout au long de notre travail.

Le simulateur TrueTime est une boîte à outils basée sur Matlab/Simulink, permettant la simulation de systèmes de contrôle distribués temps réel. Il est développé par un groupe de recherche suédois à Lund Institute of Technology depuis 1999. Il permet de simuler le comportement temps réel des systèmes multi-tâches exécutant des tâches de contrôle-commande modélisées par des blocs Simulink. Ces différentes tâches peuvent être distribuées et inter-connectées à travers un réseau.

Les différentes politiques d'ordonnancement implantées dans TrueTime sont : FP (Fixed Priority), RM (Rate monotonic), DM (Deadline Monotonic) et EDF (Earliest Deadline First).

TrueTime permet de simuler des modèles simples de réseaux de communication et donc d'étudier leur influence sur les performances des applications réalisées à travers le réseau.

Les protocoles MAC les plus répandus sont supportés par TrueTime. Pour le réseau filaire, il est possible de spécifier les protocoles CSMA/CD (ETHERNET), CSMA/AMP (CAN), Round Robin, FDMA, TDMA et Switched Ethernet. Pour les réseaux non filaires, les protocoles MAC implantés dans TrueTime sont : IEEE 802.11 b/g (WiFi) et IEEE 802.15.4 (ZigBee). Seuls les modes de fonctionnement *ad hoc* sont possibles pour ces deux réseaux. Au niveau du bloc réseau sans fil de TrueTime il est possible de spécifier le protocole MAC, la vitesse de transmission, la taille des trames, la durée d'attente de l'acquittement, ...

Pour prendre en compte l'affaiblissement du signal en fonction de la distance, la position de chaque nœud est spécifiée par les coordonnées  $x, y$ . Les interférences avec d'autres terminaux sont détectées.

C'est cet outil que nous avons utilisé tout au long de notre travail de thèse, pour des simulations de SCR à travers les réseaux, CAN, WiFi, ZigBee et que nous avons également modifiés pour introduire les priorités hybrides dans l'assignation des identificateurs dans CAN (Chap. 2) et pour mettre en œuvre les protocoles sans collision et à priorités (Chap. 4 et Chap. 5).



# Réseau CAN : concept de priorité hybride (relation $QdC \rightarrow QdS$ )

---

## Sommaire

---

|            |   |           |
|------------|---|-----------|
| <b>2.1</b> | <b>Introduction</b> . . . . .   | <b>32</b> |
| <b>2.2</b> | <b>Contexte d'étude</b> . . . . .   | <b>32</b> |
| 2.2.1      | Application de contrôle-commande distribuée considérée . . . . .                    | 32        |
| 2.2.2      | Réseau CAN avec priorités statiques . . . . .                                       | 34        |
| 2.2.3      | Idées de la priorité hybride . . . . .  | 36        |
| <b>2.3</b> | <b>Trois schémas de priorité hybride</b> . . . . .                                  | <b>38</b> |
| 2.3.1      | Schéma ph . . . . .   | 39        |
| 2.3.2      | Schéma ph + sts . . . . .   | 40        |
| 2.3.3      | Schéma ph+std . . . . .   | 41        |
| <b>2.4</b> | <b>Implantation d'une application de contrôle-commande et d'un flux externe</b> . . | <b>44</b> |
| 2.4.1      | Introduction . . . . .  | 44        |
| 2.4.2      | Références à considérer : QdS et QdC avec les priorités statiques . . . . .         | 44        |
| 2.4.3      | QdS fournie par les trois schémas de priorité hybride . . . . .                     | 46        |
| 2.4.4      | QdC fournie par les différents schémas de priorité hybride . . . . .                | 50        |
| <b>2.5</b> | <b>Implantation de plusieurs applications de contrôle-commande</b> . . . . .        | <b>52</b> |
| <b>2.6</b> | <b>Commande d'un drone à quatre rotors</b> . . . . .                                | <b>54</b> |
| 2.6.1      | Le système étudié . . . . .   | 54        |
| 2.6.2      | Implantation sur CAN . . . . .  | 55        |
| <b>2.7</b> | <b>Conclusion</b> . . . . .   | <b>58</b> |

---

## 2.1 Introduction

L'objectif de ce chapitre est de montrer, dans le contexte de l'implantation d'applications de contrôle-commande à travers un réseau, l'intérêt de mettre en œuvre une adaptation dynamique de mécanismes de transfert de messages sur la base de la connaissance des besoins de ces applications. Le mécanisme considéré ici est l'ordonnancement des messages qui est basé sur des priorités hybrides qui dépendent, en particulier, de l'urgence de la transmission des messages, l'urgence découlant des contraintes imposées par les besoins des applications.

Ce chapitre comprend sept sections : la deuxième section concerne le contexte de l'étude ; la troisième section définit et présente trois modalités de priorités hybrides basées sur une adaptation dynamique aux besoins des applications ; les quatrième et cinquième sections montrent respectivement la QdS et la QdC fournies par ces trois modalités et l'intérêt par rapport aux priorités statiques qui sont les plus souvent utilisées ; la sixième section présente l'utilisation de ces modalités de priorité hybride dans le cadre d'une application de commande d'un drone ; la septième section conclut le chapitre.

La plupart des résultats de ce chapitre a été présentée dans [Nguyen Xuan 2010a].

## 2.2 Contexte d'étude

### 2.2.1 Application de contrôle-commande distribuée considérée

#### Application continue de départ

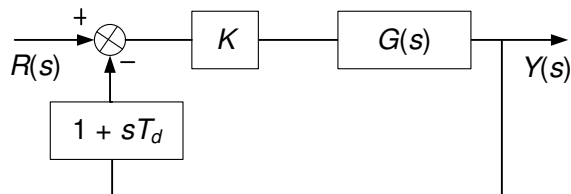


FIGURE 2.1 – Modèle.

Cette application est représentée sur la figure 2.1. Le procédé à contrôler est un servo-moteur de fonction de transfert  $G(s) = \frac{1000}{s(1+s)}$  et le contrôleur est un Proportionnel-Dérivé (PD) avec une action dérivée sur la sortie. L'algorithme PD a la forme suivante :  $U(s) = K(R(s) - (1 + sT_d)Y(s))$  où  $U(s)$ ,  $R(s)$  et  $Y(s)$  représentent les transformées de Laplace de respectivement le signal de contrôle  $u(t)$ , le signal d'entrée  $r(t)$  et le signal de sortie  $y(t)$ . Relativement au contrôleur,  $K$  représente le gain Proportionnel et  $T_d$  représente la constante de temps associée à la dérivée du signal de sortie. La fonction de transfert du système bouclé  $F(s)$  est une fonction du second ordre :

$$F(s) = \frac{1000K}{s^2 + (1 + 1000KT_d)s + 1000K} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Cette fonction de transfert est caractérisée par la pulsation propre non amortie  $\omega_n$  et le coefficient d'amortissement  $\zeta$  ( $\omega_n^2 = 1000K$  et  $2\zeta\omega_n = 1 + 1000KT_d$ ).

On étudie le cas d'un signal d'entrée de type échelon de position démarrant à l'instant  $t_0 = 0$ .

On s'impose les performances suivantes : dépassement de 5% et temps de réponse à 5% inférieur à 100 ms. Ceci nécessite les caractéristiques dynamiques suivantes ( $\zeta = 0.7$  et  $\omega_n = 43$  rad/s) et donne un temps de montée ( $t_m$ ) de 40 ms. Dans ces conditions, on a besoin des valeurs suivantes pour  $K$  et  $T_d$  :  $K = 1.8$ ,  $T_d = 0.032$  s.

### Discrétisation du système continu

Comme on veut implanter cette application à travers un réseau, on doit donc échantillonner le signal de sortie. Le choix de la période d'échantillonnage  $h$  est un choix fondamental. On utilise la formule 1.1 d'où  $2.3 \text{ ms} \leq h \leq 13.9 \text{ ms}$ . On choisit  $h = 10 \text{ ms}$ .

On discrétise aussi le contrôleur avec la fréquence d'échantillonnage. Les performances mesurées avec le contrôleur discrétisé sont : un dépassement inférieur à 5% ( $\approx 1\%$ ) ; un temps de montée de 34 ms ; un temps de réponse à 5% de 45 ms.

### Réseau de communication

Le réseau considéré est le réseau local CAN [Bosch 1991] limité à la couche physique et à la couche MAC qui implante l'ordonnement des messages.

Concernant la couche physique, on considère le cas d'un débit de 125 Kbits/s.

Concernant l'ordonnement de messages, nous rappelons que la couche MAC du réseau CAN est basée sur le principe d'un protocole MAC de type CSMA gérant l'accès au médium sur la base de priorités statiques qui apparaissent dans le champ Identificateur des trames (« ID field »). L'ordonnement est fait, dans chaque nœud, par la comparaison bit par bit (entre bit de ID émis et bit de ID reçu). Comme le bit 0 est un bit dominant et le bit 1 est un bit récessif, la priorité est d'autant plus élevée que la valeur du champs ID est faible. Nous considérons ici la longueur standard de 11 bits pour le champ ID.

### Application discrétisée distribuée

L'implantation distribuée à travers le réseau CAN est caractérisée par deux flux de trames :

- le flux  $f_{cc}$  c.-à-d. le flux des trames Capteur-Contrôleur ;
- le flux  $f_{ca}$  c.-à-d. le flux des trames Contrôleur-Actionneur.

Nous considérons ici que les champs de données des trames des flux  $f_{cc}$  et  $f_{ca}$  ont la même longueur de 16 bits, ce qui donne une longueur de trame de 75 bits (Equa. 1.2). Cependant comme on utilise l'ancienne version de TrueTime (qui considère des trames de longueur multiple d'octets), on considère donc pour les trames des flux  $f_{cc}$  et  $f_{ca}$  des longueurs de 80 bits (= 10 octets). Ces flux peuvent être les seuls à utiliser le réseau (on dit que l'on a un réseau dédié à l'application) ou peuvent partager le réseau avec des flux d'autres applications.

### Qualité de Service fournie par le réseau à une application de contrôle-commande

La Qualité de Service (QoS) est évaluée sur la base des retards subis par les trames des flux  $f_{cc}$  et  $f_{ca}$  : on évalue, par période, le retard moyen  $\bar{D}$  et l'écart type  $\sigma$  pendant toutes les périodes d'échantillonnage sur une durée égale au temps de réponse de l'application.



Appelons  $D_i$  le délai de communication dans la boucle pendant la période d'échantillonnage  $i$  dans un temps de réponse qui compte  $n$  périodes d'échantillonnage ( $D_i =$  retard de la trame du flux  $f_{cc}$  + retard de la trame du flux  $f_{ca}$ ). Les valeurs de  $\bar{D}$  et  $\sigma$  sont :

$$\bar{D} = \frac{\sum_{i=1}^n D_i}{n}, \quad \sigma = \sqrt{\frac{\sum_{i=1}^n (D_i - \bar{D})^2}{n}}$$

### Qualité de Contrôle de l'application de contrôle-commande

a/ La Qualité de Contrôle (QdC) est évaluée au moyen d'une fonction de coût, on prend le critère ITSE (Integral of Time-weighted Square Error) :

$$J = \int_0^T t(r(t) - y(t))^2 dt$$

$T$  temps de simulation est pris égal à 500 ms (10 fois le temps de réponse du système échantillonné et donc le système a bien atteint son régime permanent).

La fonction de coût donne uniquement une représentation quantifiée de QdC. En considérant seulement l'application de contrôle-commande avec le contrôleur discrétisé mais sans le réseau, on obtient la valeur  $J = 2.3458 \times 10^{-4}$ . Cette valeur de  $J$  (notée  $J_0$ ) sera considérée comme référence dans la suite de ce travail.

b/ Notons qu'outre cette mesure quantitative, nous utilisons le temps de réponse à 5% de l'application de contrôle-commande que l'on obtient à partir de la réponse temporelle à un échelon de position qui est une visualisation de la QdC.

### 2.2.2 Réseau CAN avec priorités statiques

Les trames ont des priorités statiques c.-à-d. que la priorité d'une trame est la priorité associée au flux auquel elle appartient (donc toutes les trames d'un même flux ont la même priorité).

Nous voulons rappeler les principales études qui ont été faites et les résultats qui ont été obtenus avec l'application de contrôle-commande que nous considérons ici [Juanole 2005b], [Nguyen Xuan 2010a].

#### Types d'études

La première concerne le réseau CAN dédié à une application de contrôle commande (l'objectif est de voir l'influence du format des trames des flux  $f_{cc}$  et  $f_{ca}$  sur la QdC) ; la deuxième concerne le réseau CAN partagé par les deux flux  $f_{cc}$  et  $f_{ca}$  avec un autre flux, appelé flux externe ( $f_{ex}$ ) qui est une abstraction pour représenter d'autres applications (l'objectif est de montrer l'influence de la charge du réseau sur la QdC).

Le flux externe est un flux périodique (période  $T_{ex}$ ), synchrone avec le flux  $f_{cc}$ , de longueur de champ de données de 6 octets ce qui donne une trame de 115 bits (Equa. 1.2. Comme on ne peut considérer sur notre version de l'outil TrueTime que des trames multiples d'octets, on prend donc ici une longueur de trame de 120 bits (= 15 octets)). On considère que la fréquence  $1/T_{ex}$  est un multiple de  $1/h$ . La variation de la fréquence  $1/T_{ex}$  est le moyen de faire varier la charge

du réseau qui est représentée par ce qui est appelé le Taux de Requêtes Utilisateur (TRU) et qui est défini ainsi  $TRU = \frac{D_{cc}}{h} + \frac{D_{ca}}{h} + \frac{D_{ex}}{T_{ex}}$  où  $D_{cc}$ ,  $D_{ca}$  et  $D_{ex}$  sont la durée de la trame respectivement des flux  $f_{cc}$ ,  $f_{ca}$  et  $f_{ex}$ . On a ici  $D_{cc} = D_{ca} = 0.64$  ms et  $D_{ex} = 0.96$  ms.

Nous représentons sur le tableau 2.1 les TRUs qui ont été considérés et que nous reprendrons dans le cadre de ce chapitre (Sect. 2.4) pour l'étude des performances des schémas de priorité hybride.

| Multiple de $1/h$ | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $T_{ex}$ (ms)     | 1.111 | 1.250 | 1.428 | 1.667 | 2.000 | 2.500 | 3.333 | 5.000 | 10.00 |
| TRU (%)           | 99.2  | 89.6  | 80.0  | 70.4  | 60.8  | 51.2  | 41.6  | 32.0  | 22.4  |

TABLE 2.1 – Périodes du  $f_{ex}$  et différents TRUs.

## Résultats

**Réseau dédié :** La fonction de coût  $J$  caractérisant la QdC de l'application de contrôle-commande donne  $J = 2.565 \times 10^{-4}$ . Cette valeur est proche de la valeur  $J_0$  ce qui montre que le réseau dédié a peu d'influence sur l'application de contrôle-commande. Ceci résulte de la faible durée de chaque trame des flux  $f_{cc}$  et  $f_{ca}$  ( $D_{cc} = D_{ca} = 0.64$  ms) par rapport à la période  $h$  (10 ms).

Le temps de réponse est toujours de l'ordre de 45 ms (Fig. 2.2-a).

**Réseau partagé :** Les études faites ont donné les résultats suivants (appelons  $P_{cc}$ ,  $P_{ca}$  et  $P_{ex}$  les priorités respectivement des flux  $f_{cc}$ ,  $f_{ca}$  et  $f_{ex}$ ) :

- les meilleures performances de l'application de contrôle-commande sont obtenues quand  $P_{ca} > P_{cc}$ . L'intuition est que le contrôleur doit envoyer sa trame le plus tôt possible (nous reviendrons sur ce point) ;
- si  $P_{ex} < P_{cc} < P_{ca}$ , les performances de l'application de contrôle-commande (QdC) ne sont pas influencées par le flux externe ;
- si  $P_{ex} > P_{ca} > P_{cc}$ , le flux externe influence la QdC de l'application de contrôle-commande et cette influence peut même dégrader fortement la QdC lorsque la charge du flux externe devient élevée (Figs. 2.2-b et 2.2-c). Ce sont ces résultats qui ont induit nos travaux sur les priorités hybrides.

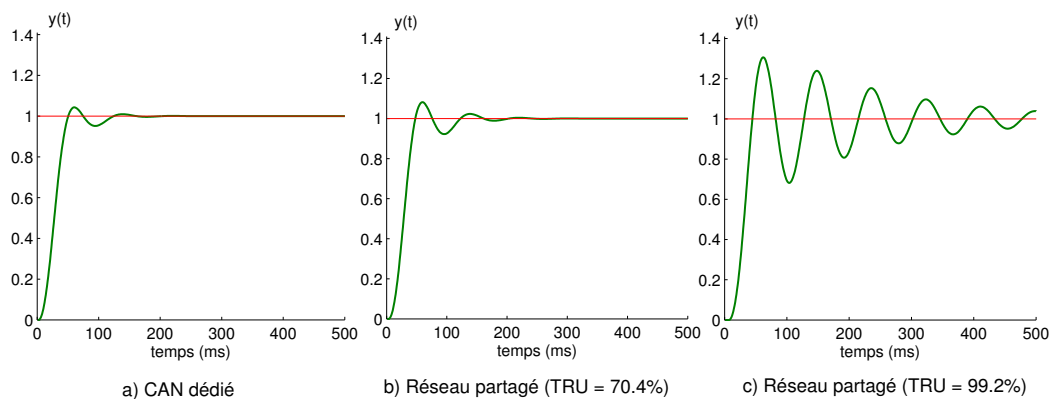


FIGURE 2.2 – Résultats.

### 2.2.3 Idées de la priorité hybride

Quand on a des priorités statiques (priorités a priori), on voit d'après l'étude précédente que lorsque la charge du réseau est importante et si les flux de l'application de contrôle-commande n'ont pas les plus grandes priorités, on peut ne pas obtenir une QdC acceptable. Cependant, en général, il n'est pas toujours possible de donner la plus haute priorité à une application de contrôle-commande : on peut avoir des applications plus importantes et de plus si on a deux ou plusieurs applications de contrôle-commande, seule une pourrait avoir la plus haute priorité (et donc les autres ne l'auraient pas).

L'idée de la notion de priorité hybride résulte de ce problème que nous venons d'expliquer et aussi d'une importante observation : dans le cas le plus général d'un système distribué, on a de nombreuses applications qui génèrent différentes classes de flux qui ont des besoins différents en termes d'urgence de transmission (urgence constante c.-à-d. toujours la même ; urgence variable (de faible à forte)). Une classe de flux est un ensemble de flux qui est spécifié hors ligne. Les besoins représentent des caractéristiques opérationnelles qui dépendent du comportement des applications concernées. Les besoins sont spécifiés hors ligne s'ils sont constants et en ligne s'ils sont variables. Dans ce dernier cas, on dit que l'on a des besoins dynamiques.

Une application de contrôle-commande appartient à la classe des applications de contrôle-commande qui ont chacune deux flux (flux  $f_{cc}$  et  $f_{ca}$ ) et qui ont des besoins dynamiques : urgence forte pendant le régime transitoire après un changement de la référence d'entrée (il faut suivre ce changement) ou après une perturbation (il faut toujours commander le système en dépit de la perturbation) ; faible urgence en régime permanent.

En prenant toujours l'exemple du réseau CAN, ces considérations nous amènent à introduire le concept de priorité hybride pour la trame d'un flux et donc de structurer le champ ID.

#### Structuration du champ ID

Le champ ID ( $n$  bits) d'une trame est divisé en deux niveaux (Fig. 2.3) : le premier niveau (champ de  $(n - m)$  bits) représente la priorité d'un flux (c'est une priorité statique définie hors ligne) ; le second niveau (champ de  $m$  bits) représente la priorité de l'urgence de la transmission.

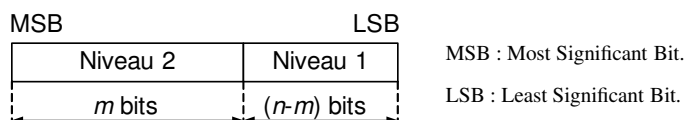


FIGURE 2.3 – Structure du champ ID.

Cette idée de structurer le champ ID est présente dans le « Mixed Traffic Scheduler » [Zuberi 1997, Zuberi 2000] qui combine EDF (« Dynamic field ») et FP (« Static field »). D'autres travaux [Walsh 2001], [Yepez 2003] utilisent également la structuration du champ ID.

En ce qui concerne la priorité de l'urgence de transmission : une urgence constante est caractérisée par une priorité statique (une combinaison de  $m$  bits) ; une urgence variable est caractérisée par une priorité dynamique (qui peut prendre toutes combinaisons de  $m$  bits).

Les trames des flux  $f_{cc}$  et  $f_{ca}$  seront donc caractérisées par :

- la priorité statique (une valeur du champ de  $(n - m)$  bits, c.-à-d. une valeur du niveau 1) qui représente le flux ;
- une priorité dynamique, traduisant l'urgence de transmission, et qui peut avoir n'importe quelle combinaison de  $m$  bits du niveau 2.

L'ordonnement est exécuté dans chaque nœud en comparant d'abord les bits du champ de  $m$  bits (prédominance des besoins) et ensuite, si les besoins des flux sont identiques, en comparant les bits du champ de  $(n - m)$  bits (prédominance du flux).

### **Cohabitation de flux avec des besoins constants et de flux d'applications de contrôle-commande (besoins variables)**

Nous avons, comme objectif, que les besoins très urgents en régime transitoire pour les applications de contrôle-commande puissent être satisfaits rapidement. Pour ceci, nous imposons une priorité maximale aux besoins constants des flux (concept de Seuil de priorité : Spr). De cette manière, la forte urgence de transmission d'un flux d'une application de contrôle-commande (c.-à-d. de priorité supérieure au seuil Spr) sera ordonnancée en premier.

### **Mise en œuvre de la priorité hybride**

La mise en œuvre de la notion de priorité hybride pour une application de contrôle-commande implique :

- d'une part, le choix des priorités statiques associées aux flux  $f_{cc}$  et  $f_{ca}$  ; on prend (suite aux études faites sur les priorités statiques [Juanole 2005b]) priorité  $f_{ca} >$  priorité  $f_{cc}$  ;
- d'autre part, la détermination de la priorité dynamique ; ceci est l'aspect fondamental de notre travail.

La notion de priorité dynamique nécessite, d'une part, de définir le composant de l'application de contrôle-commande qui calcule la valeur de cette priorité dynamique et, d'autre part, de dire quand et comment cette valeur est calculée.

C'est le contrôleur, qui est le composant qui a la vue la plus complète du fonctionnement de l'application (signal de différence entre entrée et sortie ; loi de commande et signal de commande), qui nous paraît le mieux à même de fournir le signal qui permet de spécifier les besoins.

Nous proposons ici de définir les besoins à partir du signal de commande  $u$  et plus exactement d'une fonction du signal de commande  $u$ . Nous appelons  $f(u)$  cette fonction que nous définissons dans la section suivante.

On peut ensuite traduire ces besoins en termes de priorité et également définir les instants où cette priorité est réévaluée.

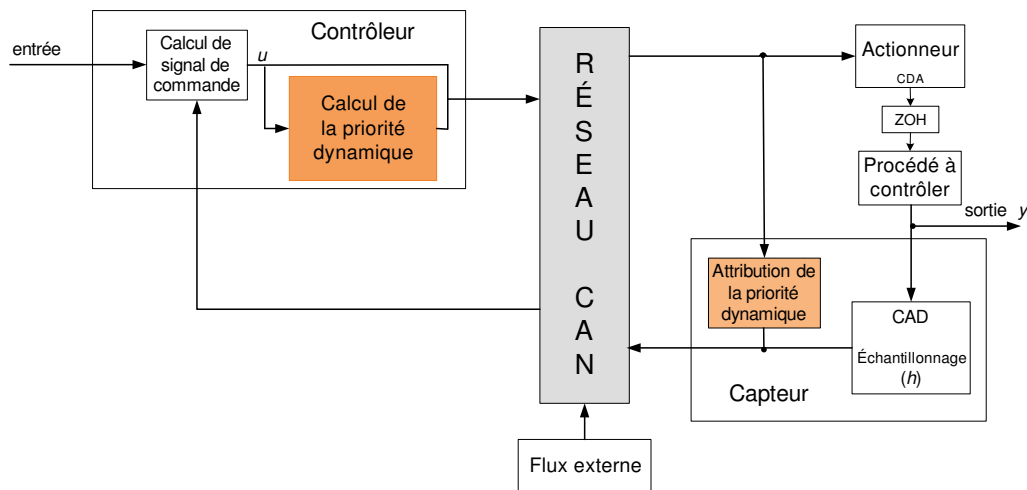


FIGURE 2.4 – Mécanisme de la priorité dynamique.

### Mise en œuvre du mécanisme de la priorité dynamique

Le calcul de la priorité dynamique est effectué par le contrôleur chaque fois qu'il reçoit une trame que le capteur lui envoie après l'échantillonnage de la sortie. Après la réception de la trame du capteur et le calcul de la priorité dynamique, le contrôleur envoie une trame à l'actionneur. Cette trame utilise un ID qui contient la priorité statique utilisé par le flux  $f_{ca}$  et la priorité dynamique calculée. Cette trame atteint tous les sites (CAN est un bus) et comme le site capteur connaît la priorité statique du flux  $f_{ca}$  (c'est une contrainte de notre proposition), il apprend la valeur de la priorité dynamique qu'il utilisera dans la trame qu'il enverra après le prochain échantillonnage. *Donc sur une période comprise entre deux instant d'échantillonnage : le capteur utilise la priorité dynamique évaluée par le contrôleur à la période précédente et le contrôleur utilise la priorité dynamique nouvellement calculée (qui sera utilisée par le capteur à la prochaine période).*

La mise en œuvre du mécanisme de la priorité dynamique (calcul par le contrôleur, utilisation par le capteur) est représentée sur la figure 2.4.

#### Remarque importante :

Au démarrage de l'application (c.-à-d. au premier instant d'échantillonnage), c'est le capteur qui transmet la première trame. Pour cette première trame, le capteur n'a pas d'information sur la priorité dynamique et donc il va faire un choix arbitraire. Nous proposons que le capteur utilise la priorité dynamique maximale. Ainsi la première trame du flux  $f_{cc}$  atteint le contrôleur aussi vite que possible.

## 2.3 Trois schémas de priorité hybride

Nous définissons trois schémas qui découlent de trois modalités différentes de mise en œuvre de la priorité dynamique :

- le schéma appelé priorité hybride (ph) ;
- le schéma appelé priorité hybride + stratégie temporelle statique (ph+sts) ;
- le schéma appelé priorité hybride + stratégie temporelle dynamique (ph+std).

### 2.3.1 Schéma ph

La priorité dynamique est obtenue au moyen d'une fonction croissante de la valeur absolue du signal de contrôle  $u$  et caractérisée par une saturation à une valeur de  $|u|$  appelé  $u_s$  (ici nous choisissons arbitrairement  $u_s = 2/3 \times |u_{\max}|$ ). Nous avons considéré, avant la saturation, trois fonctions  $f(u)$  (Fig. 2.5) : une fonction linéaire  $f_1$  (Equa. 2.1), une fonction parabole  $f_2$  (Equa. 2.2), et une fonction racine carrée  $f_3$  (Equa. 2.3). La fonction  $f_3$  est la meilleure d'un point de vue réactivité (réaction importante c.-à-d. priorité importante dès les petites valeurs de  $u$ ) et est donc la seule considérée ici.

$$f_1(u) = \begin{cases} \left\lceil P_{\max} \times \frac{|u|}{u_s} \right\rceil, & 0 \leq |u| \leq u_s \\ P_{\max}, & |u| > u_s \end{cases} \quad (2.1)$$

$$f_2(u) = \begin{cases} \left\lceil P_{\max} \times \left(\frac{|u|}{u_s}\right)^2 \right\rceil, & 0 \leq |u| \leq u_s \\ P_{\max}, & |u| > u_s \end{cases} \quad (2.2)$$

$$f_3(u) = \begin{cases} \left\lceil P_{\max} \times \sqrt{\frac{|u|}{u_s}} \right\rceil, & 0 \leq |u| \leq u_s \\ P_{\max}, & |u| > u_s \end{cases} \quad (2.3)$$

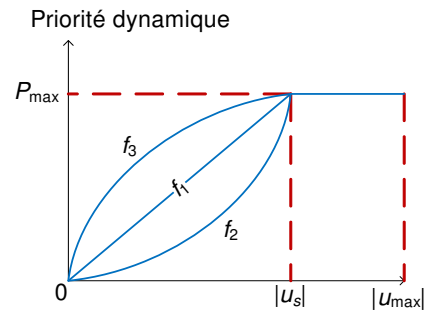


FIGURE 2.5 – fonctions  $f(u)$ .

#### Exemple :

Soit l'application de contrôle-commande présentée dans la section 2.2 et considérons l'implantation de cette application à travers le réseau CAN avec un champ ID de 11 bits structuré de la manière suivante : 4 bits pour la partie statique (donc on peut identifier 16 flux) et 7 bits pour la partie dynamique (donc 128 niveaux de priorités dynamiques). On suppose que l'on a un flux externe tel que  $\text{TRU} = 99.2\%$  et on choisit  $\text{Spr} = 0.5P_{\max}$ .

Les figures 2.6 et 2.7 représentent les évolutions temps réel du signal de contrôle  $u$  et de la priorité dynamique suite à l'action du signal d'entrée. On voit sur ces figures qu'à la période notée  $a$  le signal de commande  $|u| > 2/3u_{\max}$  donc la priorité dynamique est égale à  $P_{\max}$ . Par contre à la période  $c$ ,  $|u| < 2/3u_{\max}$ , on n'a plus la priorité dynamique maximale.

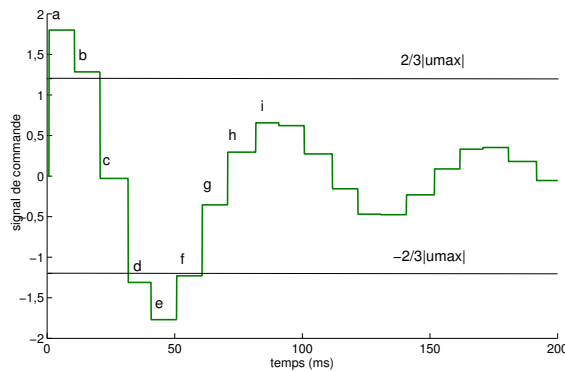


FIGURE 2.6 – ph : Signal de commande  $u$ .

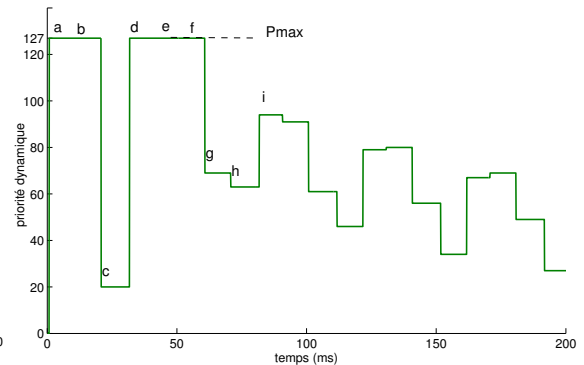


FIGURE 2.7 – ph : Priorité dynamique.

### 2.3.2 Schéma ph + sts

#### Analyse critique du schéma ph

Le schéma ph+sts résulte d'une critique du schéma ph. Considérons les figures 2.6 et 2.7. Dû à un régime transitoire oscillatoire, le signal de contrôle  $u$  peut prendre des valeurs positives ou négatives telles que  $|u|$  est supérieur à  $2/3|u_{\max}|$  (voir les instants  $a$  et  $d$  sur la figure 2.6). Mais entre ces deux instants, le signal de contrôle  $u$  peut prendre des valeurs telles que  $|u| < 2/3|u_{\max}|$  (voir l'instant  $c$ ). Compte tenu de ce comportement du signal  $u$ , la priorité dynamique devient élevée à l'instant  $a$ , prend une valeur intermédiaire à l'instant  $c$  et redevient élevée à l'instant  $d$ . Ce comportement oscillatoire de la priorité dynamique montre que l'action de correction initiale (priorité très élevée) semble inadéquate en terme de durée car après avoir laissé à l'instant  $c$  la priorité élevée pour une priorité intermédiaire, on revient à cette priorité élevée à l'instant  $d$ .

Cette observation suggère d'accroître la durée de la correction initiale (priorité élevée) afin d'améliorer le régime transitoire.

#### La proposition du schéma ph+sts

L'idée est que, contrairement au précédent schéma ph où la priorité dynamique est réévaluée après chaque réception d'une trame du flux  $f_{cc}$  (donc après chaque échantillonnage), la réévaluation n'est pas si étroitement liée aux instants d'échantillonnage. La durée de l'intervalle de temps entre deux réévaluations doit être pertinente du point de vue du régime transitoire de l'application de contrôle-commande avant son implantation à travers le réseau. Nous proposons donc l'algorithme suivant :

- si la priorité dynamique a une valeur entre  $\frac{P_{\max}}{2}$  et  $P_{\max}$ , on garde cette valeur durant 4 périodes d'échantillonnage et on réévalue la priorité dynamique ensuite ; cette durée de 4 périodes d'échantillonnage (40 ms) est de l'ordre du temps de montée ( $t_m = 34$  ms) et représente donc une caractéristique importante du régime transitoire ;
- si la priorité dynamique a une valeur inférieure à  $\frac{P_{\max}}{2}$ , on réévalue après chaque période d'échantillonnage comme dans le schéma ph.

#### Résultats

Le signal de commande  $u$  et la priorité dynamique obtenue en considérant toujours l'exemple traité avec le schéma ph sont représentés sur les figures 2.8 et 2.9. L'aspect oscillatoire a disparu.

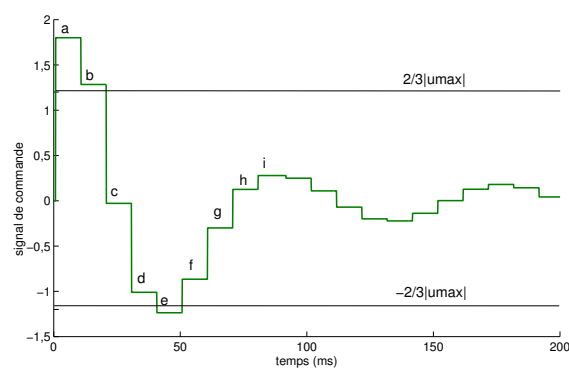


FIGURE 2.8 – ph+sts : Signal de commande  $u$ .

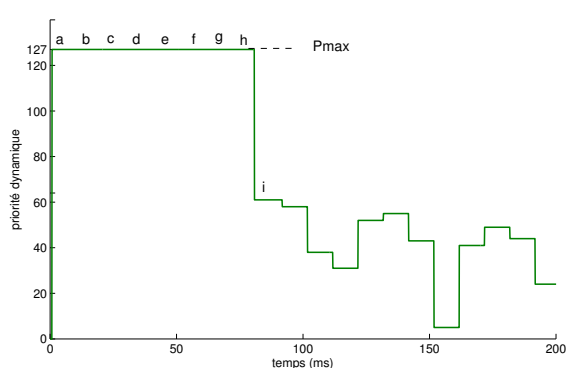


FIGURE 2.9 – ph+sts : Priorité dynamique.

### 2.3.3 Schéma ph+std

Ce schéma résulte d'une critique du schéma ph+sts. En effet, la stratégie temporelle du schéma ph+sts, qui est utilisée dès que l'on constate que la priorité dynamique est supérieure à  $P_{\max}/2$ , est statique (on garde cette priorité pendant 4 périodes d'échantillonnage) et donc on peut garder une valeur élevée de la priorité dynamique plus longtemps que nécessaire.

L'objectif du schéma ph+std est toujours d'éviter le comportement oscillatoire de la priorité dynamique mais en gardant la réévaluation de la priorité dynamique à chaque période (comme pour le schéma hp).

#### Les idées du schéma ph+std

Ce schéma est basé sur deux concepts de base :

- le concept de profil de référence de décroissance des priorités dynamiques à partir d'une priorité  $P_{\max}$  (qui doit contrôler la plus grande urgence qui peut survenir dans un régime transitoire) jusqu'à une priorité  $P_{\min}$  (associée au fonctionnement en régime permanent). La plus grande urgence est obtenue lors de l'application d'un échelon de position à l'entrée de l'application de contrôle-commande ;
- le concept de supervision en ligne des besoins de l'application de contrôle-commande pour piloter le déplacement pendant la durée d'un régime transitoire, sur le profil de référence.

Le profil de référence consiste en une fonction temporelle continue décroissante utilisée par le contrôleur (Equa. 2.4 et Fig. 2.10). Cette fonction est définie sur un domaine temporel  $[t_0 = 0, t = t_{\max}]$  : l'instant  $t_0 = 0$  fixe la priorité maximale  $P_{\max}$  qui est utilisée au début du régime transitoire ; l'instant  $t_{\max}$  fixe la priorité minimale  $P_{\min}$  qui est utilisée à la fin du régime transitoire et donc pendant le régime permanent (à la configuration du système, le contrôleur est positionné au point  $P_{\min}$ ). Notons encore que l'on choisit pour  $t_{\max}$  la valeur du temps de réponse à 5% de l'application de contrôle-commande sans le réseau (c'est l'objectif maximum que l'on peut espérer en termes de durée du régime transitoire puisque la présence du réseau est pénalisante).

$$P(t) = \begin{cases} \left[ P_{\max} + (P_{\min} - P_{\max}) \left( \frac{t}{t_{\max}} \right)^2 \right] & \text{si } 0 \leq t \leq t_{\max} \\ P_{\min} & \text{si } t > t_{\max} \end{cases} \quad (2.4)$$

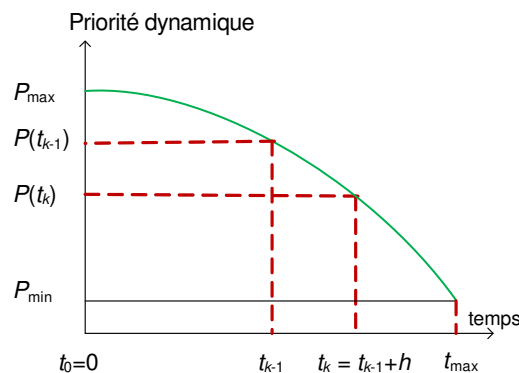


FIGURE 2.10 – Profil de référence.



Le principe d'utilisation du profil de référence par le contrôleur, suite à l'application de l'échelon de position est maintenant décrit :

- à la réception de la trame initiale du flux  $f_{cc}$ , le contrôleur se positionne au point  $(P_{\max}, t_0)$  et transmet une trame du flux  $f_{ca}$  avec la priorité  $P_{\max}$  ;
- suite à la réception des trames suivantes du flux  $f_{cc}$ , les priorités données par le profil de référence et envoyées dans les trames du flux  $f_{ca}$  sont :
  - après la première trame du flux  $f_{cc}$  : priorité donnée par le profil de référence au temps  $t_1 = t_0 + h$  ;
  - après la deuxième trame du flux  $f_{cc}$  : priorité donnée par le profil de référence au temps  $t_2 = t_1 + h$  ;
  - ...
  - après la  $k^e$  trame du flux  $f_{cc}$  : priorité donnée par le profil de référence au temps  $t_k = t_{k-1} + h$ .

L'évolution temporelle de la priorité dynamique pendant le régime transitoire que nous venons de décrire est toujours décroissante (décroissance plus faible au début du régime transitoire que sur la fin ; il faut en effet maintenir des priorités élevées au début car il faut être réactif).

Cependant, si on considère le comportement réel du système étudié, il faut tenir compte de l'influence du réseau (en particulier quand la charge devient importante) et de perturbations qui peuvent se produire pendant le régime transitoire ce qui allonge la durée du régime transitoire par rapport à celui obtenu quand on considère seulement le profil de référence. Plus précisément l'évolution temporelle des priorités dynamiques ne peut pas être toujours décroissante pendant le régime transitoire. Après certains instant  $t_k$ , on peut avoir besoin de priorités dynamiques supérieures et donc il faut se déplacer vers la gauche sur le profil de référence.

Afin d'intégrer ce problème dans la génération des priorités dynamiques, on introduit le concept de supervision en ligne du régime transitoire sur la base d'une observation du signal de contrôle  $u$ . Cette observation est caractérisée par une fonction  $g(u)$  qui permet d'ajuster le positionnement temporel des valeurs des priorités dynamiques.

Ainsi, alors que si on considère seulement le profil de référence on a  $t_k = t_{k-1} + h$ , la fonction  $g(u)$  donne une valeur temporelle qui doit être soustraite à  $t_{k-1} + h$  et donc peut introduire un mouvement vers la gauche sur le profil de référence.

La fonction  $g(u)$  est représentée sur l'équation 2.5 et la figure 2.11.

$$g(u) = \begin{cases} t_{\max} \sqrt{\frac{|u|}{|u_{\max}|}}, & 0 \leq |u| \leq |u_s| \\ t_{\max}, & |u| > |u_s| \end{cases} \quad (2.5)$$

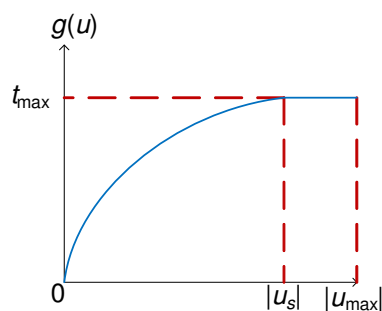


FIGURE 2.11 – Fonction  $g(u)$ .

### L'algorithme de calcul des priorités dynamiques

1. à la réception d'une trame du flux  $f_{cc}$ , le contrôleur évalue  $g(u)$  ;
2. calcul de  $t_k$  :  $t_k = t_{k-1} + h - \alpha g(u)$  avec  $\alpha = \frac{t_{k-1}}{t_{\max}}$ ,  $0 \leq \alpha \leq 1$  ;  
 si  $t_k < 0 \rightarrow t_k = 0$  ;  
 si  $t_k > t_{\max} \rightarrow t_k = t_{\max}$  ;
3. évaluation de  $P(t_k)$ .

La visualisation de l'action de croissance ou de décroissance de la priorité dynamique, entre deux instants d'échantillonnages, est représentée sur les figures 2.12 et 2.13.

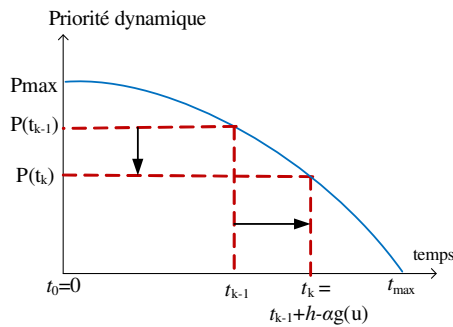


FIGURE 2.12 – Décroissance de priorité  
( $h - \alpha g(u) > 0$ )

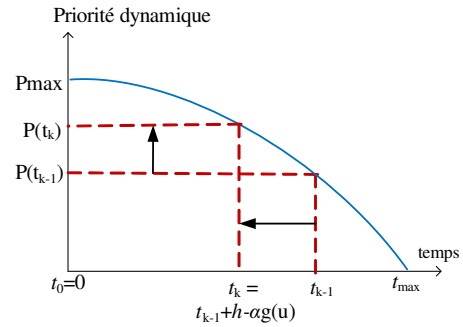


FIGURE 2.13 – Croissance de priorité  
( $h - \alpha g(u) < 0$ )

### Remarques importantes :

1. la considération de la supervision induit (par rapport à la seule considération du profil de référence) un ralentissement de l'évolution temporelle de la priorité dynamique (le ralentissement, qui se produit quand  $h - \alpha g(u) < 0$ , est d'autant plus élevé que la valeur de  $g(u)$  est grande).
2. la considération d'un seuil de priorité  $Spr$  pour un flux à besoin constant accentue encore le ralentissement d'autant plus que  $Spr$  a une valeur élevée : en effet plus  $Spr$  est élevé, plus le signal  $u$  subira un retard élevé amenant des valeurs de  $u$  et donc de  $g(u)$  élevées. La figure 2.14 montre le rôle joué par  $Spr$  dans l'évolution de la priorité dynamique.

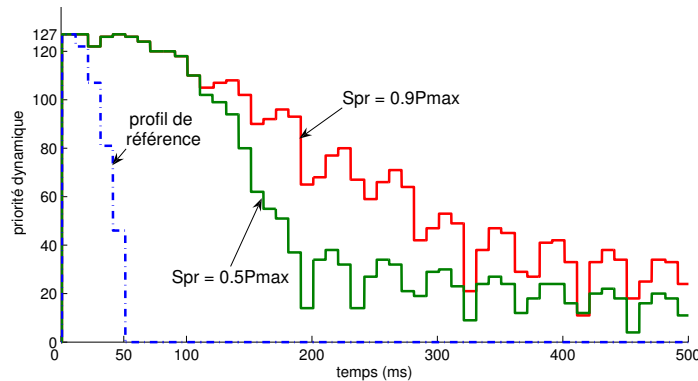


FIGURE 2.14 – ph+std : Priorités dynamiques.

## 2.4 Implantation d'une application de contrôle-commande et d'un flux externe

### 2.4.1 Introduction

L'objectif de cette section est de montrer, d'une part, l'intérêt de ces trois schémas par rapport à l'utilisation de la priorité statique (cas où on a  $P_{ex} > P_{ca} > P_{cc}$  qui a justifié l'étude des priorités hybrides) et, d'autre part, de montrer leurs caractéristiques respectives.

Cette étude est faite sur la base d'éléments définis dans la section 2.2 et que nous rappelons ici : l'application de contrôle-commande (période d'échantillonnage  $h = 10$  ms) est implantée dans un réseau CAN (flux  $f_{cc}$  et  $f_{ca}$  ; durée des trames de chaque flux 0.64 ms) partagé avec un flux externe ( $f_{ex}$ ) périodique (durée de la trame de 0.96 ms) et synchrone avec le flux  $f_{cc}$  (la fréquence de  $f_{ex}$  est un multiple de  $1/h$ , voir le tableau 2.1 - page 35). Le tableau 2.1 donne les période  $T_{ex}$  du flux  $f_{ex}$  pour les multiple de  $1/h$  considérés ainsi que les charges du réseau (TRU%) qui en résultent.

Les conditions de l'étude sur les priorités hybrides sont :

- la partie statique du flux  $f_{ex}$  est supérieure à celle du flux  $f_{ca}$ . La partie statique du flux  $f_{ca}$  est supérieure à celle du flux  $f_{cc}$  ;
- nous considérons trois niveaux de Seuil de priorité Spr : un petit seuil  $0.25P_{max}$  ; un seuil moyen  $0.5P_{max}$  et un grand seuil  $0.9P_{max}$  ;
- le critère d'évaluation de QdC est  $\Delta J/J_0(\%)$  ;
- la QdS est évaluée au moyen du retard moyen et de l'écart type.

Nous voulons mettre l'accent sur deux points qui sont importants pour expliquer les résultats que nous obtiendrons :

- *point 1* : jusqu'à la valeur TRU de 70.4% (valeur de 1.667 ms pour  $T_{ex}$ ) on peut voir que, durant une période  $T_{ex}$  et quelle que soit la priorité des trames, une trame de chaque flux peut accéder au médium. En effet, on a  $0.96 + 0.64 = 1.6$  ms  $<$  1.667 ms ou  $0.64 + 0.64 <$  1.667 ms et donc la trame du troisième flux peut toujours commencer à être transmise et ne peut être interrompue.
- *point 2* : au début du régime transitoire (application de l'échelon de position), comme le signal de commande est à son maximum, la priorité dynamique de l'application de contrôle-commande est  $P_{max}$ .

### 2.4.2 Références à considérer : QdS et QdC avec les priorités statiques

#### a/ Qualité de Service

Le tableau 2.2 donne la QdS (le retard moyen  $\bar{D}$  et l'écart type  $\sigma$ ) en fonction de la charge du réseau (TRU).

**Retard moyen  $\bar{D}$  :**

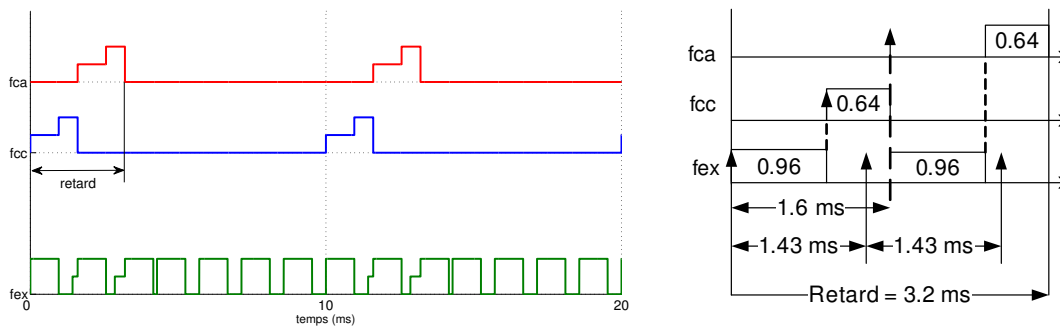
- le retard moyen est toujours de 2.24 ms tant que  $TRU \leq 70.4\%$  (c.-à-d.  $T_{ex} \geq 1.667$  ms). Ceci est logique car, après l'envoi du premier message du flux externe (0.96 ms) et tant que la période du flux externe est supérieure à  $(D_{ex} + D_{cc})$  soit  $0.96 + 0.64 = 1.6$  ms, les 2 trames (flux  $f_{cc}$  et flux  $f_{ca}$ ) peuvent passer ;

| TRU %       | $\bar{D}$ | $\sigma$ |
|-------------|-----------|----------|
| 99.2        | 6.08      | 0        |
| 89.6        | 4.16      | 0        |
| 80.0        | 3.20      | 0        |
| $\leq 70.4$ | 2.24      | 0        |

TABLE 2.2 – priorité statique :  $\bar{D}$  et  $\sigma$ .

- pour  $\text{TRU} > 70.4\%$  c.-à-d. pour une période du flux externe de plus en plus inférieure à 1.6 ms, le retard moyen augmente. L'analyse des échanges des flux de message au moyen de l'outil TrueTime permet de visualiser la croissance du retard en fonction de la charge TRU. La figure 2.15<sup>1</sup> représente le retard pour  $\text{TRU} = 80\%$  : le retard, à chaque période d'échantillonnage, est dû à la durée des messages d'un flux  $f_{ex}$ , d'un flux  $f_{cc}$ , d'un autre flux  $f_{ex}$  et d'un flux  $f_{ca}$  soit, le retard  $\tau = 0.96 + 0.64 + 0.96 + 0.64 = 3.2$  ms.

**Ecart type  $\sigma$**  : l'écart type est toujours nul car on a une reproductibilité de scénario quelle que soit la charge ( $D_i = \bar{D}$  quelle que soit la période d'échantillonnage  $i$ ).

FIGURE 2.15 – priorité statique avec  $\text{TRU} = 80\%$ ,  $T_{ex} = 1.43$  ms (retard = 3.2 ms).

### b/ Qualité de Contrôle

Le tableau 2.3 donne la QdC ( $\frac{\Delta J}{J_0}\%$  et le temps de réponse  $t_r$  (ms)) en fonction de TRU. Nous constatons :

- la QdC est identique tant que  $\text{TRU} \leq 70.4\%$  ;
- pour  $\text{TRU} \geq 70.4\%$ , plus TRU est grand, moins bonne est la QdC ( $\frac{\Delta J}{J_0}\%$  et  $t_r$  augmentent).

| TRU %         | $\frac{\Delta J}{J_0}\%$ | $t_r$ (ms) |
|---------------|--------------------------|------------|
| 99.2          | 492                      | 419        |
| 89.6          | 79.9                     | 189        |
| 80.0          | 37.6                     | 114        |
| $\leq 70.4\%$ | 17.3                     | 108        |

TABLE 2.3 – priorité statique :  $\Delta J/J_0\%$  et  $t_r$  (ms).

1. Sur ces chronogrammes obtenus de TrueTime le niveau haut représente un message en cours d'émission, le niveau intermédiaire représente (suite à une demande d'émission) une attente du médium libre et le niveau bas est une absence de demande d'émission.

### 2.4.3 QdS fournie par les trois schémas de priorité hybride

#### 2.4.3.1 Schéma ph

Le tableau 2.4 qui donne les valeurs du retard moyen  $\bar{D}$  et de l'écart type  $\sigma$  amène les commentaires suivants :

| TRU (%)     | Spr           |          |               |          |                |          |
|-------------|---------------|----------|---------------|----------|----------------|----------|
|             | $0.9P_{\max}$ |          | $0.5P_{\max}$ |          | $0.25P_{\max}$ |          |
|             | $\bar{D}$     | $\sigma$ | $\bar{D}$     | $\sigma$ | $\bar{D}$      | $\sigma$ |
| 99.2        | 5.33          | 1.68     | 3.74          | 2.26     | 1.80           | 1.38     |
| 89.6        | 3.26          | 1.28     | 2.24          | 1.22     | 1.62           | 0.84     |
| 80.0        | 2.48          | 0.88     | 1.97          | 0.82     | 1.54           | 0.59     |
| $\leq 70.4$ | 1.89          | 0.46     | 1.71          | 0.47     | 1.47           | 0.38     |

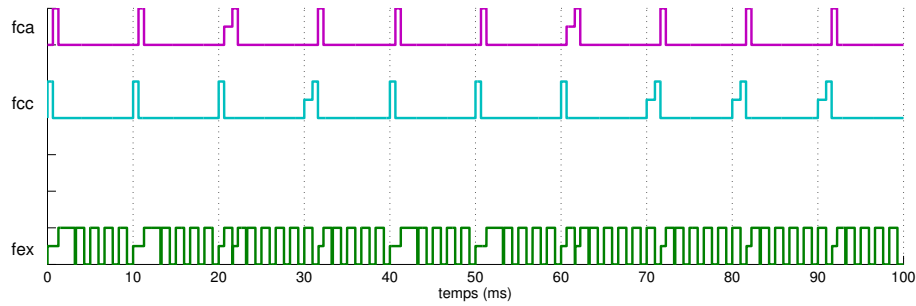
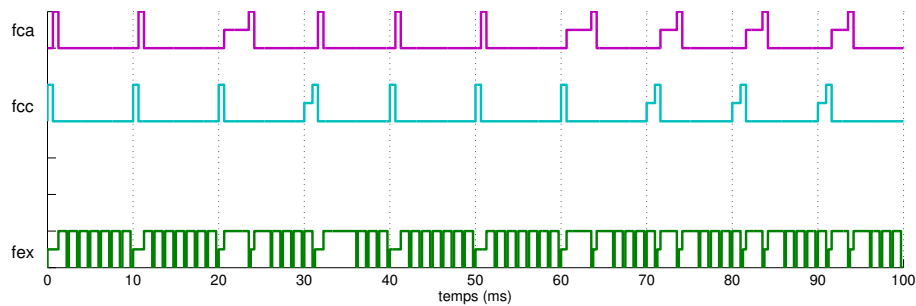
TABLE 2.4 – ph :  $\bar{D}$  et  $\sigma$  (ms).

#### Retard moyen $\bar{D}$ :

- pour chaque valeur de Spr :
  - . si  $\text{TRU} \leq 70.4\%$  on a les mêmes valeurs de  $\bar{D}$  (et de  $\sigma$ ) quelle que soit la valeur de TRU. Ceci est la conséquence du fait que (point 1 signalé dans la sous section 2.4.1) les trames des flux  $f_{cc}$  et  $f_{ca}$  peuvent être émises durant chaque période  $T_{ex}$  (et donc au maximum, une trame du flux  $f_{ex}$  peut retarder la trame du flux  $f_{cc}$  ou du flux  $f_{ca}$ ) ce qui n'est pas le cas avec  $\text{TRU} > 70.4\%$  où  $\bar{D}$  croît avec la valeur de TRU (voir sur le tableau 2.4  $\text{TRU} = 80\%$ ,  $89.6\%$ ,  $99.2\%$ );
  - . la différence entre les conditions  $\text{TRU} \leq 70.4\%$  et  $\text{TRU} > 70.4\%$  est expliquée au moyen des diagrammes temporels fournis par l'outil TrueTime (Figs. 2.16 et 2.17,  $\text{TRU} = 70.4\%$  et  $89.6\%$ ,  $\text{Spr} = 0.9P_{\max}$ ). Sur la figure 2.16, on voit que les trames du flux  $f_{cc}$  ou du flux  $f_{ca}$  peuvent être retardées, durant une période d'échantillonnage, au maximum de la durée d'une trame d'un flux  $f_{ex}$  (0.96 ms). Sur la figure 2.17, on voit que les deux trames des flux  $f_{cc}$  et  $f_{ca}$  peuvent être retardées sur la même période d'échantillonnage et les retards de la trame du flux  $f_{ca}$  peuvent être d'une durée supérieure à la durée d'une trame du flux  $f_{ex}$ ;
  - . notons que quand  $\text{TRU} > 70.4\%$  et pour des valeurs croissantes de TRU,  $\bar{D}$  augmente parce que la charge du réseau croît (donc les trames du flux externe retardent de plus en plus les trames des flux  $f_{cc}$  et  $f_{ca}$ ).
- pour les valeurs croissantes du seuil Spr,  $\bar{D}$  croît aussi parce que les priorités dynamiques des trames  $f_{cc}$  et  $f_{ca}$  ont moins de chance d'être plus élevées que le seuil (excepté au début du régime transitoire).

#### Ecart type $\sigma$ :

- pour chaque valeur de Spr, l'écart type croît avec la charge du réseau  $\text{TRU} \geq 70.4\%$ . Ceci s'explique au moyen des diagrammes temporels fournis par l'outil TrueTime (Fig. 2.16 et Fig. 2.17 :  $\text{TRU} = 70.4\%$  et  $89.6\%$ ,  $\text{Spr} = 0.9P_{\max}$ ). Sur la figure 2.17 ( $\text{TRU} = 89.6\%$ ) on voit que les temps d'attente de trames du flux  $f_{ca}$  (3<sup>e</sup> période et 8<sup>e</sup> période) sont plus élevés que sur la figure 2.16 ( $\text{TRU} = 70.4\%$ ).

FIGURE 2.16 – ph : échange de message (TRU = 70.4%, Spr = 0.9P<sub>max</sub>).FIGURE 2.17 – ph : échange de message (TRU = 89.6%, Spr = 0.9P<sub>max</sub>).

### 2.4.3.2 Schéma ph+sts

Le tableau 2.5 donne le retard moyen et l'écart type.

| TRU<br>(%) | Spr                 |          |                     |          |                      |          |
|------------|---------------------|----------|---------------------|----------|----------------------|----------|
|            | 0.9P <sub>max</sub> |          | 0.5P <sub>max</sub> |          | 0.25P <sub>max</sub> |          |
|            | $\bar{D}$           | $\sigma$ | $\bar{D}$           | $\sigma$ | $\bar{D}$            | $\sigma$ |
| 99.2       | 2.58                | 2.13     | 2.58                | 2.13     | 1.28                 | 0        |
| 89.6       | 1.85                | 1.15     | 1.85                | 1.15     | 1.28                 | 0        |
| 80.0       | 1.66                | 0.76     | 1.66                | 0.76     | 1.28                 | 0        |
| ≤ 70.4     | 1.28                | 0        | 1.28                | 0        | 1.28                 | 0        |

TABLE 2.5 – ph+sts :  $\bar{D}$  et  $\sigma$  (ms).

On peut signaler les différences suivantes avec le schéma ph :

#### Retard moyen $\bar{D}$ :

- pour TRU ≤ 70.4%, le retard est toujours constant et égal à 1.28 ms quelle que soit la valeur du seuil Spr et ce pour deux raisons (le point 1 signalé dans l'introduction de cette section et le fait que maintenant, pendant le régime transitoire, la priorité dynamique maximale est utilisée pour une durée égale au moins à 4h et donc sur toutes ces périodes d'échantillonnage, les trames des flux  $f_{cc}$  et  $f_{ca}$  sont transmises en premier, d'où le retard 0.64 ms + 0.64 ms = 1.28 ms) ;

- pour  $Spr = 0.25P_{max}$ , le retard  $\bar{D}$  est constant et égal à 1.28 ms quel que soit TRU. Ceci résulte du fait que, quelle que soit la charge, la priorité dynamique est supérieure au seuil  $Spr$  ; la figure 2.18 (TRU = 99.2%) qui représente la priorité dynamique et le diagramme temporel explique ce résultat ;
- pour un  $Spr$  donné  $> 0.25P_{max}$ ,  $\bar{D}$  augmente avec TRU pour  $TRU > 70.4\%$  ;
- analyse d'une ligne du tableau 2.5 : quel que soit  $Spr > 0.25P_{max}$ , nous avons la même valeur de  $\bar{D}$  car les priorités dynamiques sont identiques (cf. Fig. 2.19). A partir de cette figure, on peut déduire que pour un TRU donné, on a le même résultat pour tous les  $Spr \in [0.5P_{max}; P_{max}]$ .

#### Écart type $\sigma$ :

- pour  $Spr = 0.25P_{max}$  et pour  $TRU \leq 70.4\%$ , le retard est toujours minimal (1.28 ms), l'écart type est donc nul ;
- pour un  $Spr$  donné  $> 0.25P_{max}$ , l'écart type augmente avec TRU (l'explication est la même que dans le cas du schéma ph).

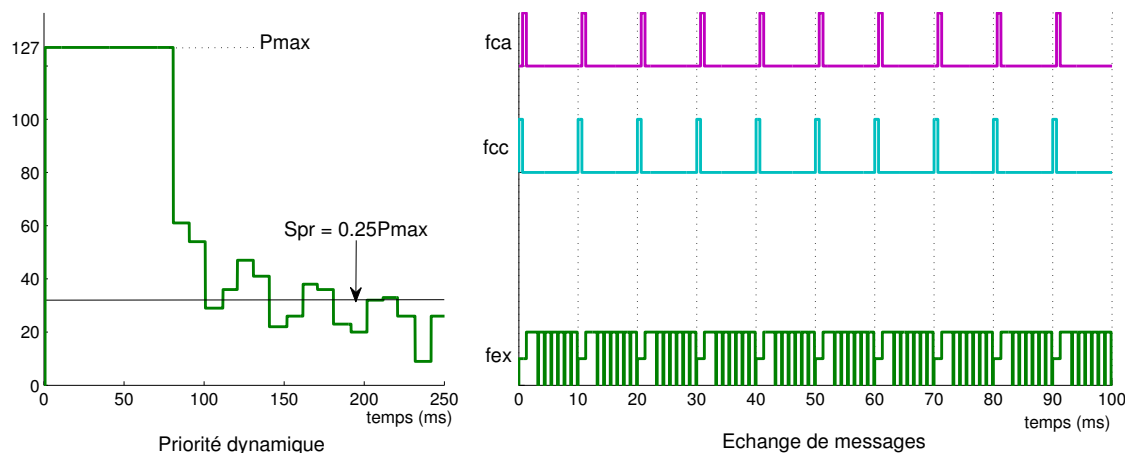


FIGURE 2.18 – ph+sts : TRU = 99.2%,  $Spr = 0.25P_{max}$ .

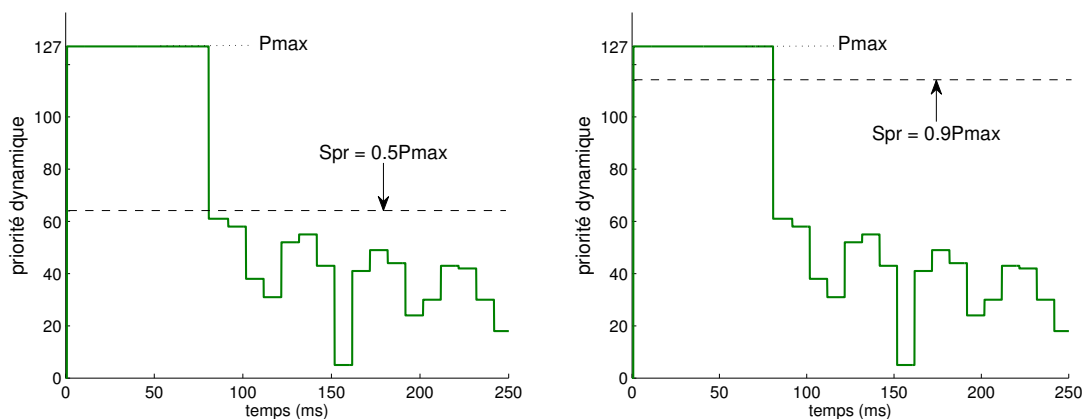


FIGURE 2.19 – ph+sts : priorité dynamique (TRU = 99.2%,  $Spr = 0.5P_{max}$  et  $Spr = 0.9P_{max}$ ).

### 2.4.3.3 Schéma ph+std

Le tableau 2.6 donne le retard moyen et l'écart type. Nous obtenons ici le retard  $\bar{D}$  minimum (durée de la trame  $f_{cc}$  (0.64 ms) + durée de la trame  $f_{ca}$  (0.64 ms)), ce retard est constant et  $\sigma = 0$ . Ceci est la conséquence du fait que la priorité dynamique est contrôlée de manière continue (par le signal de contrôle  $u$ ) et est supérieure au seuil sur une durée supérieure au temps de réponse (Fig. 2.20).

| TRU<br>(%)  | Spr           |          |               |          |                |          |
|-------------|---------------|----------|---------------|----------|----------------|----------|
|             | $0.9P_{\max}$ |          | $0.5P_{\max}$ |          | $0.25P_{\max}$ |          |
|             | $\bar{D}$     | $\sigma$ | $\bar{D}$     | $\sigma$ | $\bar{D}$      | $\sigma$ |
| 99.2        | 1.28          | 0        | 1.28          | 0        | 1.28           | 0        |
| 89.6        | 1.28          | 0        | 1.28          | 0        | 1.28           | 0        |
| 80.0        | 1.28          | 0        | 1.28          | 0        | 1.28           | 0        |
| $\leq 70.4$ | 1.28          | 0        | 1.28          | 0        | 1.28           | 0        |

TABLE 2.6 – ph+std :  $\bar{D}$  et  $\sigma$  (ms).

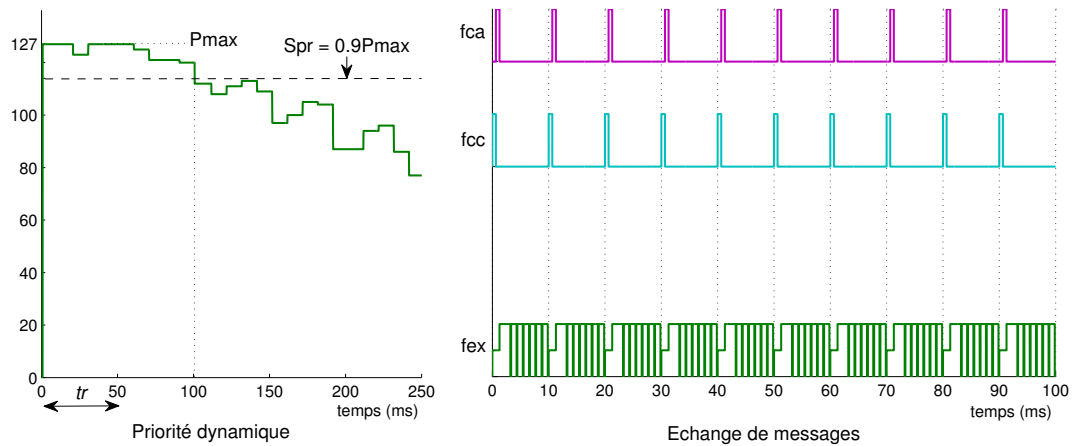


FIGURE 2.20 – ph+std : priorité dynamique (TRU = 99.2%, Spr =  $0.9P_{\max}$ ).

### 2.4.3.4 Conclusion

Pour les différentes stratégies nous avons étudié le retard moyen et l'écart type qui sont liés à la stratégie d'ordonnancement et qui caractérisent la QdS de cette dernière. Cette étude montre l'intérêt de la stratégie de priorité hybride, où l'arbitrage se fait d'abord sur un premier champ de l'identificateur fonction de l'urgence de la transmission, par rapport à une priorité statique. Des trois schémas considérés nous pouvons souligner l'intérêt du schéma ph+std qui a un double aspect : priorité dynamique basée sur une supervision temporelle du signal de contrôle et profil de référence des priorités.

Si l'utilisation de tels schémas de priorité hybride conduit à maîtriser le retard moyen nous allons évaluer dans les mêmes conditions la QdC des applications à partir du critère de performance  $J$  et du temps de réponse à 5%  $t_r$ .



#### 2.4.4 QdC fournie par les différents schémas de priorité hybride

Les performances ( $\Delta J/J_0\%$  et  $t_r$ ) fournies par les trois schémas de priorité sont représentées sur les tableaux 2.7 et 2.8.

| TRU(%) |                      | 99.2% | 89.6% | 80%  | 70.4% |
|--------|----------------------|-------|-------|------|-------|
| ph     | Spr = $0.9P_{\max}$  | 172   | 31.3  | 15.8 | 9.3   |
|        | Spr = $0.5P_{\max}$  | 56.1  | 16.7  | 10.6 | 8.0   |
|        | Spr = $0.25P_{\max}$ | 13.4  | 7.7   | 7.0  | 6.6   |
| ph+sts | Spr = $0.9P_{\max}$  | 16.7  | 8.7   | 7.7  | 7.1   |
|        | Spr = $0.5P_{\max}$  | 16.7  | 8.7   | 7.7  | 7.1   |
|        | Spr = $0.25P_{\max}$ | 9.6   | 7.4   | 7.2  | 6.9   |
| ph+std | Spr = $0.9P_{\max}$  | 14.2  | 8.0   | 7.3  | 6.9   |
|        | Spr = $0.5P_{\max}$  | 6.99  | 6.87  | 6.85 | 6.84  |
|        | Spr = $0.25P_{\max}$ | 6.95  | 6.86  | 6.85 | 6.84  |

TABLE 2.7 – Evaluation de performance  $\Delta J/J_0\%$ .

| TRU(%) |                      | 99.2% | 89.6% | 80% | 70.4% |
|--------|----------------------|-------|-------|-----|-------|
| ph     | Spr = $0.9P_{\max}$  | 359   | 148   | 111 | 107   |
|        | Spr = $0.5P_{\max}$  | 228   | 110   | 108 | 105   |
|        | Spr = $0.25P_{\max}$ | 105   | 106   | 101 | 99    |
| ph+sts | Spr = $0.9P_{\max}$  | 103   | 100   | 98  | 46    |
|        | Spr = $0.5P_{\max}$  | 103   | 100   | 98  | 46    |
|        | Spr = $0.25P_{\max}$ | 46    | 46    | 46  | 46    |
| ph+std | Spr = $0.9P_{\max}$  | 46    | 46    | 46  | 46    |
|        | Spr = $0.5P_{\max}$  | 46    | 46    | 46  | 46    |
|        | Spr = $0.25P_{\max}$ | 46    | 46    | 46  | 46    |

TABLE 2.8 – Temps de réponse  $t_r$  (ms).

Nous avons les résultats suivants pour chaque schéma de priorité :

- pour un Spr donné, plus TRU est grand, moins bonne est la QdC (considération des lignes sur le tableau 2.7) ;
- pour un TRU donné, plus Spr est grand, moins bonne est la QdC (considération des colonnes sur le tableau 2.7).

Les résultats obtenus sont intuitivement logiques car quand TRU et Spr augmentent, le flux externe perturbe davantage les flux  $f_{cc}$  et  $f_{ca}$ , et les retards deviennent donc plus élevés (c.-à-d. mauvaise QdS, voir la section 2.4.3) et donc la QdC est plus mauvaise.

En ce qui concerne la comparaison des trois schémas de priorité avec le schéma statique, d'une part, et entre les schémas hybrides, d'autre part (voir les tableaux 2.3, 2.7 et 2.8), on trouve :

- le schéma statique est moins bon que les schémas de priorité hybride car pour le schéma statique, on est toujours retardé par le flux externe par contre pour les schémas de priorité hybride, grâce à la priorité dynamique, les flux  $f_{cc}$  et  $f_{ca}$  peuvent avoir des priorités élevées et donc peuvent passer rapidement ;
- parmi les priorités hybrides, ph+sts est meilleur que ph et ph+std est meilleur que ph+sts. Les schémas ph+sts et ph+std fonctionnent très bien malgré la charge élevée du réseau et le grand seuil (TRU = 99.2% et Spr  $0.9P_{max}$  : les  $\Delta J/J_0\%$  (Tab. 2.7) de ph+sts et ph+std sont respectivement 16.7% et 14.2% par contre celui de ph est 172%). L'explication est faite par le principe de calcul de la priorité dynamique : ph+std permet de maintenir la priorité dynamique élevée plus longtemps que ph+sts et ph+sts la maintient plus longtemps que ph (la figure 2.21 représente les priorités dynamiques de trois schémas de priorité hybride avec TRU = 99.2% et Spr =  $0.9P_{max}$ ) ; les retards de ph+sts et de ph+std (et donc les temps de réponses) sont plus faibles que ceux de ph ;
- nous représentons sur la figure 2.22 les réponses temporelles  $y(t)$  de quatre schémas de priorité pour TRU = 99.2% et Spr =  $0.9P_{max}$  (cas des priorités hybrides). Il est à noter que les réponses de ph+sts et de ph+std sont presque identiques, en effet les  $\Delta J/J_0\%$  sont juste un petit peu différents (16.7% pour ph+sts et de 14.2% pour ph+std).

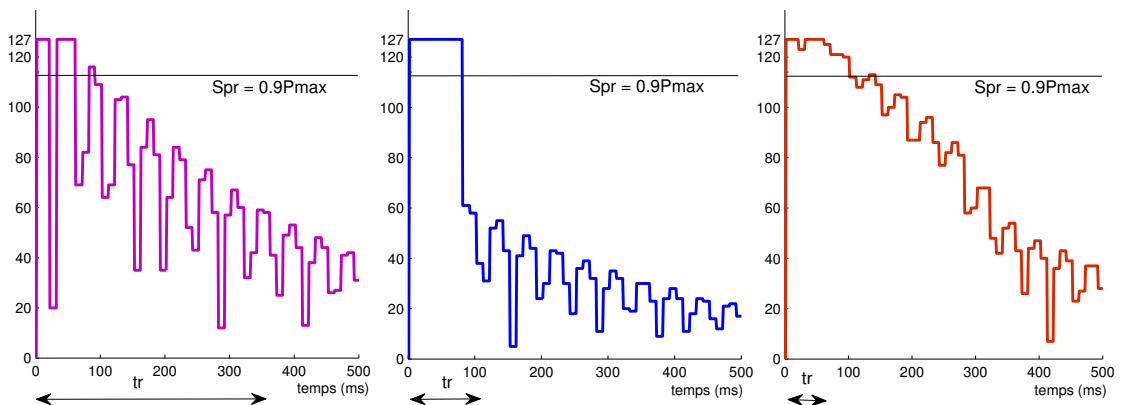


FIGURE 2.21 – Priorités dynamiques : ph, ph+sts, ph+std (TRU = 99.2% et Spr =  $0.9P_{max}$ ).

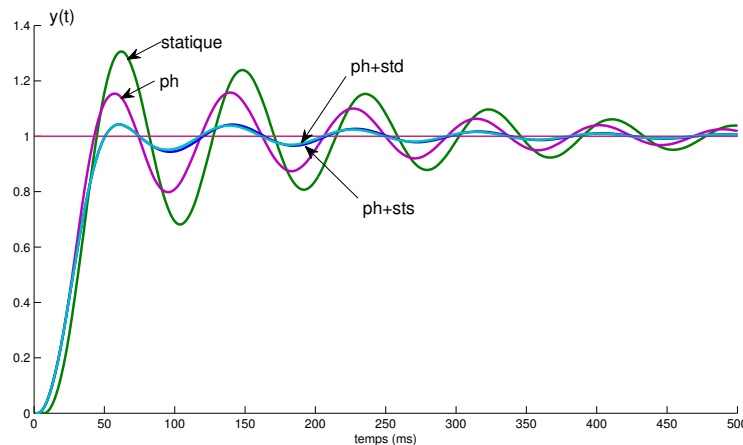


FIGURE 2.22 – Réponses temporelles : TRU = 99.2% et Spr =  $0.9P_{max}$ .

## 2.5 Implantation de plusieurs applications de contrôle-commande

Nous considérons ici l'implantation de plusieurs applications identiques (servo-moteur étudié précédemment) et comparons la QdC évaluée à partir du critère  $J$  et résultant des différents schémas de priorité.

La période d'échantillonnage est la même pour tous les procédés ( $h = 10$  ms); les flux  $f_{cc}$  sont synchrones. Notons  $P_i$  le procédé  $i$ , nous choisissons une priorité statique pour  $P_i$  plus élevée que pour  $P_j$  si  $i < j$  ( $P_1$  a la plus grande priorité statique). Les résultats en terme de dégradation de performances ( $\Delta J/J_0\%$ ) sont présentés sur les tableaux 2.9 et 2.10 pour respectivement 4 et 5 procédés et pour les différentes stratégies d'ordonnancement (statique, ph, ph+sts, ph+std).

|                | statique | ph   | ph+sts | ph+std |
|----------------|----------|------|--------|--------|
| P <sub>1</sub> | 6.83     | 43.8 | 10.1   | 38.1   |
| P <sub>2</sub> | 22.6     | 53.5 | 27.2   | 52.1   |
| P <sub>3</sub> | 61.9     | 62.2 | 56.0   | 59.5   |
| P <sub>4</sub> | 181      | 71.9 | 101    | 81.0   |

TABLE 2.9 – 4 procédés :  $\Delta J/J_0\%$ .

|                | statique | ph  | ph+sts | ph+std |
|----------------|----------|-----|--------|--------|
| P <sub>1</sub> | 6.83     | 111 | 14.9   | 72.9   |
| P <sub>2</sub> | 22.6     | 127 | 35.4   | 90.0   |
| P <sub>3</sub> | 61.9     | 138 | 68.3   | 97.5   |
| P <sub>4</sub> | 181      | 128 | 115    | 145    |
| P <sub>5</sub> | 724      | 125 | 189    | 125    |

TABLE 2.10 – 5 procédés :  $\Delta J/J_0\%$ .

Ces résultats sont traduits graphiquement sur les figures 2.23 et 2.24, ce qui nous permet de visualiser et de caractériser la différence de QdC résultant des différents schémas.

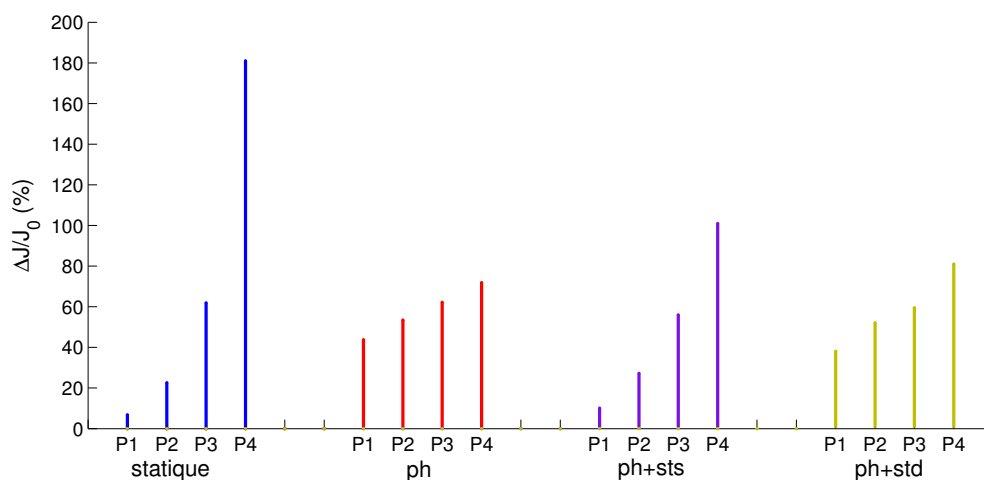


FIGURE 2.23 – 4 procédés :  $\Delta J/J_0$  (%) pour différents schémas de priorité.

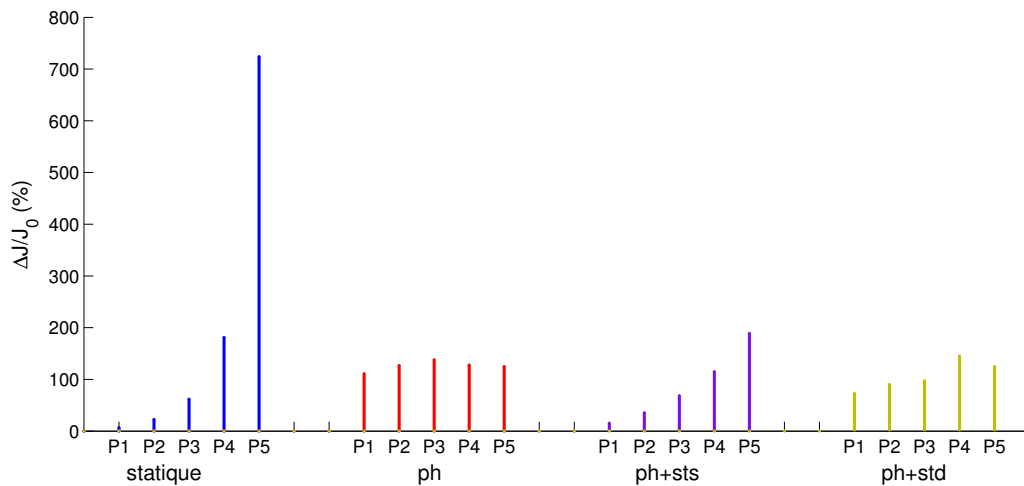


FIGURE 2.24 – 5 procédés :  $\Delta J/J_0$  (%) pour différents schémas de priorité.

### Priorité statique

La dégradation des performance est directement liée à l'ordre des priorités ce qui est logique :  $prio(P_i) > prio(P_j) \Rightarrow \Delta J_i/J_0 < \Delta J_j/J_0$ . La plus grande priorité obtient le meilleur résultat qui est très différent par rapport à celui obtenu par la plus petite priorité ; on a  $\frac{\Delta J}{J_0} \max \approx 25 \frac{\Delta J}{J_0} \min$  pour 4 procédés et  $\frac{\Delta J}{J_0} \max \approx 100 \frac{\Delta J}{J_0} \min$  pour 5 procédés ;

### Priorités hybrides

D'une manière générale, elles atténuent les différences entre les extrêmes (ceci découle de la priorité dynamique qui est très grande au début du régime transitoire quel que soit le procédé). Cette atténuation est d'autant plus marquée que le nombre de procédés est élevé (comparer 4 et 5 procédés).

- *schéma ph* : ce schéma est celui qui amène la répartition la plus égalitaire de la ressource réseau entre les différents procédés (ceci est d'autant plus marqué que le nombre de procédés est élevé) ; ceci résulte du rôle prépondérant de la priorité dynamique ;
- *schéma ph+sts* : ce schéma est celui qui montre une tendance comportementale la plus semblable à celle des priorités statiques. Ceci résulte du fait qu'à l'initialisation, comme les priorités dynamiques sont maximales, le procédé qui a la plus grande partie priorité statique des priorités hybrides garde la priorité dynamique maximale pendant  $4h$  ; donc on retrouve un ordre des performances qui suit l'ordre des parties priorités statiques des priorités hybrides ; le rapport des performances (max/min) est grand mais atténué par rapport au cas des priorités statiques ;
- *schéma ph+std* : ce schéma se situe entre le schéma ph et le schéma ph+sts ; ceci résulte d'une contrainte moins forte, concernant la priorité dynamique, que celle exercée par le schéma ph+sts (les priorités dynamiques suivent le profil de référence et ne sont pas contraintes au début à être  $P_{\max}$  pendant 4 périodes d'échantillonnage).

## 2.6 Commande d'un drone à quatre rotors

### 2.6.1 Le système étudié

Le projet Safe-NeCS (Conception coordonnée de systèmes contrôlés en réseau tolérants aux défauts)<sup>2</sup> a porté sur la co-conception commande et ordonnancement des tâches et des messages et sur les méthodes de diagnostic des SCR. Une plate-forme expérimentale constituée d'un drone à quatre rotors (quadrotor) a permis de valider les approches proposées. Cette validation pouvant se faire à plusieurs niveaux : simulation, « hardware in the loop » (dialogue par bus CAN entre le simulateur et la carte où sont implantés les algorithmes de commande, observation et de diagnostic) et enfin la maquette expérimentale. Des résultats relatifs à la commande et au diagnostic du quadrotor sont donnés dans [Aubrun 2010] et en particulier dans un chapitre consacré à l'implantation [Berbra 2010].

Ce minidrone a été étudié au GIPSA-lab, le modèle et la loi de commande sont présentés dans [Guerrero Castellanos 2008], [Guerrero Castellanos 2011]. Une version distribuée autour d'un réseau CAN a été développée et nos partenaires du GIPSA-lab participant au projet ont fourni une version du simulateur réalisé à partir de TrueTime. Le réseau CAN assure la communication du capteur au contrôleur et du contrôleur à l'actionneur. Le module capteur regroupe des données issues d'une centrale d'attitude et fournit trois données issues d'un magnétomètre, trois données issues d'un accéléromètre et trois données issues d'un gyromètre. La mesure des vitesses de chaque moteur est utilisée pour une boucle locale de commande. Le module de commande est constitué d'un observateur qui reconstitue l'attitude du quadrotor, et d'un contrôleur qui, en fonction de l'attitude et de la référence, fournit pour chaque moteur une vitesse angulaire de référence. L'architecture de ce SCR est représentée sur la figure 2.25

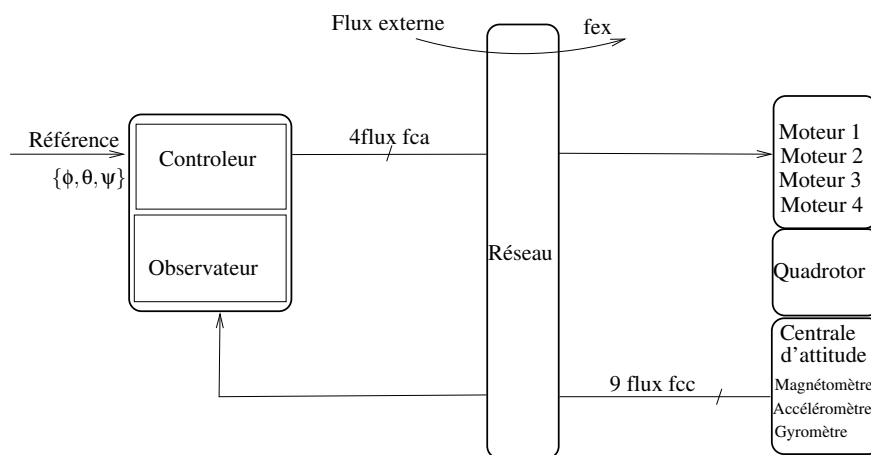


FIGURE 2.25 – Commande en réseau du quadrotor.

Nous avons fait une étude en simulation de la commande du drone et nous avons comparé la QdC découlant de l'application de différentes techniques d'ordonnancement des messages sur le réseau CAN : priorité statique, priorités hybrides ph et ph+std.

2. Ce projet ANR a été présenté dans la section 1.3.

### 2.6.2 Implantation sur CAN

Le problème que nous nous posons est de ramener le drone à une position d'équilibre telle que les trois angles, roulis (roll, noté  $\phi$ ), tangage (pitch, noté  $\theta$ ) et lacet (yaw, noté  $\psi$ ) soient nuls en partant d'une position initiale  $\{\phi_0, \theta_0, \psi_0\}$ . La condition initiale choisie pour nos simulations est  $\{\phi_0, \theta_0, \psi_0\} = \{-25^\circ, -35^\circ, -10^\circ\}$ .

Pour étudier l'influence du réseau les réponses temporelles obtenues avec le réseau seront comparées avec les réponses temporelles obtenues sans réseau. La figure 2.26 représente cette réponse temporelle de référence. Les dépassements obtenus sont respectivement de  $\{20\%, 15\%, 50\%\}$  pour les différents angles. Le temps de réponse à 5% est de l'ordre de 2 s, le temps de montée  $t_m \simeq 500$  ms. Si l'on recherche la période d'échantillonnage à partir de la règle [Åström 1997] :  $4 < t_m/h < 10$ , la période maximum admissible est de l'ordre de 100 ms.

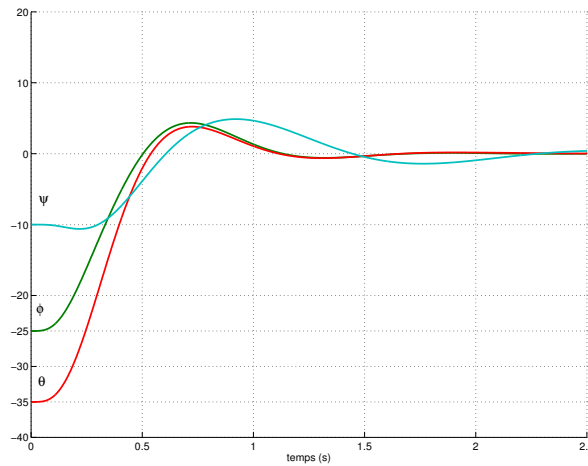


FIGURE 2.26 – Angles d'Euler, commande sans réseau

Dans le cadre de la mise en œuvre sur CAN, l'application de contrôle-commande est caractérisée par 9 flux capteur-contrôleur  $f_{cc}$  correspondant aux données issues de la centrale d'attitude, 4 flux contrôleur-actionneur  $f_{ca}$ , correspondant aux 4 commandes en vitesse des moteurs. Un flux externe  $f_{ex}$  permettant de charger le réseau et de perturber l'ordonnancement est également introduit.

Les hypothèses suivantes sont considérées :

- les envois des capteurs et du contrôleurs sont guidés par le temps (horloge d'échantillonnage) ; le contrôleur transmet les commandes calculées, pendant une période, suite à la réception des trames des flux  $f_{cc}$ , à la période suivante ; l'actionneur est guidé par les événements (c.-à-d. les actions sont exécutées dès la réception des trames du flux  $f_{ca}$ ) ;
- le débit physique du réseau CAN est de 500 Kbits/s ;
- les trames des flux  $f_{cc}$  et  $f_{ca}$  sont de longueur 75 bits et les trames du flux externe sont de longueur 135 bits, les durées d'émission des trames sont donc de  $D_{cc} = D_{ca} = 150 \mu\text{s}$ ,  $D_{ex} = 270 \mu\text{s}$  ;
- la priorité statique du flux externe est toujours supérieure aux priorités des flux  $f_{cc}$  et  $f_{ca}$  ;
- les priorités des flux  $f_{ca}$  sont supérieures aux priorités des flux  $f_{cc}$ , dans le cas contraire

les retards capteur-actionneur sont plus importants et nous avons pu vérifier par simulation que les réponses temporelles sont plus dégradées (voire, que sous certaines conditions de charge du réseau, le système devient instable pour une période  $h = 100$  ms).

Les différents schémas de priorité considérés sont la priorité statique, la priorité hybride ph et la priorité hybride avec stratégie temporelle dynamique (ph+std). Pour calculer la priorité dynamique nous nous appuyons sur le signal de commande  $u$ . Ici 4 signaux de commande  $u_1, u_2, u_3, u_4$  sont calculés et envoyés vers les 4 moteurs. Nous choisissons de calculer la priorité à partir de la moyenne de ces 4 signaux de commande et de la moyenne de  $u_{\max}$  obtenu pour la condition initiale  $\{\phi_0, \theta_0, \psi_0\} = \{-25^\circ, -35^\circ, -10^\circ\}$  degrés.

Différentes simulations ont été faites pour différentes périodes d'échantillonnage allant du sur-échantillonnage (2 ms, 10 ms) jusqu'à des valeurs plus critiques (100 ms) et pour différentes charges de réseau, du réseau dédié au réseau partagé (TRU = 100%). Pour  $h = 100$  ms (4 échantillons dans le temps de montée) et une charge de réseau de 100% le schéma statique conduit à une réponse dégradée (fort dépassement, temps de réponse prohibitif), nettement améliorée par le schéma ph+std (et ce même avec un seuil pour le flux externe élevé  $\text{Spr} = 90\%P_{\max}$ ). Nous choisissons de montrer ici des résultats pour  $h = 50$  ms (10 échantillons dans le temps de montée) et une charge de réseau de 100%. Le flux externe a un seuil élevé  $\text{Spr} = 90\%P_{\max}$ .

L'analyse des réponses temporelles obtenues pour le schéma de priorité statique (Fig. 2.27), les schémas de priorité hybride ph (Fig. 2.28) et ph+std (Fig. 2.29) montrent l'efficacité de ces dernières stratégies et plus particulièrement de ph+std ; cette amélioration est quantifiable, en particulier, en terme de dépassement.

La mise en œuvre de la priorité hybride est justifiée dans le cas où la commande est critique (4 à 10 échantillons dans le temps de montée) alors qu'elle apporte moins lorsque le système est sur-échantillonné. En effet dans ce cas ( $h = 2$  ms ou  $h = 10$  ms par exemple) les retards et les pertes sont tolérées par le système alors que pour une période plus critique ( $h = 100$  ms) le système est plus sensible aux retards.

**Remarque.** Dans ce travail nous avons montré que le caractère d'urgence d'une transmission peut être traduit en terme de priorité dynamique afin de mobiliser la ressource réseau pour une application critique (pendant un régime transitoire par exemple) et d'améliorer la QdC.

Un autre type d'étude est possible qui consisterait à laisser plus d'espace sur le bus, en faisant varier dynamiquement la période d'échantillonnage<sup>3</sup>, pour des applications moins prioritaires (applications émettant le flux noté externe, par exemple) lorsque la situation de l'application n'est pas critique (régime permanent atteint). Ce type d'étude sortant du cadre des priorités hybrides ne sera pas présenté ici.

---

3. L'intérêt de travailler à période dynamiquement variable est d'utiliser le bus prioritairement pour la commande quand c'est nécessaire (période d'échantillonnage faible) et de libérer de l'espace sur le bus, pour d'autres flux tel le flux externe par exemple, quand la commande est moins urgente (régime permanent). Compte-tenu de la loi de commande la période peut varier de 2 ms (période la plus faible compte-tenu du temps de transfert des messages) à 100 ms (valeur maximale admissible). Nous proposons donc de faire varier dynamiquement la période : si une urgence est détectée à partir du signal de commande ( $|u|$  important) la période sera faible, dans le cas du régime établi ( $|u|$  faible) on peut échantillonner à la valeur maximale admissible. Sur une durée correspondant au temps de réponse, on peut évaluer l'espace occupé par le flux externe (en mesurant leur nombre et leur durée cumulée). Le taux d'occupation du bus par le flux externe passe de 80% pour une période de 10 ms à 96% pour une période variant de 2 ms à 100 ms et ce pour des QdC du même ordre.

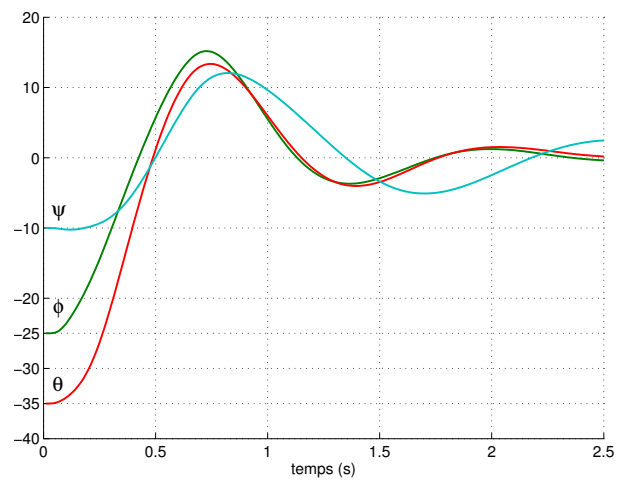


FIGURE 2.27 – Angles d'Euler , priorité statique

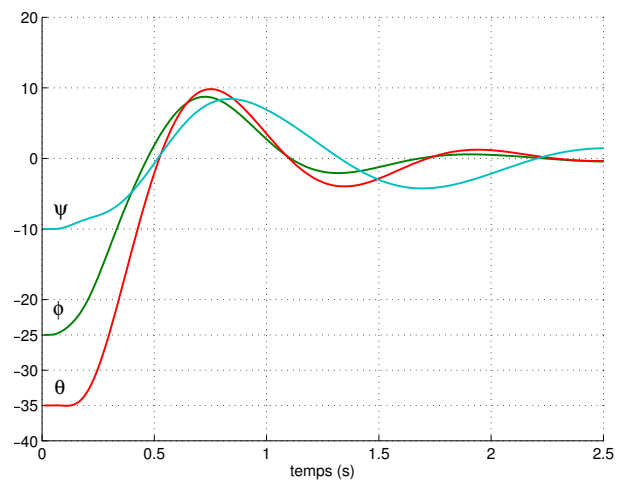


FIGURE 2.28 – Angles d'Euler, schéma ph

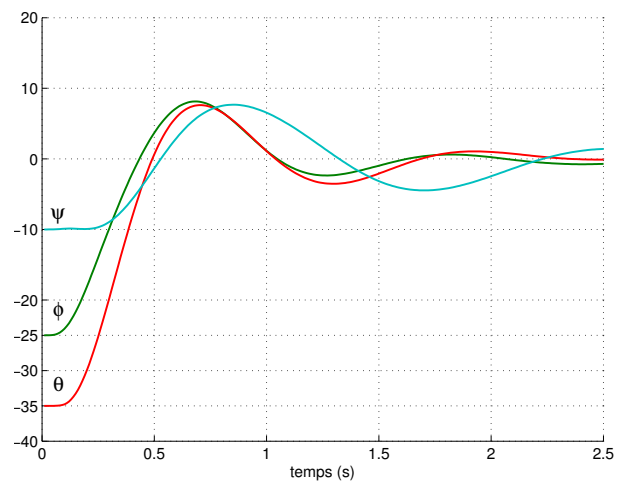


FIGURE 2.29 – Angles d'Euler, schéma ph+std



## 2.7 Conclusion

Ce chapitre a montré l'intérêt du concept de priorité hybride, par rapport aux priorités statiques, pour faire l'ordonnancement des messages des flux des applications de commande distribuées (c.-à-d. implantées à travers un réseau de communication).

Avec le concept de priorité statique, une priorité (a priori) est associée à un flux et tous les messages de ce flux ont la même priorité. On a défini, pour les messages des flux, des priorités hybrides qui sont des priorités comportant deux composantes : une composante « priorité statique » qui représente la priorité du flux auquel le message appartient et une composante « priorité dynamique » qui représente l'urgence de transmission d'un message d'un flux c.-à-d. qui est basée sur la connaissance des besoins de l'application (« Application aware »). Cette notion de priorité dynamique permet de bien assurer le déroulement de la phase d'un régime transitoire d'une application de contrôle-commande, phase durant laquelle les besoins et donc les urgences varient.

Trois schémas de priorités hybrides (différents dans la mise en œuvre de la priorité dynamique (c.-à-d. son calcul et sa réévaluation) ont été proposés et analysés :

- le schéma ph qui est le schéma de base ; le contrôleur calcule sur la base d'une fonction du signal de commande  $u$  (fonction  $f(u)$ ) la priorité dynamique chaque fois qu'il reçoit une donnée capteur (donc réévaluation à chaque période d'échantillonnage) ; ce schéma fait apparaître un comportement oscillatoire de la priorité dynamique ce qui a suggéré la définition des autres schémas ;
- les schémas ph+sts et ph+std introduisent des contraintes par rapport au schéma ph. En ce qui concerne le schéma ph+sts, dès qu'il trouve une priorité dynamique supérieure à  $P_{\max}/2$ , le contrôleur ne réévalue qu'au bout de 4 périodes d'échantillonnage ; si la priorité dynamique est inférieure à  $P_{\max}/2$ , il la réévalue comme dans le schéma ph. En ce qui concerne le schéma ph+std, il est totalement différent du schéma ph en ce sens qu'il définit un profil de référence (pour les priorités dynamiques) à utiliser et une fonction de supervision temporelle basée sur une fonction du signal de commande  $u$  (fonction  $g(u)$ ) qui permet de contrôler l'évolution sur le profil de référence.

Les schémas ph+sts et ph+std fournissent d'excellents résultats lorsqu'on a une seule application de contrôle-commande et un flux externe (qui représente une abstraction d'applications avec des urgences constantes). Le schéma ph est par contre le schéma qui, lorsqu'on a plusieurs applications de contrôle-commande, procure la répartition la plus égalitaire de la ressource réseau entre les différentes applications (le rôle de la partie « priorité dynamique » atténue le rôle de la partie « priorité statique », ce qui est moins marqué pour ph+sts et ph+std). L'application « Commande d'un drone à quatre rotors », réalisée avec les schémas ph et ph+std, confirme ces résultats et l'intérêt de la priorité hybride.

Notons encore que ce concept de priorité hybride a été utilisé dans le cadre de la problématique du diagnostic dans les SCR et cette technique a montré son intérêt, par rapport à des priorités statiques, pour éliminer les fausses alarmes [Juanole 2010].

Notons enfin que quelques aspects relatifs à la notion de priorité dynamique restent des problèmes ouverts à des discussions :

- le premier aspect concerne la caractéristique de l'application de contrôle-commande à utiliser pour représenter l'urgence de transmission ; nous avons considéré le signal de commande  $u$  mais d'autres travaux [Walsh 2001], [Yepez 2003] ont utilisé le signal d'erreur  $e = r - y$  ; bien que nous ayons vérifié que, dans le cas de la fonctions de transfert considérée, nous obtenons de meilleurs résultats en considérant le signal  $u$  qu'en considérant le signal  $e$ , il faudrait faire une étude plus complète tenant compte de différentes fonctions de transfert et aussi de différents contrôleurs ;
- les autres aspects concernent le choix de la saturation  $u_s$  du signal de commande  $u$ , le choix dans le schéma ph+sts du nombre de périodes d'échantillonnage où on ne réévalue pas la priorité dynamique dès qu'on la trouve supérieure à  $P_{\max}/2$ , le choix dans le schéma ph+std de la durée du profil de référence ; nous avons fait dans le travail présenté des choix justifiés mais arbitraires qu'il faudrait critiquer et analyser toujours en fonction de différentes fonctions de transfert et de différents contrôleurs.



## Réseau CAN :

Mise en œuvre de la relation  $QdS \rightleftharpoons QdC$ 

---

**Sommaire**

---

|            |   |           |
|------------|---|-----------|
| <b>3.1</b> | <b>Introduction</b> . . . . .   | <b>62</b> |
| <b>3.2</b> | <b>Contexte</b> . . . . .   | <b>62</b> |
| 3.2.1      | Le type d'application de contrôle-commande considéré . . . . .                        | 62        |
| 3.2.2      | Implantations considérées sur un réseau CAN . . . . .                                 | 64        |
| 3.2.3      | Modèle d'une application de contrôle-commande implantée sur un réseau . . . . .       | 64        |
| 3.2.4      | Critère d'évaluation de performance d'une implantation . . . . .                      | 65        |
| <b>3.3</b> | <b>Mise en œuvre de la relation <math>QdS \rightarrow QdC</math></b> . . . . .        | <b>65</b> |
| 3.3.1      | Idées principales . . . . .   | 65        |
| 3.3.2      | Méthode des pôles dominants . . . . .   | 66        |
| 3.3.3      | Calculs inhérents à la méthode des pôles dominants . . . . .                          | 66        |
| 3.3.4      | Calcul de $u_k$ . . . . .   | 69        |
| 3.3.5      | Validation de la méthode des pôles dominants . . . . .                                | 69        |
| 3.3.6      | Considération des différentes implantations sur le réseau CAN . . . . .               | 71        |
| <b>3.4</b> | <b>Mise en œuvre de la relation <math>QdS \rightleftharpoons QdC</math></b> . . . . . | <b>74</b> |
| 3.4.1      | Idée de base . . . . .  | 74        |
| 3.4.2      | Implantation de la relation $QdS \rightleftharpoons QdC$ . . . . .                    | 75        |
| 3.4.3      | Evaluation de performance . . . . .   | 76        |
| <b>3.5</b> | <b>Récapitulation des résultats obtenus</b> . . . . .                                 | <b>76</b> |
| <b>3.6</b> | <b>Conclusion</b> . . . . .   | <b>79</b> |

---

### 3.1 Introduction

L'objectif de ce chapitre est de montrer, toujours dans le contexte de l'implantation d'applications de contrôle-commande à travers un réseau CAN, l'intérêt de la mise en œuvre conjointe de deux adaptations dynamiques :

- *adaptation 1* : l'ordonnancement des messages sur la base de priorités hybrides (relation  $QdC \rightarrow QdS$  présentée au chapitre 2). Nous considérons ici le cas de la priorité dynamique  $ph$  que nous avons définie au chapitre 2 (c'est la technique de priorité dynamique qui procure le comportement le plus égalitaire de toutes les applications de contrôle-commande) ;
- *adaptation 2* : le calcul en ligne, à chaque période d'échantillonnage, des paramètres du contrôleur sur la base de la connaissance du retard dans la boucle des applications (relation  $QdS \rightarrow QdC$  présentée dans ce chapitre). Le principe de cette adaptation a été présenté dans [Nguyen Xuan 2011a].

Ce chapitre comprend quatre sections : la première section présente le contexte de l'étude ; la deuxième section présente la mise en œuvre de l'adaptation 2 ( $QdS \rightarrow QdC$ ) en considérant des priorités statiques pour l'ordonnancement des messages ; la troisième section présente la mise en œuvre conjointe des deux adaptations (relation  $QdC \rightarrow QdS$ , basée sur l'utilisation de priorités hybrides et relation  $QdS \rightarrow QdC$ ), c'est à dire la mise en œuvre de la relation bidirectionnelle  $QdS \rightleftharpoons QdC$ , et effectue une analyse comparative des performances obtenues par rapport aux cas où on utilise seulement l'adaptation 1 ( $QdC \rightarrow QdS$ ) ou l'adaptation 2 ( $QdS \rightarrow QdC$ ) en considérant les priorités statiques ; la quatrième section présente un résumé des principaux résultats.

L'étude, concernée par ce chapitre, a été présentée dans [Nguyen Xuan 2012]

### 3.2 Contexte

#### 3.2.1 Le type d'application de contrôle-commande considéré

C'est le type d'application de contrôle-commande continue qui est représenté sur la figure 3.1. Le procédé à contrôler a toujours la fonction de transfert considérée dans le chapitre précédent ( $G(s) = \frac{1000}{s(s+1)}$ ). Nous considérons également un correcteur PD mais cette fois-ci nous mettons l'action dérivée dans la chaîne directe, ceci dans le but de faire des études avec des contrôleurs implantés différemment.

Avant l'insertion du contrôleur, la marge de phase ( $Mp$ ) est  $Mp_0 = 1.8^\circ$  à la pulsation  $\omega_c = 31.6$  rad/s ( c'est-à-dire  $\omega_c$  tels que  $|G(j\omega_c)| = 1$ ). On considère que cette marge de phase

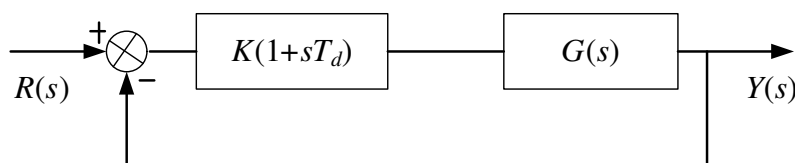


FIGURE 3.1 – Modèle.

est insuffisante et donc on utilise le correcteur PD pour avoir une marge de phase de  $45^\circ$  à  $\omega_c$ . Le contrôleur doit apporter à  $\omega_c$  une phase positive  $\varphi_c = 45^\circ - Mp_0$ . Pour ceci, on doit avoir :

$$\begin{cases} \text{tg}(\varphi_c) = T_d \omega_c \\ 20|\log(K(1 + j\omega_c T_d))| = 0 \end{cases}$$

Nous avons donc l'équation 3.1 qui permet d'obtenir les valeurs de  $T_d$  s et  $K$ .

$$\begin{cases} T_d = \frac{\text{tg}(\varphi_c)}{\omega_c} \\ K = \frac{1}{\sqrt{1 + (\omega_c T_d)^2}} \end{cases} \quad (3.1)$$

Nous obtenons :  $T_d = 0.0297$  s et  $K = 0.7291$ .

La fonction de transfert du système en boucle fermée est :

$$\begin{aligned} F(s) &= \frac{K(1 + T_d s)G(s)}{1 + K(1 + T_d s)G(s)} \\ &= \frac{1000K(1 + T_d s)}{s^2 + (1 + 1000KT_d)s + 1000K} \\ &= \frac{\omega_n^2(1 + T_d s)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned} \quad (3.2)$$

où  $\omega_n$  est la pulsation propre non amortie et  $\zeta$  est le coefficient d'amortissement.

Les caractéristiques du système sont donc les suivantes : marge de phase  $Mp = 45^\circ$  ; les pôles  $p_{1,2} = -11 \pm j24.5$  ; le dépassement  $O = 33.8\%$  ( $O$  pour « Overshoot ») ; le temps de réponse (à 2%)  $t_r = 284$  ms ;  $\omega_n = 27$  rad/s ;  $\zeta = 0.4$  et l'erreur en régime permanent (« steady state error »)  $e_{ss} = \lim_{s \rightarrow 0} s \frac{1}{1 + K(1 + T_d s)G(s)} R(s) = 0$ .

La réponse temporelle  $y(t)$  de l'application de contrôle-commande est représentée sur la figure 3.2.

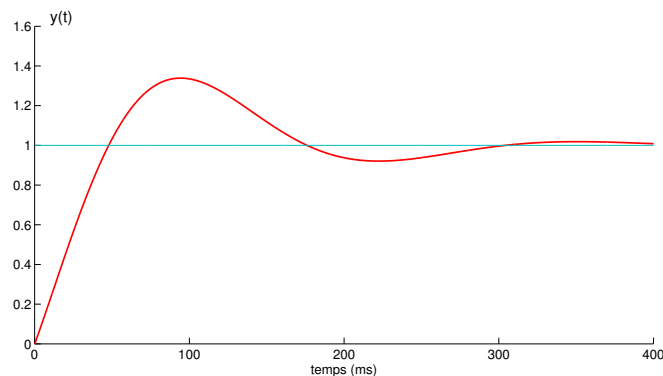


FIGURE 3.2 – réponse temporelle.

### 3.2.2 Implantations considérées sur un réseau CAN

Après la discrétisation de l'application de contrôle-commande (on utilise toujours, pour le choix de la période d'échantillonnage  $h$ , la formule  $\omega_n h \in [0.1; 0.6]$  [Åström 1997] qui permet de prendre encore  $h = 10$  ms), nous considérons les trois implantations suivantes sur le réseau CAN (le réseau CAN a toujours les caractéristiques considérées au chapitre 2) :

- l'implantation de la seule application de contrôle-commande (donc contexte de réseau dédié) avec les mêmes longueurs des trames des flux  $f_{cc}$  et  $f_{ca}$  que celles spécifiées au chapitre 2 ;
- l'implantation de l'application de contrôle-commande qui se partage le réseau avec un flux externe  $f_{ex}$  ( qui a la même longueur de trame que celle spécifié au chapitre 2) ;
- l'implantation de plusieurs applications de contrôle-commande qui se partagent le réseau (ces applications sont toutes identiques à celles présentée à la sous-section 3.2.1).

Rappelons que le retard de communication (retard introduit par le réseau) dans la boucle de chaque application de contrôle-commande est  $(\tau_{cc} + \tau_{ca})$ . Nous avons aussi l'influence du Bloqueur d'Ordre Zéro (« Zero Order Hold (ZOH) ») qui peut être considérée comme l'introduction d'un retard pur supplémentaire  $\tau_{ZOH} = h/2$  [Franklin 2002].

Le retard total dans la boucle d'une application de contrôle-commande est donc

$$\tau = \tau_{cc} + \tau_{ca} + \tau_{ZOH}$$

Notons enfin que nous supposons (seulement pour des raisons de présentation plus simple) que l'actionneur et le capteur (qui sont associées au procédé à contrôler) sont dans le même site (calculateur C1). Le contrôleur est dans un calculateur distant C2.

### 3.2.3 Modèle d'une application de contrôle-commande implantée sur un réseau

Ce modèle, qui va servir de base à notre étude, est représenté sur la figure 3.3 (un retard  $\tau$  est représenté par  $e^{-\tau s}$ ).

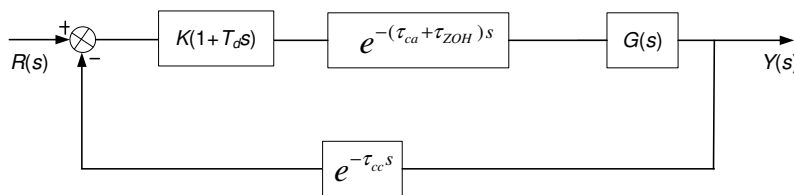


FIGURE 3.3 – Modèle intégrant les retards : retards  $\tau_{ca}$  et  $\tau_{ZOH}$  dans la chaîne directe ; retard  $\tau_{cc}$  dans la chaîne de retour.

La fonction de transfert du système bouclé est la suivante :

$$F(s) = \frac{K(1 + T_d s) e^{-(\tau_{ca} + \tau_{ZOH})s} G(s)}{1 + K(1 + T_d s) e^{-\tau s} G(s)}$$

Les fonctions exponentielles peuvent être remplacées en utilisant l'approximation Padé de 1<sup>er</sup> ordre par les expressions suivantes :

$$e^{-\tau s} \approx \frac{-s + 2/\tau}{s + 2/\tau} \quad \text{et} \quad e^{-(\tau_{ca} + \tau_{ZOH})s} \approx \frac{-s + 2/(\tau_{ca} + \tau_{ZOH})}{s + 2/(\tau_{ca} + \tau_{ZOH})}$$

Appelons  $a = 2/\tau$  et  $b = 2/(\tau_{ca} + \tau_{ZOH})$ , nous avons :

$$\begin{aligned}
 F(s) &= \frac{\frac{1000K(1+T_d s)(-s+b)}{s(s+1)(s+b)}}{\frac{s^3 + (1+a-1000KT_d)s^2 + (1000KT_d a + a - 1000K)s + 1000Ka}{s(s+1)(s+a)}} \\
 &= \frac{1000Ka(1+T_d s)(1+s/a)(1-s/b)}{[s^3 + (1+a-1000KT_d)s^2 + (1000KT_d a + a - 1000K)s + 1000Ka](1+s/b)} \quad (3.3)
 \end{aligned}$$

Le dénominateur est du 4<sup>e</sup> ordre et donc nous avons quatre pôles :  $p_1, p_2, p_3$  pour la partie entre crochets et  $p_4 = -b$ .

Le numérateur a trois zéros :  $z_1 = -1/T_d, z_2 = -a$  et  $z_3 = b$ .

### 3.2.4 Critère d'évaluation de performance d'une implantation

Nous reprenons le critère ITSE  $J = \int_0^T t(r(t) - y(t))^2 dt$ . Nous prenons comme référence la valeur de  $J$  trouvée pour le système continu  $J_0 = 9.4562 \times 10^{-4}$  (durée de simulation  $T = 500$  ms et donc le régime permanent de la référence est atteint). Le critère d'évaluation de performance sera représenté par le terme  $\frac{(J-J_0)}{J_0} \%$  ou  $\frac{\Delta J}{J_0} \%$ . Plus  $\frac{\Delta J}{J_0} (\%)$  est grand, moins bonne est la QdC.

On considère aussi, pour la QdC, le dépassement  $O(\%)$  et la réponse temporelle  $y(t)$ .

## 3.3 Mise en œuvre de la relation QdS→QdC

L'étude stricte de cette relation est faite en considérant le cas où on a des priorités statiques (priorités *a priori* sans lien avec la QdC) et où le retard, généré dans la boucle à chaque période d'échantillonnage, par le transfert des trames des flux  $f_{cc}$  et  $f_{ca}$ , doit être apporté à la connaissance du contrôleur afin que le contrôleur puisse compenser (contrôler) l'effet du retard de boucle ( $\tau$ ) en modifiant les paramètres  $K$  et  $T_d$  de manière à maintenir une caractéristique de l'application de contrôle-commande.

On peut penser, par exemple, au maintien de la marge de phase ou au maintien des pôles qui caractérisent le régime transitoire (méthode dite des pôles dominants). C'est cette dernière technique que nous considérons dans ce travail.

### 3.3.1 Idées principales

Comme la tâche capteur est guidée par le temps (instants d'échantillonnage  $t_k = kh$  avec  $k = 0, 1, 2, \dots$ ) et comme cette tâche est dans le même calculateur C1 que la tâche actionneur, le calculateur C1 peut évaluer le retard  $\tau$  dans la boucle à chaque période.

Tout d'abord, le calculateur C1 évalue la valeur  $(\tau_{cc} + \tau_{ca})$  au moyen du temps écoulé depuis l'instant  $t_k$  (début d'une période d'échantillonnage) jusqu'à l'instant de réception de la trame du flux  $f_{ca}$  correspondante par l'actionneur. Ensuite, le calculateur C1 peut déduire la valeur du retard total dans la boucle  $\tau = \tau_{cc} + \tau_{ca} + \tau_{ZOH}$ .

Si on considère la trame du flux  $f_{cc}$  envoyée par le capteur à l'instant d'échantillonnage  $kh$  (c.-à-d. au début de la  $(k+1)$ <sup>e</sup> période), cette trame contient dans la partie donnée : la valeur de la sortie  $y_k$  et la valeur du retard global  $\tau_k$  évalué sur la période précédente.



Quand le contrôleur reçoit cette trame, il a donc la connaissance de ces deux informations :

- avec la connaissance de  $\tau_k$ , le contrôleur calcule en ligne les nouveaux paramètres  $K$  et  $T_d$  ;
- après le calcul de  $K$  et  $T_d$  et avec la connaissance de  $y_k$ , le contrôleur calcule le signal de commande  $u_k$  et essaie d'envoyer immédiatement (car le contrôleur est guidé par les événements) ce  $u_k$  à l'actionneur.

#### Remarques importantes :

1. on peut constater et s'étonner que le calcul fait par le contrôleur au cours d'une période, pour modifier ses paramètres  $K$  et  $T_d$  est fait en fonction du retard dans la boucle durant la période précédente. Ceci peut poser la question de la correction de cette modification. Cette modification est correcte car on considère que le réseau n'est pas surchargé  $TRU < 100\%$  ( $\tau_{cc} + \tau_{ca} < h$ ) et dans ce cas le retard est constant et le même sur chaque période ;
2. à l'instant initial  $t_0 = 0$  (c.-à-d. à l'initialisation donc à la 1<sup>re</sup> période), le capteur n'a aucune information de  $\tau_{cc}$  et  $\tau_{ca}$ , donc, il prend le retard de la boucle de  $\tau = \tau_{ZOH} = h/2$ .

### 3.3.2 Méthode des pôles dominants

L'objectif de la méthode est de maintenir le même type de régime transitoire que celui avant l'implantation dans le réseau c.-à-d. caractérisé par les deux pôles de la fonction de transfert de l'application de contrôle-commande continue (Equa. 3.2). Il faut donc, en considérant la fonction de transfert du système bouclé à travers le réseau (Equa. 3.3 avec les 4 pôles  $p_1, p_2, p_3, p_4$ ), faire apparaître dans le dénominateur de l'équation 3.3 les deux pôles  $p_1, p_2$  de l'équation 3.2 que l'on appelle pôles dominants (c.-à-d.  $p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = R \pm jI$  avec  $R = 11$  et  $J = 24.5$ ) et considérer la condition qui donne un rôle négligeable aux pôles  $p_3$  et  $p_4$  (que l'on appelle pôles insignifiants). Pour être insignifiants, les pôles  $p_3$  et  $p_4$  doivent avoir leur partie réelle très petite par rapport à la partie réelle des pôles dominants (Fig. 3.4).

Notons que la méthode des pôles dominants ne se préoccupe que de l'évolution du régime transitoire (ce qui dépend des pôles évidemment) mais comme le numérateur de l'équation 3.3 est aussi différent de celui de l'équation 3.2, c'est à dire on a 3 zéros, il faut également lorsqu'on applique cette méthode, voir quelle est l'influence de ces zéros sur le dépassement (« Overshoot »).

### 3.3.3 Calculs inhérents à la méthode des pôles dominants

Considérons la partie entre crochets du dénominateur de l'équation 3.3 de  $F(s)$  :

$$f_3(s) = s^3 + (1 + a - 1000KT_d)s^2 + (1000KT_d a + a - 1000K)s + 1000Ka \quad (3.4)$$

Cette partie ( $f_3(s)$ ) peut se mettre sous la forme  $(s - p_1)(s - p_2)(s - p_3)$  et peut être réécrite en tenant compte des valeurs des deux pôles  $p_{1,2} = R \pm jI$  :

$$(s - p_1)(s - p_2)(s - p_3) = s^3 - (2R + p_3)s^2 + (2Rp_3 + R^2 + I^2)s - (R^2 + I^2)p_3 \quad (3.5)$$

En identifiant les seconds membres des équations 3.5 et 3.4, on obtient les relations suivantes :

$$\begin{cases} 1 + a - 1000KT_d = -(2R + p_3) \\ 1000KT_d a + a - 1000K = R^2 + I^2 + 2Rp_3 \\ 1000Ka = -(R^2 + I^2)p_3 \end{cases}$$

Ces relations permettent de déterminer les valeurs de  $p_3$  (pôle réel),  $K$  et  $T_d$  :

$$\begin{cases} p_3 = -\frac{a^3 + (2 + 2R)a^2 - (R^2 + I^2)a}{a^2 - 2Ra + R^2 + I^2} \\ K = -\frac{(R^2 + I^2)p_3}{1000a} \\ T_d = \frac{1 + a + p_3 + 2R}{1000K} \end{cases} \quad (3.6)$$

Le pôle  $p_4$  est aussi un pôle réel :  $p_4 = -b$ .

#### Remarques sur les calculs des paramètres du contrôleur :

- les modifications de  $K$  et  $T_d$  dépendent seulement du retard total de la boucle ;
- la méthode de compensation amène à modifier en ligne le gain proportionnel  $K$ . Cependant, la modification de  $K$  n'affecte pas la précision du système car l'erreur en régime permanent est nulle quelle que soit la valeur de  $K$  (Sect. 3.2.1).

#### Remarque sur la fonction de transfert $F(s)$ :

L'expression 3.3 de la fonction de transfert  $F(s)$  peut être réécrite sous la forme suivante :

$$\begin{aligned} F(s) &= \frac{1000Ka(1 + T_d s)(1 + s/a)(1 - s/b)}{(s - p_1)(s - p_2)(s - p_3)(1 + s/b)} \\ &= \frac{1000Ka(1 + T_d s)(1 + s/a)(1 - s/b)}{-p_3(s - p_1)(s - p_2)(1 - s/p_3)(1 + s/b)} \end{aligned}$$

Comme on a trouvé la condition  $K = \frac{-(R^2 + I^2)p_3}{1000a}$  et comme  $R^2 + I^2 = \omega_n^2$ , on a donc  $\frac{1000Ka}{-p_3} = \omega_n^2$ . Sachant que  $(s - p_1)(s - p_2) = s^2 + 2\zeta\omega_n s + \omega_n^2$ ,  $F(s)$  peut être réécrite :

$$F(s) = \frac{\omega_n^2(1 + T_d s)(1 + s/a)(1 - s/b)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)(1 - s/p_3)(1 + s/b)} \quad (3.7)$$

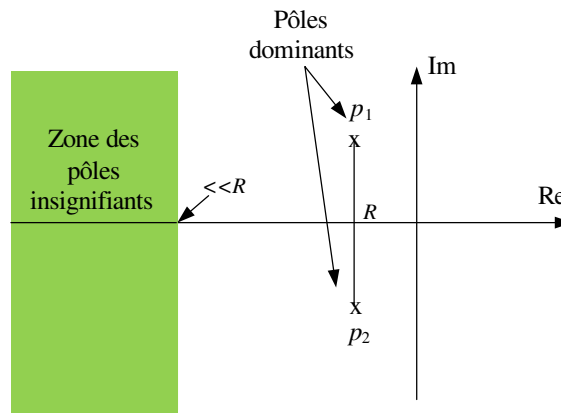


FIGURE 3.4 – Location des pôles dominants et des pôles insignifiants ( $R$  est la valeur de la partie réelle des pôles dominants).

### Conditions pour que les pôles $p_3$ et $p_4$ soient insignifiants

Nous reprenons les conditions exprimées par [Kuo 2003] (page 281) à savoir la partie réelle des pôles insignifiants 5 fois plus petite que la partie réelle des pôles dominants. Donc les conditions pour l'insignifiance des pôles  $p_3$  et  $p_4$  sont les suivantes :

- le pôle  $p_3$  :  $p_3 = -\frac{a^3 + (2+2R)a^2 - (R^2 + I^2)a}{a^2 - 2Ra + R^2 + I^2} \leq 5R$  soit  $p_3 < -55 \Rightarrow \tau < 40$  ms.  
Comme  $\tau_{ZOH} = 5$  ms, il faut que  $\tau_{cc} + \tau_{ca} < 35$  ms ;
- le pôle  $p_4$  :  $p_4 = -2/(\tau_{ca} + \tau_{ZOH}) \leq 5R \Rightarrow \tau_{ca} + \tau_{ZOH} < 36$  ms.  
Comme  $\tau_{ZOH} = 5$  ms, il faut que  $\tau_{ca} < 31$  ms.

Ces conditions sont toujours satisfaites car la période d'échantillonnage  $h$  est de 10 ms et le retard ( $\tau_{cc} + \tau_{ca}$ ) est toujours inférieur à  $h$ .

En considérant que les pôles  $p_3$  et  $p_4$  sont insignifiants, la fonction de transfert (Equa. 3.7) est la suivante :

$$F(s) = \frac{\omega_n^2(1 + T_d s)(1 + s/a)(1 - s/b)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (3.8)$$

où  $a = 2/\tau$  et  $b = 2/(\tau_{ca} + \tau_{ZOH})$ .

### Analyse de l'influence des zéros

Dans la fonction de transfert représentée sur l'équation 3.8, on trouve trois zéros : deux zéros négatifs  $z_1 = -1/T_d$ ,  $z_2 = -2/\tau$  et un zéro positif  $z_3 = 2/(\tau_{ca} + \tau_{ZOH})$ .

L'effet d'un zéro sur la performance du système de second ordre peut être trouvé dans [C.Dorf 1995] et [Kwoon 2000]. Nous présentons dans l'annexe A une démonstration graphique et simple de l'effet des zéros et nous avons constaté que quand le zéro approche de l'origine, le dépassement  $O$  augmente, autrement dit, quand le zéro est très éloigné de l'origine, son influence est très faible et peut être négligée. On doit caractériser quantitativement cet éloignement :

- pour un zéro négatif (appelons  $z_0$  la valeur absolue de ce zéro) ; dans [C.Dorf 1995], le zéro négatif a un effet faible et peut être négligé si  $z_0/\zeta\omega_n \geq 5$  ;
- pour un zéro positif (appelons  $z_0$  la valeur de ce zéro) ; dans [Kwoon 2000], le dépassement est évalué sous la forme suivante :

$$O = e^{-\frac{\zeta(\Phi+\pi)}{\sqrt{1-\zeta^2}}} \sqrt{1 + \frac{2\zeta\omega_n}{z_0} + \left(\frac{\omega_n}{z_0}\right)^2} \text{ où } \Phi = \tan^{-1} \frac{\sqrt{1-\zeta^2}}{z_0 + \zeta\omega_n}$$

On voit que si  $\frac{z_0}{2\zeta\omega_n} \gg 1$  et  $\left(\frac{z_0}{\omega_n}\right)^2 \gg 1$ , le dépassement peut être réécrit comme  $O = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$  c.-à-d. on a le dépassement du système du second ordre sans zéro.

On considère donc que si  $\frac{z_0}{2\zeta\omega_n} > 5$  et  $\left(\frac{z_0}{\omega_n}\right)^2 > 5$  (ces conditions induisent que  $z_0 > 110$  avec  $\omega_n = 27$ ,  $\zeta = 0.4$ ), le zéro positif a un effet faible et peut être négligé.

On considère maintenant les trois zéros :

- le zéro négatif  $z_2 = -2/\tau$  : comme  $\tau = \tau_{cc} + \tau_{ca} + \tau_{ZOH}$  est inférieur à  $h + h/2 = 15$  ms,  $z_2$  est inférieur à  $\frac{-2}{0.015}$  soit -133. On voit que  $z_2/\zeta\omega_n \geq 5$ . L'influence de ce zéro est donc faible et peut être négligée ;
- le zéro positif  $z_3 = 2/(\tau_{ca} + \tau_{ZOH})$  : car  $\tau_{ca} + \tau_{ZOH} < 15$  ms,  $z_3$  est supérieur à  $\frac{2}{0.015} = 133$ , donc  $z_3 > 110$ . Alors, ce zéro peut être négligé ;

- le zéro négatif  $z_1 = -1/T_d$  : notons que  $z_1$  est basé sur un paramètre du contrôleur ( $T_d$ ).  
L'influence du zéro  $z_1 = -1/T_d$  dépend de la valeur de  $T_d$  ; plus grande est la valeur de  $T_d$ , le zéro est plus proche de l'origine et donc plus forte est l'influence. Il faut donc considérer l'influence de ce zéro.

#### Remarque sur la fonction de transfert du système bouclé $F(s)$ :

En considérant que les zéros  $z_2$  et  $z_3$  sont négligés, la fonction de transfert dans l'équation 3.8 peut être réécrite comme l'équation 3.9. On voit que on a la même forme comme la fonction de transfert du système sans retards (Equa. 3.2) dans la section 3.2.1 mais la valeur de  $T_d$  est maintenant changée en fonction du retard à compenser.

$$F(s) = \frac{\omega_n^2(1 + T_d s)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.9)$$

#### 3.3.4 Calcul de $u_k$

Le contrôleur calcule le signal de commande  $u_k$  (qui se compose d'une composante proportionnelle  $P_k$  et d'une composante dérivée  $D_k$ ) en utilisant la formule 3.10.

$$\begin{cases} P_k = Ke_k \\ D_k = \frac{T_d}{Nh + T_d} D_{k-1} + \frac{NKT_d}{Nh + T_d} (e_k - e_{k-1}) \\ u_k = P_k + D_k \end{cases} \quad (3.10)$$

où  $e_k = r_k - y_k$ ,  $N$  est une constante ([Åström 1997], page 307).

#### 3.3.5 Validation de la méthode des pôles dominants

L'objectif de cette section est, en considérant le modèle de la figure 3.3 qui intègre les retards, qui apparaissent dans la chaîne directe et dans la chaîne de retour, de vérifier l'aspect compensation des retards permis par la méthode des pôles dominants. Ceci est fait en utilisant l'outil MATLAB.

Afin de bien illustrer le rôle de la méthode des pôles dominants, nous prenons différentes valeurs des retards (on prend  $\tau_{ZOH} = 5$  ms et on fait varier  $\tau_{cc}$  et  $\tau_{ca}$ ) et nous analysons deux cas :

- cas 1 : on ne change pas le contrôleur (c'est-à-dire qu'on garde le contrôleur défini à la section 3.2.1), c'est ce que nous appelons le cas « Contrôleur fixe » ;
- cas 2 : on change le contrôleur par la méthode des pôles dominants, c'est ce que nous appelons le cas « Contrôleur adaptatif ».

##### Cas 1 : Contrôleur fixe

Nous représentons ce cas (les dépassements  $O$  sont donnés) sur le tableau 3.1a (où  $\tau_{cc}$  est variable ;  $\tau_{ca}$  est constant et égal à 1 ms) et le tableau 3.1b (où  $\tau_{ca}$  est variable ;  $\tau_{cc}$  est constant et égal à 1 ms).

Deux points sont à remarquer :

- plus le retard est grand, moins bonne est la QdC (ce qui est logique) c'est-à-dire qu'on a une croissance du dépassement ;

| $\tau_{cc}$ (ms) | $\tau$ (ms) | $O$ (%) |
|------------------|-------------|---------|
| 1                | 7           | 46.65   |
| 2                | 8           | 49.18   |
| 3                | 9           | 51.88   |

(a)  $\tau_{cc}$  variable,  $\tau_{ca}$  constant = 1 ms.

| $\tau_{ca}$ (ms) | $\tau$ (ms) | $O$ (%) |
|------------------|-------------|---------|
| 2                | 8           | 49.14   |
| 3                | 9           | 51.82   |
| 4                | 10          | 54.65   |

(b)  $\tau_{ca}$  variable,  $\tau_{cc}$  constant = 1 ms.

TABLE 3.1 – Contrôleur fixe.

- l'influence de  $\tau_{cc}$  et  $\tau_{ca}$  est identique (voir les lignes des deux tableaux 3.1a et 3.1b où le retard total  $\tau$  est égal à 8 ms et 9 ms).

### Cas 2 : Contrôleur adaptatif

Nous représentons ce cas sur les tableaux 3.2 et 3.3 ( $K$ ,  $T_d$ , les pôles dominants  $p_{1,2}$ , les pôles insignifiants  $p_3$  et  $p_4$ , le dépassement  $O$  et les zéros  $z_1$ ,  $z_2$ ,  $z_3$  sont donnés). Ces tableaux nous amènent les commentaires suivants :

- on maintient la spécification souhaitée :  $-11 \pm j24.5$  pour les pôles dominants  $p_{1,2}$  et les pôles insignifiants  $p_3$ ,  $p_4$  car toujours inférieur à  $5R$ . Les analyses et les calculs des paramètres du contrôleur présentés dans la section 3.3.3 sont donc corrects ;
- quand  $\tau$  augmente le dépassement  $O$  croit. Cette dégradation de  $O$  est due au zéro  $z_1 = -1/T_d$ . Les deux zéros  $z_2 = -2/\tau$  et  $z_3 = 2/(\tau_{ca} + \tau_{ZOH})$  ont les effets faibles (Sect. 3.3.3) mais le zéro  $z_1 = -1/T_d$  croit avec le retard global et donc c'est ce zéro qui est la cause de la croissance du dépassement. Par exemple pour  $\tau = 8$  ms,  $T_d = 0.0377$  s et donc  $z_1 = -26.5$ , le rapport  $|z_1|/\zeta\omega_n = 2.4$ , ce rapport est inférieur à 5 et donc  $z_1$  ne peut pas être négligé.
- l'influence du zéro  $z_1 = -1/T_d$  est aussi marquée par la croissance du dépassement  $O$  par rapport à celui du système sans retard dans la section 3.2.1 (où  $O = 33.8\%$ ) ;
- on constate ici aussi que pour des valeurs identiques du retard total  $\tau$  (lignes des 2 tableaux 3.2 et 3.3 où  $\tau$  est égal à 8 ms et 9 ms), les dépassements sont sensiblement identiques.

| $\tau_{cc}$ (ms) | $\tau$ (ms) | $K$    | $T_d$ (s) | $p_{1,2}$       | $p_3$ | $p_4$ | $O$ (%) | $z_1$ | $z_2$ | $z_3$ |
|------------------|-------------|--------|-----------|-----------------|-------|-------|---------|-------|-------|-------|
| 1                | 7           | 0.6119 | 0.0365    | $-11 \pm j24.5$ | -242  | -333  | 40.66   | -27.3 | -285  | 333   |
| 2                | 8           | 0.5959 | 0.0377    | $-11 \pm j24.5$ | -206  | -333  | 41.74   | -26.5 | -250  | 333   |
| 3                | 9           | 0.5799 | 0.0389    | $-11 \pm j24.5$ | -178  | -333  | 42.84   | -25.7 | -222  | 333   |

TABLE 3.2 –  $\tau_{cc}$  variable,  $\tau_{ca}$  constant = 1 ms (Contrôleur adaptatif).

| $\tau_{ca}$ (ms) | $\tau$ (ms) | $K$    | $T_d$ (s) | $p_{1,2}$       | $p_3$ | $p_4$ | $O$ (%) | $z_1$ | $z_2$ | $z_3$ |
|------------------|-------------|--------|-----------|-----------------|-------|-------|---------|-------|-------|-------|
| 2                | 8           | 0.5959 | 0.0377    | $-11 \pm j24.5$ | -206  | -285  | 41.70   | -26.5 | -250  | 285   |
| 3                | 9           | 0.5799 | 0.0389    | $-11 \pm j24.5$ | -178  | -250  | 42.78   | -25.7 | -222  | 250   |
| 4                | 10          | 0.5639 | 0.0401    | $-11 \pm j24.5$ | -156  | -222  | 43.95   | -24.9 | -200  | 222   |

TABLE 3.3 –  $\tau_{ca}$  variable,  $\tau_{cc}$  constant = 1 ms (Contrôleur adaptatif).

### 3.3.6 Considération des différentes implantations sur le réseau CAN

#### Une seule application de contrôle-commande

Les réponses temporelles pour un contrôleur fixe et un contrôleur adaptatif sont représentées sur la figure 3.5. On voit bien que le contrôleur adaptatif donne un meilleur résultat que le contrôleur fixe.

On peut noter cependant que le dépassement avec le contrôleur adaptatif est plus grand que celui dans le système sans réseau (Fig. 3.2) : ceci résulte de la croissance de  $T_d$  pour compenser le retard introduit par la transmission des trames des flux  $f_{cc}$  et  $f_{ca}$  (donc le zéro  $-1/T_d$  se rapproche de l'origine).

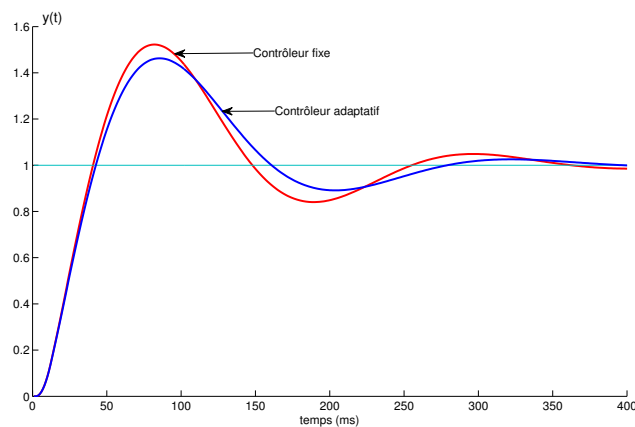


FIGURE 3.5 – Un seul procédé : Contrôleur fixe et Contrôleur adaptatif.

#### Application de contrôle-commande partageant le réseau avec un $f_{ex}$

##### Contrôleur fixe

Nous avons sur la figure 3.6 les réponses temporelles  $y(t)$  correspondant à des TRUs différents dans le cas d'un contrôleur fixe. On peut constater l'impact de la charge du réseau (TRU) par la croissance du dépassement. Plus TRU est grand, plus le retard est grand, plus le dépassement est élevé.

##### Contrôleur adaptatif

Nous avons, dans le cas d'un contrôleur adaptatif, les réponses temporelles correspondant à des TRUs différents qui sont représentées sur la figure 3.7. On remarque que :

- la méthode des pôles dominants améliore bien la QdC (comparaison avec la figure 3.6) ;
- les dépassements croissent avec le retard, ceci résultant de l'effet du zéro  $-1/T_d$ .

Pour les deux types du contrôleur (fixe et adaptatif), les performances sont évidemment moins bonnes que celles du cas sans flux externe (comparer les figures 3.5, 3.6 et 3.7).

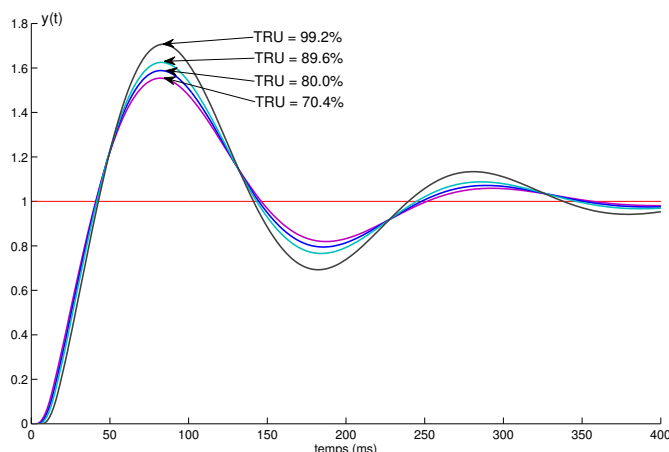


FIGURE 3.6 – Réponses temporelles avec Contrôleur fixe (un procédé ; plusieurs TRUs).

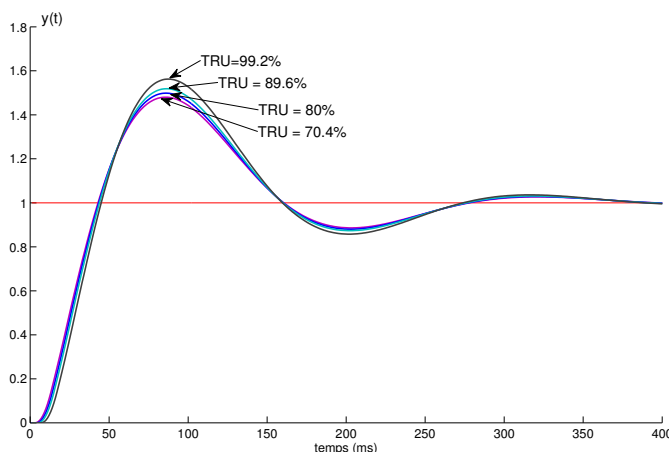


FIGURE 3.7 – Réponses temporelles avec Contrôleur adaptatif (un procédé ; plusieurs TRUs).

### Quatre applications de contrôle-commande se partageant le réseau

Appelons  $P_1, P_2, P_3, P_4$  les 4 applications considérées. Les flux  $f_{cc}$  de 4 applications sont synchrones avec la période d'échantillonnage  $h = 10$  ms.

En ce qui concerne les priorités, nous considérons pour chaque application de contrôle-commande  $P_{ca} > P_{cc}$  (cf. Chap. 2). Appelons  $P_{cci}$  et  $P_{cai}$  les priorités des flux  $f_{cc}$  et  $f_{ca}$  du procédé  $P_i$ . Les priorités des flux des différents procédés  $P_i$  sont ordonnées de la manière suivante :

$$P_{ca1} > P_{ca2} > P_{ca3} > P_{ca4} > P_{cc1} > P_{cc2} > P_{cc3} > P_{cc4} \quad (3.11)$$

C'est à dire que le procédé  $P_i$  est considéré plus important que le procédé  $P_j$  avec  $i < j$ .

Les résultats de cette étude, toujours basée sur la comparaison entre le contrôleur fixe (pas de relation  $QdS \rightarrow QdC$ ) et le contrôleur adaptatif (relation  $QdS \rightarrow QdC$ ), sont présentés sur le tableau 3.4 (au moyen du critère ITSE) et sur les figures 3.8 et 3.9 (réponses temporelles).

Nous constatons :

- d'une part, à partir des résultats dans le cas du contrôleur fixe : les performances des applications suivent l'ordre des priorités ce qui est logique (plus la priorité est élevée, moins grand est le retard et donc meilleure est la performance) ;

- d'autre part, par comparaison entre les résultats du cas contrôleur fixe et du cas contrôleur adaptatif : l'influence de la mise en place du contrôleur adaptatif pour améliorer les performances et les resserrer. On ne peut pas cependant avoir des performances identiques pour toutes les applications à cause de l'effet du zéro négatif  $-1/T_d$  (comme les applications les moins prioritaires ont des retards plus élevés, les compensations requièrent des valeurs de  $T_d$  plus élevées ce qui entraîne que le zéro négatif  $-1/T_d$  se rapproche davantage de l'origine).

|  | P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> | P <sub>4</sub> |
|--|----------------|----------------|----------------|----------------|
| Contrôleur fixe + priorité statique                            | 75.3           | 107            | 150            | 208            |
| Contrôleur adaptatif + priorité statique<br>(relation QdS→QdC) | 47.6           | 62.8           | 80.6           | 101            |

TABLE 3.4 – Performances ( $\Delta J/J_0\%$ ) de 4 procédés avec le contrôleur fixe et le contrôle adaptatif (les priorités des flux  $f_{ca}$  et  $f_{cc}$  sont statiques).

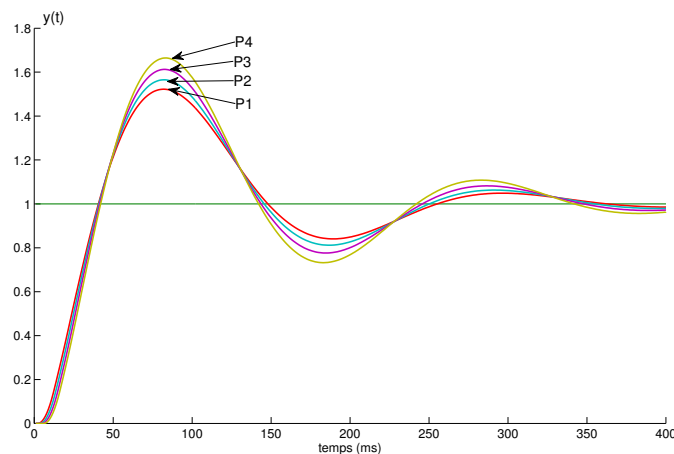


FIGURE 3.8 – Réponses temporelles de 4 procédés avec Contrôleur fixe.

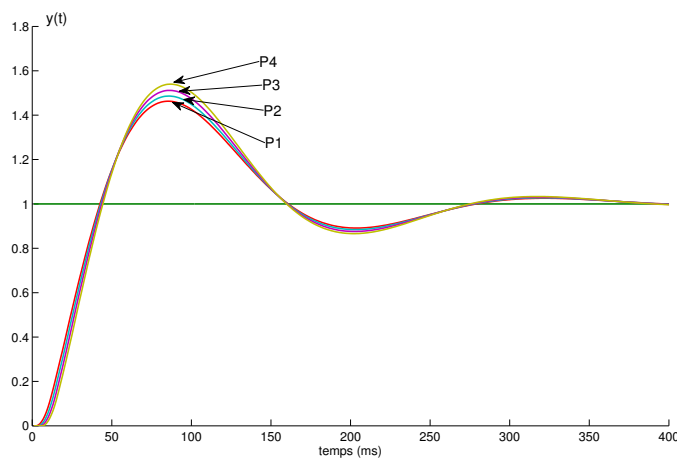


FIGURE 3.9 – Réponses temporelles de 4 procédés avec Contrôleur adaptatif.



### 3.4 Mise en œuvre de la relation $QdS \rightleftharpoons QdC$

#### 3.4.1 Idée de base

L'idée de base est de combiner le schéma de priorité hybride ( $QdC \rightarrow QdS$ ) et la méthode de compensation de retard ( $QdS \rightarrow QdC$ ). L'objectif est d'avoir un système plus efficace en terme de nombre d'applications que l'on peut implanter sur le réseau. Plus précisément, le schéma de priorité hybride ph (Chap. 2) est le schéma de priorité hybride qui permet au mieux par rapport au cas statique d'équilibrer les retards et donc d'envisager l'implantation d'un nombre plus important d'applications (dans le cas des priorités statiques, les applications qui ont des priorités faibles ont des retards très importants qui peuvent ne plus être bien compensables).

Cependant avec le schéma de priorité hybride ph, qui est basé sur l'urgence de la transmission et donc donne des retards variables sur chaque période d'échantillonnage, la méthode de compensation présentée dans la section 3.3 (basée sur l'utilisation du retard de la période précédente) ne peut plus être utilisée maintenant. Pour faire une bonne compensation et donc modifier correctement les paramètres  $K$  et  $T_d$ , le contrôleur doit connaître le retard de la boucle ( $\tau = \tau_{cc} + \tau_{ca} + \tau_{ZOH}$ ) dans la période actuelle :

- $\tau_{ZOH}$  est une donnée connue qui est considérée comme un retard de communication de  $h/2$  ; il peut être à disposition du contrôleur ;
- $\tau_{cc}$  peut être porté à la connaissance du contrôleur au moyen de la trame du flux  $f_{cc}$  envoyée par le capteur après l'instant d'échantillonnage  $kh$  ( $k = 0, 1, 2, \dots$ ) ; nous considérons que l'instant  $kh$  est mis dans la trame envoyée par le capteur et donc le contrôleur, lorsqu'il reçoit cette trame, en déduit, en faisant référence à son horloge locale (nous supposons que les horloges sont bien synchronisés), la valeur de  $\tau_{cc}$  ;
- $\tau_{ca}$ , par contre, n'est pas connu. Il faut l'anticiper. *Nous proposons de donner aux contrôleurs les priorités dynamiques maximales  $P_{\max}$ <sup>1</sup> et donc un contrôleur peut, dès qu'il reçoit une trame du capteur, envoyer tout de suite (nous négligeons le temps de calcul) sa trame du flux  $f_{ca}$  (avec la priorité dynamique  $P_{\max}$ ). Avec cette hypothèse, le retard  $\tau_{ca}$  est égal à la durée de la transmission de la trame du flux  $f_{ca}$ , donc  $\tau_{ca} = D_{ca}$ .*

*Remarque :* le retard de la boucle est maintenant calculé par  $\tau = \tau_{cc} + D_{ca} + h/2$ .

Notons que si le contrôleur utilise la priorité dynamique  $P_{\max}$  pour l'envoi de sa trame du flux  $f_{ca}$ , il calcule toujours la priorité dynamique obtenue à partir du signal de commande  $u$  (après la modification des paramètres du contrôleur) et met cette valeur dans la partie donnée de la trame du flux  $f_{ca}$  destinée à l'actionneur. Comme le capteur et l'actionneur sont dans le même calculateur, le capteur utilise cette priorité dynamique pour envoyer sa prochaine trame du flux  $f_{cc}$ .

Nous appelons  $ph^+$  le schéma ph en donnant aux contrôleurs la priorité dynamique maximale  $P_{\max}$ .

1. On peut noter que cette proposition rejoint la structuration en classe de [Velasco 2006].

### 3.4.2 Implantation de la relation QdS $\Rightarrow$ QdC

Le principe de cette implantation, relativement à la période  $[t_k, t_{k+1}]$ , est représenté sur la figure 3.10.

L'implantation de cette relation est distribuée sur les deux sites, le site Actionneur-Capteur et le site Contrôleur :

- site Actionneur-Capteur :
  - à l'instant  $t_k = kh$ , le capteur génère une trame du flux  $f_{cc}$  avec un champ ID où la priorité dynamique est la priorité dynamique envoyée par le contrôleur dans la période précédente ( $P_{k-1}$ ). Ceci caractérise la réalisation de relation QdC $\rightarrow$ QdS ;
  - cette trame du flux  $f_{cc}$  contient dans sa partie donnée la valeur de l'instant  $t_k$  et la valeur  $y_k$  de la sortie. Ceci caractérise la contribution à la relation QdS $\rightarrow$ QdC.
- site Contrôleur :
  - à la réception de la trame du flux  $f_{cc}$ , le contrôleur utilise la valeur de l'instant  $t_k = kh$  pour déduire (par comparaison avec l'horloge locale) la valeur de  $\tau_{cc}$ . Avec sa connaissance des valeurs  $\tau_{ZOH}$  ( $= h/2$ ) et  $\tau_{ca}$  ( $= D_{ca}$ ), il peut donc calculer le retard global dans la boucle  $\tau = \tau_{cc} + \tau_{ca} + \tau_{ZOH}$  et calculer les nouveaux paramètres  $K$  et  $T_d$  qui sont nécessaires. Ceci caractérise la réalisation de la relation QdS $\rightarrow$ QdC ;
  - ensuite, le contrôleur calcule, avec les valeurs  $y_k$  et ses nouveaux paramètres  $K$  et  $T_d$ , la valeur du signal de commande  $u_k$  (ceci est typiquement l'action classique de la commande) qu'il va mettre dans le champ de données de la prochaine trame du flux  $f_{ca}$ . Avec ce signal de commande  $u_k$ , le contrôleur calcule la priorité dynamique  $P_k$  qu'il va mettre également dans le champ de données de la prochaine trame du flux  $f_{ca}$  et que le capteur utilisera pour envoyer la trame du flux  $f_{cc}$  à la prochaine période. Ceci caractérise la mise en place de la relation QdC $\rightarrow$ QdS ;
  - le contrôleur envoie ensuite la trame du flux  $f_{ca}$  avec la priorité dynamique  $P_{max}$ .

La figure 3.11 visualise la priorité dynamique du champ ID dans les trames issues du capteur (flux  $f_{cc}$ ) et du contrôleur (flux  $f_{ca}$ ) sur les périodes successives débutant aux instants  $t_k$  et  $t_{k+1}$ .

**Remarque :** A l'instant initial  $t_0 = 0$ , le capteur n'a aucune information venant du contrôleur quant à la priorité dynamique qu'il doit utiliser. Le capteur prend donc la priorité dynamique  $P_{max}$ .

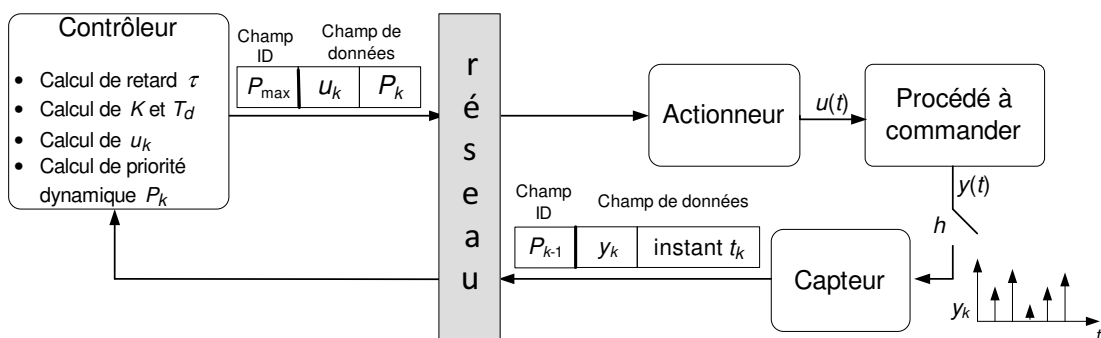


FIGURE 3.10 – Principe de fonctionnement de la relation QdS $\Rightarrow$ QdC (période  $[t_k, t_{k+1}]$ ).

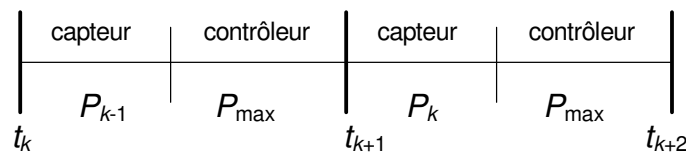


FIGURE 3.11 – Diagramme des priorités dynamiques utilisées par le capteur et le contrôleur sur les périodes  $[t_k, t_{k+1}]$  et  $[t_{k+1}, t_{k+2}]$ .

### 3.4.3 Evaluation de performance

Nous reprenons le cas de l'implantation de 4 applications de contrôle-commande identiques à celle définie dans la section 3.2.1. En ce qui concerne la partie priorité statique de la priorité hybride, nous considérons les relations 3.11.

Le tableau 3.5 montre sur la base du critère  $\Delta J/J_0(\%)$ , les résultats obtenus en considérant soit uniquement la relation  $QdC \rightarrow QdS$  soit la relation  $QdS \rightleftharpoons QdC$ .

|  | P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> | P <sub>4</sub> |
|--|----------------|----------------|----------------|----------------|
| QdC → QdS<br>(priorité hybride + contrôleur fixe)      | 122            | 125            | 131            | 145            |
| QdS ⇌ QdC<br>(priorité hybride + contrôleur adaptatif) | 72.9           | 70.3           | 68.7           | 78.4           |

TABLE 3.5 – Performances ( $\Delta J/J_0(\%)$ ) de 4 procédés.

Si on compare par rapport au tableau 3.4, on voit un aspect d'équilibre qui se dégage pour les différentes applications : ceci est dû à l'influence de la priorité hybride ( $QdC \rightarrow QdS$ ) qui crée une différence importante par rapport à la priorité statique (performances plus resserrées). Cet aspect d'équilibre est ensuite maintenu dans la relation  $QdS \rightleftharpoons QdC$ , par l'effet compensateur de retard.

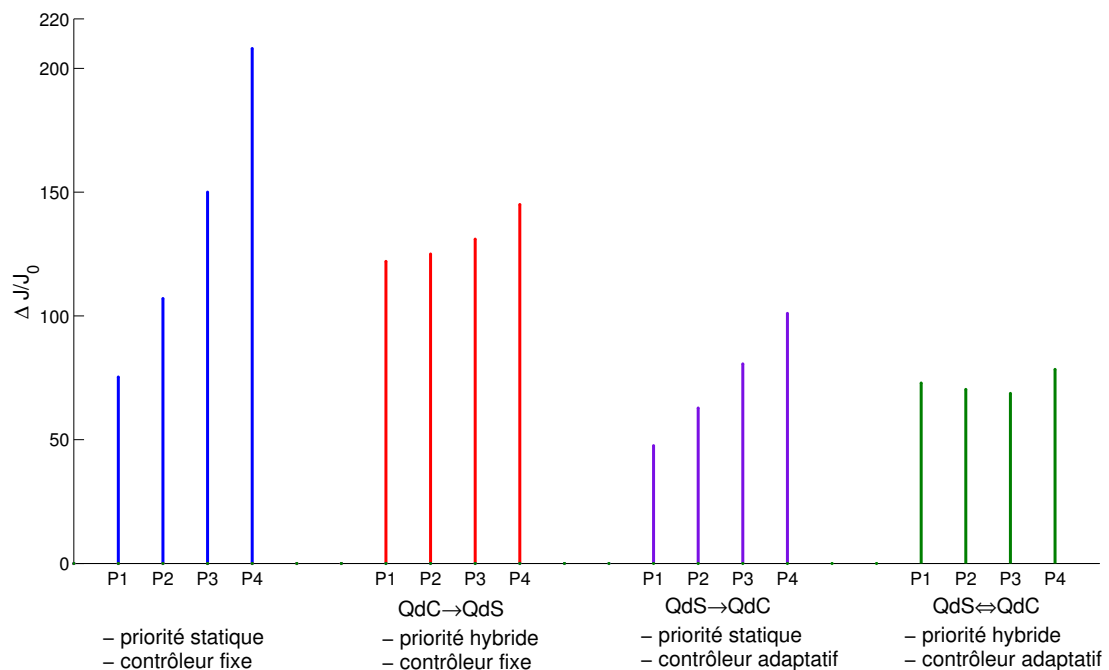
On peut donc dire que si on se fixe comme contrainte une certaine valeur de  $\Delta J/J_0(\%)$  (pas trop petite) la relation  $QdS \rightleftharpoons QdC$  permet d'implanter davantage d'applications que la relation  $QdS \rightarrow QdC$ .

## 3.5 Récapitulation des résultats obtenus

Le tableau 3.6, qui rassemble les tableaux 3.4 et 3.5, veut résumer les principaux résultats. Ces résultats sont représentés graphiquement sur la figure 3.12 et montrent bien l'équilibrage de performances des applications procuré par les priorités hybride ( $QdC \rightarrow QdS$ ) par rapport aux priorités statiques et propagés par la compensation des retards ( $QdS \rightleftharpoons QdC$ ).

La visualisation du rôle des priorités hybrides par rapport aux priorités statiques est montrée sur les « zooms » des réponses temporelles des applications représentés sur les figures 3.13 et 3.14 : on voit que l'on a avec les priorités hybrides pour toutes les applications un dépassement de l'ordre de 60% (Fig. 3.14), alors que l'on a un dépassement allant de 53% à 67% avec les priorités statiques (Fig. 3.13). Ceci traduit le fait que l'on a avec les priorités hybrides un retard sur la boucle sensiblement identique pour toutes les applications et donc qui donne un même dépassement. Par contre avec les priorités statiques, on a des retards différents pour chaque application ce qui introduit des dépassements différents.

|   | P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> | P <sub>4</sub> |
|---|----------------|----------------|----------------|----------------|
| priorité statique + contrôleur fixe                   | 75.3           | 107            | 150            | 208            |
| QdC→QdS<br>(priorité hybride + contrôleur fixe)       | 122            | 125            | 131            | 145            |
| QdS→QdC<br>(priorité statique + contrôleur adaptatif) | 47.6           | 62.8           | 80.6           | 101            |
| QdS⇌QdC<br>(priorité hybride + contrôleur adaptatif)  | 72.9           | 70.3           | 68.7           | 78.4           |

TABLE 3.6 – Récapitulation des résultats de 4 procédés ( $\Delta J/J_0\%$ ).FIGURE 3.12 – Récapitulation graphique des résultats de 4 procédés ( $\Delta J/J_0\%$ ).

La propagation de la conséquence de ce rôle des priorités hybrides apparaît dans la mise en œuvre de la compensation des retards (Figs. 3.15 et 3.16) :

- la figure 3.15 montre que la compensation des retard maintient des différences entre les applications avec des dépassement plus faibles que sur la figure 3.13 ; ces différences résultent des retards de boucles différents (Fig. 3.13) qui induisent des modifications différentes des contrôleurs ( $K$  et  $T_d$ ) et donc en particulier des zéros négatifs ( $-1/T_d$ ) différents ;
- la figure 3.16 montre que les compensations des retards donnent sensiblement les mêmes dépassements ( $\approx 50\%$ ) ; ceci découle du fait que les retards avant compensation étaient sensiblement les mêmes (Fig. 3.14) et donc on a sensiblement les mêmes modifications pour les contrôleurs ( $K$  et  $T_d$ ), ce qui donne donc des zéros négatifs ( $-1/T_d$ ) sensiblement les mêmes.

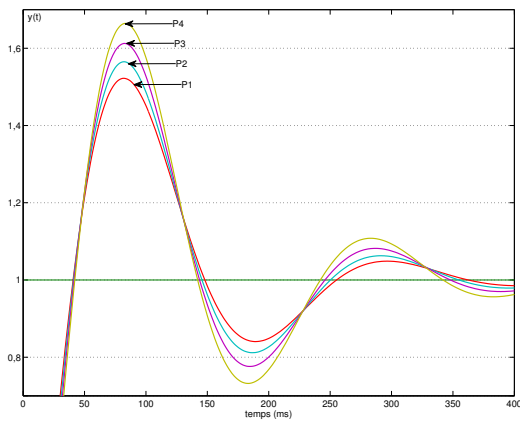


FIGURE 3.13 –  
priorité statique + contrôleur fixe.

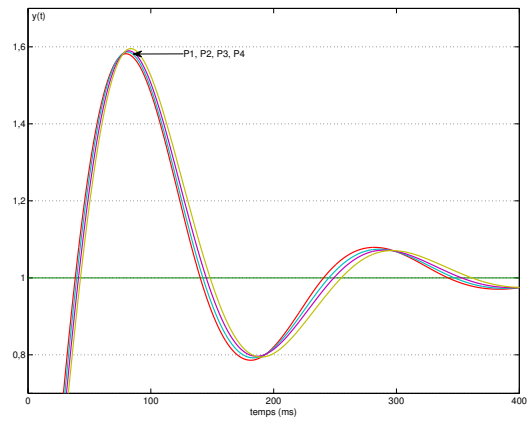


FIGURE 3.14 –  $QdC \rightarrow QdS$  :  
priorité hybride + contrôleur fixe.

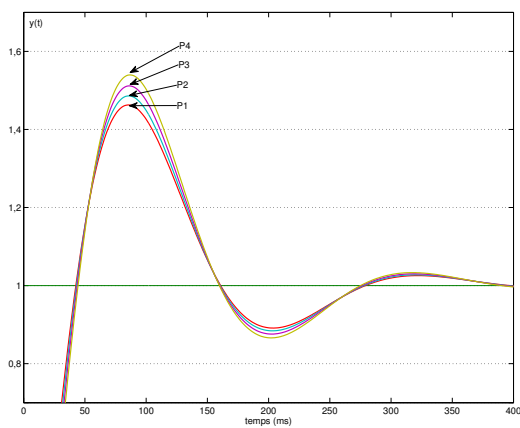


FIGURE 3.15 –  $QdS \rightarrow QdC$  :  
priorité statique + contrôleur adaptatif.

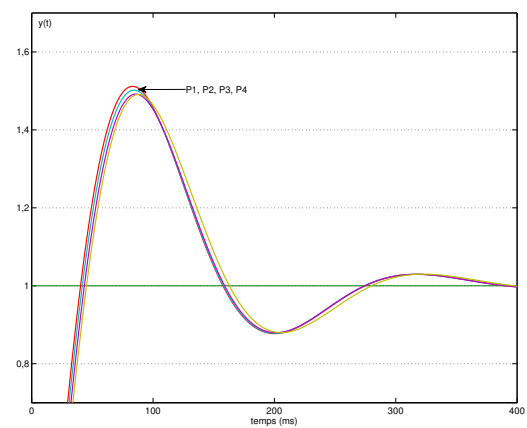


FIGURE 3.16 –  $QdS \rightleftharpoons QdC$  :  
priorité hybride + contrôleur adaptatif.

## 3.6 Conclusion

Il nous paraît important de mettre en exergue les deux points suivants qui ont été traités dans ce chapitre :

- tout d’abord, la présentation de la mise en œuvre de la relation  $QdS \rightarrow QdC$  sur la base des priorités statiques pour l’ordonnancement des messages (ceci est basé sur le fait que la compensation du retard, dans la boucle, sur une période d’échantillonnage peut utiliser l’information du retard, dans la boucle, sur la période d’échantillonnage précédente, puisque ces retards, compte tenu que le réseau n’est pas surchargé ( $TRU < 100\%$ ), sont constants quelle que soit la période d’échantillonnage). Cette mise en œuvre est basée sur la méthode des pôles dominants qui assure, à la fois, la dominance des deux pôles fondamentaux et d’autre part, l’insignifiance des pôles introduits par les retards dans la boucle. Les relations garantissant cette assurance ont été données ;
  - ensuite et, principalement, la présentation de la mise en œuvre de la bi-relation  $QdS \rightleftharpoons QdC$  qui nécessite de réaliser à la fois les relations  $QdC \rightarrow QdS$  (basé sur la notion de la priorité hybride  $ph$ ) et  $QdS \rightarrow QdC$ . Cependant comme la priorité hybride donne des retards variables sur chaque période, on ne peut pas appliquer maintenant la méthode de l’utilisation du retard de la période précédente pour mettre en œuvre la relation  $QdS \rightarrow QdC$  comme exposée avec des priorités statiques. Il faut donc mettre en œuvre la relation  $QdS \rightarrow QdC$  en considérant le retard dans la période actuelle ce qui nécessite de modifier la technique de priorité  $ph$  (relation  $QdC \rightarrow QdS$ ) et de la transformer en ce que nous avons appelé priorité  $ph^+$  (le contrôleur utilise maintenant la priorité dynamique maximale  $P_{max}$ , ce qui lui permet d’envoyer immédiatement la trame du flux  $f_{ca}$  et savoir que la durée  $\tau_{ca}$  est égale à la durée de la trame du flux  $f_{ca}$ , en négligeant le temps de propagation ce qui est réaliste). La combinaison de la relation  $QdC \rightarrow QdS$ , sur la base de la priorité  $ph^+$ , et de la relation  $QdS \rightarrow QdC$ , sur la base de la méthode des pôles dominants, a montré tout son intérêt pour amener des performances convenables et sensiblement identiques pour les différentes applications de contrôle-commande.
- Un point de recherche future serait évidemment de trouver la limite d’un tel comportement (dépassement convenable - nombre maximum d’applications).



# Réseaux locaux sans fil : Protocoles MAC de type CSMA sans collision

---

## Sommaire

---

|            |   |            |
|------------|---|------------|
| <b>4.1</b> | <b>Introduction</b> . . . . .   | <b>82</b>  |
| <b>4.2</b> | <b>Préliminaires</b> . . . . .  | <b>83</b>  |
| 4.2.1      | La couche physique . . . . .  | 83         |
| 4.2.2      | La sous-couche MAC . . . . .  | 86         |
| <b>4.3</b> | <b>Spécification des protocoles MAC CSMA sans collision basés sur des messages<br/>BlackBurst</b> . . . . . | <b>87</b>  |
| 4.3.1      | Considération de priorités statiques . . . . .  | 88         |
| 4.3.2      | Considération de priorités hybrides . . . . .   | 92         |
| <b>4.4</b> | <b>Spécification des protocoles MAC CSMA sans collision du type « CANlike »</b> .                           | <b>97</b>  |
| 4.4.1      | Idées de base : Adaptation du protocole CAN au contexte sans fil . . . . .                                  | 97         |
| 4.4.2      | Notion de signal de synchronisation . . . . .   | 97         |
| 4.4.3      | Evaluation des bits de l'ID . . . . .   | 99         |
| 4.4.4      | Les protocoles . . . . .  | 100        |
| 4.4.5      | Bilan des paramètres du protocole CANlike . . . . .   | 101        |
| <b>4.5</b> | <b>Conclusion</b> . . . . .   | <b>101</b> |

---



## 4.1 Introduction

Les réseaux sans fil, et en particulier les réseaux locaux sans fil (Wireless Local Area Network : WLAN), sont aujourd'hui de plus en plus utilisés dans différents secteurs de la société et, en particulier, le secteur industriel où on a de nombreuses applications temps réel qui requièrent des garanties en termes de Qualité de Service.

Dans ce contexte, les protocoles MAC, qui implantent l'ordonnement des trames (dans lesquelles les messages des applications sont encapsulés) dans un canal radio partagé par plusieurs stations, ont un rôle primordial.

Deux importants types de protocoles MAC sont le type TDMA (Time Division Multiple Access) et le type CSMA (Carrier Sense Multiple Access). Le type CSMA est totalement distribué alors que le type TDMA requiert une certaine centralisation. En particulier, par rapport au type TDMA, dans le type CSMA les modifications d'un réseau (par exemple l'ajout ou le retrait de stations) sont plus faciles à mettre en œuvre. De plus le type CSMA est tout à fait adapté pour le trafic sporadique ce qui n'est pas le cas du type TDMA. Cependant l'inconvénient d'un strict type CSMA, c.-à-d. seulement basé sur le principe « Listen Before Send » est que des collisions peuvent survenir et, en conséquence, il ne peut pas fournir des garanties de Qualité de Service (QoS). Les protocoles IEEE 802.11 (WiFi) et IEEE 802.15.4 (ZigBee) (que nous avons brièvement présentés dans la section 1.4) sont les principaux exemples basés sur le type CSMA (malgré la présence de l'attribut CA (Collision Avoidance) on a toujours des collisions). Si on veut obtenir des garanties de QoS (par exemple, une borne supérieure du délai de transmission) et, donc être à même de pouvoir considérer des trafics avec des contraintes temporelles, on a besoin de la propriété « sans collision » ce qui demande d'incorporer au strict type CSMA des mécanismes supplémentaires.

Notons que des travaux ont déjà été réalisés pour prendre en compte du trafic temps réel dans le cadre de protocoles basés sur le type CSMA. Nous voulons mentionner tout d'abord, des travaux qui proposent des extensions du cadre opérationnel du standard IEEE 802.11 DCF [Crow 1997] qui utilise les concepts de IFS (InterFrame Space), de fenêtre de contention (« Contention Window ») et de Backoff. Ce standard ne peut cependant pas éviter des situations de collision et en outre ne différencie pas les types de flux de messages. Les principales idées des travaux, pour considérer du trafic temps réel, consistent en l'utilisation, d'une part, d'un IFS plus court pour le trafic temps réel que pour les autres trafics et, d'autre part, de priorités qui sont associées aux messages de données temps réel. Ces priorités peuvent représenter différentes caractéristiques des messages de données temps réel : le temps d'attente avant de pouvoir accéder au canal de transmission [Sobrinho 1996, Sobrinho 1999], le « deadline » du message de donnée [Pal 2002]. Les priorités sont représentées par des messages d'annonce qui sont des messages envoyés avant les messages de données temps réel proprement dits. Ces messages d'annonce sont des messages dits « messages BlackBurst » (séquences d'impulsions d'énergie dont la longueur est proportionnelle à la priorité des messages de données temps réel). L'inconvénient des messages BlackBurst est que, si nous avons un grand nombre de priorités, la longueur des messages BlackBurst, qui représentent des priorités élevées, peut être très élevée, ce qui induit des délais importants.

D'autres travaux [You 2003b, You 2003a, Pereira 2007a, Pereira 2007b], toujours basés sur la

notion de priorité, évitent l'inconvénient de la longueur des messages BlackBurst en considérant une adaptation au contexte sans fil de la technique utilisée dans le réseau filaire CAN. Nous appelons encore « CANlike » cette technique.

Après une lecture très approfondie des travaux mentionnés, nous estimons que tous ces travaux ne montrent pas clairement et rigoureusement, d'une part, la problématique spécifique d'un protocole du strict type CSMA (potentialité de la collision : pourquoi, quand et comment) et, d'autre part, les différentes modalités (et leurs contraintes) pour implanter des priorités qui doivent permettre d'éliminer l'occurrence de collisions. C'est la motivation de ce chapitre dans lequel notre objectif est de faire une présentation pédagogique des différents protocoles de type CSMA sans collision pour permettre la gestion de trafic temps réel dans le contexte d'un réseau local sans fil mono-canal radio. Le réseau local est basé sur une topologie physique totalement interconnectée, c.-à-d. toute station est dans le « transmission range » des autres stations et donc toute station peut envoyer un message à une autre station en un seul saut (« un hop »); on n'a pas le problème de la station cachée et de la station exposée (cas de la topologie physique multi-sauts) [Xu 2002].

Ce chapitre comprend trois parties : la première partie concerne des préliminaires qui sont nécessaires pour bien comprendre et spécifier complètement la problématique ; la deuxième partie présente la spécification de protocoles basés sur des messages BlackBurst ; la troisième partie présente la spécification du protocole dit « CANlike ».

## 4.2 Préliminaires

Nous présentons les connaissances de base qui concernent la couche physique et la sous-couche MAC basée sur un protocole CSMA. Ces connaissances sont fondamentales pour la spécification de protocoles MAC de type CSMA sans collision.

### 4.2.1 La couche physique

#### Sur les transceivers

Un transceiver (Transmetteur-Récepteur) est le composant de la couche physique d'un nœud directement en prise avec le canal radio pour l'émission et la réception des trames échangées avec les autres nœuds.

Dans un contexte réseau sans fil (contrairement au contexte filaire), on ne peut pas simultanément émettre et recevoir sur un canal. Un transceiver peut être dans trois états : Transmetteur, Récepteur, Dormant (nous ne considérons pas ici l'état Dormant qui est utilisé pour limiter l'énergie dépensée). Par défaut, le transceiver est dans l'état Récepteur (état d'écoute du canal).

Deux attributs temporels caractérisent le fonctionnement d'un transceiver : le temps de test du canal (« Sensing Time »  $\tau_{ST}$ ), le temps de retournement (« Turnaround Time »  $\tau_{TT}$ ). Le temps  $\tau_{ST}$  intervient lorsque la couche MAC doit émettre une trame et donc il faut tester l'état du canal afin de voir (pour pouvoir émettre) si le canal est libre ou occupé c.-à-d. si l'énergie détectée durant le temps  $\tau_{ST}$  est inférieure ou supérieure à un seuil d'énergie préfixé (cette activité est appelée le

« Clear Channel Assessment » CCA). Le temps de retournement du transceiver  $\tau_{TT}$  est le temps pour passer de l'état récepteur à l'état transmetteur ou inversement.

Suite au CCA, si le canal est détecté occupé, le transceiver reste dans l'état Récepteur. Par contre si le canal est détecté libre, le transceiver passe (après le temps  $\tau_{TT}$ ) dans l'état Transmetteur afin de permettre la transmission de la trame de la couche MAC. Après la transmission de la trame, le transceiver repasse dans l'état Récepteur.

### Sur les canaux radios

Un canal radio, qui est le médium de transmission entre les transceivers des nœuds, est défini par deux paramètres (bande de fréquence, atténuation des signaux électriques représentant les trames émises) et est caractérisé relativement à la fonction transmetteur du transceiver d'un nœud par deux portées (« Ranges ») : « Carrier Sense Range » ( $R_{CS}$ ) et « Transmission Range » ( $R_{TX}$ ).

La portée « Carrier Sense Range » ( $R_{CS}$ ) représente la distance maximale entre deux nœuds  $x$  et  $y$  où l'envoi d'une trame MAC par le nœud  $x$  déclenche à l'arrivée de cette trame dans le nœud  $y$  situé à cette distance l'état canal occupé (cette portée dépend de la sensibilité de l'antenne dans le nœud  $y$ ). Donc si le nœud  $y$  voulait transmettre durant l'arrivée de la trame, il ne le pourrait pas (énergie supérieure au seuil fixé durant  $\tau_{ST}$ ). Notons que, dans le contexte CSMA, un nœud ne peut démarrer une transmission que s'il a détecté le canal libre.

La portée « Transmission Range » ( $R_{TX}$ ) représente la distance maximale entre deux nœuds  $x$  et  $y$  où une trame MAC envoyée par le nœud  $x$  est bien reçue et décodée par le nœud  $y$ . Cette portée dépend d'un rapport signal sur bruit suffisant, en particulier, pour permettre à la couche physique du nœud  $y$  de bien détecter la partie du début de la trame MAC qui contient des informations relatives à la couche physique (en-tête de synchronisation qui permet de se synchroniser sur le flux des bits série qui arrivent ; en-tête physique qui contient le nombre d'octets de la trame MAC proprement dite). Suite à cette détection, la couche physique pourra passer la trame MAC reçue à la couche MAC.

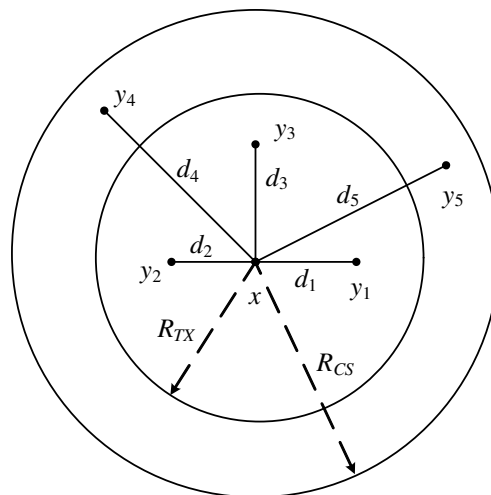
On a évidemment  $R_{CS} > R_{TX}$ , la portée  $R_{TX}$  traduisant des conditions plus dures (on peut décoder ce qui arrive) que la portée  $R_{CS}$  (on trouve simplement un bruit).

Les portées  $R_{TX}$  et  $R_{CS}$  relatives à un nœud transmetteur  $x$  sont modélisées par deux cercles centrés sur le nœud  $x$  et de rayon respectivement  $R_{TX}$  et  $R_{CS}$ .

Nous donnons sur la figure 4.1 un exemple de modélisation des portées  $R_{TX}$  et  $R_{CS}$  associées à un nœud  $x$ . Le nœud  $x$  peut échanger directement (sur un saut) des trames avec les nœuds  $y_1$ ,  $y_2$ ,  $y_3$ . Les trames peuvent être bien reçues dès le premier envoi (s'il n'y a pas de collision avec les envois de  $y_1$ ,  $y_2$  et  $y_3$ ) ou après un processus de détection de collision et de réémission suite à un mécanisme de Backoff (technique CSMA/CA). Le nœud  $x$  ne peut pas envoyer des trames aux nœuds  $y_4$  et  $y_5$  mais rend, par contre lorsqu'il envoie une trame à  $y_1$  ou  $y_2$  ou  $y_3$ , le canal occupé au niveau des nœuds  $y_4$  et  $y_5$  (les empêchant de pouvoir émettre).

### Sur des topologies

On peut avoir plusieurs types de topologies : des topologies mono-saut (tout nœud communique directement avec les autres nœuds) et des topologies multi-sauts (des nœuds peuvent communiquer avec d'autres nœuds à travers des nœuds intermédiaires). Les topologies multi-sauts

FIGURE 4.1 – Portées  $R_{TX}$  et  $R_{CS}$ .

s'étendent sur plusieurs  $R_{CS}$ . Les chaînes sont des exemples de topologies multi-sauts [Ng 2004], [Gao 2005]. Dans les topologies multi-sauts on a, en particulier, les problématiques du nœud caché et du nœud exposé.

Ici nous ne considérons pas les topologies multi-sauts. Nous considérons uniquement une topologie maillée à un saut et totalement inter-connectée.

Dans ce cas, nous considérons que le  $R_{TX}$  à choisir pour tous les nœuds est donné par le  $R_{TX_i}$  nécessaire pour le saut  $i$  le plus long :  $R_{TX} = \max_i \{R_{TX_i}\}$ . Comme on veut que tout nœud dialogue avec tout autre nœud et qu'il n'y ait pas de perturbations venant d'autres nœuds, on peut prendre  $R_{CS} = R_{TX}$ .

Nous représentons sur la figure 4.2 une topologie maillée ABCD (4 nœuds A, B, C, D) qui se trouve dans un domaine délimité par l'intersection des domaines définis par les cercles de rayon  $R_{TX} = R_{CS}$  associés aux différents nœuds (chaque cercle est centré sur un nœud et le rayon est légèrement supérieur au plus long saut AC). Cette topologie représente bien le cas d'un réseau local où la population est bien définie (A, B, C, D) et où on suppose qu'il n'y a pas de perturbations extérieures (nœuds distants).

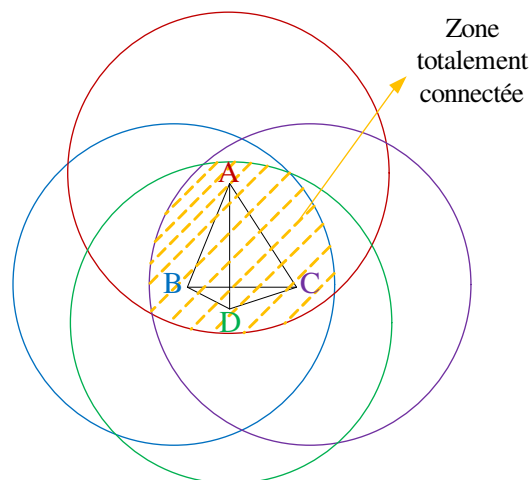


FIGURE 4.2 – Topologie à 1 saut totalement connectée.

## 4.2.2 La sous-couche MAC

### Concept de « Fenêtre temporelle d'ambiguïté »

Ce concept est introduit afin de considérer l'ambiguïté dans le contexte CSMA de l'expression « canal libre dans une entité MAC d'un nœud à un instant  $t$  », expression qui est à la base d'une transmission effectuée par cette entité MAC. En effet cette expression est ambiguë parce qu'elle exprime seulement une vue locale alors que le canal est géographiquement distribué et donc une vue locale peut être différente de la vue globale ce qui crée les situations de collisions.

Nous caractérisons quantitativement cette ambiguïté en introduisant le concept de fenêtre d'ambiguïté qui représente la durée maximale possible entre la décision d'envoi d'une trame par l'entité MAC d'un nœud et l'inévitabilité de l'occurrence d'une collision sur cette trame. La caractérisation quantitative doit nécessairement considérer les paramètres des transceivers ( $\tau_{ST}$ ,  $\tau_{TT}$ ) et le temps de propagation maximal entre deux nœuds c.-à-d. entre les 2 nœuds les plus éloignés (appelons  $\tau_{PT}$  ce temps de propagation).

La figure 4.3 représente le point de vue théorique quantitatif du concept de « fenêtre temporelle d'ambiguïté ».

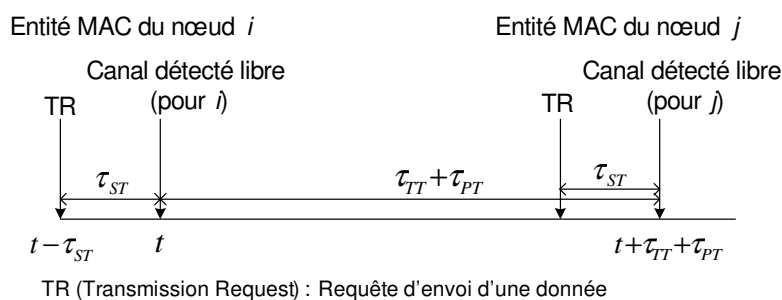


FIGURE 4.3 – Fenêtre temporelle d'ambiguïté.

L'entité MAC  $i$  reçoit une Requête (« Transmission Request » : TR), du niveau supérieur à l'instant  $(t - \tau_{ST})$  pour envoyer une donnée et teste le canal pendant la durée  $\tau_{ST}$ . On suppose que le canal est détecté libre à l'instant  $t - \tau_{ST} + \tau_{ST} = t$  et donc l'entité MAC  $i$  décide d'envoyer une trame contenant la donnée. Après le temps de retournement ( $\tau_{TT}$ ), la trame est envoyée et arrive au niveau de l'entité MAC  $j$  la plus éloignée juste à l'instant  $(t + \tau_{TT} + \tau_{PT})$ . Si l'entité MAC  $j$  est juste en train de finir de tester l'état du canal à cet instant là (suite à une requête TR pour envoyer une donnée à l'instant  $(t + \tau_{TT} + \tau_{PT} - \tau_{ST})$ ) et comme le canal a été vu libre durant  $\tau_{ST}$ , l'entité MAC  $j$  décide d'envoyer une trame à l'instant  $(t + \tau_{TT} + \tau_{PT})$ . On a donc une situation inévitable de future collision physique. Le début de l'arrivée de la trame venant de l'entité MAC  $i$  coïncide avec l'instant où le transceiver de l'entité MAC  $j$  se retourne pour se mettre en position de transmetteur. Durant le temps de retournement  $\tau_{TT}$ , l'entité MAC  $j$  est aveugle. La collision physique ne se produira effectivement qu'après le retournement.

La durée  $(\tau_{TT} + \tau_{PT})$  représente ce que nous appelons la « fenêtre temporelle d'ambiguïté ».

### Sur les priorités associées aux messages des flux de données

On peut avoir des priorités statiques et des priorités hybrides comme dans l'étude sur le réseau CAN (Chap. 2).

Dans le cadre de la mise en œuvre de la technique BlackBurst, la priorité est représentée par un message d'annonce BlackBurst envoyé avant la trame de données.

Dans le cadre de la mise en œuvre de la technique « CANlike », la priorité est dans le champ ID associé à la trame de données comme dans le réseau filaire CAN.

### Notion de tournoi relatif au transfert d'une trame de données

Avec la technique BlackBurst, le tournoi concerne la compétition entre les messages d'annonces qui véhiculent les priorités des trames de données candidates au transfert.

Avec la technique CANlike, le tournoi se résume à la comparaison des champs ID envoyés au début des trames de données.

#### Mise en œuvre d'un tournoi

Un tournoi a pour objectif de transformer une situation qui aurait été une situation de collision en une situation vainqueur-vaincu(s). Dans le cas de la technique BlackBurst, l'entité MAC vainqueur d'un tournoi (celle qui a le message d'annonce le plus long) est alors autorisée à envoyer sa trame de données. Dans le cas de la technique CANlike, l'entité MAC vainqueur (celle qui a le plus longtemps des bits dominants) continue le transfert de sa trame de données.

*Le bon déroulement d'un tournoi est conditionné par les durées des messages d'annonce (technique BlackBurst) et par la durée des bits du champ ID (technique CANlike). C'est la considération de la fenêtre temporelle d'ambiguïté qui permet la détermination de ces durées.*

## 4.3 Spécification des protocoles MAC CSMA sans collision basés sur des messages BlackBurst

Nous considérons successivement les cas des priorités statiques et des priorités hybrides. Dans le cas de priorités statiques, un seul message d'annonce est associé à une trame de données. Soit  $k$  la priorité d'une trame de données, le message d'annonce qui lui est associé a une durée  $kTBB$ ,  $TBB$  représentant le temps (durée) d'un BlackBurst. Ce cas est présenté dans [Nguyen Xuan 2010b].

Dans le cas des priorités hybrides, deux messages d'annonce sont associés à une trame de données : le message d'annonce représentant la partie priorité statique (noté message d'annonce statique) et le message d'annonce représentant la partie priorité dynamique (noté message d'annonce dynamique). Soit  $k_s$  et  $k_d$  respectivement la priorité statique et la priorité dynamique d'une trame de données et notons  $TBB_s$  et  $TBB_d$  la durée d'un BlackBurst concernant respectivement la priorité statique et la priorité dynamique. Les messages d'annonce statique et d'annonce dynamique de cette trame de données ont respectivement une durée  $k_s TBB_s$  et une durée  $k_d TBB_d$ .

### 4.3.1 Considération de priorités statiques

#### 4.3.1.1 Etapes du protocole

Le protocole basé sur la priorité statique (noté BBsta) se compose de 4 étapes représentées sur la figure 4.4 :

**Etape 1.** Toutes les entités MAC, qui sont candidates pour l'envoi d'une trame de données, doivent d'abord observer, l'état canal libre pendant un temps donné. Nous appelons TOBS1 ce temps d'**o**bservation (nous justifions ci-dessous le rôle de cette observation initiale).

**Etape 2.** Toutes ces entités MAC démarrent le tournoi par l'envoi du message d'annonce (multiple de TBB) représentant la priorité de la trame de données ;

**Etape 3.** suite à l'envoi du message d'annonce, les entités MAC observent l'état du canal pendant un temps TOBS2. Ce temps observation permet à chaque entité MAC de se déclarer soit gagnante (si le canal est dans l'état libre durant TOBS2) soit perdante (si le canal est occupé durant TOBS2).

- il y a une seule entité gagnante. C'est l'entité MAC qui a envoyé le message BlackBurst le plus long (plus grande priorité) ;
- les entité MAC perdantes reviennent à l'état d'observation du médium libre pendant TOBS1.

**Etape 4.** La seule entité MAC gagnante envoie sa trame de données.

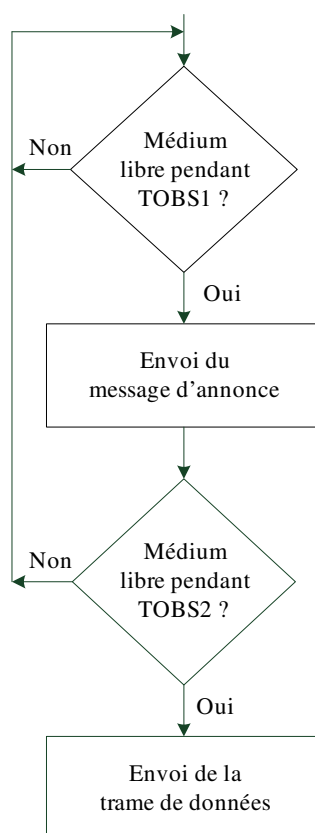


FIGURE 4.4 – Etapes du protocole BBsta.

Remarques :

1. il est important de noter les commutations qui se produisent dans les transceivers aux frontières des différentes étapes :
  - à la fin de l'étape 1 c.-à-d. après TOBS1, les entités MAC doivent se retourner ( $\tau_{TT}$ ) pour pouvoir émettre le message d'annonce ;
  - à la fin de l'étape 2, les entités MAC doivent se retourner ( $\tau_{TT}$ ) afin de pouvoir observer l'état du canal ;
  - à la fin de l'étape 3, l'entité MAC gagnante doit se retourner ( $\tau_{TT}$ ) afin d'émettre la trame de données ;
2. le rôle de la durée TOBS1 est d'éviter des intrusions durant le tournoi et avant le transfert de la trame de données par l'entité MAC gagnante. On doit avoir  $TOBS1 > 2\tau_{TT} + TOBS2$  ;  $2\tau_{TT}$  représente les temps de retournement qui encadrent TOBS2 ;
3. les grandeurs TBB, TOBS2, TOBS1 sont les grandeurs fondamentales qu'il faut déterminer pour spécifier le protocole ;

#### 4.3.1.2 Détermination de TBB

Cette détermination passe tout d'abord par la considération, d'une part, de la fenêtre d'ambiguïté (Fig. 4.3) et d'autre part, des deux situations extrêmes qui pourraient générer une collision sur la trame que l'entité MAC  $i$  déciderait d'envoyer à l'instant  $t$  :

- une collision au moment de la décision d'envoi de cette trame par l'entité MAC  $i$  qui résulte de l'envoi d'une trame par l'entité MAC  $j$  à l'instant  $(t - (\tau_{TT} + \tau_{PT}))$  ;
- une collision au début de l'arrivée de cette trame dans l'entité MAC  $j$  qui résulte de la décision d'envoi d'une trame par l'entité MAC  $j$  à l'instant  $(t + (\tau_{TT} + \tau_{PT}))$ .

Nous devons ensuite trouver les contraintes sur la valeur de TBB qui transforment ces situations de collision en une situation « gagnant-perdant ». Pour ceci, il faut considérer les scénarios suivants :

- l'entité MAC  $i$ , qui a une priorité  $(k + 1)$  (c.-à-d. le message d'annonce a une longueur  $(k + 1)TBB$ ) décide d'envoyer à l'instant  $t$  ce message (début de l'émission à  $(t + \tau_{TT})$  ;
- l'entité MAC  $j$ , qui a une priorité immédiatement inférieure à celle de l'entité MAC  $i$  (c.-à-d. une priorité  $k$  et donc le message d'annonce a une longueur  $kTBB$ ) décide d'envoyer ce message soit à l'instant  $t - (\tau_{TT} + \tau_{PT})$  (début de l'émission à  $t - (\tau_{TT} + \tau_{PT}) + \tau_{TT}$ ), soit à l'instant  $t + (\tau_{TT} + \tau_{PT})$  (début de l'émission à  $t + (\tau_{TT} + \tau_{PT}) + \tau_{TT}$ ), c'est à dire soit en avance soit en retard d'une fenêtre d'ambiguïté.

Le scénario, qui exprime la plus grande contrainte pour que la situation « entité MAC  $i$  gagnante - entité MAC  $j$  perdante » se produise, est celui où l'entité MAC  $j$  envoie en retard ; cette situation représente le cas où le message d'annonce de l'entité MAC  $j$  a le plus de chance de dépasser, quand il arrive dans l'entité MAC  $i$ , le message d'annonce que l'entité MAC  $i$  envoie. Nous considérons uniquement ce scénario qui est représenté sur la figure 4.5 où les deux diagrammes temporels au-dessus représentent les envois de messages d'annonces des entités MAC  $i$  et  $j$  (les instants de fin d'envoi sont  $t_1$  et  $t_2$ ) et les deux diagrammes temporels au-dessous représentent les propagations de messages d'annonces envoyées par les entités MAC  $i$  et  $j$  qui arrivent respectivement aux entités MAC  $j$  et  $i$  (les instants de fin de propagation sont  $t_4$  et  $t_3$ ).



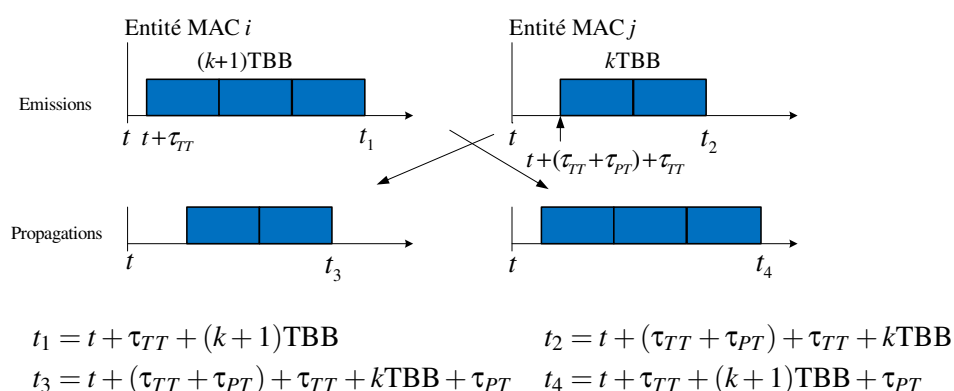


FIGURE 4.5 – BBsta : Scénario d'évaluation de TBB.

La situation « entité MAC  $i$  gagnante - entité MAC  $j$  perdante » est caractérisée par les contraintes suivantes :

- au niveau de l'entité MAC  $i$  : le message d'annonce de l'entité MAC  $i$  (gagnante) ne peut pas être dépassé par le message d'annonce de l'entité MAC  $j$  (perdante) soit :

$$t_1 \geq t_3$$

$$\Rightarrow t + \tau_{TT} + (k+1)TBB \geq t + (\tau_{TT} + \tau_{PT}) + \tau_{TT} + kTBB + \tau_{PT}$$

$$\Rightarrow TBB \geq 2\tau_{PT} + \tau_{TT} \quad (4.1)$$

- au niveau de l'entité MAC  $j$  : le message d'annonce de l'entité MAC  $i$  (gagnante) doit dépasser le message d'annonce de l'entité MAC  $j$  (perdante) et l'entité MAC  $j$  (perdante) doit l'observer (durée  $\tau_{ST}$ ) :

$$t_4 \geq t_3 + (\tau_{TT} + \tau_{ST}) \quad (4.2)$$

$$\Rightarrow t + \tau_{TT} + (k+1)TBB + \tau_{PT} \geq t + (\tau_{TT} + \tau_{PT}) + \tau_{TT} + kTBB + (\tau_{TT} + \tau_{ST})$$

$$\Rightarrow TBB \geq 2\tau_{TT} + \tau_{ST} \quad (4.3)$$

La dernière composante ( $\tau_{TT} + \tau_{ST}$ ) dans l'équation 4.2 représente le fait que l'entité MAC  $j$  (perdante) doit observer qu'elle a perdu (après l'envoi de son message d'annonce, elle doit se retourner ( $\tau_{TT}$ ) pour passer dans l'état d'écoute et tester pendant le temps  $\tau_{ST}$  que le canal est occupé).

Donc compte tenu des deux contraintes (Equas. 4.1 et 4.3), la contrainte à considérer est :

$$TBB \geq 2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$$

Nous considérons la valeur minimale :

$$TBB = 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \quad (4.4)$$

### 4.3.1.3 Détermination de TOBS2, TOBS1

#### Le temps d'observation TOBS2

L'évaluation de TOBS2 nécessite de calculer, en considérant la valeur de TBB, la durée du dépassement, dans l'entité MAC  $j$  perdante (annonce  $kTBB$ ) du message d'annonce émis par cette entité MAC par le message d'annonce venant de l'entité MAC  $i$  gagnante (annonce  $(k+1)TBB$ ).

Durée de dépassement =

$$\begin{aligned}
 &= \text{fin d'arrivée du message d'annonce de } i (t_4) - \text{fin d'envoi du message d'annonce de } j (t_2) \\
 &= (t + \tau_{TT} + (k+1)TBB + \tau_{PT}) - (t + (\tau_{TT} + \tau_{PT}) + \tau_{TT} + kTBB) \\
 &= 2\tau_{PT} + \tau_{TT} + \tau_{ST}
 \end{aligned}$$

Comme dans cette durée du dépassement, on a le temps de retournement  $\tau_{TT}$  qui est nécessaire avant de pouvoir observer le canal, on a donc :

$$\text{TOBS2} = 2\tau_{PT} + \tau_{ST} \quad (4.5)$$

Notons que comme ce temps d'observation TOBS2 est également utilisé par l'entité gagnante pour observer le canal libre avant de pouvoir émettre la trame de données, l'émission de cette trame de données débute quand le canal est globalement libre (il n'y a plus de résidu du tournoi au niveau d'aucune entité MAC).

#### Le temps d'observation TOBS1

En ce qui concerne TOBS1, on doit avoir  $\text{TOBS1} > 2\tau_{TT} + \text{TOBS2}$  (Sect. 4.3.1.1), plus précisément, TOBS1 doit être supérieur à  $(2\tau_{TT} + \text{TOBS2})$  d'au moins un  $\tau_{ST}$  pour pouvoir détecter correctement l'état du canal. On a donc  $\text{TOBS1} \geq 2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST}$ . Nous prenons ici :

$$\text{TOBS1} = 2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST} \quad (4.6)$$

### 4.3.1.4 Bilan du protocole BBsta

Nous présentons sur le tableau 4.1 les valeurs des paramètres (TBB, TOBS1, TOBS2) du protocole BBsta (à partir des équations 4.4, 4.5 et 4.6).

Ces paramètres et les commutations (retournements) qui se produisent entre les étapes permettent de calculer la durée de compétition (ou autrement dit le temps d'accès au médium), noté  $\tau_a$ , de l'entité MAC vainqueur (en supposant qu'elle a la priorité  $k$ ) :

$$\begin{aligned}
 \tau_a &= \text{TOBS1} + \tau_{TT} + kTBB + \tau_{TT} + \text{TOBS2} + \tau_{TT} \\
 \tau_a &= 4\tau_{PT} + 5\tau_{TT} + 3\tau_{ST} + kTBB
 \end{aligned}$$

| TBB                                   | TOBS1                                  | TOBS2                    |
|---------------------------------------|--|--------------------------|
| $2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$ | $2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST}$ | $2\tau_{PT} + \tau_{ST}$ |

TABLE 4.1 – Paramètres du protocole BBsta.

## 4.3.2 Considération de priorités hybrides

### 4.3.2.1 Etapes du protocole

Appelons ce protocole BBhyb. Nous avons les étapes (Fig. 4.6) suivantes :

**Etape 1.** Toutes les entités MAC, qui sont candidates pour l'envoi d'une trame de données doivent d'abord observer l'état canal libre pendant une durée notée TOBS1 (comme pour le cas du protocole basé sur les priorités statiques et pour les mêmes raisons).

**Etape 2.** Toutes ces entités MAC démarrent un tournoi par l'envoi de leurs messages d'annonce dynamique (c.-à-d. on examine d'abord l'urgence de transmission des messages).

**Etape 3.** Le tournoi dans l'étape 2, qui concerne les messages d'annonce dynamique, peut avoir une entité MAC gagnante ou plusieurs entités MAC gagnantes. En effet comme les priorités dynamiques n'ont pas la propriété d'unicité on peut avoir un tournoi où les messages de plusieurs entités MAC ont les mêmes priorités dynamiques avec la valeur la plus grande. Donc toutes ces entités MAC doivent être gagnantes de ce tournoi c.-à-d. elles doivent observer l'état canal libre pendant un temps TOBS2.

Nous montrons dans la prochaine sous-section 4.3.2.2 que dans une telle situation si on veut avoir pour toutes les entités MAC gagnantes la condition « canal libre durant TOBS2 », il est nécessaire, compte tenu de l'asynchronisme entre les entités MAC, de définir la notion de « temps de garde  $G$  » associée aux messages d'annonce dynamique (un temps de garde est un temps où une entité MAC est aveugle). On montre en particulier que le temps de garde contient la durée de temps de retournement  $\tau_{TT}$ .

Les perdantes de ce tournoi doivent attendre de nouveau une observation de l'état canal libre de durée TOBS1 (c.-à-d. retourner à l'étape 1) pour redémarrer le tournoi.

**Etape 4.** Cette étape concerne, pour le ou les gagnants du tournoi relatif aux messages d'annonce dynamique, le tournoi basé sur les messages d'annonce statique. Les entités MAC gagnantes trouvées dans l'étape 3 envoient les messages d'annonce statique.

**Etape 5.** Suite à l'envoi des messages d'annonce statique dans l'étape 4, les entités MAC observent le canal pendant une durée TOBS3. Il y a une seule station gagnante, celle qui a envoyé le message BlackBurst statique le plus long. Toutes les entités MAC perdantes doivent retourner à l'étape 1 et elles doivent attendre d'observer l'état canal libre pendant la durée TOBS1 afin de pouvoir redémarrer la compétition.

**Etape 6.** La seule entité MAC gagnante après l'étape 5 envoie sa trame de données.

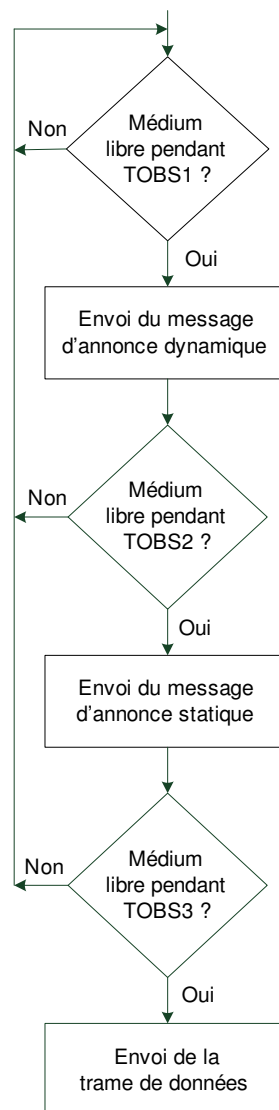


FIGURE 4.6 – Étapes du protocole BBhyb

*Remarques :* Concernant les commutations qui se produisent dans les transceivers aux frontières des différentes étapes, nous avons :

- à la fin de l'étape 1 (c.-à-d. après TOBS1), les entités MAC doivent se retourner ( $\tau_{TT}$ ) pour pouvoir émettre les messages d'annonce dynamique ;
- à la fin de l'étape 2 (c.-à-d. après l'envoi des messages d'annonce dynamique suivis du temps de garde  $G$ ), on n'a pas de retournement puisqu'il est inclus dans le temps de garde ;
- à la fin de l'étape 3 (c.-à-d. après TOBS2), les entités MAC gagnantes du tournoi basé sur les priorités dynamiques, se retournent pour émettre les messages d'annonce statique ;
- à la fin de l'étape 4 (c.-à-d. après l'envoi des messages d'annonce statique), les entités MAC doivent se retourner ( $\tau_{TT}$ ) pour pouvoir observer l'état du médium pendant TOBS3 ;
- à la fin de l'étape 5 (c.-à-d. après TOBS3), la seule entité MAC vainqueur doit se retourner pour entreprendre le transfert d'une trame de données.

### 4.3.2.2 Nécessité d'un temps de garde associé aux messages d'annonce dynamique

Considérant la situation particulière où on a plusieurs entités MAC qui doivent gagner à l'étape 2 (car elles ont la même priorité dynamique qui est la valeur la plus élevée des entités MAC qui font le tournoi) et compte tenu de l'asynchronisme de l'envoi des messages d'annonce dynamique (asynchronisme caractérisé par le concept de fenêtre temporelle d'ambiguïté), l'occupation du canal vue par une entité MAC, qui doit gagner, peut être plus longue que le message d'annonce dynamique que cette entité MAC a envoyé. Ceci est dû au chevauchement entre les messages d'annonce dynamique des entités MAC qui doivent gagner.

Compte tenu de ces situations, nous devons mettre un temps de garde (temps où l'entité est aveugle) après l'envoi d'un message d'annonce dynamique par une entité MAC afin d'avoir, après ce temps de garde, l'état canal libre pour toutes les entités MAC qui doivent être gagnantes.

Le temps de garde nécessaire est évalué, d'une part, en considérant le concept de fenêtre temporelle d'ambiguïté et, d'autre part, en analysant les scénarios suivants où on a deux entités MAC ( $i$  et  $j$ ) qui ont la même priorité dynamique (priorité qui est la plus élevée du tournoi qui se déroule ; cette priorité est noté  $k_d$  et donc  $k_d TBB_d$  est la longueur du message d'annonce dynamique) :

- l'entité MAC  $i$  décide d'envoyer le message  $k_d TBB_d$  à l'instant  $t$  ;
- l'entité MAC  $j$  décide d'envoyer le message  $k_d TBB_d$ 
  - plus tard (cas 1), à la date limite ( $t + (\tau_{TT} + \tau_{PT})$ ) ;
  - plus tôt (cas 2), à la date limite ( $t - (\tau_{TT} + \tau_{PT})$ ).

Dans le cas 1 (Fig. 4.7), la date de fin de propagation, à l'entité MAC  $i$ , du message d'annonce émis par l'entité MAC  $j$  est supérieure à la date de fin de l'émission du message d'annonce de  $i$ . Cette différence caractérise le temps de garde :  $G = t_3 - t_1 = 2\tau_{PT} + \tau_{TT}$ .

Dans le cas 2, la date de fin de propagation à l'entité MAC  $j$  du message d'annonce émis par l'entité MAC  $i$  est supérieure à la date de fin de l'émission du message d'annonce de  $j$ . La différence est encore  $2\tau_{PT} + \tau_{TT}$ . On a donc le temps de garde :

$$G = 2\tau_{PT} + \tau_{TT} \quad (4.7)$$

Donc, chaque entité MAC ajoutera, après l'envoi d'un message d'annonce dynamique, un temps de garde ( $G = 2\tau_{PT} + \tau_{TT}$ ). Comme la durée  $\tau_{TT}$  est comprise dans  $G$ , on considère que le retournement se fait pendant le temps de garde et, donc suite au temps de garde, l'entité MAC peut observer pendant TOBS2.

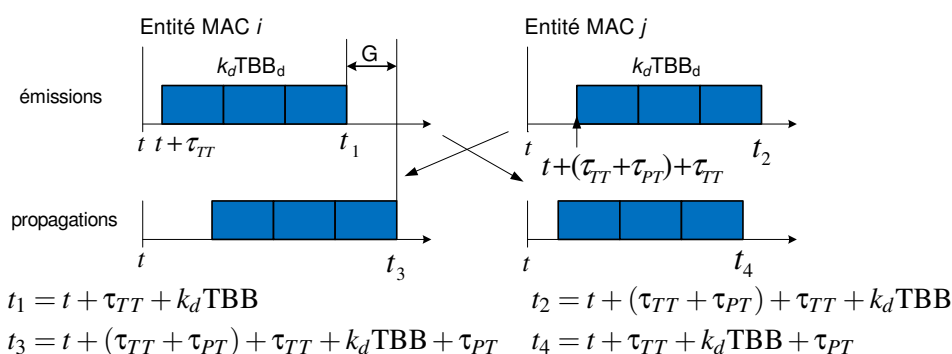


FIGURE 4.7 – BBhyb : Scénario d'évaluation de  $G$  - cas 1 ( $G = t_3 - t_1 = 2\tau_{PT} + \tau_{TT}$ ).

### 4.3.2.3 Détermination de $TBB_d$

Nous reprenons le type d'analyse qui a été fait à la sous-section 4.3.1.2 en considérant uniquement le scénario qui exprime la contrainte la plus dure et en intégrant la notion de temps de garde.

Le scénario considéré, en reprenant le concept de la fenêtre d'ambiguïté (Fig. 4.3), est représenté sur la figure 4.8 : l'entité MAC  $i$  décide d'envoyer à l'instant  $t$  le message d'annonce dynamique de longueur  $(k_d + 1)TBB_d$  ; l'entité MAC  $j$  décide d'envoyer à l'instant  $(t + (\tau_{TT} + \tau_{PT}))$  le message d'annonce dynamique de longueur  $k_d TBB_d$ .

L'entité MAC  $i$  doit être gagnante et l'entité MAC  $j$  doit être perdante. Donc on a les contraintes suivantes :

- dans l'entité MAC  $i$  ; le message d'annonce dynamique de cette entité, complété par le temps de garde, ne doit pas être dépassé par le message d'annonce arrivant de l'entité MAC  $j$  :

$$\begin{aligned} t_1 + G &\geq t_3 \\ \Rightarrow t + \tau_{TT} + (k_d + 1)TBB_d + G &\geq t + (\tau_{TT} + \tau_{PT}) + \tau_{TT} + k_d TBB_d + \tau_{PT} \\ \Rightarrow TBB_d &\geq 0 \end{aligned} \quad (4.8)$$

- dans l'entité MAC  $j$  ; le message d'annonce dynamique venant de l'entité MAC  $i$  doit dépasser le message d'annonce dynamique de l'entité MAC  $j$  augmenté de son temps de garde  $G$ , et l'entité MAC  $j$  doit observer ce dépassement (donc voir, après le retournement fait pendant le temps de garde, le médium occupé au moins pendant  $\tau_{ST}$ ) :

$$\begin{aligned} t_4 &\geq t_2 + G + \tau_{ST} \\ \Rightarrow t + \tau_{TT} + (k_d + 1)TBB_d + \tau_{PT} &\geq t + (\tau_{TT} + \tau_{PT}) + \tau_{TT} + k_d TBB_d + G + \tau_{ST} \\ \Rightarrow TBB_d &\geq \tau_{TT} + G + \tau_{ST} \\ \Rightarrow TBB_d &\geq 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \end{aligned} \quad (4.9)$$

Donc on voit qu'entre la contrainte (4.8) et la contrainte (4.9), c'est la contrainte (4.9) qui fixe la valeur de  $TBB_d$ . Nous considérons la valeur minimale de la contrainte (4.9) :

$$TBB_d = 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \quad (4.10)$$

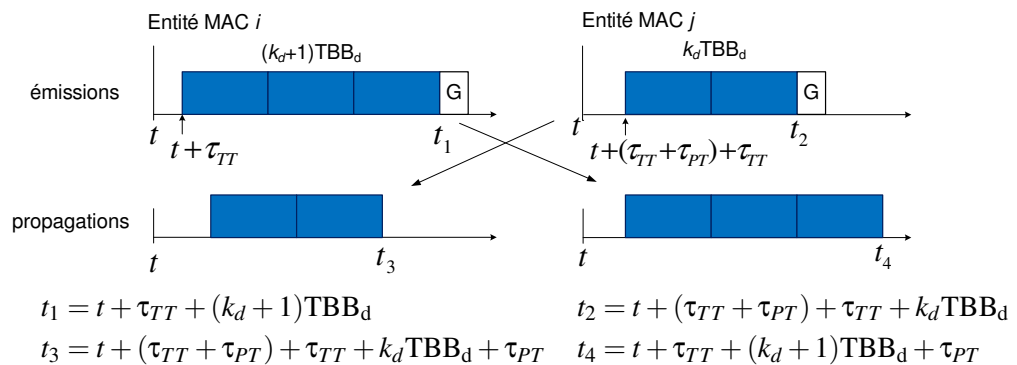


FIGURE 4.8 – BBhyb : Scénario d'évaluation de  $TBB_d$ .

#### 4.3.2.4 Détermination de TOBS1, TOBS2, TOBS3, TBB<sub>s</sub>

**Durée de TOBS2 :** Nous reprenons le type d'analyse fait à la section 4.3.1.3 (page 91). L'évaluation de TOBS2 nécessite de calculer la durée du dépassement, dans l'entité MAC  $j$  perdante (annonce  $k_d TBB_d$ ), du message d'annonce émis par cette entité MAC par le message d'annonce venant de l'entité MAC  $i$  gagnante (annonce  $(k_d + 1)TBB_d$ ) en considérant la valeur de  $TBB_d$  que l'on vient de calculer. Ce scénario est représenté sur la figure 4.8.

La durée de dépassement est  $t_4 - t_2 = 2\tau_{PT} + \tau_{TT} + \tau_{ST}$  (voir Fig. 4.8). Comme dans cette durée, on a le temps de garde  $G = 2\tau_{PT} + \tau_{TT}$ , on obtient :

$$TOBS2 = \tau_{ST} \quad (4.11)$$

**Durées de TBB<sub>s</sub> et TOBS3 :** Pour  $TBB_s$  et TOBS3 (c'est à dire les grandeurs qui concernent le tournoi entre les messages d'annonce statique et l'observation conséquente), on peut reprendre les résultats obtenus aux sections 4.3.1.2 et 4.3.1.3 :

$$TBB_s = 2\tau_{PT} + 2\tau_{TT} + \tau_{ST} \quad (4.12)$$

$$TOBS3 = 2\tau_{PT} + \tau_{ST} \quad (4.13)$$

Il est à noter que  $TBB_s = TBB_d$  (Equa. 4.10 et Equa. 4.12). Appelons donc TBB ces durées.

**Durée de TOBS1 :** Le rôle de TOBS1 est d'éviter des intrusions durant le tournoi. TOBS1 doit être supérieur aux durées des étapes d'observation (étapes 3 et 5) et le temps de garde ainsi que les temps de retournement qui encadrent les durées d'observation. Donc :

$$TOBS1 > \max\{G + TOBS2 + \tau_{TT} ; \tau_{TT} + TOBS3 + \tau_{TT}\}$$

On trouve que  $(G + TOBS2 + \tau_{TT})$  et  $(\tau_{TT} + TOBS3 + \tau_{TT})$  sont égaux à  $2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$ .

Donc  $TOBS1 > 2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$ . Plus précisément, le temps TOBS1 doit être supérieur à  $2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$  d'au moins un  $\tau_{ST}$  pour pouvoir détecter correctement l'état du médium. Nous avons donc  $TOBS1 \geq 2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST}$ . Nous prenons :

$$TOBS1 = 2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST} \quad (4.14)$$

#### 4.3.2.5 Bilan du protocole BBhyb

Le tableau 4.2 résume les paramètres du protocole BBhyb. Ces paramètres et les commutations (garde, retournements) qui se produisent entre les étapes permettent de calculer le temps d'accès ( $\tau_a$ ) au médium de l'entité MAC vainqueur qui a la priorité dynamique  $k_d$  et la priorité statique  $k_s$  :

$$\tau_a = TOBS1 + \tau_{TT} + k_d TBB + G + TOBS2 + \tau_{TT} + k_s TBB + \tau_{TT} + TOBS3 + \tau_{TT}$$

$$\tau_a = 6\tau_{PT} + 7\tau_{TT} + 4\tau_{ST} + (k_s + k_d)TBB$$

|     |                                       |       |  |
|-----|---------------------------------------|-------|--|
| TBB | $2\tau_{PT} + 2\tau_{TT} + \tau_{ST}$ | TOBS1 | $2\tau_{PT} + 2\tau_{TT} + 2\tau_{ST}$ |
| $G$ | $2\tau_{PT} + \tau_{TT}$              | TOBS2 | $\tau_{ST}$                            |
|     |                                       | TOBS3 | $2\tau_{PT} + \tau_{ST}$               |

TABLE 4.2 – Paramètres du protocole BBhyb.

## 4.4 Spécification des protocoles MAC CSMA sans collision du type « CANlike »

Tout d'abord, nous présentons les idées de base nécessaires pour spécifier ces protocoles : plus précisément, la nécessité de l'adaptation du protocole « Dominance/Binary Countdown » au contexte sans fil, la nécessité d'un signal de synchronisation et les caractéristiques des bits de l'identificateur. Ces éléments ont été publiés dans [Nguyen Xuan 2011b].

### 4.4.1 Idées de base : Adaptation du protocole CAN au contexte sans fil

Le protocole CAN, mis en œuvre sur réseau filaire, effectue la comparaison bit par bit des Identificateurs (ID) des trame émises et reçues (un bit dominant l'emporte sur un bit récessif).

Dans un réseau sans fil, on ne peut pas émettre et écouter en même temps et donc on doit définir un comportement différent pour une entité MAC et sa couche physique (PHY) suivant qu'elle a un bit dominant ou un bit récessif. Nous considérons ici la solution présentée dans [You 2003a], [Pereira 2007a] c'est à dire : *l'entité MAC envoie une porteuse (signal d'énergie) sur le canal lorsqu'elle a un bit dominant et écoute le canal quand elle a un bit récessif.*

En conséquence, dans toute entité MAC, le tournoi, basé sur les IDs des trames de données que l'on veut émettre, a les caractéristiques suivantes :

- on a un bit dominant ; l'entité MAC est gagnante par définition et donc elle continue le tournoi (prochain bit de l'ID) ;
- on a un bit récessif ; compte tenu que l'on écoute le canal soit on détecte une porteuse (bit dominant émis par une autre entité MAC) et donc on est perdant et on ne continue pas le tournoi, soit on ne détecte rien (donc il n'y pas de bit dominant émis par une autre entité MAC) et par conséquent on continue le tournoi.

### 4.4.2 Notion de signal de synchronisation

Le démarrage d'un tournoi par une entité MAC (c.-à-d. l'envoi de l'ID) doit être précédé par l'envoi d'un signal de synchronisation  $l_s$  (impulsion d'énergie représentée par une porteuse) dont le rôle est d'annoncer aux autres entités MAC l'arrivée des bits de l'ID et donc de constituer la référence temporelle pour analyser l'ID qui arrive.

Cependant, si on considère le concept de fenêtre temporelle d'ambiguïté et si on suppose que l'entité MAC  $i$  décide d'envoyer un signal de synchronisation  $l_s$  à l'instant  $t$ , l'entité MAC  $j$  peut également décider d'envoyer un signal de synchronisation  $l_s$  entre  $(t - \tau_{TT} + \tau_{PT})$  et  $(t + \tau_{TT} + \tau_{PT})$ . Cette situation crée un croisement (collision) de signaux de synchronisation (mais une collision d'impulsions d'énergie donne encore des impulsions d'énergie) et donc dans une entité MAC la fin de l'arrivée d'un signal de synchronisation d'une autre entité MAC peut dépasser la fin de l'envoi de son propre signal de synchronisation.

Compte tenu de telles situations, une entité MAC doit, afin d'avoir un système propre (c.-à-d. que quand le premier bit du champ de priorité de l'ID est envoyé, il ne doit plus rien rester de l'activité de synchronisation sur le canal), ajouter, après l'envoi de son signal de synchronisation, *un temps de garde  $t_g$*  (temps où l'entité est aveugle) avant de pouvoir procéder ensuite à l'envoi du



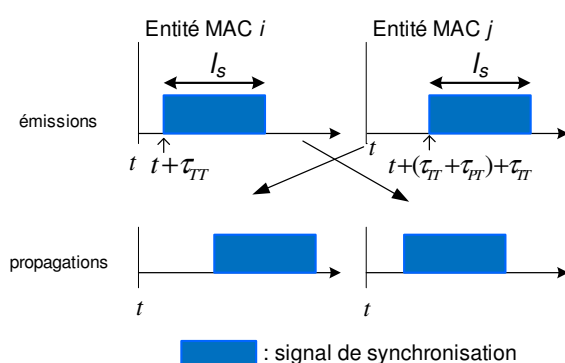


FIGURE 4.9 – CANlike : Signal de synchronisation.

premier bit de son ID. Le temps de garde utilisé par une entité MAC  $i$  est la plus grande différence entre l'instant de la fin d'envoi de son signal de synchronisation et l'instant de la fin d'arrivée du signal de synchronisation de l'entité MAC  $j$ .

Afin de définir le temps de garde, reprenons le concept de la fenêtre temporelle d'ambiguïté et considérons le scénario le plus contraignant c.-à-d. l'entité MAC  $i$  décide d'envoyer un signal de synchronisation  $l_s$  à l'instant  $t$  et l'entité MAC  $j$  décide d'envoyer un signal de synchronisation  $l_s$  au plus tard (à l'instant  $t + \tau_{TT} + \tau_{PT}$ ) (Fig. 4.9).

Les fins des envois de ces signaux de synchronisation pour les entités MAC émettrices sont :

- entité MAC  $i$  :  $t + \tau_{TT} + l_s$  ;
- entité MAC  $j$  :  $t + (\tau_{TT} + \tau_{PT}) + \tau_{TT} + l_s$ .

Les fins d'arrivée de ces signaux de synchronisation dans les entités MAC destinataires sont :

- entité MAC  $i$  :  $t + (\tau_{TT} + \tau_{PT}) + \tau_{TT} + l_s + \tau_{PT}$  ;
- entité MAC  $j$  :  $t + \tau_{TT} + l_s + \tau_{PT}$ .

On peut voir qu'il n'y a pas de dépassement dans l'entité MAC  $j$  (parce qu'elle a envoyé plus tard) et qu'il y a un dépassement dans l'entité MAC  $i$  (parce qu'elle a envoyé plus tôt). Le temps de garde nécessaire est donc  $t_g = 2\tau_{PT} + \tau_{TT}$ .

Donc toute entité MAC qui veut entreprendre le tournoi ne peut le faire qu'après l'envoi de son signal de synchronisation  $l_s$  suivi du temps de garde ( $2\tau_{PT} + \tau_{TT}$ ).

Concernant la longueur  $l_s$  du signal de synchronisation, comme il est suffisant qu'il soit détecté par une entité MAC réceptrice, on doit donc avoir  $l_s \geq \tau_{ST}$ .

*Remarque 1* : Une entité MAC qui reçoit un signal de synchronisation et n'en pas envoyé avant, n'est pas concernée par le tournoi.

*Remarque 2* : La valeur du temps de retournement  $\tau_{TT}$  étant incluse dans le temps de garde, on considère qu'une entité MAC utilise ce temps pour se retourner durant le temps de garde si besoin est pour effectuer ensuite la compétition relative au premier bit de l'ID : ceci dépend du type du premier bit de l'ID ; si c'est un bit dominant, on n'a pas besoin de se retourner (le bit dominant est une porteuse comme le signal de synchronisation) et donc ce temps  $\tau_{TT}$  est une attente ; si c'est un bit récessif, on utilise ce temps pour se retourner.

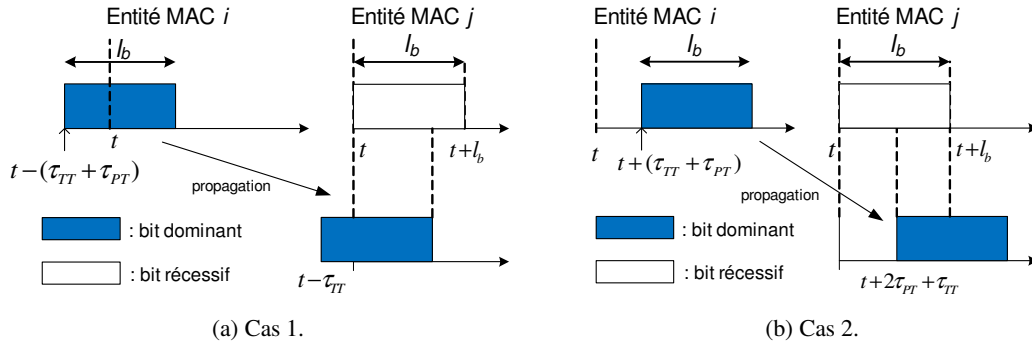


FIGURE 4.10 – CANlike : Evaluation de bit ID.

#### 4.4.3 Evaluation des bits de l'ID

Le point fondamental du tournoi, qui nécessite une présentation très précise, réside dans l'analyse de l'écoute faite par une entité MAC, qui a un bit récessif, alors qu'une autre entité MAC émet un bit dominant. Il faut que l'entité qui a le bit récessif détecte le bit dominant en intégrant le fait qu'un décalage peut exister entre le début de l'écoute et le début de l'émission du bit dominant.

Nous considérons pour faire cette analyse le début du tournoi c.-à-d. la compétition sur le premier bit de l'ID qui arrive juste après le signal de synchronisation. Nous notons  $l_b$  la durée d'un bit de l'ID.

Nous faisons cette analyse, évidemment sur la base du concept de la fenêtre temporelle d'ambiguïté (qui exprime l'asynchronisme avec les décalages maximums possibles), en considérant le scénario suivant :

- l'entité MAC  $j$ , qui a un bit récessif, débute l'écoute du canal à l'instant  $t$  et finit cette écoute à l'instant  $t + l_b$  ;
- l'entité MAC  $i$ , qui a un bit dominant, émet la porteuse qui le représente : soit (cas 1 - Fig. 4.10a) à l'instant  $(t - (\tau_{TT} + \tau_{PT}))$  c.-à-d. avec une avance maximale, soit (cas 2 - Fig. 4.10b) à l'instant  $(t + (\tau_{TT} + \tau_{PT}))$  c.-à-d. avec un retard maximum.

Analysons maintenant les conséquences, dans l'entité MAC  $j$ , des deux cas :

- **Cas 1** : la porteuse émise par l'entité MAC  $i$  arrive au niveau de l'entité MAC  $j$  à l'instant  $t - (\tau_{TT} + \tau_{PT}) + \tau_{PT} = t - \tau_{TT}$ , et dure jusqu'à l'instant  $t - \tau_{TT} + l_b$ . On a donc une arrivée de la porteuse avant la décision de se mettre à l'écoute (mais ceci tombe dans le temps de garde du signal de synchronisation et donc n'a pas d'importance) et une présence de cette porteuse au début de l'écoute seulement si  $t - \tau_{TT} + l_b > t$ , il faut donc pour avoir un recouvrement au début de l'écoute la contrainte (Equa. 4.15).

$$l_b > \tau_{TT} \quad (4.15)$$

- **Cas 2** : la porteuse émise par l'entité MAC  $i$  arrive au niveau de l'entité MAC  $j$  à l'instant  $t + (\tau_{TT} + \tau_{PT}) + \tau_{PT} = t + \tau_{TT} + 2\tau_{PT}$  et dure jusqu'à l'instant  $t + \tau_{TT} + 2\tau_{PT} + l_b$ . On a une arrivée de la porteuse après le début de l'écoute et donc la nécessité de recouvrement sur la fin de l'écoute implique la contrainte  $t + \tau_{TT} + 2\tau_{PT} < t + l_b$  soit :

$$l_b > 2\tau_{PT} + \tau_{TT} \quad (4.16)$$

De plus l'occupation temporelle de la porteuse va au-delà du temps d'écoute (dépassement de  $2\tau_{PT} + \tau_{TT}$ ) et donc il est nécessaire d'avoir aussi un temps de garde associé à un bit (comme dans le cas du signal de synchronisation) afin d'avoir un système propre (c.-à-d. quand la compétition entre bits de même rang débute, il ne reste rien sur le canal qui concerne la compétition des bits de rang précédent).

De l'analyse de ces deux cas, nous pouvons déduire maintenant les deux caractéristiques fondamentales relatives à un bit de l'ID :

- durée  $l_b$  d'un bit : outre les deux contraintes (Equa. 4.15 et Equa. 4.16), on doit avoir aussi la contrainte que la durée du recouvrement de la période de l'écoute par la porteuse soit supérieur ou égal à  $\tau_{ST}$ . En conséquence la contrainte globale est  $l_b \geq 2\tau_{PT} + \tau_{TT} + \tau_{ST}$ , on prend donc :

$$l_b = 2\tau_{PT} + \tau_{TT} + \tau_{ST}$$

- temps de garde  $t_g$  associé à un bit (ce temps de garde est fixé par le dépassement qui apparaît dans le cas 2) :

$$t_g = 2\tau_{PT} + \tau_{TT}$$

#### 4.4.4 Les protocoles

On a comme avec les messages d'annonce BlackBurst, deux types de protocoles qui résultent du type de priorité porté par l'ID : priorité statique ou priorité hybride.

On a aussi besoin (comme pour les protocoles avec les messages d'annonce BlackBurst) avant le début d'un tournoi (bit de synchronisation + comparaison des bits de l'ID) d'un temps d'observation appelé TOBS. Pour calculer TOBS, on considère le pire cas de la durée possible de l'état canal libre suite au passage d'un bit de synchronisation. Ceci se produit lorsque tous les bits de l'ID sont des bits récessifs (par exemple le cas où on a une seule entité MAC qui veut émettre et son ID est constitué uniquement de bits récessifs). C'est le cas extrême. Supposons un ID de  $n$  bits. Donc  $\text{TOBS} > n(l_b + t_g) = n(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$ . On peut prendre  $\text{TOBS} = (n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$ . La durée de la phase de compétition (observation, synchronisation, arbitrage) est donc TOBS + durée de la phase de synchronisation + durée de l'arbitrage. En considérant que la durée d'un bit de synchronisation + son temps de garde est égale à la durée d'un bit de l'ID + son temps de garde ( $l_s = l_b$  et même  $t_g$ ), on obtient alors :

$$\text{durée de la phase de compétition} = 2(n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$$

On n'a plus de temps d'observation durant un tournoi pour l'obtention du qualificatif gagnant-perdant (ceci apparaît tout naturellement lors des comparaisons des bits de même rang).

Le principe de base des protocoles est représenté sur la figure 4.11 (on ne tient compte ni de la fenêtre d'ambiguïté ni des temps de garde).

On considère ici un identificateurs de 7 bits et une compétition entre trois nœuds notés B, C, D. Le nœud D moins prioritaire abandonne la compétition au premier bit de priorité, étant en écoute il entend un bit dominant, le nœud C écoute pendant les bits 2 et 3 et sur ce bit il entend un bit dominant et abandonne à son tour ; le nœud B qui termine sans lire de bit dominant gagne et peut émettre ses données.

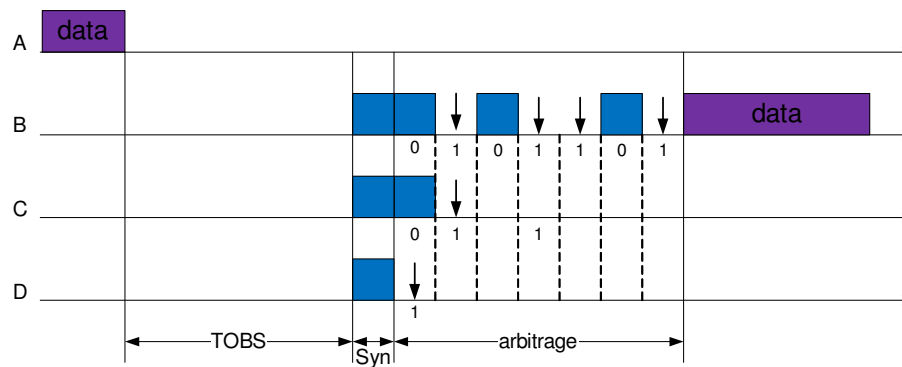


FIGURE 4.11 – CANlike : Principe du protocole (l'écoute pendant un bit récessif est symbolisée par ↓).

#### 4.4.5 Bilan des paramètres du protocole CANlike

On résume les paramètres du protocoles CANlike sur le tableau 4.3.

|   |   |
|---|---|
| bit ID $l_b$<br>= bit synchronisation $l_s$             | $2\tau_{PT} + \tau_{TT} + \tau_{ST}$            |
| temps de garde $t_g$<br>(bit ID et bit synchronisation) | $2\tau_{PT} + \tau_{TT}$                        |
| TOBS  | $(n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$  |
| phase de compétition<br>(TOBS + syn + arbitrage)        | $2(n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$ |

TABLE 4.3 – Paramètres du protocole CANlike.

## 4.5 Conclusion

Nous pensons pouvoir dire que la présentation des protocoles MAC, de type CSMA sans collision pour des réseaux locaux sans fil à un saut, que nous avons faite dans ce chapitre, est une première dans l'optique d'une présentation, à la fois, rationnelle et qui se veut aussi pédagogique que possible.

Tout d'abord, nous avons exprimé un concept fondamental « le concept de fenêtre d'ambiguïté temporelle » qui est, à la base, à la fois de la spécification d'une situation de collision (pourquoi ? quand ? et comment ?) et qui fournit les éléments de raisonnement pour contrôler sur la base de priorités, cette situation et la transformer en situation « gagnant - perdant(s) ».

Nous avons présenté les protocoles où les priorités sont envoyées soit sous la forme de messages d'annonces BlackBurst, soit sous la forme d'un champ ID comme dans CAN filaire (technique dite CANlike) mais en adaptant au sans fil. La comparaison de ces priorités définit un tournoi dont le gagnant est autorisé à transmettre une trame de données.

Nous avons défini sur la base du concept de la fenêtre d'ambiguïté et des caractéristiques de la couche physique d'un réseau sans fil (temps  $\tau_{ST}$  et  $\tau_{TT}$  d'un transceiver ; temps  $\tau_{PT}$  d'un canal de transmission) les principaux éléments des différents protocoles.



# Réseaux locaux sans fil et applications de contrôle-commande : sur l'utilisation des protocoles MAC sans collision

---

## Sommaire

---

|            |   |            |
|------------|---|------------|
| <b>5.1</b> | <b>Introduction</b>   | <b>104</b> |
| <b>5.2</b> | <b>Contexte de l'étude</b>  | <b>104</b> |
| 5.2.1      | L'application de contrôle-commande considérée   | 104        |
| 5.2.2      | Les protocoles considérés et leur paramètres  | 105        |
| <b>5.3</b> | <b>Protocoles MAC sans collision basés sur des priorités statiques</b>                        | <b>106</b> |
| 5.3.1      | Considération d'une seule application de contrôle-commande : étude de l'influence du champ ID | 106        |
| 5.3.2      | Considération d'une application de contrôle-commande et d'un flux externe                     | 109        |
| 5.3.3      | Considération de quatre applications de contrôle-commande                                     | 110        |
| 5.3.4      | Mise en œuvre de la relation $QdS \rightarrow QdC$  | 112        |
| <b>5.4</b> | <b>Protocole CANlike avec la priorité hybride <math>ph^+</math> (CANlike-hyb)</b>             | <b>113</b> |
| 5.4.1      | Mise en œuvre de la relation $QdC \rightarrow QdS$  | 113        |
| 5.4.2      | Mise en œuvre de la relation $QdS \rightleftharpoons QdC$                                     | 113        |
| <b>5.5</b> | <b>Récapitulation</b>   | <b>114</b> |
| <b>5.6</b> | <b>Conclusion</b>   | <b>117</b> |

---

## 5.1 Introduction

Ce chapitre a pour objectif, sur la base de l'implantation dans l'outil TrueTime des protocoles sans collision (protocoles basés sur des messages d'annonce BlackBurst ; protocoles CANlike), de montrer la mise en œuvre avec ces protocoles des relations QdS→QdC, QdC→QdS et QdC⇐QdS.

Ce chapitre comprend quatre parties :

- la première partie présente le contexte de l'étude, à savoir l'application de contrôle-commande et les protocoles considérés ;
- la deuxième partie qui considère les protocoles basés sur des priorités statiques, traite quatre points :
  - . le premier, concerne, en se plaçant dans le cas de l'implantation d'une seule application de contrôle-commande, l'étude de l'influence de la structure de donnée utilisée pour exprimer les priorités sur les performances en QdC ; nous appelons « champ ID » cette structure comme cela est fait et utilisé déjà dans le protocole CANlike ;
  - . le deuxième concerne l'étude de l'influence d'un flux externe sur une application de contrôle-commande ;
  - . le troisième considère l'implantation de 4 applications de contrôle-commande ; Ces trois différentes études, en considérant des priorités statiques, sont menées en faisant toujours des comparaisons avec le protocole DCF-WiFi.
  - . le quatrième point concerne la mise en œuvre de la relation QdS→QdC par la méthode des pôles dominants.
- la troisième partie considère le protocole CANlike avec la priorité hybride  $ph^+$  (c.-à-d. le schéma  $ph$  avec le contrôleur qui utilise la priorité dynamique maximale (cf. Chap. 3)) et traite de la mise en œuvre, dans le cadre de l'implantation de 4 applications de contrôle-commande, des relations QdC→QdS et QdS⇐QdC ;
- la quatrième partie fait une récapitulation des résultats obtenus.

## 5.2 Contexte de l'étude

### 5.2.1 L'application de contrôle-commande considérée

Nous rappelons pour des facilités de lecture le contexte considéré dans les précédents chapitres en y ajoutant la partie inhérente à la spécificité de ce chapitre à savoir les paramètres liés aux différents protocoles considérés et les implantations considérées.

Le modèle du système continu considéré est celui donné dans le chapitre 3 (Fig. 5.1) :  $G(s) = \frac{1000}{s(s+1)}$  ; le contrôleur PD =  $K(1 + sT_d)$  ; l'entrée  $R(s)$  est un échelon de position.

La marge de phase de  $G(s)$  est  $1.8^\circ$  à la fréquence  $\omega_c = 31.6$  rad/s (c.-à-d.  $\omega_c$  tels que  $|G(j\omega_c)| = 1$ ). On utilise le contrôleur PD pour compenser la phase afin d'avoir une marge de phase de  $45^\circ$  à  $\omega_c$ . Il faut donc  $T_d = 0.0297$  s et  $K = 0.7291$ .

Rappelons les caractéristiques du système : les pôles  $p_{1,2} = -11 \pm j24.5$  ; le dépassement  $O = 33.8\%$  ( $O$  pour « Overshoot ») ; le temps de réponse (à 2%)  $t_r = 284$  ms ;  $\omega_n = 27$  rad/s et l'erreur en régime permanent (« Steady State Error »)  $e_{ss} = 0$ .

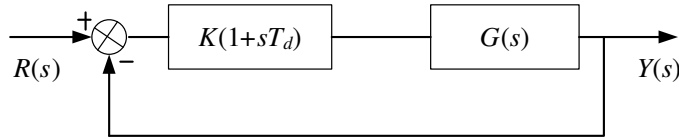


FIGURE 5.1 – Modèle.

Le système est échantillonné avec la période  $h = 10$  ms afin d'être implanté sur un réseau sans fil. Cette application implantée sur un réseau donne lieu aux deux flux  $f_{cc}$  et  $f_{ca}$ . Nous supposons toujours que le champ de données des trames de ces flux contient 16 bits.

Nous reprenons le critère ITSE  $J = \int_0^T t(r(t) - y(t))^2 dt$ . La durée de simulation est  $T = 500$  ms. La valeur de  $J$  pour ce système continu  $J_0 = 9.4562 \times 10^{-4}$ . Le critère d'évaluation de performance sera représenté par le terme  $\frac{(J-J_0)}{J_0} \% = \frac{\Delta J}{J_0} \%$ . Plus grande est la valeur de  $\frac{\Delta J}{J_0} (\%)$ , moins bonne est la QdC.

### 5.2.2 Les protocoles considérés et leur paramètres

Nous considérons, comme référence, le protocole IEEE 802.11 DCF (noté DCF-WiFi) dont nous allons utiliser la structure de trame pour spécifier les protocoles sans collision que nous allons étudier.

Une trame DCF-WiFi a les informations de contrôles suivantes : 192 bits pour la couche physique + 272 bits pour l'en tête de la couche MAC (cf. Sect. 1.4.2.1). La longueur d'une trame de données = 192 + 272 + nombre de bit du champ de données.

En considérant les priorités statiques, nous étudions les protocoles BlackBurst (noté BBsta) et CANlike (noté CANlike-sta).

En considérant les priorités hybrides, nous étudions seulement CANlike (noté CANlike-hyb).

Notons, concernant les protocoles CANlike, que les bits du champ ID ont les mêmes caractéristiques  $(l_b, t_g)$  quel que soit le type de priorité (statique, hybride).

Les caractéristiques des protocoles considérées sont :

*Protocole DCF-WiFi :*

- trame de données :  $192 + 272 + 16 = 480$  bits ;
- débit 1 Mbits/s ; DIFS =  $50 \mu s$ ,  $\tau_{TT} = 5 \mu s$ ,  $\tau_{ST} = 15 \mu s$ ,  $\tau_{PT}$  est négligé.

*Protocoles BB et CANlike :*

- trame de données : 480 bits comme DCF-WiFi ;
- les paramètres du protocole BBsta liés à la mise en œuvre du tournoi sont : TBB =  $25 \mu s$  ; TOBS1 =  $45 \mu s$  ; TOBS2 =  $25 \mu s$  (notons que par simplicité, TOBS1 et TOBS2 représentent ici les durées d'observation et les temps de retournement qui encadrent les durées d'observation) ;
- les paramètres du protocole CANlike liés à la mise en œuvre du tournoi sont :  $l_b = 20 \mu s$  ;  $t_g = 5 \mu s$  ; TOBS =  $(n+1)(l_b + t_g) = 25(n+1) \mu s$  (où  $n$  est le nombre de bit du champ ID) ;
- le débit de 1 Mbits/s et donc la durée des trames est de  $480 \mu s$  (noté  $D$ ).



## 5.3 Protocoles MAC sans collision basés sur des priorités statiques

### 5.3.1 Considération d'une seule application de contrôle-commande : étude de l'influence du champ ID

Ici nous avons donc uniquement les deux flux  $f_{cc}$  et  $f_{ca}$ .

#### Protocole BBsta

Considérons un champ ID de  $n$  bits. On peut exprimer  $2^n$  valeurs qui permettent de fixer  $2^n$  priorités et donc de considérer  $2^n$  flux.

Par contre, si on a un nombre de flux inférieur aux possibilités offertes par le champ ID, on peut se poser la question du choix des priorités : les plus petites (appelons ce choix BBsta-min) ou les plus grandes (appelons ce choix BBsta-max).

Notons que si on choisit les plus petites priorités, les messages d'annonces auront toujours la même durée quelle que soit la longueur du champ ID. Par contre si on choisit les plus grandes valeurs, les messages d'annonce auront des durées qui croîtront avec la longueur du champ ID.

Nous voulons montrer précisément ici, l'influence sur la QdC de l'application de contrôle-commande, de la longueur du champ ID lorsqu'on choisit les plus grandes priorités (BBsta-max). Le tableau 5.1 donne le cadre de cette analyse où on considère des champs ID de 2, 4, 6, 8 bits et où on donne les valeurs de priorité maximales pour les flux  $f_{cc}$  et  $f_{ca}$  (en nombre de TBB).

| Nombre de bit ID ( $n$ ) | 2 | 4  | 6  | 8   |
|--------------------------|---|----|----|-----|
| flux $f_{cc}$            | 3 | 15 | 63 | 255 |
| flux $f_{ca}$            | 4 | 16 | 64 | 256 |

TABLE 5.1 – BBsta-max : priorités des flux (nombre de TBB).

Les résultats concernant la QdC lorsque l'on considère soit BBsta-min soit BBsta-max sont donnés respectivement sur les tableaux 5.2 et 5.3.

| $n$       | $\forall n$ |
|-----------|-------------|
| BBsta-min | 73.0        |

TABLE 5.2 – BBsta-min  $\forall n : \Delta J/J_0\%$

| $n$       | 2    | 4    | 6   | 8        |
|-----------|------|------|-----|----------|
| BBsta-max | 75.2 | 89.3 | 168 | instable |

TABLE 5.3 – BBsta-max :  $\Delta J/J_0\%$

On voit donc que l'on a *évidemment* intérêt à faire le choix de BBsta-min (le tournoi dure moins longtemps, donc moins de retard et meilleures performances). Ce type d'étude avait été fait dans [Nguyen Xuan 2010b] en comparant le protocole BBsta avec DCF-WiFi et ZigBee (technique unslotted CSMA/CA).

Cependant il faut remarquer (Tab. 5.3) que si on a un grand nombre d'applications, par exemple, pour un champ ID de longueur  $n = 8$ , on a  $2^8$  applications et donc la plus prioritaire ne pourra pas être implantée car ses messages d'annonce sont trop longs. Ceci montre les limites des protocoles BlackBurst.

### Protocole CANlike-sta

Ici le champ ID est toujours échangé dans un tournoi et ce quelle que soit la valeur des priorités. Donc sa longueur influence la durée du tournoi. Notons encore que le temps TOBS dépend de la longueur  $n$  du champ ID ( $\text{TOBS} = (n + 1)(l_b + t_g)$ ) et donc le champ ID influence doublement la durée du tournoi.

Le tableau 5.4 représente l'influence de la longueur  $n$  du champ ID sur la QdC de l'application de contrôle-commande.

| Nombre de bit ID ( $n$ ) | 2    | 4    | 6    | 8    |
|--------------------------|------|------|------|------|
| CANlike-sta              | 74.8 | 79.3 | 84.0 | 88.9 |

TABLE 5.4 – CANlike-sta : performances ( $\Delta J/J_0\%$ ).

Il est important de noter qu'avec un champ ID de longueur  $n = 8$  bits on peut maintenant mettre en œuvre toutes les applications (contrairement au protocole BBsta-max - Tab. 5.3 où, pour  $n = 8$ , on a un système instable).

### Remarque importante sur les protocoles BBsta et CANlike-sta

Dans la suite de ce travail, nous considérons seulement un identificateur de 8 bits ( $n = 8$ ) et pour les protocoles BBsta, on considérera le type BBsta-min.

### Comparaison des protocoles MAC : DCF-WiFi, BBsta-min, CANlike-sta

Le tableau 5.5, qui représente les performances en termes de  $\Delta J/J_0(\%)$  pour les trois protocoles BBsta-min, CANlike-sta et DCF-WiFi, montre que DCF-WiFi est meilleur que BBsta-min et BBsta-min est meilleur que CANlike-sta.

|             |      |
|-------------|------|
| DCF-WiFi    | 68.6 |
| BBsta-min   | 73.0 |
| CANlike-sta | 88.9 |

TABLE 5.5 – Réseau dédié : performances ( $\Delta J/J_0\%$ ).

L'explication est faite par l'évaluation du retard de communication  $\tau_{com} = \tau_{cc} + \tau_{ca}$ . Les retards introduits par des protocoles sont les suivants (rappelons que  $D$  est la durée de transmission d'une trame de données) :

- **DCF-WiFi** : il n'y pas de collision.
- pour la 1<sup>re</sup> période d'échantillonnage :
  - $\tau_{cc} = \text{DIFS} + D$ ;
  - $\tau_{ca} = \text{DIFS} + D$ ;
  - $\Rightarrow \tau_{com} = 100 \mu\text{s} + 2D$  (Fig. 5.2 entre l'instant  $t_0$  et l'instant  $t_1$ ).

- pour les périodes suivantes :

$\tau_{cc} = D$  (car le post-backoff est terminé avant le début de nouvelle période d'échantillonnage) ;

$\tau_{ca} = \text{DIFS} + D$  ;

$\Rightarrow \tau_{com} = 50 \mu\text{s} + 2D$  (Fig. 5.2 à l'instant  $t_1$ ).

- **BBsta-min** : (Fig. 5.3)

$\tau_{cc} = \text{TOBS1} + \text{TBB} + \text{TOBS2} + D$  ;

$\tau_{ca} = \text{TOBS1} + 2\text{TBB} + \text{TOBS2} + D$  ;

$\Rightarrow \tau_{com} = 215 \mu\text{s} + 2D$ .

- **CANlike-sta** : (Fig. 5.4)

$\tau_{cc} = \tau_{ca} = 2(8 + 1)25 + D = 450 \mu\text{s} + D$

$\Rightarrow \tau_{com} = 900 \mu\text{s} + 2D$

On trouve bien que le retard pour CANlike-sta est plus grand que le retard pour BBsta-min, et le retard de BBsta-min est plus grand que celui de DCF-WiFi. Ceci explique les performances trouvées (Tab. 5.5).

## Conclusion

Dans le cas du réseau dédié à une seule application de contrôle-commande comme il n'y a pas de collisions, DCF-WiFi est le meilleur car son temps d'accès est le plus petit.

CANlike-sta a la dégradation la plus grande qui résulte d'un temps d'accès plus élevé que BBsta-min (en effet on a l'influence de toute la longueur  $n = 8$  du champ ID).



FIGURE 5.2 – DCF-WiFi : transfert de trames des flux  $f_{cc}$  et  $f_{ca}$ .

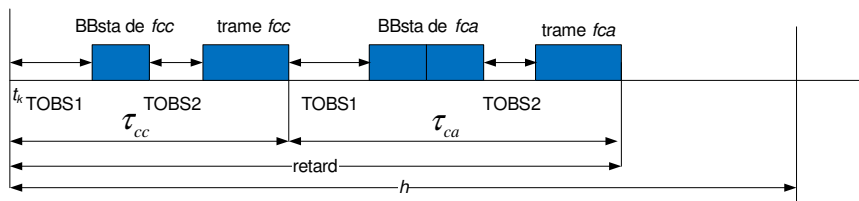


FIGURE 5.3 – BBsta-min : transfert de messages d'annonce et de trames des flux  $f_{cc}$  et  $f_{ca}$ .

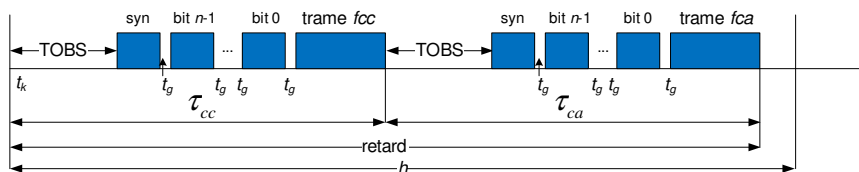


FIGURE 5.4 – CANlike-sta : transfert de champs ID et de trames des flux  $f_{cc}$  et  $f_{ca}$ .

### 5.3.2 Considération d'une application de contrôle-commande et d'un flux externe

Le flux externe  $f_{ex}$  est périodique ( $h = 10$  ms) et synchrone avec le flux  $f_{cc}$  à chaque instant d'échantillonnage  $t_k = kh$  ( $k = 0, 1, 2, \dots$ ). La longueur d'une trame du flux  $f_{ex}$  est aussi de 480 bits.

Dans la mise en œuvre des protocoles BBsta-min et CANlike-sta (protocoles à priorité), le flux externe est soit le plus prioritaire (noté cas  $f_{ex\_max}$ ) soit le moins prioritaire (noté cas  $f_{ex\_min}$ ).

Il est à noter que, pour DCF-WiFi, il existe des collisions entre les flux  $f_{ex}$  et  $f_{cc}$  et entre les flux  $f_{ex}$  et  $f_{ca}$ . La résolution de collision est faite au moyen du mécanisme Backoff (un temps d'attente aléatoire après une collision ; ceci est présenté dans la sous-section 1.4.2.1). C'est à dire que chaque simulation donne un résultat différent. Nous donnons donc, pour DCF-WiFi, la valeur moyenne sur 20 simulations et également l'écart maximal (voir le tableau B.1 dans l'annexe B).

Les performances ( $\frac{\Delta J}{J_0}$ %) données par les protocoles BBsta-min, CANlike-sta et DCF-WiFi sont représentées dans le tableau 5.6.

|                  | cas $f_{ex\_min}$ | cas $f_{ex\_max}$ |
|------------------|-------------------|-------------------|
| BBsta-min        | 73.0              | 93.3              |
| CANlike-sta      | 88.9              | 114               |
| DCF-WiFi         |                   |                   |
| - valeur moyenne | 93.9              |                   |
| - écart maximal  | 28.1              |                   |

TABLE 5.6 – 1 procédé + 1  $f_{ex}$  :  $\Delta J/J_0$  (%).

#### Protocole DCF-WiFi :

DCF-WiFi est dégradé par rapport au cas du réseau dédié (Tab. 5.5) car on a maintenant des collisions entre les flux  $f_{cc}$ ,  $f_{ca}$  et  $f_{ex}$ .

#### Protocoles BBsta-min et CANlike-sta :

- cas  $f_{ex\_min}$  : on retrouve les performances obtenues dans le cas du réseau dédié (c'est logique car le flux externe ne perturbe pas les flux  $f_{cc}$ ,  $f_{ca}$ ) ;
- cas  $f_{ex\_max}$  : les résultats sont logiquement moins bons que le cas  $f_{ex\_min}$  (le flux  $f_{ex}$  perturbe les flux  $f_{cc}$ ,  $f_{ca}$ ) ;
- notons que CANlike-sta a de moins bons résultats que BBsta-min, ceci étant dû à un temps d'accès  $\tau_a$  plus long (rappelons que  $n = 8$ ,  $l_b = 20 \mu s$ ,  $t_g = 5 \mu s$ ,  $TOBS1 = 45 \mu s$ ,  $TOBS2 = TBB = 25 \mu s$ ) :
  - CANlike-sta : pour chaque trame des flux  $f_{cc}$  et  $f_{ca}$  :  $\tau_a = 2(n+1)(l_b + t_g) = 450 \mu s$  ;
  - BBsta-min : pour chaque trame du flux  $f_{cc}$  :  $\tau_a = TOBS1 + TBB + TOBS2 = 95 \mu s$  ;  
pour chaque trame du flux  $f_{ca}$  :  $\tau_a = TOBS1 + 2TBB + TOBS2 = 120 \mu s$ .

### 5.3.3 Considération de quatre applications de contrôle-commande

Nous considérons 4 procédés  $P_1, P_2, P_3, P_4$  qui ont les flux  $f_{cc}$  synchrones. Appelons  $P_{cai}$  et  $P_{cci}$  les priorités des flux  $f_{ca}$  et  $f_{cc}$  du procédé  $P_i$ . Les priorités pour les protocoles BBsta-min et CANlike-sta sont rangées dans cet ordre :  $P_{ca1} > P_{ca2} > P_{ca3} > P_{ca4} > P_{cc1} > P_{cc2} > P_{cc3} > P_{cc4}$ . C'est à dire que le procédé  $P_i$  est considéré plus important que  $P_j$  avec  $i < j$ .

Les performances ( $\Delta J/J_0\%$ ) procurées par les 3 protocoles sont représentées sur le tableau 5.7 (Nous donnons encore, pour DCF-WiFi, la valeur moyenne sur 20 simulations et également l'écart maximal (voir le tableau B.2 dans l'annexe B)). Une visualisation graphique de ce tableau est donnée sur la figure 5.5 (les pointillés représentent les écarts maximaux donnés sur les 20 simulations). Nous représentons également les réponses temporelles sur les figures 5.6 (c'est un tirage arbitraire), 5.7 et 5.8.

|                  | $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|------------------|-------|-------|-------|-------|
| DCF-WiFi :       |       |       |       |       |
| - valeur moyenne | 139   | 136   | 134   | 161   |
| - écart maximal  | 132   | 84    | 156   | 140   |
| BBsta-min        | 78.0  | 133   | 158   | 218   |
| CANlike-sta      | 88.9  | 145   | 234   | 382   |

TABLE 5.7 – 4 procédés :  $\Delta J/J_0\%$ .

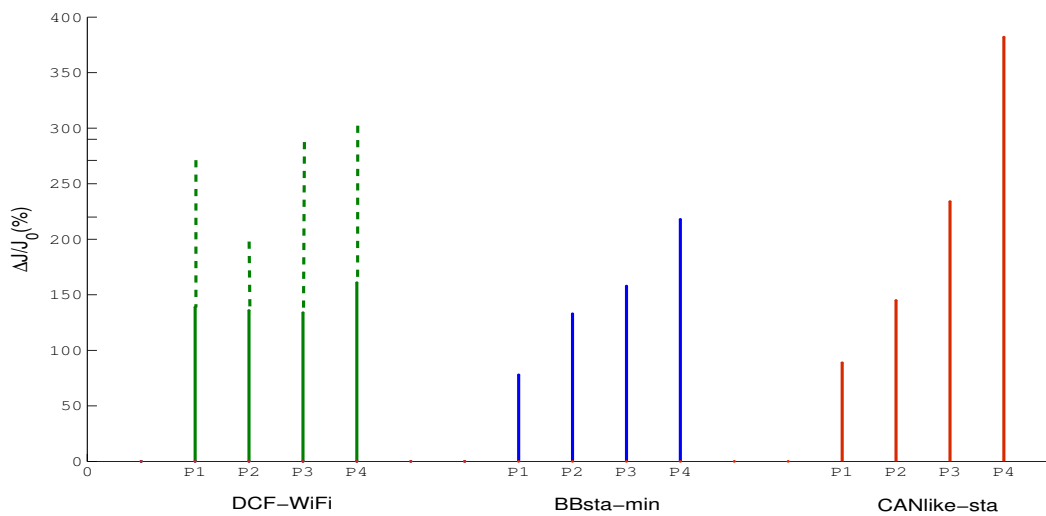


FIGURE 5.5 – Performances : priorité statique.

#### Commentaires

- DCF-WiFi donne en moyenne (sur 20 simulations) des performances les plus équilibrées pour les procédés (Fig. 5.5), ce qui est logique car la probabilité d'accès au médium des procédés est identique, l'ordre de performances des procédés est aléatoire, nous représentons ici sur la figure 5.6 un ordre obtenu pour une simulation. Par contre, BBsta-min et CANlike-sta favorisent les procédés plus prioritaires (plus grande est la priorité, meilleure est la QdC) ;
- CANlike a de moins bons résultats que BBsta-min à cause du temps d'accès plus élevé que celui de BBsta-min (toujours à cause du fait que toute la durée du champ ID intervient) ;

- la visualisations des QdC en termes de réponse temporelle pour BBsta-min et CANlike-sta sont représentées sur les figures 5.7 et 5.8. On voit bien les différences de dépassement en fonction des priorités et on remarque évidemment que les dépassements dans le cas de CANlike-sta sont plus élevés que dans le cas BBsta-min.

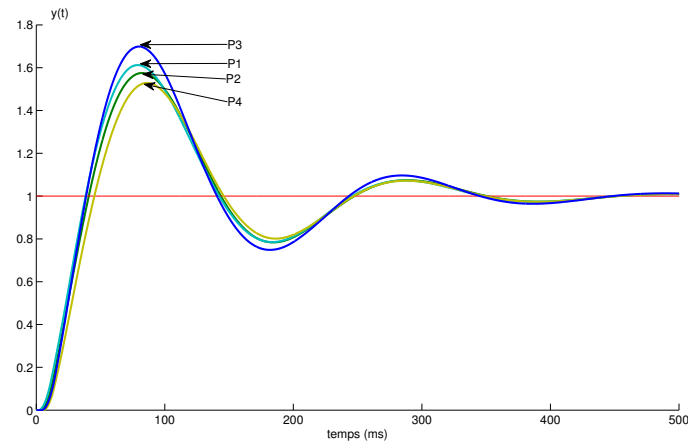


FIGURE 5.6 – DCF-WiFi : Réponses temporelles (un tirage arbitraire).

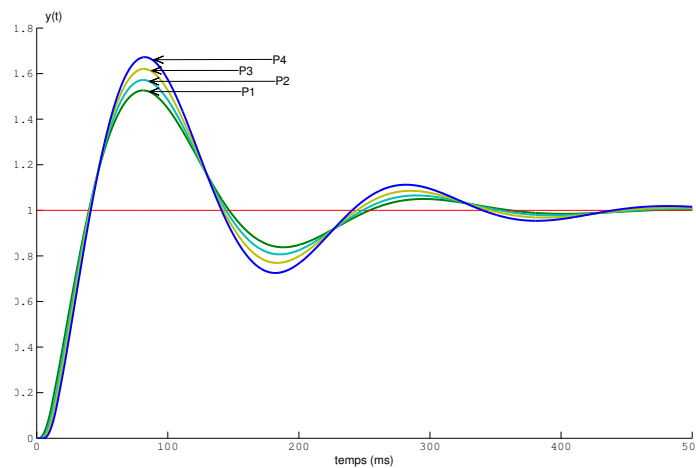


FIGURE 5.7 – BBsta-min : réponses temporelles.

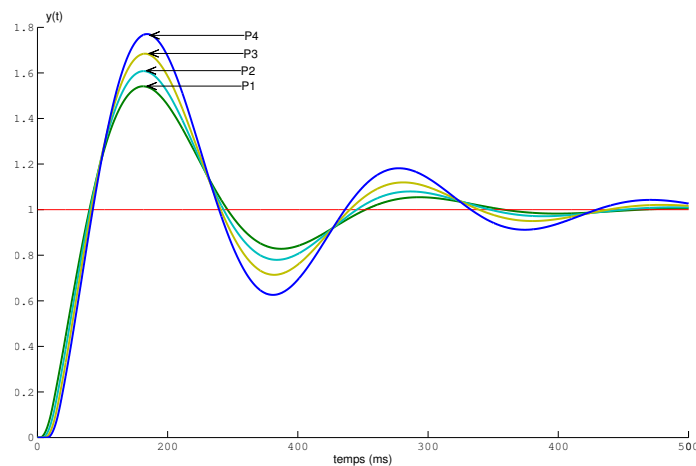


FIGURE 5.8 – CANlike-sta : réponses temporelles.

### 5.3.4 Mise en œuvre de la relation QdS→QdC

Nous rappelons (Chap. 3) que la mise en œuvre de cette relation est basée sur une mesure du retard dans la boucle et une compensation en ligne de ce retard par modification des paramètres du réseau correcteur (en appliquant la méthode des pôles dominants) à chaque période d'échantillonnage. Avec DCF-WiFi, le contrôleur n'a pas les moyens de connaître le retard dans la boucle comme ceci est permis avec les protocoles avec priorité (cf. Chap. 3) et donc cette étude ne concerne que les protocoles BBsta-min et CANlike-sta.

Nous considérons donc les 4 applications de contrôle-commande  $P_1, P_2, P_3, P_4$  et appliquons la compensation du retard basée sur la méthode des pôles dominants.

Les résultats obtenus sur la QdC sont représentés sur le tableau 5.8 : on voit évidemment l'amélioration apportée par rapport aux résultats de la sous section 5.3.3 (Tab. 5.7). Les réponses temporelles représentées sur les figures 5.9 et 5.10 visualisent ces améliorations par rapport aux figures 5.7 et 5.8 (dépassement plus faible). On ne peut pas avoir le même dépassement pour toutes les applications car on a des retards différents pour toutes les applications (Figs. 5.7 et 5.8), ce qui a amené au niveau de la compensation à la définition de contrôleurs différents (en particulier des valeurs de  $T_d$  différentes) ce qui donne des zéros ( $-1/T_d$ ) différents, donc des dépassements différents.

|             | $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|-------------|-------|-------|-------|-------|
| BBsta-min   | 48.8  | 64.6  | 82.0  | 101   |
| CANlike-sta | 54.0  | 77.6  | 106   | 142   |

TABLE 5.8 – CANlike-sta et BBsta-min : QdS→QdC.

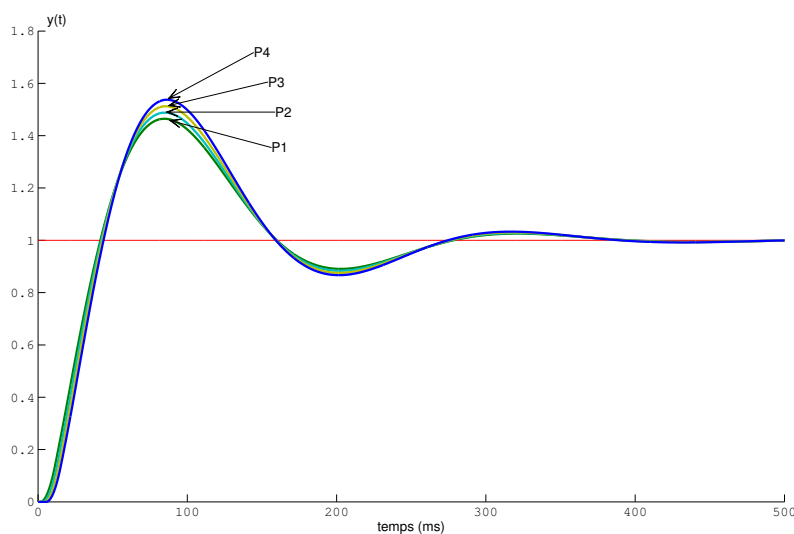


FIGURE 5.9 – BBsta-min (QdS→QdC).

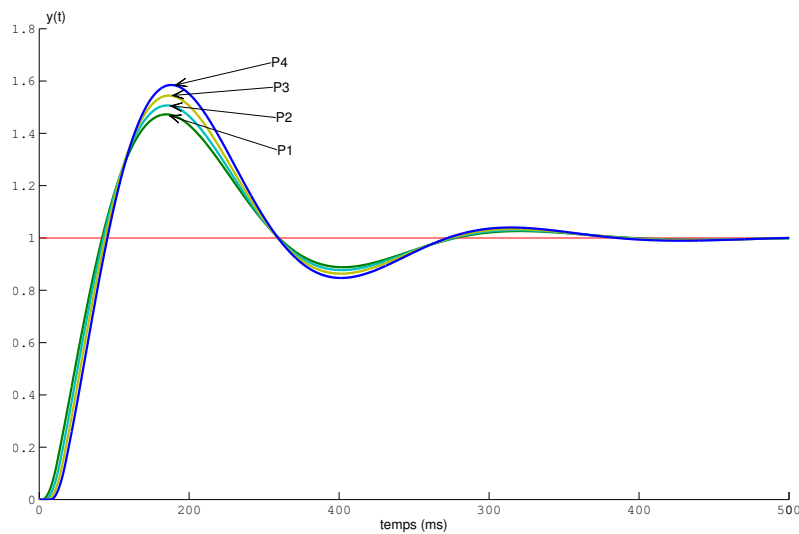


FIGURE 5.10 – CANlike-sta (QdS→QdC).

## 5.4 Protocole CANlike avec la priorité hybride $ph^+$ (CANlike-hyb)

On considère la partie priorité statique du champ ID de 4 bits (comme l'étude faite dans le réseau CAN dans le chapitre 2), donc il reste 4 bits pour la partie priorité dynamique (on a donc 16 niveaux de priorité dynamique).

Le schéma de priorité hybride utilisé est  $ph^+$  (spécifié au chapitre 3).

Nous reprenons les 4 applications  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$  considérées dans la section 5.3.3.

### 5.4.1 Mise en œuvre de la relation QdC→QdS

Le tableau 5.9 représente la QdC ( $\Delta J/J_0\%$ ) obtenue et la figure 5.11 représente les réponses temporelles. La comparaison avec CANlike-sta (Tab. 5.7 ; Fig. 5.8) montre un équilibrage des performances (c'est ce que l'on avait constaté déjà avec le réseau CAN filaire).

| $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|-------|-------|-------|-------|
| 169   | 198   | 185   | 233   |

TABLE 5.9 – Relation QdC→QdS : 4 procédés ( $\Delta J/J_0\%$ ).

### 5.4.2 Mise en œuvre de la relation QdS⇌QdC

Le tableau 5.10 et la figure 5.12, qui représentent la QdC ( $\Delta J/J_0\%$ ) et les réponses temporelles obtenues de cette relation QdS⇌QdC, montrent l'amélioration obtenue par rapport aux tableaux 5.9 et à la figure 5.11 tout en maintenant des performances équilibrées.

| $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|-------|-------|-------|-------|
| 99.3  | 76.6  | 107   | 112   |

TABLE 5.10 – Relation QdS⇌QdC : 4 procédés ( $\Delta J/J_0\%$ ).



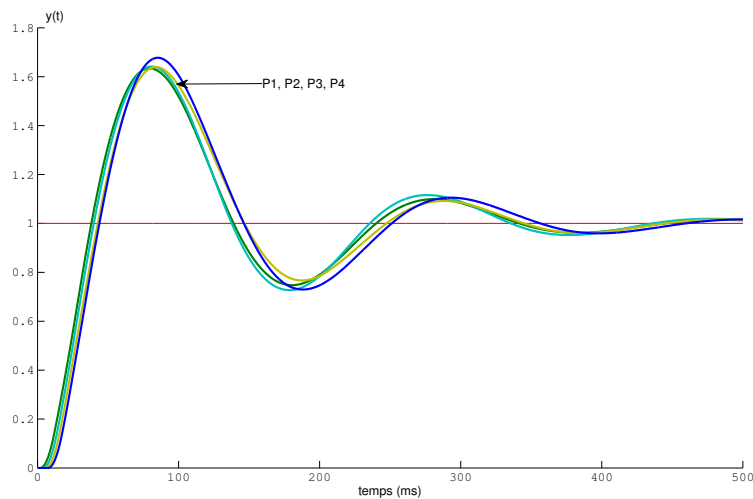


FIGURE 5.11 – CANlike-hyb (QdC→QdS).

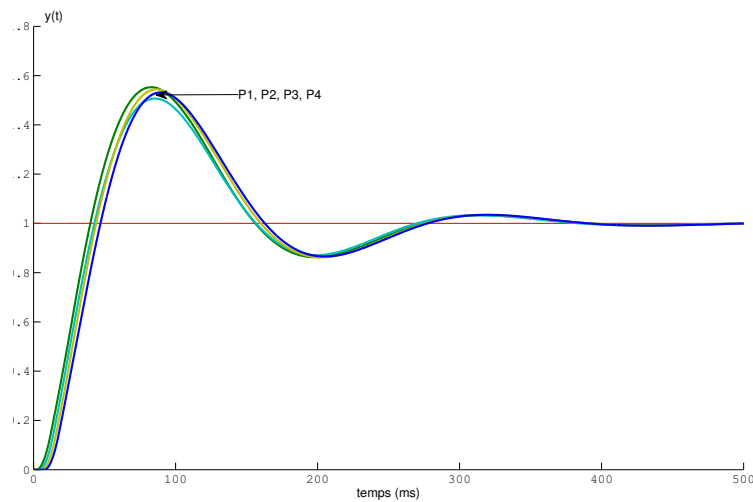


FIGURE 5.12 – CANlike-hyb (QdS⇒QdC).

## 5.5 Récapitulation

Le tableau 5.11 récapitule les principaux résultats obtenus dans ce chapitre pour CANlike et DCF-WiFi . Ces résultats sont visualisés par le graphique de la figure 5.13 (les pointillés représentent toujours les écarts maximaux donnés par 20 simulations).

Le tableau 5.11 rappelle :

- tout d’abord les résultats que l’on peut obtenir avec DCF-WiFi (valeur moyenne ; écart maximal) et qui montrent que l’on ne peut pas considérer DCF-WiFi pour implanter des applications contraintes comme les applications de contrôle-commande (compte tenu des écarts que l’on peut avoir) ;
- l’utilisation des seules priorités statiques (le contrôleur étant fixe) donne logiquement des résultats en relation directe avec la priorité du procédé et donc très dégradés pour les priorités les plus faibles ;

- l'utilisation des priorités statiques avec un contrôleur adaptatif (relation  $QdS \rightarrow QdC$ ) améliore incontestablement les performances mais elles restent toujours très marquées par l'ordre des priorités ;
- la considération de priorités hybrides a pour objectif de donner priorité aux urgences de transmission (par la composante priorité dynamique (relation  $QdC \rightarrow QdS$ )) et donc d'équilibrer les performances des applications de contrôle-commande ;
- l'équilibrage donné par les priorités hybrides est ensuite renforcé dans le sens d'avoir des performances meilleures dues au rôle du contrôleur adaptatif (relation  $QdC \rightleftharpoons QdS$ ).

La visualisation du rôle des priorités hybrides par rapport à la priorité statique est montrée sur les « zooms » des réponses temporelles sur la figure 5.14.

|   | P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> | P <sub>4</sub> |
|---|----------------|----------------|----------------|----------------|
| DCF-WiFi<br>(valeur moyenne ; écart maximal)                              | 139 ; 132      | 136 ; 84       | 134 ; 156      | 161 ; 140      |
| priorité statique + contrôleur fixe                                       | 88.9           | 145            | 234            | 382            |
| QdC $\rightarrow$ QdS<br>(priorité hybride + contrôleur fixe)             | 169            | 198            | 185            | 233            |
| QdS $\rightarrow$ QdC<br>(priorité statique + contrôleur adaptatif)       | 54.0           | 77.6           | 106            | 142            |
| QdS $\rightleftharpoons$ QdC<br>(priorité hybride + contrôleur adaptatif) | 99.3           | 76.6           | 107            | 112            |

TABLE 5.11 – 4 procédés :  $\Delta J/J_0\%$ .

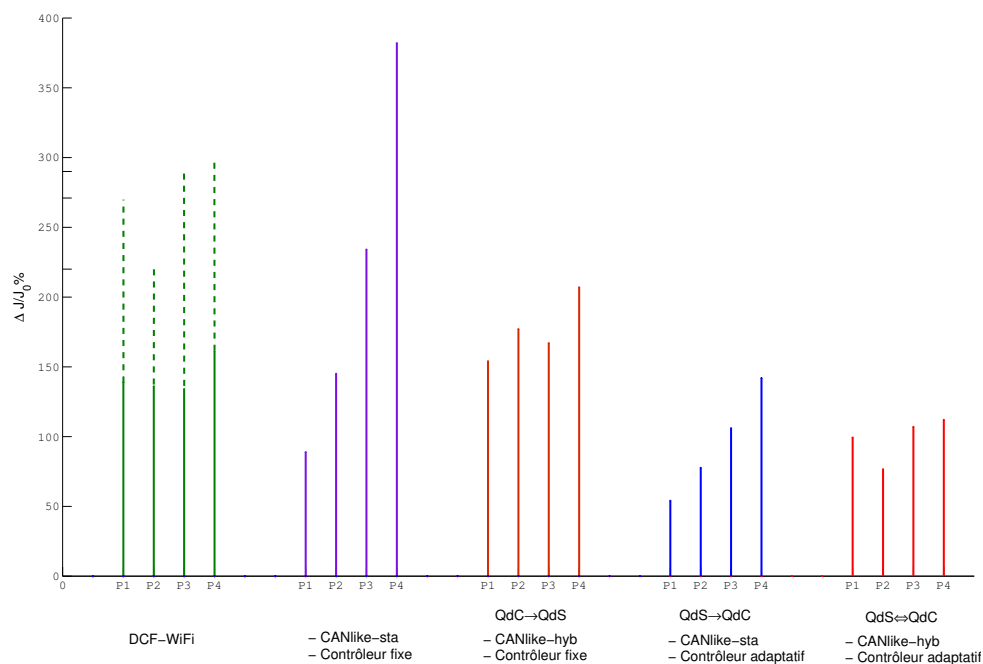


FIGURE 5.13 – 4 procédés :  $\Delta J/J_0\%$  ; visualisation graphique.

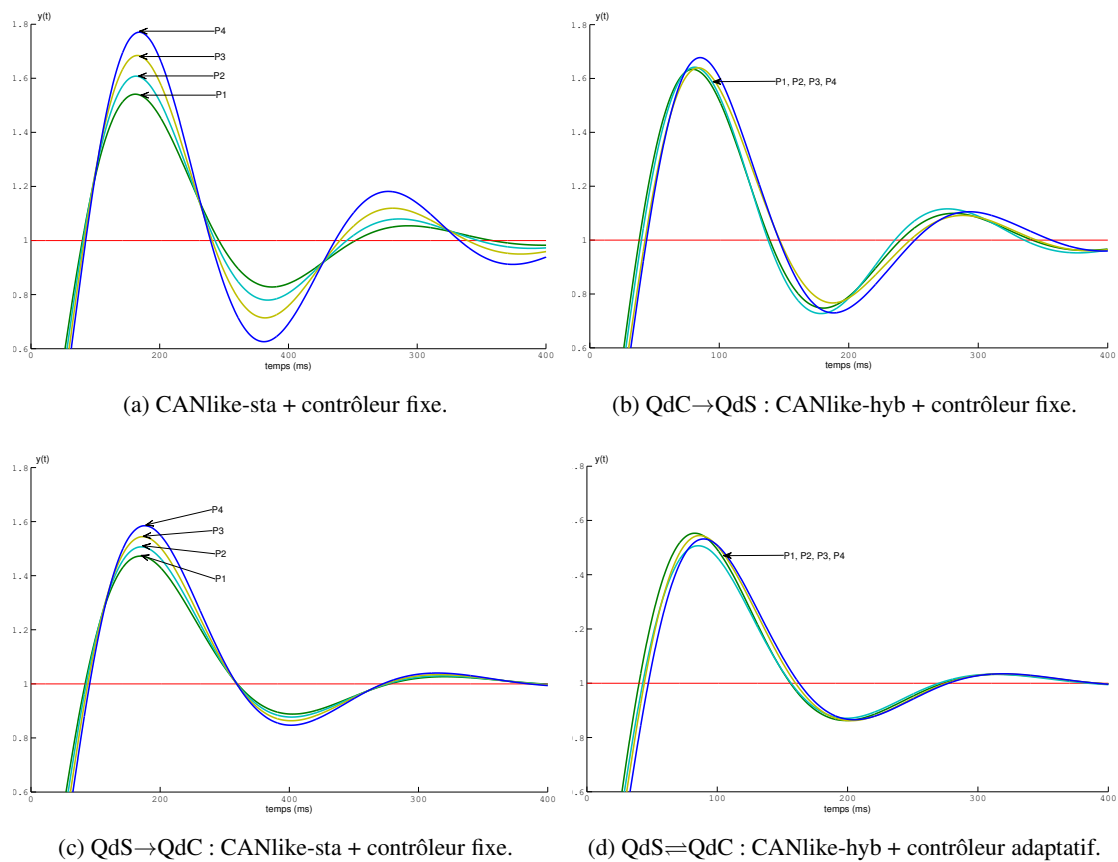


FIGURE 5.14 – CANlike : Récapitulation des réponses temporelles.

## 5.6 Conclusion

Deux points nous paraissent très importants à relever relativement à l'étude effectuée dans ce chapitre :

- tout d'abord, une différence importante par rapport aux protocoles MAC CSMA sans collision dans le réseau filaire CAN : le rôle pénalisant de la durée des priorités (que ce soit avec les messages d'annonce des protocoles de type BlackBurst ou avec le champ ID des protocoles CANlike) pour l'obtention de performances en QdC (ceci découle, pour les protocoles BlackBurst, des durées des TBB et des temps d'observations et, pour les protocoles CANlike, des durées des bits de l'ID et des temps de garde ainsi que du temps TOBS) ; les protocoles BlackBurst sont aussi plus limités que les protocoles CANlike en terme de nombre d'applications que l'on peut implanter avec des performances convenables ;
- ensuite, on retrouve dans les cas où on a quatre applications de contrôle-commande des résultats du même type que ceux que nous avons trouvés avec le réseau filaire CAN ce qui est logique (avec les priorités statiques, on a des performances en QdC qui suivent d'ordre des priorités ; avec les priorités hybrides, on a une répartition équitable des performances due à la prépondérance de la partie priorité dynamique par rapport à la partie priorité statique) ; les relations QdS→QdC et QdC→QdS donnent aussi les mêmes tendances de performances qu'avec le réseau CAN filaire. Tout ceci valide l'implantation qui a été faite des protocoles sans collision dans l'outil TrueTime.



# Conclusion et perspectives

---

## Contexte

Le travail présenté dans cette thèse concerne les systèmes commandés en réseau (« Networked Control Systems »), systèmes pluridisciplinaires (Automatique, Informatique, Réseaux de communication) qui connaissent un grand développement dans le contexte technologique actuel. Notre travail traite l'aspect (Automatique, Réseaux de communication) et concerne la problématique du lien Qualité de Contrôle (QdC) avec la Qualité de Service (QoS), fournie par les réseaux en se focalisant sur des réseaux locaux, et l'aspect ordonnancement des trames (fonction de la couche MAC) qui transportent les messages des flux capteur-contrôleur (flux  $f_{cc}$ ) et contrôleur-actionneur (flux  $f_{ca}$ ).

Les protocoles de la couche MAC qui sont considérés sont de type CSMA, type bien plus flexible que le type TDMA.

Le travail effectué se situe dans une démarche de co-conception c'est-à-dire où les composantes automatique et réseau ne sont pas considérées de manière indépendante mais sont des composantes liées.

## Contributions de la thèse

Ce travail présente trois contributions :

1. la première contribution concerne la définition de priorités hybrides pour gérer l'ordonnancement des trames des flux  $f_{cc}$  et  $f_{ca}$  des applications de contrôle-commande. Dans ces applications, nous considérons que le capteur est guidé par le temps (l'horloge qui définit l'échantillonnage) et le contrôleur est guidé par les événements (trames reçues du capteur) ; ces priorités hybrides comportent deux parties (une partie qui représente la priorité du flux (c'est une priorité statique définie *à priori*) ; une partie que l'on appelle priorité dynamique c.-à-d. partie qui veut traduire les besoins des applications en termes d'urgence de la transmission d'une trame). La priorité dynamique, qui est calculée par le contrôleur, dépend du signal de commande de l'application et varie d'une valeur  $P_{\min}$  à une valeur  $P_{\max}$ . Le concept de priorité hybride est un concept particulièrement adapté pour les application de contrôle-commande où on a les notions de régimes transitoires (besoin d'action fortes) et de régimes permanents (besoins faibles). Nous disons que ce concept permet de mettre en œuvre la relation QdC→QoS c'est-à-dire de traduire qu'une QdC faible génère un besoin

important de QdS. Nous avons expérimenté trois modalités de priorités hybrides en simulant avec l'outil TrueTime, des SCR sur le réseau CAN (les priorités sont exprimées au moyen du champ ID).

Les résultats obtenus, en considérant plusieurs applications de contrôle-commande identiques (mêmes fonctions de transfert) montrent l'intérêt des priorités hybrides : on obtient, du fait du rôle de la partie priorité dynamique, des performances en QdC très voisines (c.-à-d. faible influence de la partie priorité statique). Ceci n'est pas possible si on utilise des priorités statiques pour les trames des flux (les performances en QdC d'applications de contrôle-commande identiques sont discriminées par ces priorités statiques) ;

2. la deuxième contribution concerne la mise en œuvre, toujours dans le cadre du réseau CAN, de la relation  $QdS \rightleftharpoons QdC$  qui consiste en la combinaison de la relation  $QdC \rightarrow QdS$  (notre première contribution) et la relation  $QdS \rightarrow QdC$  qui veut traduire, compte tenu d'une QdS insuffisante, une action de compensation du retard faite par le contrôleur pour maintenir une QdC que l'on s'est fixée.

Nous avons mis en œuvre la relation  $QdS \rightarrow QdC$  sur la base, d'une part, d'une évaluation du retard introduit dans la boucle d'une application de contrôle-commande et, d'autre part, d'une action de compensation par la méthode des pôles dominants.

La mise en œuvre de la relation  $QdS \rightleftharpoons QdC$  a montré, en considérant toujours plusieurs applications de contrôle-commande identiques, son intérêt pour permettre d'obtenir des QdC bonnes et voisines (effet combiné des relations  $QdC \rightarrow QdS$  et  $QdS \rightarrow QdC$ ) ;

3. la troisième contribution se situe dans le cadre de la considération que les réseaux locaux sans fil sont de plus en plus utilisés dans de nombreux domaines et, en particulier, pour implanter des applications de contrôle-commande distribuées. Cette contribution a deux composantes :
  - la première composante est la spécification pour la couche MAC de protocoles CSMA sans collisions sur la base de priorités qui sont associées aux trames de données et qui peuvent être exprimées soit sous la forme de messages d'annonce BlackBurst précédant les trames de données ou soit dans un champ identificateur (ID) au début de la trame de données (comme avec le réseau filaire CAN). Ces protocoles ont également été implantés dans l'outil TrueTime.
  - la deuxième composante concerne l'étude de l'implantation d'applications de contrôle-commande avec les différents protocoles CSMA sans collision et la comparaison entre ces différents protocoles et également le protocole IEEE 802.11 DCF. Cette étude montre, tout d'abord, en ce qui concerne les protocoles sans collision, l'influence de la durée des messages d'annonce BlackBurst et du champ ID. Cette étude montre aussi l'intérêt de la mise en œuvre de la relation  $QdS \rightleftharpoons QdC$ .

## Perspectives

Le travail présenté dans cette thèse devrait être, à notre avis, poursuivi à la fois sur l'aspect réseau sans fil et sur l'aspect automatique :

- aspect réseau sans fil où deux points nous paraissent intéressants à explorer :
  - . la spécification des protocoles MAC sans collision avec des messages d'annonce, en considérant des priorités hybrides, devrait être complétée par l'étude de protocoles qui différencient le type d'implantation de la partie priorité statique et de la partie priorité dynamique ; par exemple, considérer des « BlackBurst » (séquence d'impulsions) pour l'une ou l'autre des parties et considérer des « WhiteBurst » (durée d'écoute du canal) pour l'autre partie ; il faudrait évaluer, à la fois, les évolutions des temps d'accès et également l'aspect énergétique ;
  - . les réseaux sans fil multi-sauts connaissant actuellement un grand développement, il s'agirait de considérer la problématique de la spécification de protocoles sans collision dans un tel contexte pour appréhender l'implantation d'application avec de fortes contraintes temporelles comme les applications de contrôle-commande.
- aspect automatique : nous avons ici considéré seulement un type d'application de contrôle-commande (une intégration et une constante de temps dans la chaîne directe) et un type de réseau correcteur (PD). Il nous paraît important de considérer différents types d'applications avec différentes dynamiques et également différents types de réseaux correcteurs (PI, PID) et de se poser des questions, à la fois, sur la stabilité et la précision, compte tenu de différents types d'entrées (échelon de position, échelon de vitesse, échelon d'accélération).





# L'effet des zéros supplémentaires sur la performance du système second ordre

La fonction de transfert du système bouclé :

$$F(s) = \frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

Avec un zéro supplémentaire  $z_o = -1/T_z$  ( $z_o$  est positif si  $T_z < 0$  et  $z_o$  est négatif si  $T_z > 0$ ).  
La fonction de transfert devient :

$$F_1(s) = \frac{Y_1(s)}{R(s)} = \frac{\omega_n^2(1 + T_z s)}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

On a donc :

$$F_1(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} + \frac{T_z \omega_n^2 s}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

Pour l'entrée de type échelon, la réponse temporelle est :

$$y_1(t) = y(t) + T_z \frac{dy(t)}{dt} = \begin{cases} y(t) - |T_z| \frac{dy(t)}{dt} & \text{si } T_z < 0 \text{ (zéro positif)} \\ y(t) + |T_z| \frac{dy(t)}{dt} & \text{si } T_z > 0 \text{ (zéro négatif)} \end{cases}$$

La figure A.1 représente l'effet du zéro positif ( $T_z < 0$ ) et la figure A.2 représente l'effet du zéro négatif ( $T_z > 0$ ). On voit sur ces figures que le dépassement croît.

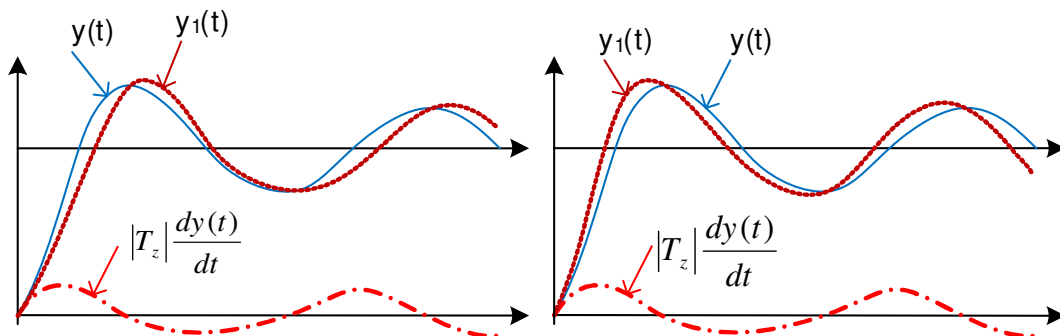


FIGURE A.1 – l'effet du zéro positif  
( $T_z < 0$ )

FIGURE A.2 – L'effet du zéro négatif  
( $T_z > 0$ )

**Remarque :** quand le zéro s'approche de l'origine (c.-à-d.  $|T_z|$  croît), la composante  $T_z \frac{dy(t)}{dt}$  devient croissante, donc le dépassement  $O$  augmente.



## Résultats $\Delta J/J_0\%$ pour DCF-WiFi

| n°        | P <sub>1</sub> |
|-----------|----------------|
| 1         | 118            |
| 2         | 80.9           |
| 3         | 88.9           |
| 4         | 116            |
| 5         | 102            |
| 6         | 100            |
| 7         | 114            |
| 8         | 82.3           |
| 9         | 73.6           |
| 10        | 91.4           |
| 11        | 85.1           |
| 12        | <b>122</b>     |
| 13        | 78.7           |
| 14        | 99.5           |
| 15        | 69.8           |
| 16        | 94.9           |
| 17        | 80.0           |
| 18        | 99.3           |
| 19        | 107            |
| 20        | 76.1           |
| moyenne   | 93.9           |
| écart max | 28.1           |

TABLE B.1 – Un procédé ( $\Delta J/J_0\%$ ).

| n°        | P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> | P <sub>4</sub> |
|-----------|----------------|----------------|----------------|----------------|
| 1         | 93.5           | 125            | 77.7           | 209            |
| 2         | 150            | 87.6           | 97             | 151            |
| 3         | 81.6           | 128            | 129            | <b>301</b>     |
| 4         | 88.3           | 188            | 75.2           | 288            |
| 5         | 255            | 126            | 187            | 61.1           |
| 6         | 164            | 125            | 56,4           | 223            |
| 7         | 117            | 214            | 178            | 99.5           |
| 8         | <b>271</b>     | 158            | 68,4           | 60.3           |
| 9         | 57.1           | 179            | 204            | 150            |
| 10        | 167            | 113            | 140            | 103            |
| 11        | 219            | 169            | 119            | 78.4           |
| 12        | 87.2           | 79.7           | 115            | 155            |
| 13        | 82.8           | 80.2           | 178            | 271            |
| 14        | 90.1           | 196            | 117            | 106            |
| 15        | 66.9           | 148            | 152            | 253            |
| 16        | 182            | 146            | <b>290</b>     | 64.8           |
| 17        | 220            | 70.2           | 111            | 168            |
| 18        | 155            | 48.9           | 191            | 209            |
| 19        | 119            | 132            | 127            | 154            |
| 20        | 122            | <b>220</b>     | 81.8           | 133            |
| moyenne   | 139            | 136            | 134            | 161            |
| écart max | 132            | 84             | 156            | 140            |

TABLE B.2 – Quatre procédés ( $\Delta J/J_0\%$ ).



# Bibliographie

- [Ariba 2009] Yassine Ariba. *Sur la stabilité des systèmes à retards variant dans le temps : théorie et application au contrôle de congestion d'un routeur*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, Novembre 2009.
- [Åström 1997] Karl J. Åström et Björn Wittenmark. *Computer-controlled systems : Theory and design*. Prentice Hall, 3<sup>rd</sup> edition, 1997.
- [Aubrun 2010] Christophe Aubrun, Daniel Simon et Ye-Qiong Song. *Co-design Approaches for Dependable Networked Control Systems*. ISTE/John Wiley, 2010.
- [Baillieul 2007] J. Baillieul et P.J. Antsaklis. *Control and Communication Challenges in Networked Real-Time Systems*. Proceedings of the IEEE, vol. 95, no. 1, pages 9–28, January 2007.
- [Ben Gaïd 2010] Mohamed El Mongi Ben Gaïd, David Robert, Olivier Sename, Alexandre Seuret et Daniel Simon. *Computing aware control*. In Co-Design Approaches to Dependable Networked Control Systems [Aubrun 2010], pages 63–103.
- [Berbra 2010] Cédric Berbra, Sylvianne Gentil, Suzanne Lesecq et Daniel Simon. *Implementation : Control and Diagnosis for an Unmanned Aerial Vehicle*. In Co-Design Approaches to Dependable Networked Control Systems [Aubrun 2010], pages 267–304.
- [Bosch 1991] GmbH Bosch. *CAN specification 2.0*, 1991.  
url : [www-micro.deis.unibo.it/~magagni/can2spec.pdf](http://www-micro.deis.unibo.it/~magagni/can2spec.pdf) (accessed on 10/10/2011).
- [Boughanmi 2009] N. Boughanmi, Y.Q. Song et E Rondeau. *Online adaptation of the IEEE 802.15.4 parameters for wireless networked control systems*. Guy Juanole et Seung Ho Hong, editors, 8<sup>th</sup> IFAC international conference on Fieldbuses and networks in industrial and embedded systems (FeT 2009), Ansan, Republic of Korea, Mai 2009.
- [Buttazzo 2004] Giorgio Buttazzo, Manel Velasco, Pau Marti et Gerhard Fohler. *Managing Quality-of-Control Performance Under Overload Conditions*. 16<sup>th</sup> Euromicro Conference on Real-Time Systems (ECRTS'04), pages 53–60, Catania, Sicily, Italy, 2004.
- [Buttazzo 2007] Giorgio Buttazzo, Manel Velasco et Pau Marti. *Quality-of-Control Management in Overloaded Real-Time Systems*. IEEE Transactions on Computers, vol. 56, pages 253–266, 2007.
- [Callaway 2002] E. Callaway, P. Gorday, L. Hester, Gutierrez J.A., Naeve M., Heile B. et V. Bahl. *Home networking with IEEE 802.15.4 : a developing standard for low-rate wireless personal area networks*. IEEE Communications Magazine, vol. 40, pages 70–77, August 2002.
- [Castane 2006] Rosa Castane, Pau Marti, Manel Velasco, Anton Cervin et Dan Henriksson. *Resource Management for Control Tasks Based on the Transient Dynamics of Closed-Loop Systems*. 18<sup>th</sup> Euromicro Conference on Real-Time Systems, pages 171–182, Washington, DC, USA, 2006. IEEE Computer Society.
- [C.Dorf 1995] Richard C.Dorf et Robert H.Bishop. *Modern Control Systems*. Addison-Wesley, 7<sup>th</sup> edition, 1995.

- [Cervin 2002] Anton Cervin, Johan Eker, Bo Bernhardsson et Karl-Erik Arzen. *Feedback-Feedforward Scheduling of Control Tasks*. Real-Time Syst., vol. 23, pages 25–53, July 2002.
- [Cervin 2003] Anton Cervin. *Integrated Control and Real-Time Scheduling*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, April 2003.
- [Cervin 2011] A. Cervin, M. Velasco, P. Marti et A. Camacho. *Optimal Online Sampling Period Assignment : Theory and Experiments*. Control Systems Technology, IEEE Transactions on, vol. 19, no. 4, pages 902–910, July 2011.
- [CiA 2002] CAN in Automation group CiA. *CAN, CANopen, DeviceNet*, 2002.  
url : [www.CAN-CiA.de](http://www.CAN-CiA.de) (accessed on 10/10/2011).
- [Colandairaj 2007] Jeremy Colandairaj, George W. Irwin et William G. Scanlon. *A Co-Design Solution for Wireless Feedback Control*. IEEE International Conference on Networking, Sensing and Control, ICNSC 2007, pages 404–409, London, UK, April 2007.
- [Crow 1997] B.P. Crow, Widjaja I., Kim L.G. et Sakai P.T. *IEEE 802.11 Wireless Local Area Networks*. IEEE Communications Magazine, vol. 35, pages 116–126, September 1997.
- [Deng 1999] Dr-Jiunn Deng et Ruay-Shiung Chang. *A Priority Scheme for IEEE 802.11 DCF Access Method*. IEICE Trans Commun (Inst Electron Inf Commun Eng), vol. E82-B, no. 1, pages 96–102, 1999.
- [Diouri 2007] Idriss Diouri, Jean-Philippe Georges et Eric Rondeau. *Accommodation of delays for networked control systems using classification of service*. IEEE International Conference On Networking, Sensing and Control - ICNSC'07, London, UK, pages 410–415, April 2007.
- [Eker 2000] Johan Eker, Per Hagander et Karl-Erik Årzén. *A feedback scheduler for real-time controller tasks*. Control Engineering Practice, vol. 8, no. 12, pages 1369 – 1378, 2000.
- [Franklin 2002] Gene F. Franklin, J. David Powell et Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, 4<sup>th</sup> edition, 2002.
- [Gao 2005] Yan Gao, Dah-Ming Chiu et John C. S. Lui. *The Fundamental Role of Hop Distance in IEEE802.11 Multi-Hop Ad Hoc Networks*. Proceedings of the 13TH IEEE International Conference on Network Protocols, pages 75–84, Washington, DC, USA, 2005. IEEE Computer Society.
- [Guerrero Castellanos 2008] José Fermi Guerrero Castellanos. *Estimation de l'attitude et commande bornée en attitude d'un corps rigide : application à un mini-hélicoptère à quatre rotors*. Thèse de Doctorat, Département Automatique de GIPSA-lab, Université Joseph Fourier, Grenoble, Janvier 2008.
- [Guerrero Castellanos 2011] José Fermi Guerrero Castellanos, Nicolas Marchand, Ahmad Hably, Suzanne Lesecq et Jérôme Delamare. *Bounded attitude control of rigid bodies : Real-time experimentation to a quadrotor mini-helicopter*. Control Engineering Practice, pages 790–797, August 2011.
- [Gupta 2010] Rachana Ashok Gupta et Mo-Yuen Chow. *Networked Control System : Overview and Research Trends*. IEEE Transactions on Industrial Electronics, vol. 57, no. 7, pages 2527–2535, 2010.

- [Halevi 1988] Yoram Halevi et Asok Ray. *Integrated communication and control systems : part 1 - Analysis*. Journal of Dynamic Systems, Measurement, and Control, vol. 110, pages 367–373, December 1988.
- [Henriksson 2003] Dan Henriksson, Anton Cervin et Karl Eric Arzen. *Real-time Control System Simulation with Matlab/Simulink*. Nordic MATLAB Conference, Copenhagen, Denmark, October 2003.
- [Henriksson 2005] D. Henriksson et A. Cervin. *Optimal On-line Sampling Period Assignment for Real-Time Control Tasks Based on Plant State Information*. 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 (CDC-ECC'05), pages 4469–4474, Seville, Spain, December 2005.
- [IEEE 802.15 2003] IEEE-SA Standards Board IEEE 802.15. *Part 15.4 : wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)*. IEEE Computer Society, 2003. url : [www.cs.jhu.edu/~cliang4/public/datasheets/802.15.4-2003.pdf](http://www.cs.jhu.edu/~cliang4/public/datasheets/802.15.4-2003.pdf) (accessed on 10/10/2011).
- [Juanole 2005a] Guy Juanole, Christophe Calmettes, Gérard Mouney et Marek Peca. *On the implementation of a process control system on a CAN network : linking the process control parameters to the network parameters*. 10<sup>th</sup> IEEE International conference on Emerging Technologies and Factory Automation (ETFA 2005), Catania, Italy, September 2005.
- [Juanole 2005b] Guy Juanole, Gérard Mouney, Christophe Calmettes et Marek Peca. *Fundamental Considerations for Implementing Control Systems on a CAN Network*. 6<sup>th</sup> IFAC International conference on Fieldbus Systems and their Applications (FET 2005), Puebla, Mexico, November 2005.
- [Juanole 2007a] Guy Juanole et Gérard Mouney. *Networked Control Systems : Definition and Analysis of a Hybrid Priority Scheme for the Message Scheduling*. 13<sup>th</sup> IEEE conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007), Daegu, Korea, August 2007.
- [Juanole 2007b] Guy Juanole et Gérard Mouney. *Using an hybrid traffic scheduling in networked control systems*. European Control Conference 2007, Kos, Greece, July 2007.
- [Juanole 2008] Guy Juanole, Gérard Mouney et Christophe Calmettes. *On different priority schemes for the message scheduling in Networked Control Systems : Definition and Analysis of a Hybrid Priority Scheme for the Message Scheduling*. 16<sup>th</sup> Mediterranean Conference on Control and Automation (MED'08), Ajaccio, Corsica, France, June 2008.
- [Juanole 2010] Guy Juanole, Gérard Mouney, Dominique Sauter, Christophe Aubrun et Christophe Calmettes. *Decision Making Improvement for Diagnosis in Networked Control Systems based on Dynamic Message Scheduling*. Proceedings of the 18<sup>th</sup> Mediterranean Conference on Control and Automation, MED'10, Marrakech, Maroc, June 2010.
- [Kumar 2006] Sunil Kumar, Vineet S. Raghavan et Deng Jing. *Medium Access Control protocols for ad hoc wireless networks : A survey*. Ad Hoc Networks, vol. 4, pages 326–358, May 2006.
- [Kuo 2003] Benjamin C. Kuo et Farid Golnaraghi. *Automatic Control System*. John Wiley & Sons, 8<sup>th</sup> edition, 2003.



- [Kwoon 2000] Byung-Moon Kwoon, Hee-Seob Ryu et Oh-Kyu Kwo. *Transient Response Analysis and Compensation of the Second Order System with One RHP Real Zero*. Transactions on Control, Automation and Systems Engineering, vol. 2, no. 4, December 2000.
- [Lian 2002] Feng-Li Lian, J. Moyne et D. Tilbury. *Network design consideration for distributed control systems*. IEEE Transactions on Control Systems Technology, vol. 10, no. 2, pages 297–307, March 2002.
- [Luck 1990] Rogelio Luck et Asok Ray. *An observer-based compensator for distributed delays*. Automatica, vol. 26, pages 903–908, September 1990.
- [Mangold 2003] S. Mangold, Sunghyun Choi, G.R. Hiertz, O. Klein et B. Walke. *Analysis of IEEE 802.11e for QoS support in wireless LANs*. IEEE Wireless Communications, vol. 35, pages 40–50, December 2003.
- [Marti 2002] Pau Marti. *Analysis and design of Real-Time Control Systems with Flexible Timing Constraints*. PhD thesis, Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Spain, June 2002.
- [Marti 2004] P. Marti, J. Yopez, M. Velasco, R. Villa et J.M. Fuertes. *Managing quality-of-control in network-based control systems by controller and message scheduling co-design*. IEEE Transactions on Industrial Electronics, vol. 51, no. 6, pages 1159–1167, December 2004.
- [Marti 2009] Pau Marti, Caixue Lin, Scott A. Brandt, Manel Velasco et Josep M. Fuertes. *Draco : Efficient Resource Management for Resource-Constrained Control Tasks*. IEEE Transactions on Computers, vol. 58, pages 90–105, 2009.
- [Moyne 2007] J.R. Moyne et D.M. Tilbury. *The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data*. Proceedings of the IEEE, vol. 95, no. 1, pages 29–47, January 2007.
- [Navet 1999] Nicolas Navet. *Evaluation de performances temporelles et optimisation de l'ordonnancement de tâches et messages*. PhD thesis, Institut National Polytechnique de Lorraine, Novembre 1999.
- [Ng 2004] Ping Chung Ng et Soung Chang Liew. *Offered Load control in IEEE 802.11 Multi-hop Ad-hoc Networks*. Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor System, MASS'04, October 2004.
- [Nguyen Xuan 2010a] Hung Nguyen Xuan, Guy Juanole, Gérard Mouney et Christophe Calmettes. *Networked Control System (NCS) on a network CAN : on the Quality of Service (QoS) and Quality of Control (QoC) provided by different message scheduling schemes based on hybrid priorities*. 8<sup>th</sup> IEEE International Workshop on Factory Communication Systems (WFCS 2010), pages 261–270., Nancy, France, May 2010.
- [Nguyen Xuan 2010b] Hung Nguyen Xuan, Guy Juanole, Gérard Mouney et Christophe Calmettes. *Wireless communication networks and process control applications : studying the influence of MAC protocols*. Proceedings of IEEE Globecom 2010, Workshop SaCoNAS, pages 1632–1637, Miami, Florida, USA, December 2010.
- [Nguyen Xuan 2011a] Hung Nguyen Xuan. *Networked Control Systems : Quality of Service aware Quality of Control adaptation*. 16<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011), Toulouse, France, Septembre 2011.

- [Nguyen Xuan 2011b] Hung Nguyen Xuan et Gérard Mouney. *Spécification du protocole « CANlike » pour la couche MAC des réseaux locaux sans fil*. École d'Été Temps Réel 2011, pages 195–198, Brest, France, September 2011.
- [Nguyen Xuan 2012] Hung Nguyen Xuan et Guy Juanolet. *Networked Control Systems (NCSs) : on the interest of a co-design approach based on interplays between Quality of Control and Quality of Service*. Submitted to 9<sup>th</sup> IEEE International Workshop on Factory Communication Systems (WFCS 2012), Lemgo/Detmold, Germany, May 2012.
- [Nilsson 1998] Johann Nilsson. *Real-Time Control with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, 1998.
- [Ohlin 2007] Martin Ohlin, Dan Henriksson et Anton Cervin. *TrueTime 1.5 – Reference Manual*. Lund Institute of Technology, Sweden, January 2007.
- [Pal 2002] Abhishek Pal, Atakan Dogan et Füsün Özgüner. *MAC Layer Protocols for Real-Time Traffic in Ad-Hoc Wireless Networks*. Proceedings of the 2002 International Conference on Parallel Processing (ICPP'02), pages 539–546, Washington, DC, USA, 2002. IEEE Computer Society.
- [Pereira 2007a] N. Pereira, B. Andersson et E. Tovar. *WiDom : A Dominance Protocol for Wireless Medium Access*. IEEE Transactions on Industrial Informatics, vol. 3, no. 2, pages 120–130, May 2007.
- [Pereira 2007b] N. Pereira, B. Andersson, E. Tovar et A. Rowe. *Static-Priority Scheduling over Wireless Networks with Multiple Broadcast Domains*. 28<sup>th</sup> IEEE Real-Time Systems Symposium (RTSS 2007), pages 447–458, December 2007.
- [Richard 2007] Jean-Pierre Richard et Thierry Divoux, editors. *Systèmes commandés en réseau*. Hermès, Paris, Mars 2007.
- [Robert 2006] David Robert, Olivier Sename et Daniel Simon. *Synthesis of a sampling period dependent controller using LPV approach*. 5<sup>th</sup> IFAC Symposium on Robust Control Design (ROCOND 2006), Toulouse, France, July 2006.
- [Seto 1996] D. Seto, J.P. Lehoczky, L. Sha et K.G. Shin. *On task schedulability in real-time control systems*. IEEE Real-Time Systems Symposium'96, pages 13–21, December 1996.
- [Sevely 1973] Yves Sevely. *Systèmes et asservissements linéaires et échantillonnés*. Dunod Université, 1973.
- [Simon 2003] Daniel Simon, Olivier Sename, David Robert et Olivier Testa. *Real-time and delay-dependent control co-design through feedback scheduling*. CERTS'03 : Co-design in Embedded Real-time Systems, Porto, Portugal, 2003.
- [Sobrinho 1996] J.L. Sobrinho et A.S. Krishnakumar. *Real-time traffic over the IEEE 802.11 medium access control layer*. Bell Labs Technical Journal, vol. 10, no. 1, pages 172–187, 1996.
- [Sobrinho 1999] J.L. Sobrinho et A.S. Krishnakumar. *Quality-of-service in ad hoc carrier sense multiple access wireless networks*. IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pages 1353–1368, August 1999.

- [Song 2009] Ye-Qiong Song. *Networked Control Systems : From Independent Designs of the Network QoS and the Control to the Co-design*. Guy Juanolet et Seung Ho Hong, editors, 8<sup>th</sup> IFAC international conference on Fieldbuses and networks in industrial and embedded systems (FeT 2009), Hanyang University, Republic of Korea, Mai 2009.
- [Stankovic 1999] John A. Stankovic, Chenyang Lu, Sang H. Son et Gang Tao. *The case for feedback control real-time scheduling*. EuroMicro Conference on Real-Time Systems, pages 11–20, 1999.
- [Tipsuwan 2003] Y Tipsuwan et M.Y. Chow. *Control methodologies in networked control systems*. Control Engineering Practice, vol. 11, no. 10, pages 1099–1111, 2003.
- [Vaidya 2000] Nitin H. Vaidya, Paramvir Bahl et Seema Gupta. *Distributed fair scheduling in a wireless LAN*. 6<sup>th</sup> annual international conference on Mobile computing and networking (MobiCom '00), pages 167–178, New York, NY, USA, 2000. ACM.
- [Vaidya 2005] Nitin Vaidya, Anurag Dugar, Seema Gupta et Paramvir Bahl. *Distributed Fair Scheduling in a Wireless LAN*. IEEE Transactions on Mobile Computing, vol. 4, no. 6, pages 616–629, 2005.
- [Vatanski 2006] Nikolai Vatanski, Jean-Philippe Georges, Christophe Aubrun, Eric Rondeau et Sirkka-Liisa Jämsä Jounela. *Control compensation based on upper bound delay in networked control systems*. 17<sup>th</sup> International Symposium on Mathematical Theory of Networks and Systems (MTNS), Kyoto, Japan , July 2006.
- [Velasco 2006] M. Velasco, P. Marti, R. Castane, J. Guardia et J.M. Fuertes. *A CAN Application Profile for Control Optimization in Networked Embedded Systems*. IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on, pages 4638–4643, November 2006.
- [Walsh 2001] G.C. Walsh et Hong Ye. *Scheduling of networked control systems*. Control Systems Magazine, IEEE, vol. 21, no. 1, pages 57–65, February 2001.
- [Willig 2005] A. Willig, K. Matheus et A. Wolisz. *Wireless Technology in Industrial Networks*. Proceedings of the IEEE, vol. 93, pages 1130–1151, June 2005.
- [Willig 2008] Andreas Willig. *Recent and Emerging Topics in Wireless Industrial Communications : A Selection*. IEEE Transactions on Industrial Informatics, vol. 4, no. 2, pages 102–124, 2008.
- [Xu 2002] Kaixin Xu, M. Gerla et Sang Bae. *How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?* Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE, volume 1, pages 72–76, November 2002.
- [Yepez 2003] J. Yepez, P. Marti et J. Fuertes. *The Large Error First (LEF) Scheduling Policy for Real-Time Control Systems*. In Work-in-Progress Session of the 24<sup>th</sup> IEEE Real-Time Systems Symposium (RTSS'03), Cancun, Mexico, Decembre 2003.
- [You 2003a] Tiantong You, Chi-Hsiang Yeh et Hossam Hassanein. *CSMA/IC : A New Class of Collision-free MAC Protocols for Ad Hoc Wireless Networks*. IEEE Symposium on Computers and Communications, vol. 0, pages 843–848, 2003.

- 
- [You 2003b] Tiantong You, Chi-Hsiang Yeh et Hossam Hassanein. *A New Class of Collision Prevention MAC Protocols for Wireless Ad Hoc Networks*. IEEE International Conference on Communications (ICC'03), pages 1135–1140, 2003.
- [Zampieri 2008] Sandro Zampieri. *Trends in Networked Control Systems*. 17<sup>th</sup> IFAC World Congress, pages 2886–2894. IFAC, 2008.
- [Zhang 2001] Wei Zhang, Michael S. Branicky et Stephen M. Phillips. *Stability of Networked Control Systems*. IEEE Control Systems Magazine, vol. 21, pages 84–99, 2001.
- [Zuberi 1997] K. Zuberi et K. Shin. *Scheduling messages on controller area network for real-time CIM applications*. IEEE Transactions On Robotics And Automation, vol. 13, no. 2, pages 310–314, 1997.
- [Zuberi 2000] K. Zuberi et K. Shin. *Design and Implementation of Efficient Message Scheduling for Controller Area Network*. IEEE Transactions on Computers, vol. 49, no. 2, pages 182–188, 2000.



# Liste des publications

## Conférences internationales et revues

**Xuan Hung NGUYEN.** *Networked Control Systems : Quality of Service aware Quality of Control adaptation.* 16<sup>th</sup> IEEE International conference on Emerging Technologies and Factory Automation (ETFAs 2011), Toulouse (France), 5-9 September 2011.

**Xuan Hung NGUYEN, G.JUANOLE, G.MOUNEY, C.CALMETTES.** *Wireless communication networks and process control applications : study the influence of MAC protocols.* Global Communications Conference (GLOBECOM 2010), Miami (USA), pp.1689-1694, 6-10 December 2010.

**Xuan Hung NGUYEN, G.JUANOLE, G.MOUNEY, C.CALMETTES.** *Networked Control System (NCS) on a network CAN : on the Quality of Service (QoS) and Quality of Control (QoC) provided by different message scheduling schemes based on hybrid priorities.* 8<sup>th</sup> IEEE International Workshop on Factory Communication Systems (WFCS 2010), pp.261-270, Nancy (France), 18-21 May 2010.

**C.AUBRUN, B.BRAHIMI, GEORGES, G.JUANOLE, G.MOUNEY, X-H.NGUYEN RONDEAU.** *QoS-aware dynamic Network QoS Adaptation.* Ouvrages (contribution) : Co-Design Approaches to Dependable Networked Control Systems, ISTE/ John Wiley , pp.105-147.

## Conférences nationales

**Xuan Hung NGUYEN, Gérard MOUNEY.** *Spécification du protocole « CANlike » pour la couche MAC des réseaux locaux sans fil.* École d'été Temps Réel (ETR 2011), pp.195-198 - Université de Bretagne Occidentale - Brest, 29 Août au 2 Septembre 2011.

**Xuan Hung NGUYEN.** *Réseaux de communication sans fil et applications de contrôle commande : étude de l'influence du protocole MAC.* Congrès EDSYS, Toulouse, 6-7 Mai 2010.

**Xuan-Hung Nguyen.** *Réseaux de communication sans fil et applications de contrôle - commande.* 3e Journée du GT ARC (Automatique et Réseaux de Communication) GDR CNRS MACS, Paris, 2 Avril 2010.



**Auteur :** NGUYEN Xuan Hung

**Titre :** Réseaux de Communication et Applications de Contrôle-Commande

**Directeurs de thèse :** Guy JUANOLE et Gérard MOUNEY

---

**RÉSUMÉ :**

Cette thèse se situe dans le domaine des Systèmes Commandés en Réseaux, systèmes pluridisciplinaires qui connaissent un grand développement dans le contexte technologique actuel et qui introduisent de nouveaux problèmes de recherche, en particulier, pour les communautés Automatique et Réseau de Communication. L'objectif est de montrer l'intérêt de considérer un lien bidirectionnel entre la Qualité de Contrôle (QdC) (stabilité et performance des applications de contrôle-commande) et la Qualité de Service (QdS) (retards dans les transmissions) en se focalisant sur les réseaux locaux et plus particulièrement sur l'ordonnancement des trames dans la sous couche MAC où on considère des protocoles MAC de type CSMA.

Notre travail se situe tout d'abord dans le cadre du réseau CAN où nous faisons trois propositions. La première proposition consiste en des solutions de QdS pour des applications de contrôle-commande sur la base du concept de priorité hybride, pour l'ordonnancement de trames, dans laquelle nous définissons une partie priorité dynamique qui dépend d'un paramètre de QdC. Ceci permet de réaliser une relation  $QdC \rightarrow QdS$ . La deuxième proposition consiste en des solutions de QdC par une méthode de compensation de retard (maintenance des pôles dominants) ce qui permet de réaliser une relation  $QdS \rightarrow QdC$ . La troisième proposition consiste à réaliser une relation  $QdC \rightleftharpoons QdS$  qui assure lors de l'implantation de plusieurs applications, une propriété d'équité comportementale. Nous abordons ensuite le cadre particulier des réseaux locaux sans fil où, d'une part, nous définissons des protocoles MAC de type CSMA sans collision sur la base de priorités et, d'autre part, nous montrons la faisabilité des idées développées dans le cadre du réseau CAN.

---

**Mots clés :**

Systèmes Commandés en Réseaux, Protocoles MAC, Réseaux sans fil, Automatique, Qualité de Contrôle, Qualité de Service, Co-conception.

---

**Discipline :** Systèmes Embarqués

**Date et lieu de soutenance :** le 15 décembre 2011 à Toulouse

**Unité de Recherche :** Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS),  
7 avenue du Colonel Roche 31077 Toulouse Cedex 4





**Author :** NGUYEN Xuan Hung

**Title :** Communication Networks and Process Control Applications

**Supervisors :** Guy JUANOLE and Gérard MOUNEY

---

**ABSTRACT :**

This thesis is in the field of Networked Controlled Systems, which are multidisciplinary systems and which experience a great development in the context of the current technology and introduce new research problems, particularly for the communities of Automatic Control and Communication Network. The aim is to show the interest in considering a bidirectional link between the Quality of Control (QoC) (stability and performance of process control applications) and Quality of Service (QoS) (delays in the transmissions) and to focus on Local Area Networks and, more particularly, on the scheduling of frames in the MAC layer in which we consider CSMA MAC protocols.

Our work first concerns a CAN network where we have three proposals. The first proposal consists of solutions for QoS, for process control applications, which are based on the concept of hybrid priority for the frame scheduling in which we define a dynamic priority part which depends on a QoC parameter. This allows to implement a relation  $QoC \rightarrow QoS$ . The second proposal consists of QoC solutions by using a compensation method for time delays (maintenance of dominant poles) which allows to implement a relation  $QoS \rightarrow QoC$ . The third proposal consists in implementing a relation  $QoS \rightleftharpoons QoC$  that ensures, for the implementation of several applications, a fairness property. We then discuss the specific context of Wireless LANs, where, on the one hand, we define collision-free CSMA MAC protocols on the basis of priorities and, on the other hand, we show the feasibility of the ideas developed in the CAN network.

---

**Keywords :**

Networked Control Systems, MAC protocols, Wireless Networks, Automatic Control, Quality of Control, Quality of Service, Co-design.

---

**Discipline :** Embedded Systems

**Date and place of the PhD defense :** December 15, 2011 in Toulouse

**Research Unit :** Laboratory for Analysis and Architecture of Systems (LAAS-CNRS), 7 avenue du Colonel Roche 31077 Toulouse Cedex 4



