ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉE À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE
M. Ing.

PAR
Safwen BOUANEN

INTERFACE DE TRANSDUCTEURS INTELLIGENTS TOLERANTE AUX PANNES POUR DES APPLICATIONS AVIONIQUES CRITIQUES

MONTRÉAL, LE 6 JANVIER 2014

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Claude Thibeault, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Yvon Savaria, codirecteur de mémoire
Département de génie électrique à l'École Polytechnique de Montréal

M. Georges Kaddoum, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Guchuan Zhu, membre du jury
Département de génie électrique à l'École Polytechnique de Montréal

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 12 DÉCEMBRE 2013

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens à remercier vivement mes professeurs Claude Thibeault et Yvon Savaria pour leur aide tout au long de cette maîtrise.

Je tiens ensuite à remercier Bombardier Aéronautique, Thales Canada Inc., et CRIAQ pour le financement du projet AVIO-402 et pour m'avoir offert l'occasion d'explorer le domaine des systèmes et réseaux avioniques. Je remercie également Mitacs pour le support financier qui m'a permis de passer un stage chez Bombardier Aéronautique. Je remercie également M. Yann Le Masson, ingénieur chez Bombardier, de m'avoir encadré durant mon stage. Je voudrais remercier mes collègue membre du projet AVIO 402: José-Philippe, Meng, David Federico et Davide, qui avaient enrichit mon expérience durant mon projet de maîtrise.

Finalement, je tiens spécialement à remercier mes parents, mes frères et ma sœur pour leur support inconditionnel tout au long de mes études.

INTERFACE DE TRANSDUCTEURS INTELLIGENTS TOLERANTE AUX PANNES POUR DES APPLICATIONS AVIONIQUES CRITIQUES

Safwen BOUANEN

RÉSUMÉ

Un nombre croissant de capteurs et d'actuateurs sont nécessaires dans les avions modernes afin de fournir une variété de fonctions de contrôle et de surveillance. Divers réseaux de communication ont été adoptés dans les nouveaux systèmes avioniques pour soutenir les transferts de données tout en assurant une importante bande passante et une réduction des paquets volumineux de câblage. L'intégration d'un large éventail de transducteurs dans un environnement de réseaux avioniques hétérogènes pose des problèmes nécessitant la normalisation des interfaces de transducteurs pour les systèmes avioniques. Une des solutions considérées consiste à appliquer la norme IEEE 1451 dans le développement d'interfaces de transducteurs intelligents pour les aéronefs de nouvelle génération. Par ailleurs, en ce qui concerne les applications avioniques critiques, les interfaces de transducteurs dotés de capacités de tolérance aux pannes sont nécessaires pour répondre aux exigences de fiabilité et de sûreté imposées par la réglementation et les normes. Dans ce mémoire, nous explorons les possibilités d'une interface de transducteurs intelligents qui cible une plate-forme FPGA. L'architecture proposée est basée sur la norme IEEE1451.0 et elle intègre plusieurs techniques de tolérance aux pannes qui renforcent la fiabilité et la sûreté de l'interface. Une redondance modulaire au niveau micro-architectural et un mécanisme de test en ligne assurent la détection, le diagnostic et le recouvrement des pannes. L'interface est capable de gérer des pannes transitoires et permanentes. L'architecture proposée a été conçue pour être aussi générique que possible afin de supporter une multitude d'applications avioniques critiques telles que des systèmes de commande de vol électrique (FCS), les systèmes de freinage électrique (EBS) et des systèmes de données aériennes (ADS). La fiabilité et la sûreté de l'interface sont évaluées et ont été jugées satisfaisantes. Enfin, l'interface proposée a été implémentée et validée via un prototype matériel.

Mots clés: Interface de transducteurs intelligents, IEEE1451.0, tolérance aux pannes, FPGA, fiabilité, sûreté

FAULT TOLERANT SMART TRANSDUCERS INTERFACE FOR SAFETY-CRITCAL AVIONICS APPLICATIONS

Safwen BOUANEN

ABSTRACT

An increasing number of sensing and actuation devices are needed in modern aircrafts in order to provide a variety of functions for aircrafts monitoring and control. Diverse communication networks have been adopted in new avionic systems to support data transfers while ensuring very high communication bandwidth and reducing bulky wiring bundles. Integrating a wide range of sensing and actuation elements in a heterogeneous network environment raises serious challenges requiring the standardization of transducer interfaces in avionics systems. One of the viable solutions for these problems is the adoption of the IEEE 1451 standard in the development of smart transducers interfaces for the next generation avionics systems. Another important aspect for safety-critical avionics systems is that smart transducers interfaces with embedded fault tolerance capabilities are needed to meet high reliability and safety requirements imposed by regulations and standards. In this work we propose a smart transducers interface architecture that targets an FPGA platform. This architecture is defined using the IEEE1451.0 standard and it integrates several fault tolerance techniques that strengthen the reliability and safety attributes of the interface. A core-level redundancy and an online testing mechanism ensure detection, diagnosis and recovery from faults. The interface can handle transient and permanent faults. The proposed interface has been designed to be as generic as possible in order to support a wide range of transducers that could be deployed in several avionics safety-critical applications such as flight control systems (FCSs), electric braking systems (EBSs) and air data systems (ADSs). Reliability and safety performance of the interface are investigated and shown to be satisfactory. Finally, the proposed interface design has been implemented and validated through a hardware prototype.

Keywords: Smart transducer interface, IEEE1451.0, fault tolerance, FPGA, reliability, safety.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ACE	Actuator Control Electronics
AFDX	Avionics Full-Duplex Switched Ethernet
ARINC	Aeronautical Radio Incorporated
ASIC	Application-specific integrated circuit
BIST	Built-in self-test
CAN	Convertisseur analogique-numérique
CDVE	Commande de vol électrique
CNA	Convertisseur numérique-analogique
COM	Command
COTS	Commercial off-the-shelf
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRIAQ	Consortium de Recherche et d'Innovation en Aérospatiale au Québec
DME	Distributed Modular Electronics
DSP	Digital signal processor
FCC	Flight control computer
FCRM	Flight control remote module
FD	Fault detection
FIFO	First in, first out
FMEA	Failure Mode and Effect Analysis
FPGA	Field Programmable Gate Array
IEEE	Institute of Electrical and Electronics Engineers
IMA	Integrated Modular Avionics
I2C	Inter-Integrated circuit bus
LVDT	Linear variable differential transformer

LUT	lookup table
MCU	Microcontroller unit
MEMS	Microelectromechanical system
MON	Monitor
NCAP	Network Capable Application Processor
NMR	N-Modular redundancy
PC	Personal computer
PCB	Printed circuit board
PCIe	Peripheral Component Interconnect Express
PFC	Primary Flight control
RAE	Remote Actuator Electronics
RDC	Remote Data Concentrator
RTCA	Radio Technical Commission for Aeronautics
RVDT	Rotary variable differential transformer
SCARLETT	SCAlable and ReconfigurabLe Electronics plaTform and Tools
SEU	Single Event Upset
TAI	Transducer Analog Interface
TEDS	Transducer Electronic Data Sheet
TIM	Transducer Interface Module
TMI	Transducer Measurement interface
TTP	Time triggered protocol
UART	Universal asynchronous receiver/transmitter
USB	Universal Serial Bus
VHDL	VHSIC hardware description language

INTRODUCTION

De nos jours, les aéronefs en service exploitent un nombre important de capteurs et d'actuateurs afin d'accomplir leurs missions d'une manière fiable et sûre. Le besoin des prochaines générations d'aéronefs en termes de nouvelles fonctionnalités incite les avionneurs à incorporer davantage ce type de composant. La connexion entre les différents types de transducteurs et les systèmes à bord d'un aéronef est traditionnellement de type point-à-point. Ceci demande des câbles volumineux et fait croitre la masse totale de l'avion.

Les récentes avancées technologiques ont permis de développer une variété de bus et de réseaux avioniques permettant de multiplexer les données provenant de plusieurs transducteurs. Toutefois, la diversité dans les marchés des bus et des transducteurs avioniques transforme le développement des interfaces de transducteurs en un processus assez couteux et laborieux. Par ailleurs, l'émergence de l'avionique modulaire intégrée IMA 2G favorise l'adoption de nouvelles générations d'interfaces de transducteurs qui sont dotés d'une intelligence et qui sont capables d'exécuter des fonctions avioniques complexes.

Les défis susmentionnés mettent en relief le besoin d'une approche systématique et normalisée permettant de concevoir des interfaces de transducteurs intelligents, indépendamment du réseau avionique et de la technologie de transducteurs sélectionnés. Une des solutions les plus viables consiste à appliquer la norme IEEE1451.0 pour la conception d'interfaces de transducteurs intelligents pour des applications avioniques.

L'objectif principal du projet AVIO 402 financé par le consortium CRIAQ, Bombardier Aéronautique et Thales Canada, est de proposer et développer un réseau de capteurs et actuateurs intelligents pour des applications avioniques critiques. L'architecture du réseau proposée dans le cadre de ce projet est basée sur un réseau principal AFDX. Des passerelles permettent de connecter plusieurs réseaux secondaires mis en œuvre sous la forme de bus de terrain ARINC 825 au réseau AFDX. Plusieurs capteurs ou actuateurs sont connectés au réseau ARINC 825 via des interfaces électroniques. L'architecture des passerelles réseau ainsi que celle des interfaces de transducteur est basée sur la norme IEEE1451.0. Le réseau proposé permet d'acheminer les données des transducteurs vers les calculateurs centraux à

bord de l'avion. Afin de démontrer les performances du réseau proposé, un prototype a été mis en ouvre.

Les travaux menés dans le cadre de ce mémoire visent à concevoir, implémenter et valider une interface de transducteurs intelligents pour des applications avioniques critiques, et ce afin de supporter le développement du prototype du réseau susmentionné. Une plateforme à base de FPGA (field programmable gate array) est ciblée pour le développement étant donné sa flexibilité et sa portabilité.

Initialement, la famille des normes IEEE 1451 n'a pas été définie pour des applications critiques. Étant donné que c'est la classe d'applications cible de ce mémoire, il a fallu étudier les différents aspects de tolérance aux pannes liées à l'implémentation de la norme IEEE1451.0. La première étape du processus de développement de l'interface consiste à définir et collecter les différentes spécifications et contraintes (bande passante, latence, fiabilité…). Par la suite, l'architecture d'une interface de transducteurs intelligents a été proposée. Celle-ci a été conçue pour être aussi générique que possible afin qu'elle puisse supporter plusieurs types de transducteurs. L'interface inclut des techniques de tolérance aux pannes favorisant un comportement sûr et un mode de fonctionnement correct mais avec une performance dégradée en présence de pannes. La fiabilité et la sûreté de l'interface proposée sont évaluées et comparées avec les performances d'autres types d'interfaces. L'implémentation de l'interface a été effectuée sur la plateforme FPGA LX45T de Xilinx. Des simulations sur le logiciel ISIM de Xilinx ont permis de vérifier et de valider le comportement du module implémenté. L'interface implémentée a été intégrée dans le prototype d'un encapsuleur de réseau. Celui-ci est formé par une passerelle ARINC825, une interface de transducteur et des capteurs de température COTS (commercial off the shelf). La passerelle ARINC825, elle aussi implémentée sur FPGA, permet d'acheminer les données provenant des capteurs vers le réseau principal AFDX. La communication entre l'interface de transducteurs el la passerelle est assurée via un bus ARINC825 double. Le prototype a servi pour valider certaines spécifications telles que la latence des messages et la tolérance aux pannes. Les mécanismes de tolérances aux pannes implémentés sont validés via la technique d'injection de pannes.

Le prototype conçu constitue une preuve de concept de la technologie des transducteurs intelligents et pourrait être utilisé pour valider une variété d'algorithmes et de fonctionnalités liés à l'utilisation de cette technologie dans des applications avioniques critiques.

En conclusion, ce mémoire fait l'objet des contributions suivantes:

- la conception d'une interface de transducteurs intelligents tolérante aux pannes favorisant des défaillances sûres et un mode de fonctionnement correct mais à performance dégradée;
- l'évaluation des performances de l'interface proposée est faite en termes de sûreté et fiabilité via les chaines de Markov et une analyse FMEA ;
- l'implémentation de l'architecture proposée sur une plateforme FPGA et son intégration dans un prototype d'encapsuleur de réseau ;
- l'utilisation du prototype de l'encapsuleur développé pour valider certaines caractéristiques de l'interface (Ex: latence et caractéristique tolérance aux pannes).

Le premier chapitre de ce mémoire présente une définition de la technologie des transducteurs intelligents. Les technologies de transducteurs utilisées dans les systèmes de contrôle de vol électriques existants ainsi que celles des prochaines générations y sont détaillées. Le deuxième chapitre passe en revue les caractéristiques de la norme IEEE 1451.0. De plus, l'architecture du réseau de transducteurs proposée et celle de l'encapsuleur du réseau sont exposées. Dans le troisième chapitre, une étude des différentes techniques permettant la conception d'un prototype basé sur un FPGA qui est tolérant aux pannes y est réalisée. L'architecture d'une interface de transducteurs intelligents tolérante aux pannes basée sur FPGA est également présentée. Le quatrième chapitre traite une évaluation des performances de l'interface proposée en termes de fiabilité et de sûreté. Enfin, le cinquième chapitre détaille les différents aspects liés à l'implémentation de l'interface sur une plateforme FPGA Spartan SP605 de Xilinx. Les résultats de validation des mécanismes de tolérance aux pannes ainsi que des caractéristiques temps réel de l'interface y sont également exposés.

CHAPITRE 1

TECHNOLOGIES DE TRANSDUCTEURS INTELLIGENTS POUR LES APPLICATIONS AVIONIQUES CRITIQUES

1.1 Introduction

Les transducteurs intelligents constituent une technologie prometteuse pour les futures générations d'aéronefs. L'intelligence fournie par ce type d'équipement réside dans des unités de calcul et de traitement présentes sous la forme de circuits intégrés tels que des microcontrôleurs, des processeurs de traitement numérique de signaux, des ASIC ou des FPGA. Ces unités de traitement constituent des interfaces qui relient d'une part, les capteurs et les actuateurs et d'autre part, les fonctions de contrôle de l'avion hébergées dans des calculateurs, par l'intermédiaire de réseaux avioniques numériques. Dans ce chapitre, une définition rigoureuse des transducteurs intelligents est présentée. Par ailleurs, le chapitre expose l'évolution de la technologie des transducteurs pour les applications avioniques tout en mettant l'accent sur leur utilisation dans les systèmes de commande de vol électriques.

1.2 Transducteurs intelligents

1.2.1 Définition d'un transducteur

Un transducteur est un dispositif qui convertit l'énergie d'un domaine à un autre. Les types d'énergie incluent notamment l'énergie électrique, mécanique, électromagnétique, chimique, acoustique et thermique. Les transducteurs sont subdivisés en deux catégories : les capteurs et les actuateurs. Un capteur est un transducteur qui convertit un paramètre physique, biologique ou chimique en un signal électrique [1, p.2]. À titre d'exemple, un capteur de pression d'un "Air Data Computer" installé au bord d'un avion mesure la pression, qui se manifeste sous la forme d'une énergie mécanique, et la convertit en un signal électrique. Un actuateur est un transducteur qui accepte un signal électrique et le convertit en une action physique. Dans un avion équipé d'un système de commande de vol électrique, l'ordinateur de

bord envoie une commande sous forme d'un signal électrique vers les vérins qui, à leur tour, actionnent les gouvernes pour contrôler l'avion.

1.2.2 Description d'un transducteur intelligent

Au début des années 80, la notion de capteur se limitait à la conversion de l'énergie et à l'échange de données sous un format analogique. Elle a depuis connu une évolution importante dans plusieurs domaines de l'industrie [2]. En effet, la notion de transducteur intelligent a été introduite à cette époque en dotant les éléments de capteurs et d'actuateurs d'une capacité de traitement ainsi que d'une interface de communication pour permettre la transmission de données numériques sur un bus informatique. L'unité de calcul sert principalement à apporter une correction et à fournir des mesures plus précises. Par ailleurs, le progrès dans le domaine de la microélectronique a permis d'augmenter significativement la performance et la puissance de calcul des puces électroniques. Motivé par ces progrès, l'Institut des ingénieurs électriciens et électroniciens (IEEE) qualifie par intelligent, dans la norme IEEE 1451.0 [24], «tout transducteur qui fournit des fonctions allant au-delà de celles nécessaires pour générer une représentation correcte d'une quantité mesurée ou contrôlée. Ces fonctionnalités simplifient typiquement l'intégration du capteur dans une application appartenant à un environnement réseau ». D'après cette définition, les fonctions fournies par la nouvelle génération de transducteur intelligent ne se limitent pas à la correction des mesures. La valeur ajoutée d'un transducteur intelligent servira à décentraliser l'intelligence d'un système industriel et à permettre au transducteur de participer à la prise de décision dans un contexte de calcul distribué [2] [3, p.22]. Comme le montre la Figure 1.1, un transducteur intelligent est composé de: un (ou plusieurs) capteurs (ou actionneurs), une unité de calcul, une mémoire locale, une unité de conditionnement et d'amplification, un convertisseur numérique-analogique (CNA) ou analogique-numérique (CAN) et une interface de communication. L'unité de calcul est l'unité responsable du traitement des échantillons mesurés. Celle-ci pourrait être un microcontrôleur, un microprocesseur, un DSP, un FPGA ou un ASIC. Dans le cas où un microcontrôleur/microprocesseur est utilisé, la mémoire sert à

stocker notamment des programmes et leurs paramètres de configuration. Sinon, elle pourrait être utilisée pour sauvegarder des données utiles pour le fonctionnement du transducteur.

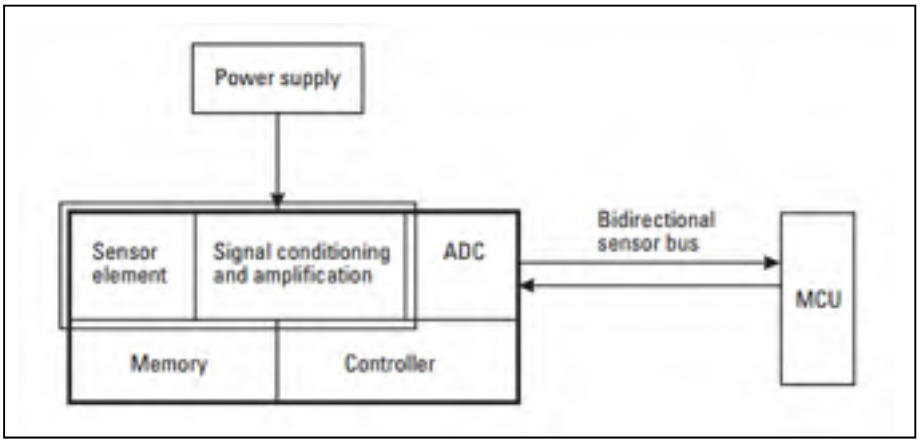

Figure 1.1 Modèle d'un transducteur intelligent.
Adaptée de Frank (2000, p.7)

Le convertisseur CAN permet de transformer la valeur analogique mesurée (par le capteur) en une valeur numérique codée sur plusieurs bits et proportionnelle à la valeur analogique. Le convertisseur CNA permet de transformer une valeur numérique (codée sur plusieurs bits) en une tension électrique proportionnelle à la valeur numérique codée. L'unité de conditionnement sert principalement à amplifier et/ou filtrer le signal venant de ou allant au transducteur. Enfin, l'interface de communication assure l'échange des données sur un bus. Le partitionnement et l'intégration du transducteur pourraient changer d'une application à l'autre. La Figure 1.2 montre quelques possibilités de partitionnement et d'intégration d'un capteur. Dans le partitionnement A, un transducteur intelligent est composé de deux entités séparées. Chaque entité est un circuit imprimé doté de composantes électroniques et/ou de circuits intégrés qui implémentent les fonctions fournies par le module en question. Le module (1) du partitionnement A comporte un ou plusieurs capteurs et est responsable de l'acquisition des mesures et du conditionnement du signal. Le module (2) est responsable de la conversion CAN, du traitement des échantillons et de la transmission des données sur un bus numérique. Dans le partitionnement B, les fonctions de conditionnement, de conversion CAN et de traitement des échantillons sont concentrées en un seul circuit. Selon ce scénario,

le capteur est un composant analogique distant (1) connecté au circuit (2) via un bus électrique. Enfin, le partitionnement C représente un transducteur intelligent où tous les composants (capteur, CAN, interface numérique, circuit de conditionnement du signal et microcontrôleur) sont intégrés sur un seul circuit imprimé.

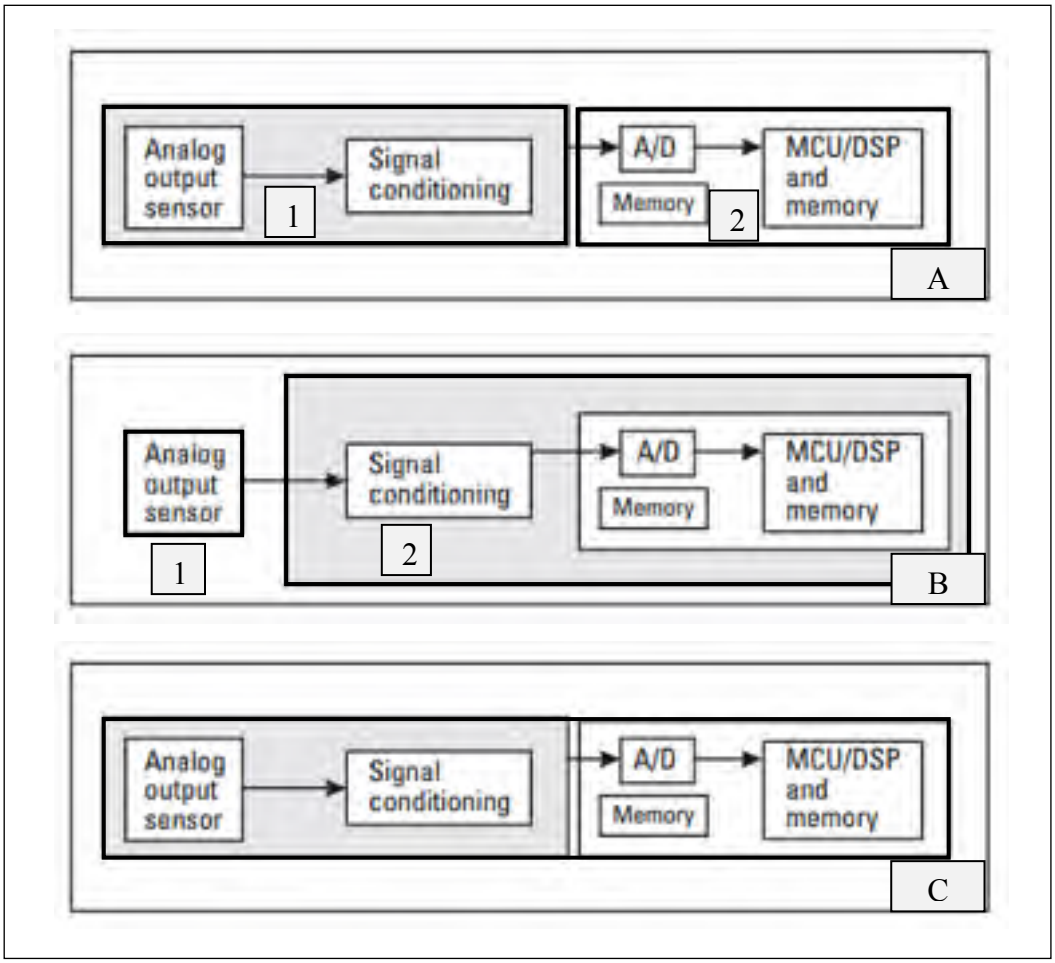

Figure 1.2 Possibilités de partitionnement et d'intégration
d'un transducteur intelligent
Adaptée de Frank (2000, p.9)

La Figure 1.3 montre à quoi ressemble un capteur de type C. Il s'agit d'un "Air Data Module" de Honeywell. Le module contient un capteur de pression utilisé dans les aéronefs modernes. Cependant, le circuit présenté n'inclut pas une unité de calcul. Le module communique les données mesurées via un bus ARINC 429.

Figure 1.3 Air Data Module de Honeywell
Tirée de Honeywell (2004, p.1)

Aujourd'hui, des capteurs intelligents de type "Commercial off the shelf" COTS intégrés dans un seul circuit intégré sont disponibles sur le marché [5]. Ces capteurs, basées sur les technologies microélectromécaniques (MEMS) et microélectroniques, permettent d'intégrer dans le même boîtier électronique tous les éléments nécessaires au fonctionnement d'un transducteur intelligent. La Figure 1.4 présente un capteur de mouvement basé sur la technologie MEMS. Comme le montre la Figure 1.5, le capteur inclut un gyroscope à 3-axes, un accéléromètre à 3-axes, des CAN, un circuit de conditionnement du signal, une interface de communication I2C ainsi qu'une unité de calcul (Digital Motion Processor) qui sert à exécuter les algorithmes de traitement du mouvement.

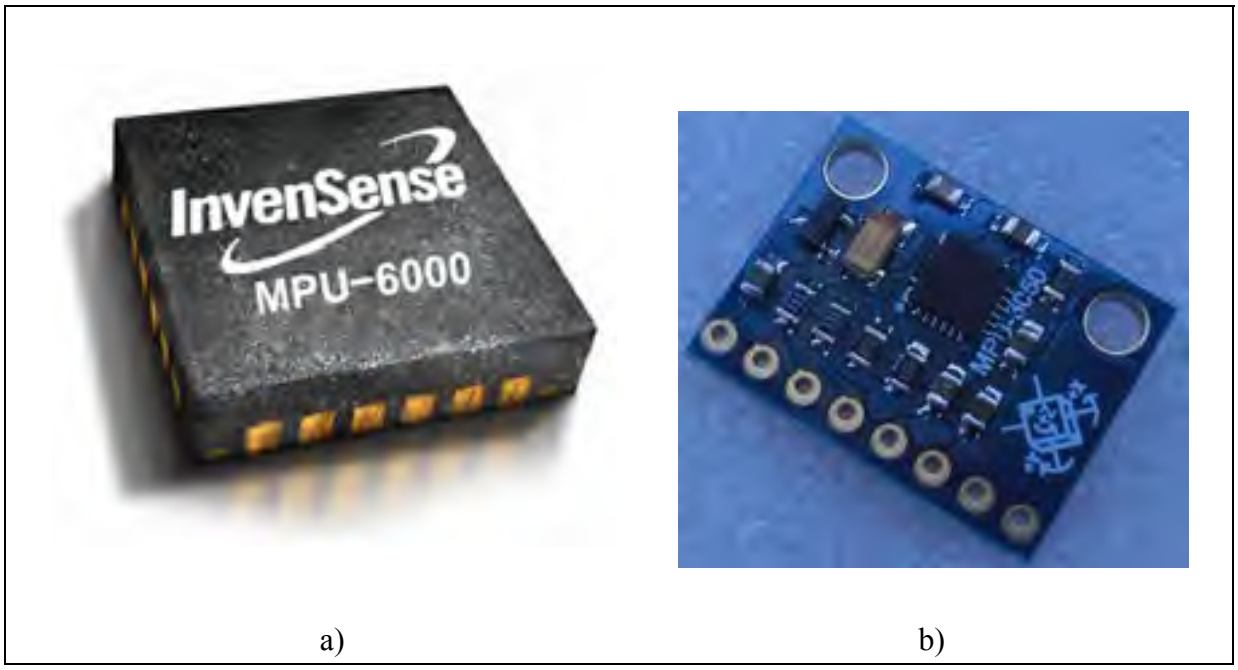

Figure 1.4 a) Capteur de mouvement InvenSense basé sur la technologie MEMS,
b) Capteur de mouvement InvenSense intégré sur un circuit imprimé (PCB),
Tirée de InvenSense (2004, p.2)

Ce niveau élevé d'intégration permet de sauver espace, poids et énergie. Cependant, pour des applications spécifiques comme l'avionique, ces solutions devraient être conçues sur mesure pour fournir certaines fonctionnalités et répondre à des exigences strictes en matière de fiabilité et de sûreté. Ceci pourra augmenter le coût de développement de ce type de transducteurs. Afin de remédier à ces problèmes, ces nouvelles générations de transducteurs COTS pourront être couplées à des unités de calculs qui implémentent les fonctions exigées par ce type d'applications.

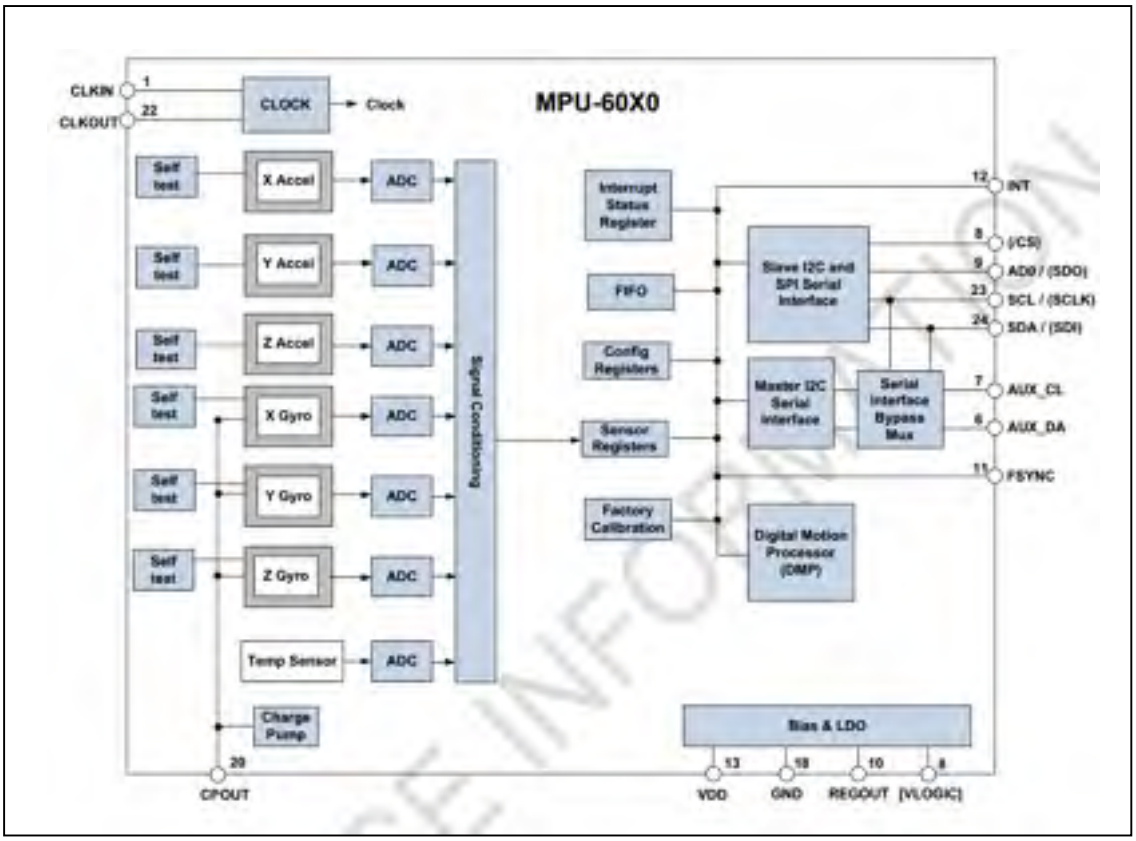

Figure 1.5 Diagramme de bloc du capteur de mouvement InvenSense MPU-60X0
Tiré de InvenSense (2010, p.22)

1.3 Technologies de transducteurs existantes utilisées dans le domaine de l'avionique

1.3.1 Médiums pour l'interconnexion des transducteurs

Les premières générations d'aéronefs modernes ont connu l'introduction de calculateurs centraux, de sous-systèmes et de transducteurs, tous analogiques, afin d'améliorer les performances de l'avion. La communication entre les différents calculateurs et transducteurs était assurée via des liens point-à-point analogiques. Au fur et à mesure de l'évolution de l'industrie aéronautique, la complexité des systèmes avioniques et le besoin en termes de puissance de calcul ont augmenté de manière remarquable. Afin de satisfaire à ces besoins, des calculateurs numériques ont été développés pour remplacer les calculateurs analogiques

anciennement utilisés. Par ailleurs, les liens analogiques point-à-point ont été conservés pour interconnecter les capteurs et les actuateurs avec des calculateurs centraux numériques. De plus, des convertisseurs CAN et CNA ont été implémentés du côté des calculateurs pour les interfacer avec les transducteurs [7]. Les premiers progrès dans le domaine des bus avioniques ont permis de remplacer une partie des liens analogiques par des liens numériques point-à-point en dotant les transducteurs d'une interface de communication numérique. Le bus avionique point-à-point le plus répandu est l'ARINC 429. Celui-ci est un bus de données série unidirectionnel (simplex). La norme ARINC 429 impose qu'il n'y ait qu'un seul émetteur par bus. Le nombre de récepteurs peut, quant à lui, aller jusqu'à 203. Cependant, comme la complexité des systèmes à bord de l'avion augmentait, le nombre de transducteurs nécessaires augmentait de même. Ceci dit, l'architecture basée sur des interconnexions point-à-point est devenue inefficace. Afin de remédier à ce problème, de nouveaux bus avioniques qui permettent un multiplexage numérique et le partage des bus ont été introduits [8]. Plusieurs transducteurs ont été dotés d'interfaces de communication qui supportent ce type de bus avioniques. Le MIL-STD-1553B est l'un des bus multi-émetteur les plus utilisés [9]. Depuis près d'une décennie, l'industrie aérospatiale a introduit un nouveau protocole de communication avionique: AFDX. Celui-ci est basé sur le protocole commercial Ethernet commuté. Un réseau AFDX est basé sur une topologie en étoile où la communication entre les différents systèmes est assurée via des équipements électroniques intermédiaires appelés commutateurs. Dans [3, p.133] [13], les auteurs proposent des transducteurs dotés de "End-System" afin de les interconnecter aux systèmes de commande de vol électriques via le protocole avionique AFDX.

1.3.2 Technologie de transducteurs dans les systèmes de commande de vol électrique existants

1.3.2.1 Architecture du système CDVE d'Airbus

Le système de commande de vol électrique (CDVE) a été introduit par Airbus pour la première fois sur la famille d'avion A320 [15]. Ce système est constitué de deux types de

calculateurs numériques ("Flight Control Computer", FCC) auto-vérifiants: des calculateurs primaires (trois calculateurs) et des calculateurs secondaires (trois calculateurs). Chaque calculateur est composé de deux voies matérielles séparées (*Voir* Figure 1.6): La voie COM (commande) et la voie MON (moniteur). Les deux voies jouent des rôles différents: la voie COM calcule les consignes de commande des actionneurs et les transmet vers la voie MON pour validation. La voie MON calcule les mêmes consignes que la voie COM, compare les commandes des deux voies et signale l'existence d'une erreur aux différents systèmes de l'avion en cas de différence entre les commandes calculées. Si une panne est détectée, le calculateur fautif sera remplacé par un autre calculateur fonctionnel.

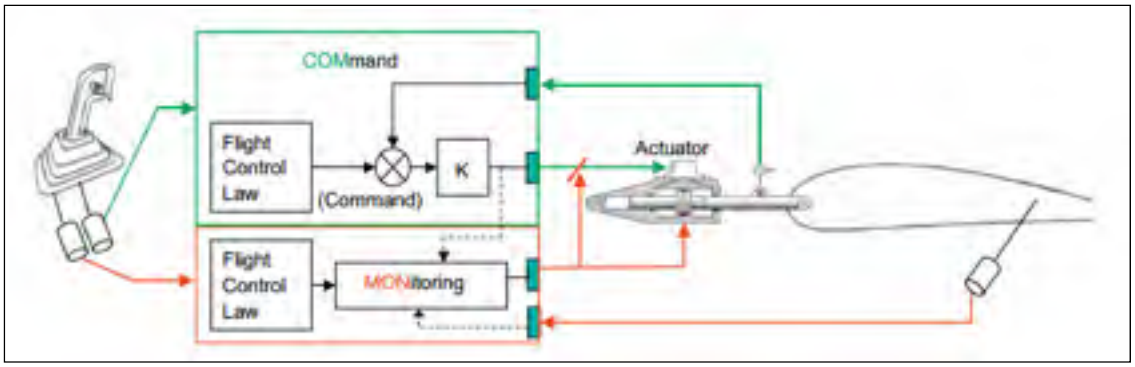

Figure 1.6 Diagramme de bloc simplifié du système CDVE d'Airbus
Tirée de Goupil (2011, p.6)

Les calculateurs constituent l'élément central du système CDVE des avions Airbus modernes. Les calculateurs traitent les consignes des pilotes et calculent des ordres pour les actionneurs. De plus, ils sont responsables de la surveillance et de la détection des pannes des transducteurs. Des liens analogiques point-à-point entre les capteurs du cockpit et les calculateurs assurent le transfert des consignes du pilote. De manière semblable aux capteurs, les actionneurs sont pilotés et asservis par les calculateurs via des liaisons directes analogiques. Une fois les consignes du pilote reçues par les FCC, ceux-ci convertissent les signaux reçus en format numérique et procèdent à l'actionnement des gouvernes. Les résultats de traitement sont acheminés vers les actionneurs soit en format analogique ou numérique. Bien que ces liens directs constituent des canaux de communication complètement indépendants et qu'ils permettent des niveaux de fiabilité et de sûreté assez

élevés, ce type de liaisons entraine un encombrement du câblage et une augmentation considérable de la complexité et du poids du système CDVE.

1.3.2.2 Architecture du système CDVE de Boeing

Le Boeing 777 est le premier avion de Boeing qui a été doté d'un système de commande de vol électrique. Le système de commande de vol électrique du Boeing 777 est subdivisé en deux niveaux [16] :

- trois calculateurs numériques centraux PFC (Primary Flight Control) pour le calcul des commandes;
- quatre calculateurs analogiques ACE (Actuator Control Electronics) pour l'asservissement des actionneurs et la conversion CNA et CAN.

Les PFC constituent les éléments centraux du système de commande de vol du Boeing 777. Chaque PFC est composé de trois voies indépendantes configurées comme suit : la voie "command", la voie "monitor" et la voie "standby". La voie "command" de chaque PFC élabore les consignes de commande, tandis que les deux autres voies surveillent son fonctionnement. En cas de panne, une des deux voies supplémentaires remplace la voie fautive. Les trois PFC échangent les consignes générées par leurs voies "command" et un mécanisme de vote permet par la suite de sélectionner la bonne commande pour la transmettre finalement sur le bus ARINC 629 vers les ACE. Les PFC sont également responsables de la surveillance des transducteurs du système CDVE ainsi que la détection de leurs pannes.

Électronique des actionneurs

Les ACE sont des calculateurs analogiques responsables de l'asservissement des actionneurs. Les ACE jouent aussi le rôle d'interface entre les organes de pilotage, les actionneurs et les PFC. Les capteurs du cockpit mesurent les intentions des pilotes et les envoient sur des liens analogiques point-à-point vers les ACE. Ces derniers convertissent les signaux de commandes analogiques en une forme numérique et les acheminent vers les PFC via un bus

ARINC 629 redondant. Les PFC utilisent ces mesures pour calculer les commandes de contrôle des surfaces qui seront par la suite transmises aux ACE sur le bus ARINC 629. Les ACE reçoivent les commandes numériques du PFC et réalisent l'asservissement des actuateurs via des liens analogiques point-à-point. La Figure 1.7 montre le diagramme de bloc d'un ACE.

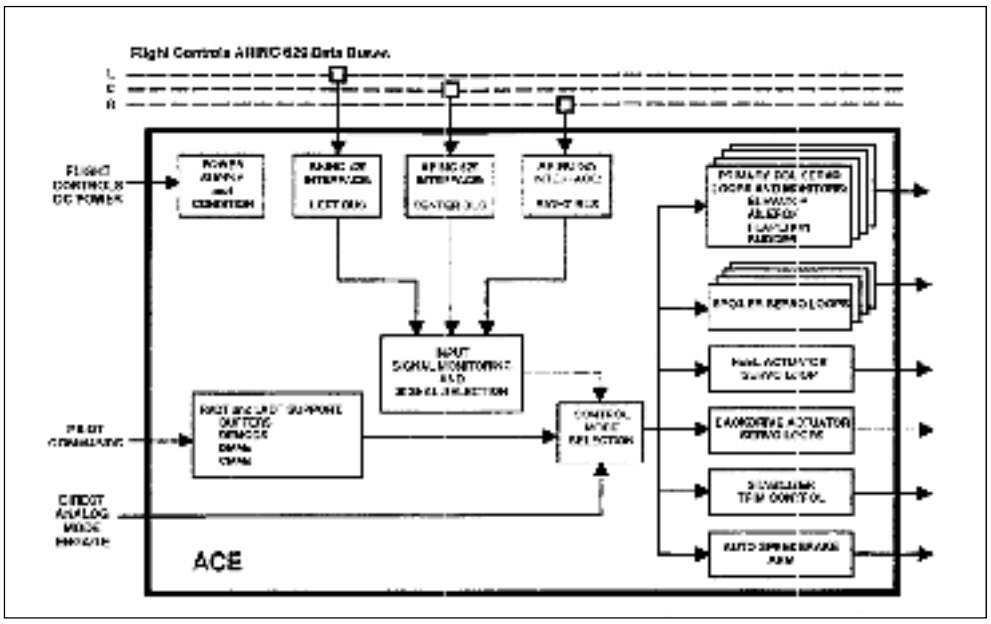

Figure 1.7 Diagramme de bloc d'un ACE
Tirée de Yeh (1996, p.5)

En cas de panne des PFC, les ACE contrôlent les actionneurs sans passer par les PFC en recevant les commandes des organes de pilotage. La Figure 1.8 montre le diagramme de bloc du système CDVE avec des ACE. L'introduction des ACE constitue une solution pour réduire la quantité de câblage nécessaire à la mise en place du système CDVE de Boeing. En effet, l'utilisation du bus ARINC629 a permis d'éliminer les connexions point-à-point reliant les PFC aux différents transducteurs et a facilité l'intégration de nouveaux sous-systèmes et fonctionnalités au système CDVE sans avoir recours à des liens dédiés aux PFC. Cependant, le fait que les ACE soient partagés entre les capteurs du cockpit et les actionneurs ne permet pas de réduire significativement la quantité de câblage surtout dans le cas des gros avions [17]. En effet, les liens point-à-point entre les transducteurs et les calculateurs PFC, ont été

simplement remplacés par de nouvelles connexions point-à-point avec les ACE (*Voir* Figure I.8).

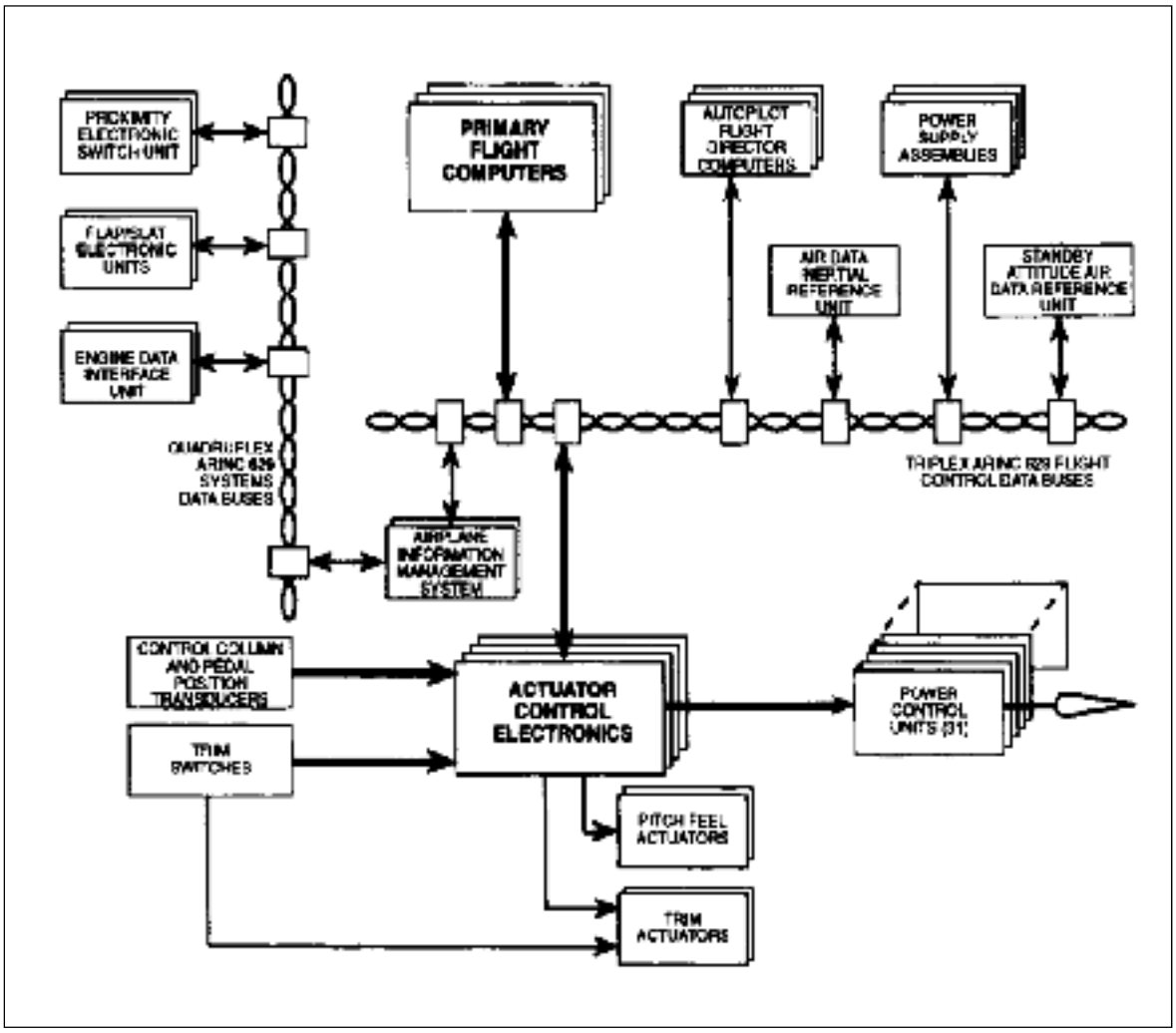

Figure 1.8 Diagramme de bloc d'un système CDVE avec des ACE
Tirée de Godo (2002, p.2)

1.4 Revue de la littérature des transducteurs intelligents pour les prochaines générations des systèmes CDVE

1.4.1 Système de commande de vol avec électronique déportée

Les auteurs de [17] proposent une nouvelle architecture d'un système CDVE basée sur celui de Boeing. L'architecture proposée vise à contourner les problèmes causés par la grande quantité de câblage utilisée pour interconnecter les actuateurs de contrôle aux ACE. En effet, dans un système CDVE d'un Boeing 777, 15 à 19 câbles sont nécessaires pour brancher chaque actuateur à un ACE. L'ensemble des câbles utilisés pour interconnecter tous les actionneurs constitue un poids énorme. Par ailleurs, l'article évoque la nécessité de fournir une bande passante élevée pour contrôler les grandes surfaces dans les gros porteurs. Pour remédier à ces problèmes, l'auteur de l'article propose une nouvelle architecture où les ACE sont remplacés par des modules électroniques numériques RAE (Remote Actuator Electronics) placés tout près des actuateurs pour assurer l'asservissement des actuateurs. Les nouveaux modules permettent de faire passer de 19 à 6 le nombre de câbles connectés à un actuateur. La communication entre un RAE et un PFC est assurée via 4 bus ARINC 429. Deux bus ARINC 429 sont utilisés pour la transmission et deux autres pour la réception des données. Deux câbles d'alimentation permettent d'assurer les besoins des RAE en matière d'électricité. Tout comme dans le système CDVE du Boeing 777, les PFC sont les éléments centraux du système qui monopolisent les prises de décisions ainsi que l'exécution des algorithmes de contrôle complexes. Les RAE communiquent aux calculateurs centraux les données de rétroaction, leurs propres pannes ainsi que les pannes simples qui peuvent survenir dans les actuateurs. Du côté du cockpit, les capteurs RVDT et LVDT mesurant des intentions des pilotes sont connectés aux calculateurs PFC via des liaisons analogiques point-à-point. La conversion des signaux analogiques transmis par les capteurs est effectuée dans les PFC. Un mécanisme similaire à celui existant dans les ACE, est implémenté dans les PFC pour permettre le pilotage des surfaces de l'avion directement à partir des consignes du pilote en cas de perte des trois calculateurs PFC. La Figure 1.9 montre le diagramme de bloc simplifié d'un RAE.

L'introduction des RAE permet de sauver du poids en remplaçant les liens analogiques entre les ACE et les actuateurs par d'autres liens numériques de type ARINC 429 reliant les PFC aux actuateurs. Par rapport au système CDVE du Boeing 777, le nombre de câbles par chaque connexion actuateur-PFC est considérablement réduit (réduction de 60% par rapport à la quantité de câblage utilisée pour connecter un actuateur dans le système CDVE du Boeing 777). La nouvelle architecture permet aussi d'éliminer complètement les ACE et de remplacer les interfaces ARINC 629 coûteuses par des interfaces ARINC 429 moins coûteuses. Les différentes améliorations introduites par cette architecture permettent de sauver plus de 400kg.

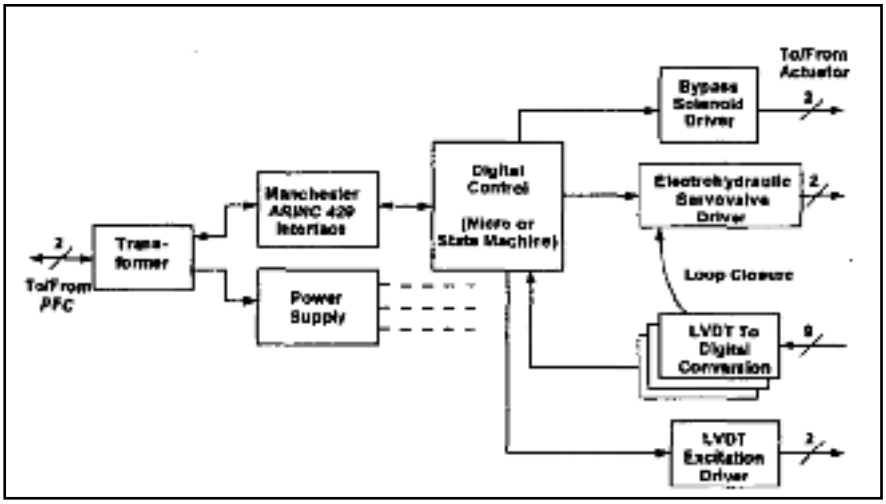

Figure 1.9 Diagramme de bloc simplifié du RAE
Tirée de Godo (2002, p.6)

Bien que les différents liens entre les PFC/ACE et les actuateurs aient été mise à niveau par l'introduction du bus numérique ARINC 429, l'architecture proposée a conservé la nature directe et rigide de ces liens. De plus, les capteurs de position de cockpit sont, eux aussi, connectés aux PFC par le biais de liaisons directes analogiques. Dans cette architecture le poids des liaisons point-à-point demeure considérable. De plus, les RAE demeurent des organes passifs dont le rôle se limite à l'exécution des commandes des calculateurs centraux et à des fonctions de supervision simples.

1.4.2 Système de commande de vol partiellement distribué

Dans [3], les auteurs proposent deux nouvelles architectures d'un système CDVE. La motivation derrière ce travail est de proposer des solutions qui permettent de transformer les architectures centralisées actuelles en architectures distribuées futuristes. Bien que les systèmes CDVE centralisés soient caractérisés par des niveaux de fiabilité et de sûreté assez satisfaisants, ceux-ci souffrent d'un certain nombre d'inconvénients. En effet, ils sont rigides, complexes et coûteux. Les architectures proposées dans ce travail répartissent l'intelligence et le traitement sur l'ensemble des éléments du système CDVE ce qui permet de concevoir des systèmes moins complexes, plus souples et moins coûteux.

Les architectures proposées reposent sur l'architecture CDVE d'Airbus et sont basées sur les principes suivants :

1) utiliser des capteurs et des actionneurs intelligents qui participent à la prise de décision et au traitement des données ;
2) remplacer les calculateurs duplex (COM/MON) par des calculateurs simplex ;
3) réduire le nombre de calculateurs ;
4) remplacer les liens directs entre les calculateurs et les actionneurs par deux réseaux numériques déterministes :

- AFDX qui joue le rôle du réseau principal ;
- MIL-STD 1553B qui joue le rôle de bus de terrain.

Architecture à vote massif

Dans cette architecture, tous les calculateurs calculent les ordres de pilotages pour tous les actionneurs. Les calculateurs communiquent leurs commandes via les réseaux avioniques vers tous les actionneurs. Chaque actionneur est doté d'une électronique locale FCRM (Flight Control Remote Module) composée de deux voie : une voie commande (COM) et une voie moniteur (MON). Après avoir reçu les consignes de tous les calculateurs, les deux unités COM et MON de chaque actionneur effectuent séparément un vote sur l'ensemble des ordres

reçus afin de sélectionner l'ordre valide. Si un désaccord existe entre les deux voies COM et MON, l'électronique locale signale aux calculateurs la défaillance de l'actionneur en question. Sinon, les FCRM acquittent aux calculateurs la validité des ordres de chaque calculateur. La Figure 1.10 montre le principe de base de l'architecture à vote massif.

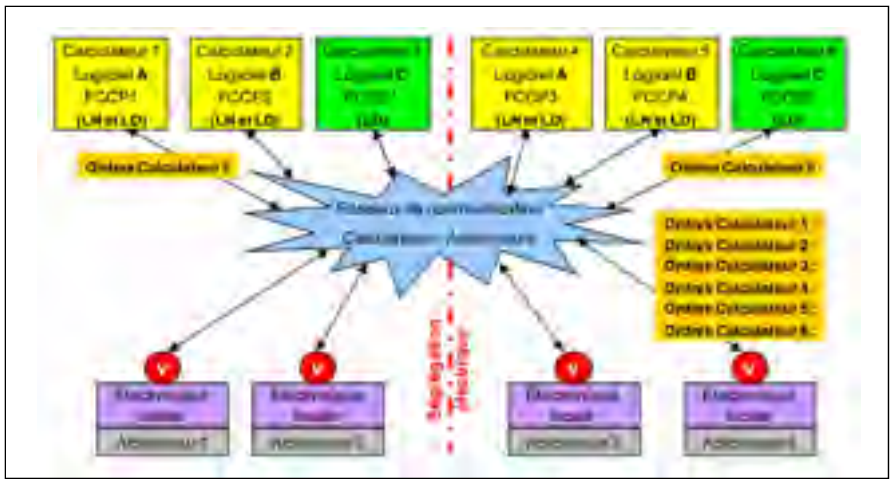

Figure 1.10 Principe de base de l'architecture à vote massif
Tirée de Sghairi (2010, p.56)

Architecture à priorité

L'architecture à priorité diffère de l'architecture à vote massif sur le plan de la prise de décision concernant la validité des commandes générées par les calculateurs. En effet, dans cette architecture, chaque calculateur joue le rôle de maître envers un groupe d'actionneurs et le rôle de valideur envers les autres actionneurs. Tout comme l'architecture à vote massif, l'ensemble des calculateurs calculent des ordres pour toutes les gouvernes. Chaque calculateur transmet ses ordres vers les actionneurs dont il est maître, et vers les autres calculateurs.

Les calculateurs valideurs comparent leurs ordres avec l'ordre du calculateur maître. Par la suite, le résultat de comparaison est transmis par le calculateur valideur aux FCRM concernés. Celles-ci regardent les résultats de validation et transmettent l'ordre vers le

servomoteur s'il a été validé par une majorité des calculateurs valideurs. Dans le cas contraire (ordre non validé), l'actionneur déclare une panne au niveau de son calculateur maître. Les actionneurs s'auto-vérifient grâce à une architecture COM/MON. L'aspect relatif à la tolérance aux pannes dans les FCRM sera discuté en détail au troisième chapitre. Bien que le fonctionnement des actuateurs soit largement détaillé sur le plan architectural et algorithmique, l'auteur de [3] ne détaille pas l'architecture matérielle et logicielle des FCRM et ne traite pas le cas des capteurs intelligents.

1.4.3 Système de commande de vol électrique totalement distribué

Pour remédier aux problèmes causés par la complexité, l'inflexibilité et le coût élevé des systèmes CDVE centralisés, les auteurs de [4] proposent une nouvelle architecture totalement distribuée. Dans cette architecture, le traitement des données/commandes est effectué par des transducteurs intelligents distribués. La communication entre les différents capteurs et actuateurs est assurée par le biais du bus avionique TTP (Time-Triggered Protocol).

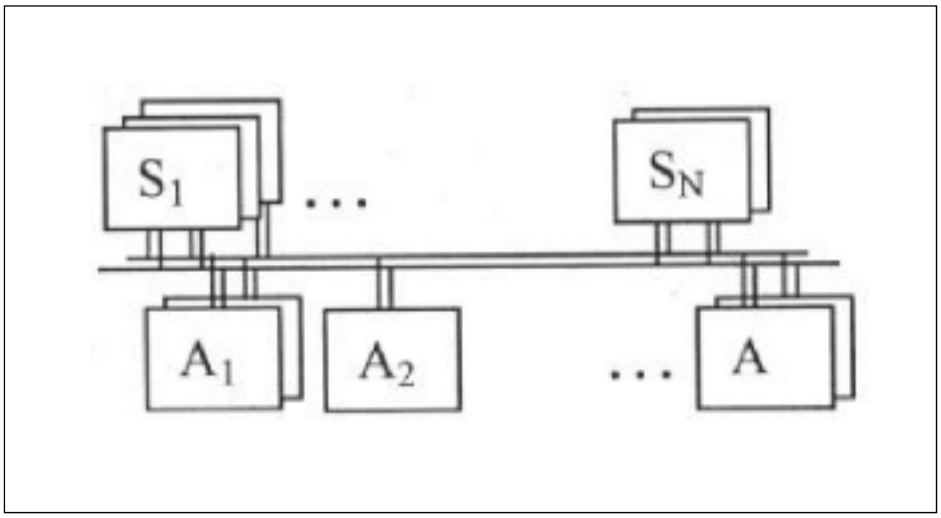

Figure 1.11 Système de commande de vol électrique distribué
Tirée de Forsberg (2003, p.22)

Une des principales raisons pour laquelle l'intelligence a été introduite au niveau des transducteurs est d'implémenter des fonctions de détection de pannes, ce qui permet de concevoir un système CDVE tolérant aux pannes avec un minimum d'unités matérielles. Ainsi, le nœud de transducteur devrait être capable de détecter les pannes des transducteurs ainsi que les pannes de l'unité de calcul. Deux types de nœuds sont proposés: les nœuds capteurs et les nœuds actuateurs. La Figure 1.11 illustre l'architecture du système CDVE distribué.

Nœud capteur

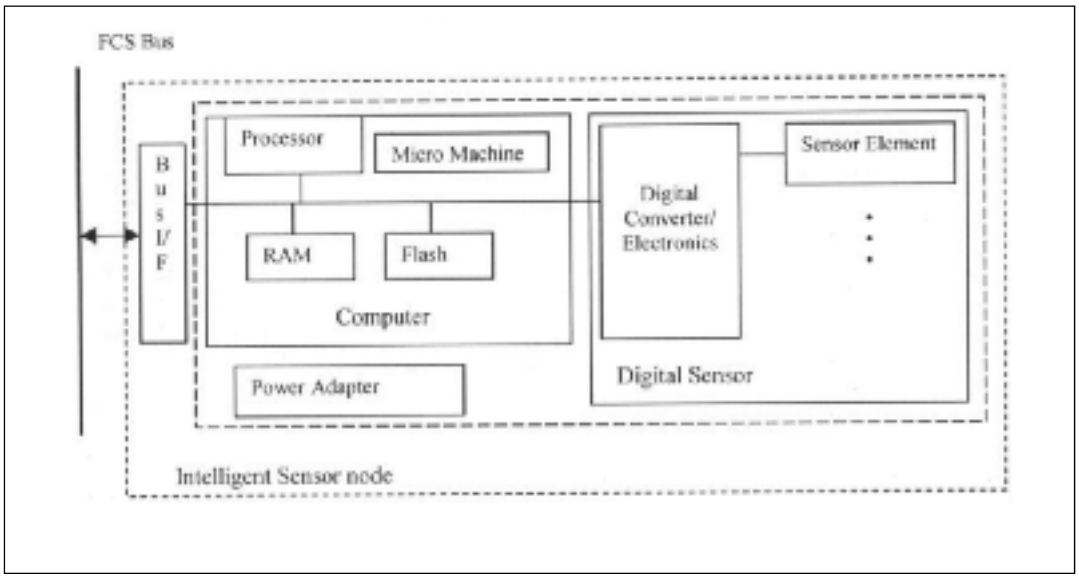

Figure 1.12 Nœud capteur
Tirée de Forsberg (2003, p.50)

Un nœud capteur est composé des éléments suivants (*Voir* Figure 1.12) :

- un capteur numérique qui inclut principalement un ou plusieurs éléments de capteurs et un circuit de conversion analogique numérique ;
- un microcontrôleur qui permet d'effectuer des fonctions de traitement de signal, de filtrer et de corriger les échantillons, de détecter les pannes et de surveiller l'état des capteurs ;
- un adaptateur de puissance pour connecter le nœud aux bus d'alimentation ;

- une interface qui permet de connecter le nœud au bus de communication.

Après avoir mesuré et traité les échantillons, ceux-ci sont transmis vers les différents sous-systèmes à travers le bus. Dans ce travail, l'auteur suppose que les différents capteurs du système CDVE, tels que les capteurs du cockpit mesurant les intentions des pilotes, sont basés sur la technologie MEMS.

Nœud actuateur

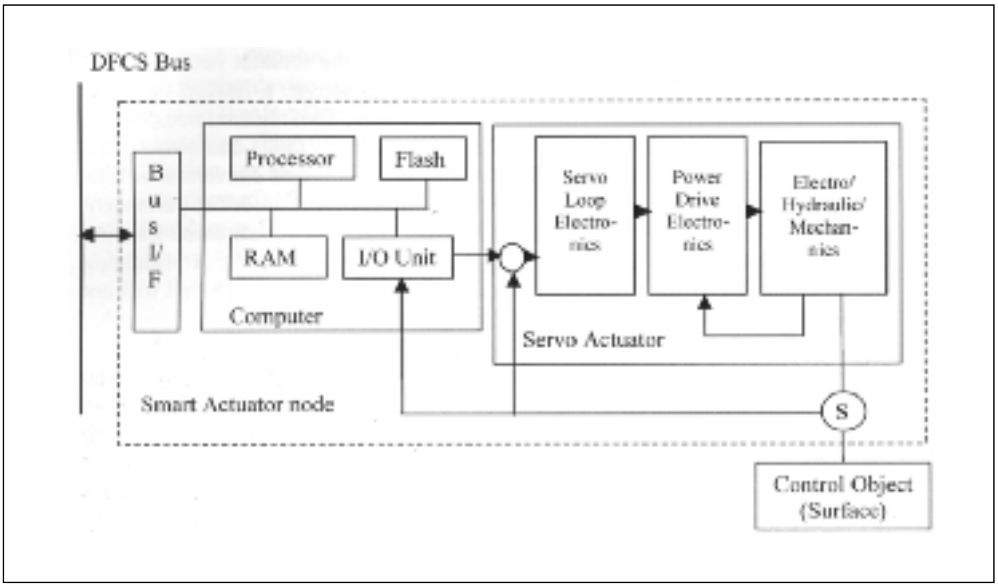

Figure 1.13 Nœud actuateur
Tirée de Forsberg (2003, p.52)

Un nœud actuateur contient les unités suivantes (*Voir* Figure 1.13):

- un servo-actuateur pour l'asservissement des actionneurs ;
- un microcontrôleur pour calculer les commandes, contrôler la gouverne, détecter les pannes, etc. ;
- un adaptateur de puissance ;
- une interface de communication .

Après avoir reçu les échantillons provenant des différents nœuds capteur, chaque nœud actuateur procède au calcul des lois de contrôle pour tous les nœuds actuateurs du système CDVE. Par la suite, les nœuds actuateurs échangent les commandes calculées et chacun d'entre eux effectue un vote sur l'ensemble des résultats pour obtenir la commande valide. La commande sélectionnée sera émise vers le servo-actuateur pour actionner la surface de contrôle.

1.5 Évolution de l'avionique modulaire intégrée

1.5.1 IMA 1G

Avant l'introduction de l'avionique modulaire intégrée (IMA), les différentes fonctions à bord d'un aéronef étaient assurées par des calculateurs dédiés faisant partie d'une architecture fédérée. Le besoin grandissant de fonctions a fait croître le nombre de calculateurs dédiés utilisés à bord d'un avion. Ceci cause plusieurs inconvénients: une masse importante de câblage et de calculateurs, une importante consommation d'énergie, un coût élevé, une gestion complexe, etc.

L'avionique modulaire intégrée (IMA) est apparu, dans les années 1990 pour les avions militaires et les années 2000 pour les avions civils, comme une solution qui consiste à ramener les fonctions, qui auparavant étaient exécutées par les calculateurs dédiés, aux calculateurs modulaires identiques. Les calculateurs de l'IMA sont assez puissants et sont capables de traiter plusieurs fonctions. La première génération d'avionique modulaire intégrée IMA 1G est caractérisée par une centralisation de l'intelligence dans les calculateurs modulaires (modules colorées en jaune et noir dans la Figure 1.14 a)). Par ailleurs, les fonctions critiques de l'avion telles que la commande de vol électrique, demeurent assurées par des calculateurs. Tous les calculateurs des systèmes avioniques basés sur IMA 1G sont installés dans la baie avionique de l'aéronef.

1.5.2 IMA 2G

La deuxième génération de l'avionique modulaire intégrée IMA 2G se base sur le nouveau concept de l'Électronique Modulaire Distribuée ("Distributed Modular Electronics", DME) [19]. En effet, la deuxième génération de l'IMA favorise davantage l'intégration des fonctions dans les calculateurs modulaires. Toutefois, dans l'IMA 2G, l'intelligence est distribuée, à des degrés divers, entre les calculateurs modulaires et des modules électroniques distants tels que des transducteurs intelligents et des concentrateurs de données distants ("Remote Data Concentrator", RDC) (les modules colorées en rouge dans la Figure 1.14 b)). Un des principaux avantages de cette architecture est qu'elle fournit une surveillance décentralisée des différents modules électroniques, ce qui permet théoriquement d'assurer une détection de pannes égale à 100% [18]. Contrairement à l'IMA 1G, les calculateurs de l'IMA 2G sont distribués sur plusieurs zones de l'aéronef (modules colorées en jaune et noir dans la Figure 1.14 b)).

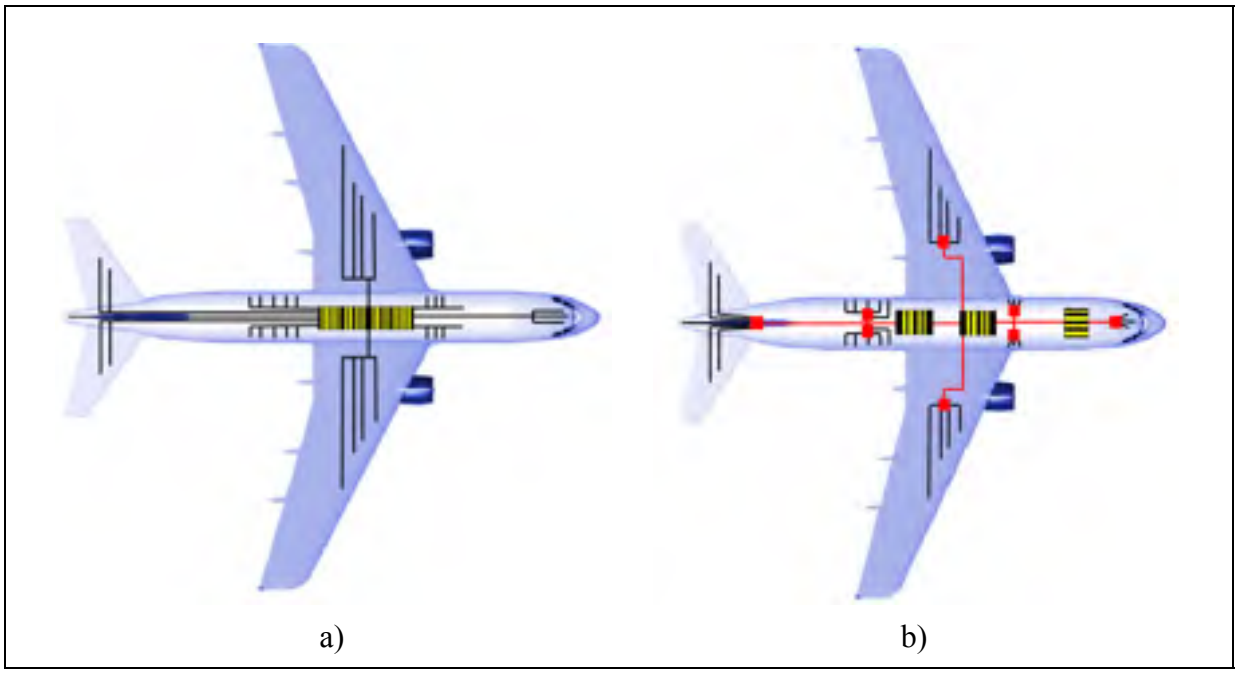

Figure 1.14 a) Architecture de l'IMA 1G, b) Architecture de l'IMA 2G
Tirée de Mats (2003, p.16 et 17)

1.5.3 Projet SCARLETT

Le projet SCARLETT (SCAlable and ReconfigurabLe Electronics plaTforms and Tools) [19] est un projet d'envergure européen qui visait à proposer des architectures et des plateformes avioniques pour la deuxième génération de l'avionique modulaire intégrée IMA 2G. Une trentaine d'industriels et institutions académiques européens ont été impliqués dans le projet et ont proposé un ensemble de produits matériels et logiciels qui pourront faire partie d'une architecture avionique futuriste basée sur IMA 2G.

Travaux et produits découlant du projet SCARLETT

Plusieurs industriels ont proposé des produits qui ont découlé de leurs travaux de recherche dans le cadre du projet SCARLETT.

RDC générique

"GE Aviation Systems" a développé un RDC générique qui permet d'interfacer plusieurs types de transducteurs distants et de les relier aux calculateurs de l'IMA via le bus MIL-STD-1553B (*Voir* Figure 1.15).

Le RDC développé offre les fonctionnalités suivantes :

- filtrer, conditionner et numériser les signaux venant des transducteurs ;
- appliquer un filtrage numérique aux échantillons ;
- transmettre les valeurs obtenues par l'intermédiaire du bus de données ;
- exécuter les commandes reçues sur le bus de données pour contrôler les transducteurs ;
- effectuer un contrôle en boucle fermée localisée ;
- surveiller en permanence l'état des transducteurs ainsi que le câblage associé.

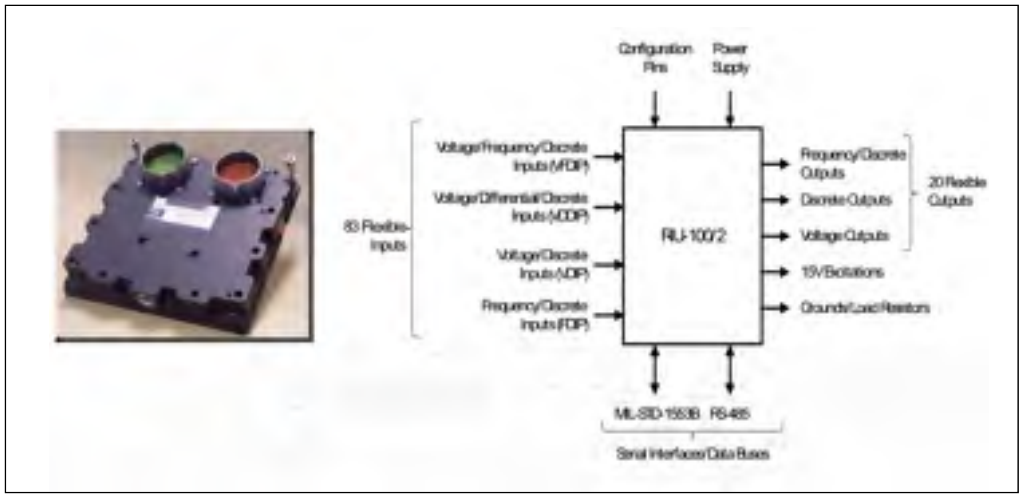

Figure 1.15 Concentrateur de données RDC
Tirée de SCARLETT (2011, p.4)

RDC pour les systèmes de freinage

Dans le cadre du projet SCARLETT, Messier-Bugatti-Dowty ont contribué à la conception d'un système de freinage basé sur des RDC. Dans ce travail, deux architectures ont été proposées. La première architecture (*Voir* Figure 1.16 a) est basée sur un calculateur central faisant partie de l'IMA et d'un RDC intelligent situé près du train d'atterrissage. Le calculateur central est responsable de l'activation et l'élaboration des commandes de freinage. Par ailleurs, le RDC intelligent héberge un système d'exploitation conforme à la norme ARINC 653. Celui-ci exécute les logiciels de freinage, de la boucle de contrôle fermée et d'antiblocage. La deuxième architecture proposée (*Voir* Figure 1.16 b) est basée sur un RDC non intelligent et concentre l'intelligence dans les calculateurs centraux de l'IMA. Dans cette architecture, les RDC sont des éléments passifs n'exécutant pas des applications logicielles et dont le rôle se limite à la génération et le conditionnement des signaux.

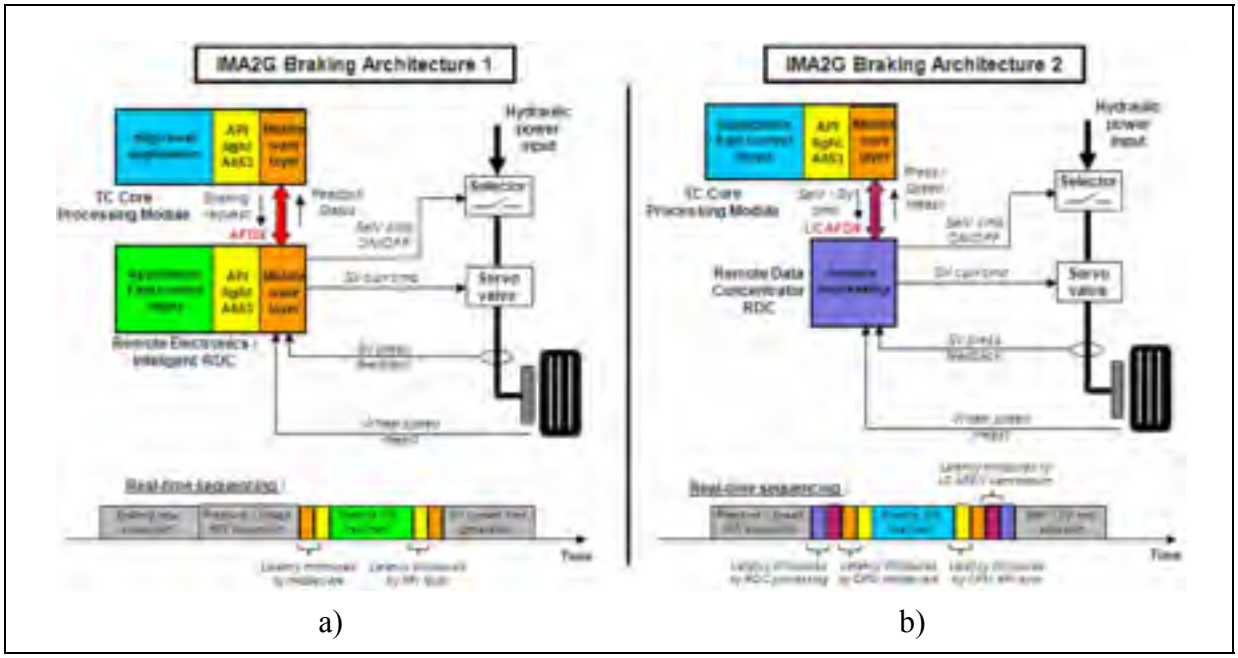

Figure 1.16 a) Système de freinage basé sur un RDC intelligent,
b) Système de freinage basé sur un RDC non intelligent
Tirée de Bernard (2011, p.4)

1.6 Conclusion

Ce chapitre a traité plusieurs aspects des technologies de transducteurs pour les applications avioniques. Le principe et le modèle des transducteurs intelligents ont été exposés. Ensuite, les technologies de transducteurs existantes utilisées dans le domaine aéronautique ont été présentées et une attention particulière a été accordée aux systèmes de commande de vol électriques. Une section entière a été dédiée à la revue de la littérature de la prochaine génération des systèmes de commande de vol électrique basés sur la technologie des transducteurs intelligents. Finalement, l'influence de l'évolution de l'avionique modulaire avionique IMA dans l'introduction de la technologie des transducteurs intelligents a été exposée.

CHAPITRE 2

NORME IEEE 1451.0

2.1 Introduction

Ce chapitre présente les normes qui constituent la famille IEEE 1451 ainsi que leurs champs d'applications. Une attention particulière est dédiée à la présentation de la norme IEEE 1451.0. Les spécifications de l'interface de transducteurs telles que définies par l'IEEE 1451.0 sont également exposées. Finalement, l'architecture du réseau de transducteurs élaborée dans le cadre du projet AVIO 402 est présentée.

2.2 Besoin de normalisation des interfaces de transducteurs intelligents

Dans les avions modernes et de prochaine génération, un nombre croissant de capteurs et d'actuateurs est nécessaire afin de fournir une variété de fonctions de surveillance et de contrôle. Comme il a été présenté au premier chapitre, divers réseaux de communication avioniques ont été adoptés pour soutenir un besoin croissant en bande passante tout en réduisant les faisceaux volumineux de câblage. L'intégration d'un large éventail de capteurs et d'actionneurs dans un environnement de réseaux hétérogènes pose des problèmes sérieux nécessitant la normalisation des interfaces de transducteurs pour les systèmes avioniques. De plus, le besoin en intelligence locale incorporée au niveau des transducteurs vient compliquer la situation. Une des solutions les plus viables permettant la résolution de ces problèmes est l'application des normes IEEE 1451 dans le développement d'interfaces de transducteurs intelligents pour des applications avioniques. Les normes IEEE1451 sont relativement récentes et n'ont pas encore été utilisées dans le développement de systèmes avioniques. Dans [13], les auteurs proposent une architecture d'un réseau de capteurs intelligents pour des applications avioniques critiques. Dans ce travail, les interfaces de capteurs sont basées sur l'IEEE1451 et permettent de connecter des capteurs à un réseau AFDX secondaire nommé Réseau Secondaire de Capteurs ("Ancillary Sensor Network", ASN). Un réseau AFDX principal permet d'acheminer les données du réseau ASN vers les différents systèmes

de l'aéronef. L'auteur de [22] présente l'IEEE 1451 comme une technologie prometteuse qui pourrait être utilisée dans le test et l'évaluation des nouveaux aéronefs. Par ailleurs, l'auteur met en relief l'utilité des normes IEEE 1451 dans le développement d'interfaces de transducteurs intelligents pour les systèmes avioniques complexes. L'auteur signale également qu'une telle technologie ne pourra être introduite dans les aéronefs commerciaux que lorsque sa sûreté sera prouvée.

2.3 Présentation des normes IEEE1451

L'IEEE 1451 [23] est un ensemble de normes développées par l'IEEE, décrivant un ensemble d'interfaces de communication ouvertes, communes et indépendantes des réseaux, pour interconnecter des transducteurs aux microprocesseurs, aux systèmes d'instrumentation et aux réseaux de terrain. Un des éléments clés de ces normes est l'introduction des "Transducer electronic data sheets" (TEDS). Les TEDS sont des dispositifs de mémoire introduits au niveau des interfaces de transducteurs, permettant la mémorisation de données nécessaires au fonctionnement du transducteur telles que les données de calibration, de correction et d'identification. La famille IEEE1451 comprend les normes suivantes: IEEE 1451.0, IEEE 1451.1, IEEE 1451.2, IEEE 1451.3, IEEE 1451.4, IEEE 1451.5, IEEE 1451.7.

La norme IEEE 1451.0 [24] est la norme principale de cette famille. C'est la norme qui définit le modèle de référence, les services et les fonctions d'un transducteur intelligent. Les normes 1451.2, 1451.3, 1451.5 et 1451.7 s'intéressent particulièrement aux protocoles de communication reliant l'interface de transducteurs ("Transducer Interface Module", TIM) aux "Network Capable Application Processor", NCAP. La norme 1451.6 est en cours d'élaboration et elle couvre la communication entre le TIM et le NCAP via le protocole CANopen. Finalement, la norme 1451.1 définit la couche applicative du NCAP.

2.4 Modèle de transducteur intelligent selon la norme IEEE1451.0

Le modèle de transducteur intelligent tel que défini par la norme IEEE 1451.0 est composé de deux parties : le "Transducer Interface Module" (TIM) et le "Network Capable Application Processor" (NCAP) (*Voir* Figure 2.1).

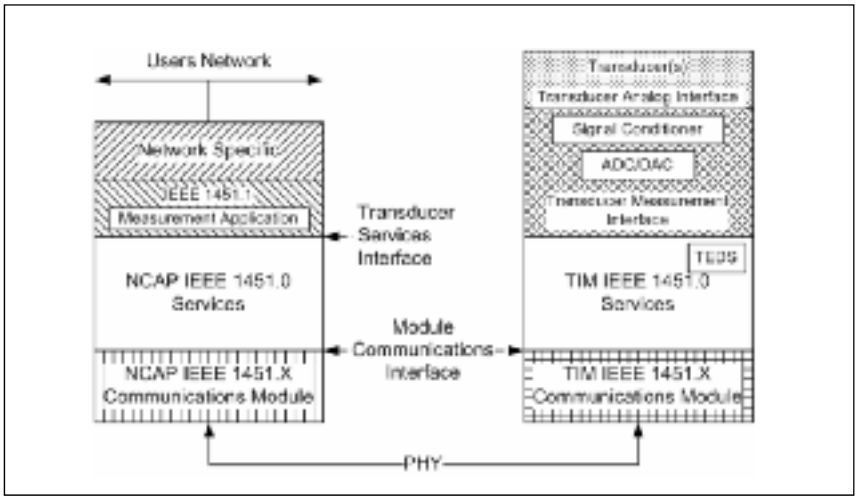

Figure 2.1 Modèle de référence simplifié
Tirée de IEEE 1451.0 (2007, p.21)

Le NCAP est un dispositif qui joue le rôle de passerelle entre les interfaces de transducteurs TIM et le réseau d'utilisateurs (*Voir* figure 2.1). Le NCAP est composé typiquement d'un processeur et d'un système d'exploitation qui servent à exécuter les applications du NCAP. Par ailleurs, deux interfaces de communication permettent au NCAP de communiquer d'une part avec le réseau d'utilisateurs et d'autre part avec les transducteurs connectés au TIM.

Le TIM est un module qui joue le rôle d'interface pour les transducteurs et qui contient (*Voir* Figure 2.1):

- un transducteur (s) ou une connexion au transducteur (s) ;
- un circuit pour la conversion et le conditionnement du signal ;
- une logique pour l'interface des mesures du transducteur ;
- une logique qui implémente les services du TIM ;

- les TEDS ;
- une interface de communication avec le NCAP.

Dans ce qui suit, nous ne nous intéressons qu'à la description du TIM.

2.5 Interface de transducteurs (TIM)

2.5.1 États de fonctionnement

Comme le montre la Figure 2.2, trois états définissent le domaine de fonctionnement du TIM: *TIM Initialization*, *TIM Active* et *TIM Sleep*. Le TIM entre immédiatement dans l'état d'initialisation *TIM Initialization* après la réception de la commande `Reset` ou après sa mise sous tension. Une fois le processus d'initialisation terminé, le TIM passe à l'état actif *TIM Active*. Le TIM passe à l'état de veille *TIM Sleep* après la réception de la commande de mise en veille `TIM_Sleep`. La réception de la commande d'activation `Wake_up` fait retourner le TIM à l'état actif.

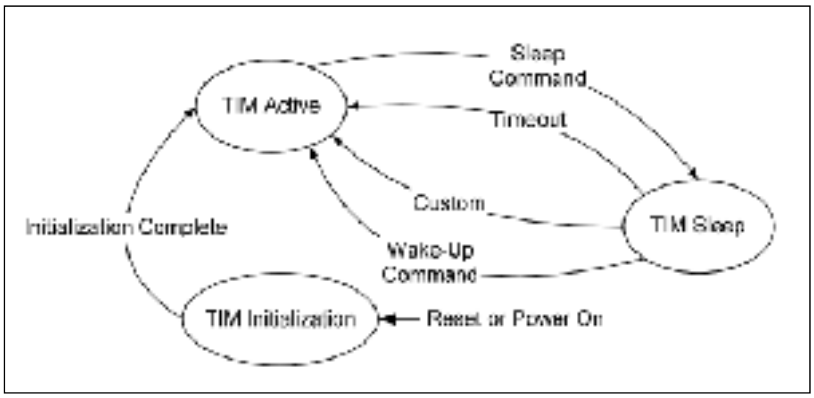

Figure 2.2 États de fonctionnement du TIM
Tirée de IEEE 1451.0 (2007, p.26)

Un TIM est capable d'interfacer un ou plusieurs transducteurs. La norme IEEE1451.0 introduit la notion de canal de transducteur ("Transducer Channel") pour désigner un

transducteur. Un canal de transducteur est composé: d'un transducteur, d'un circuit de conversion (CAN ou CNA) et de la logique de l'interface de mesure du transducteur.

Une modélisation à haut niveau d'un canal de transducteur est fournie par le diagramme d'états à la Figure 2.3. La réception de la commande d'initialisation, `Reset`, ou la mise sous tension ou l'initialisation du TIM font passer le canal de transducteur à l'état d'initialisation *Transducer Initialization*. Une fois le processus d'initialisation terminé, le canal passe à l'état de veille *Transducer Idle*. Un canal de transducteur passe à l'état de fonctionnement *Transducer Operating* une fois la commande `TransducerChannel_Operate` reçue.

Le passage du canal de transducteur de l'état *Transducer Operating* à l'état *Transducer Idle* est assuré via la réception, entre autres, des commandes d'initialisation, `Reset`, et de mise en veille du TIM, `TIM_Sleep`.

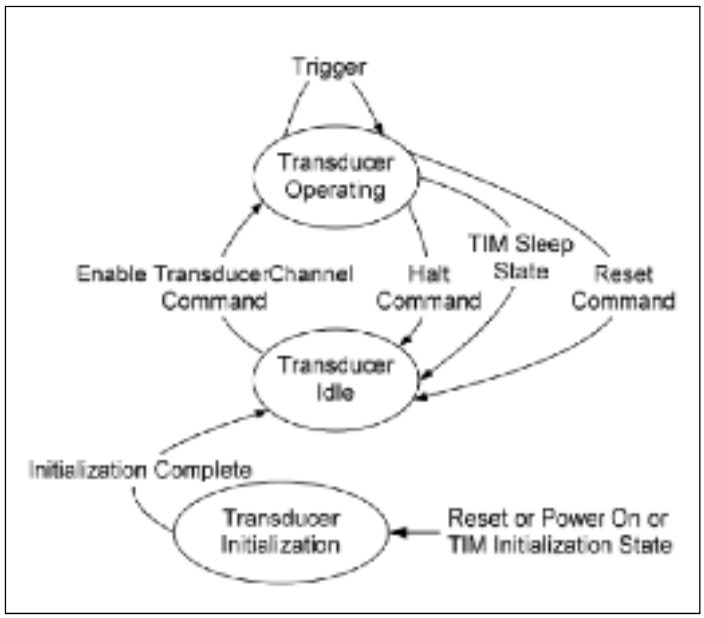

Figure 2.3 Diagramme d'états d'un canal de transducteur
Tirée de IEEE 1451.0 (2007, p.25)

2.5.2 Modes d’échantillonnage

La norme 1451.0 définit cinq modes d’échantillonnage. Un transducteur peut supporter un ou plusieurs modes d'échantillonnage. Ces modes sont stockés dans les TEDS du canal de transducteur. Les cinq modes d’échantillonnage dérivent de deux modes principaux: 1) échantillonnage initié par déclenchement ("Trigger-initiated"), 2) échantillonnage libre ("Free-running").

Échantillonnage initié par déclenchement: Dans ce mode, le transducteur commence l’exécution des fonctions de mesure ou de contrôle immédiatement après la réception d’un signal de déclenchement ("trigger"). Pour un capteur, la réception du signal de déclenchement cause le lancement de l’acquisition des mesures. Pour les variantes issues de ce mode, l’acquisition, l’enregistrement et la transmission des données diffèrent d’une variante à l’autre.

Échantillonnage libre: Dans ce mode d'échantillonnage, le capteur mesure des paramètres physiques de manière autonome et continue, tant qu’il est à l'état *Transducer Operating*. Pour les variantes issues de ce mode, l’acquisition, l’enregistrement et la transmission des données diffèrent d’une variante à une autre.

2.5.3 Déclenchement du transducteur

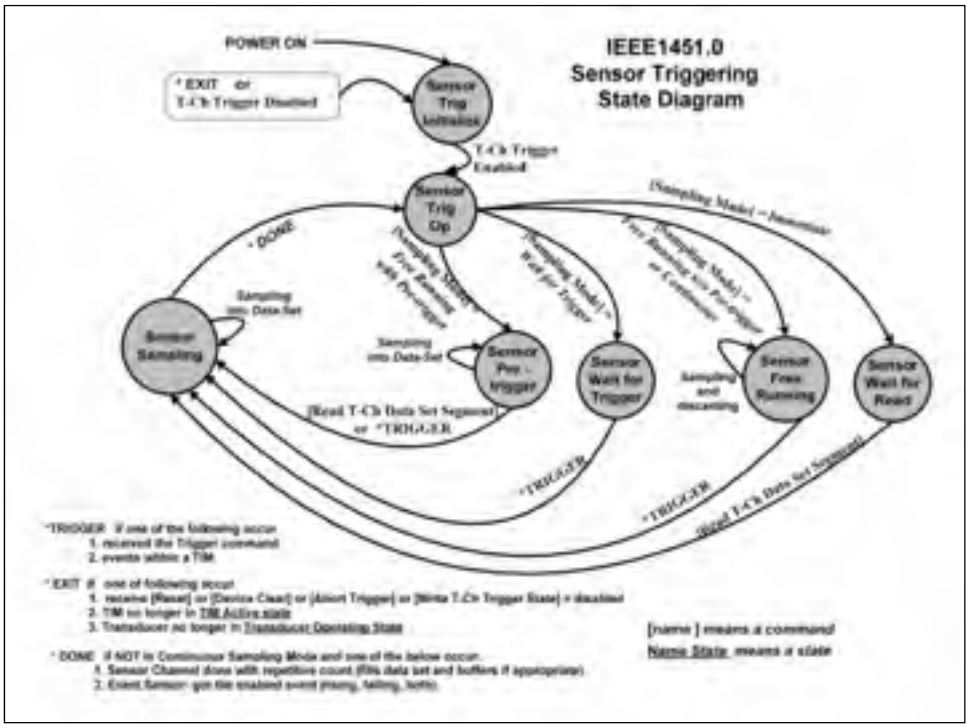

Figure 2.4 Diagramme d'états de déclenchement d'un canal de transducteur
Tirée de IEEE 1451.0 (2007, p.43)

Le déclenchement d'un transducteur pourrait être causé par la réception d'une commande ou l'occurrence d'un événement dans le TIM. Le diagramme d'états de la logique de déclenchement d'un capteur est illustré à la Figure 2.4. Celle-ci montre plusieurs scénarios de déclenchement d'un capteur dépendamment du mode d'échantillonnage choisi.

La Figure 2.5 montre le diagramme de bloc d'un canal de transducteur. Afin d'acquérir un échantillon, la *Logique de Transport de Données* ("Data Transport Logic") d'un capteur lance un signal de déclenchement `StartSamp` qui cause le début d'acquisition de données par le capteur.

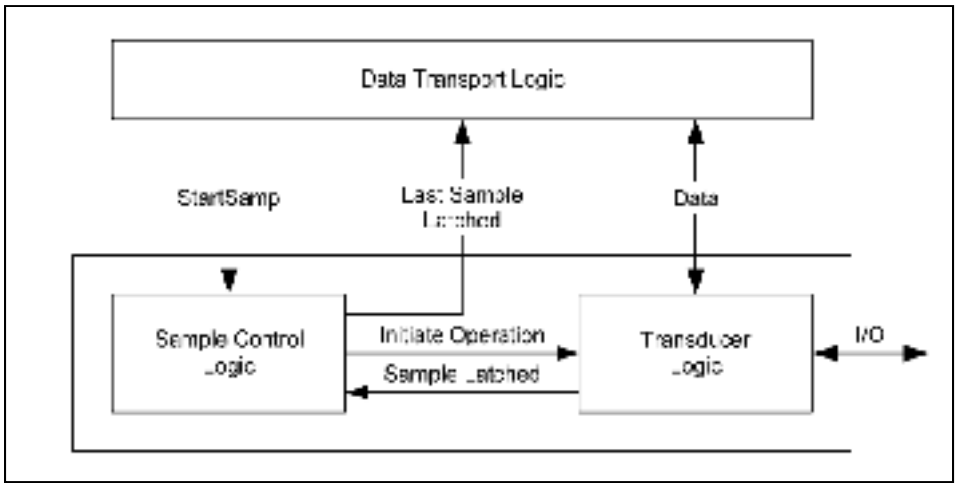

Figure 2.5 Diagramme de bloc d'un canal de transducteur
Tirée de IEEE 1451.0 (2007, p.45)

La *Logique de Contrôle d'Échantillon* ("Sample Control Logic") est responsable de l'orchestration de la séquence des opérations nécessaires à l'acquisition des données. La *Logique du Transducteur* ("Transducer Logic") est responsable du conditionnement et de la conversion des signaux. Le signal `Initiate_Operation` sert à lancer le processus de conversion d'un échantillon. Le signal `Sample_Latched` indique que l'échantillon a été acquis avec succès.

2.5.4 Structure des messages

Messages de commandes

Le tableau 2.1 montre la structure des messages de commande reçus par le TIM.

Tableau 2.1 Structure des messages de commande
Tiré de IEEE 1451.0 (2007, p.57)

1-Octet							
7	6	5	4	3	2	1	0
Destination TransducerChannel Number (most significant octet)							
Destination TransducerChannel Number (least significant octet)							
Command class							
Command function							
Length (most significant octet)							
Length (least significant octet)							
Command-dependent octets . . .							

Un message de commande contient les champs suivants :

- numéro du canal de transducteur destination ("Destination TransducerChannel number"): ce champ de 16 bits indique le numéro du canal à qui la commande est destinée ;
- classe de la commande ("Command class"): Ce champ de 8 bits indique la classe de la commande ;
- fonction de la commande ("Command function"): Ce champ de 8 bits indique la fonction de la commande ;
- longueur ("Length"): est le nombre d'octets qui dépendent de la commande ;
- octets dépendant de la commande ("Command-dependent octets"): Ce champ contient l'information liée à la commande.

Messages de réponse

Les messages de réponse ("Reply messages") sont utilisés pour répondre à une commande reçue. Le format du message de réponse est fourni dans le Tableau 2.2.

Tableau 2.2 Structure des messages de réponse
Tiré de IEEE 1451.0 (2007, p.58)

1-Octet							
7	6	5	4	3	2	1	0
Success/Fail Flag							
Length (most significant octet)							
Length (least significant octet)							
Reply-dependent octets . . .							

Le message de réponse est composé des champs suivants :

- drapeau de Succès/Échec ("Success/Fail flag"): cet octet indique si la commande a été effectuée avec succès ou non ;
- longueur ("Length") : nombre d'octets du message ;
- octets dépendant de la réponse ("Reply-dependent octets") : Ce champ contient l'information qui définit la réponse à une commande.

2.5.5 Commandes

Une commande est composée de deux octets. L'octet le plus significatif est utilisé pour définir la classe de la commande et l'octet le moins significatif identifie la fonction de la commande. Les commandes peuvent être envoyées au TIM dans son ensemble ou à un canal bien déterminé du TIM. La norme IEEE 1451.0 définit plusieurs commandes par classe.

Exemples de classes : commandes envoyées au TIM quand celui-ci est à l'état de veille, commandes envoyées à un canal de transducteur quand celui-ci est à l'état de fonctionnement;

Exemples de commandes : commande de déclenchement, commande de mise en veille, commande d'initialisation.

2.5.6 TEDS

Les TEDS fournit toutes les informations sur le transducteur, y compris:

- le fabricant, le numéro de modèle, le numéro de série ;
- le type et les limites d'utilisation ;
- les constantes de calibration ;
- le modèle de conversion du signal et la longueur du modèle ;
- la période d'échantillonnage, le temps de démarrage ;
- les exigences d'alimentation (tension et courant) ;
- la longueur du TEDS et le nombre de canaux.

La norme IEEE 1451.0 définit plusieurs types de TEDS. Les TEDS de calibration sont considérés parmi les TEDS de base pour le fonctionnement d'un transducteur intelligent. Ces TEDS mémorisent les constantes du modèle de calibration des canaux de transducteurs. Le processus de calibration consiste à définir une équation décrivant la fonction de transfert du transducteur. La méthode de calibration la plus couramment utilisée est la calibration linéaire. L'équation (2.1), montre une fonction de calibration linéaire :

$$y = ax + b \tag{2.1}$$

où :

x : valeur brute

a,b : coefficients de calibration

y : valeur réelle

Pour un capteur, la correction des mesures est le processus par lequel le modèle de calibration est appliqué à la sortie d'un capteur. Le processus de correction permet de convertir l'échantillon en une forme avec une signification physique réelle. Pour un actionneur, la correction est appliquée à la commande fournie par le système pour la convertir à la forme requise par l'actionneur.

2.5.7 Adressage

La norme IEEE 1451.0 définit deux niveaux d'adressage pour permettre l'échange de messages entre un TIM et un NCAP. Le premier niveau d'adressage est associé à l'implémentation de la couche physique du TIM. Celui-ci permet de s'adresser à un ou plusieurs TIM dans un réseau de transducteurs. Le deuxième niveau d'adressage permet de cibler un canal de transducteur bien précis dans un TIM grâce à son numéro de canal de transducteur. Ce numéro, codé sur 16 bits, est utilisé dans les messages de commandes comme adresse destination.

2.5.8 Détection et diagnostic des pannes

À la base, la norme IEEE 1451.0 n'est pas conçue pour des applications critiques. Ainsi, aucune directive n'est fournie concernant la logique nécessaire à la détection et le diagnostic des pannes dans le TIM. La norme évoque seulement la possibilité d'initier des commandes par le NCAP ou par un calculateur central pour lancer le diagnostic dans le TIM.

2.6 Répartition de l'intelligence dans un réseau de transducteurs basé sur IEEE 1451

Aucune recommandation n'est fournie par l'IEEE 1451.0 concernant la répartition de l'intelligence entre les différents composants d'un réseau de transducteurs intelligents. En fait, la répartition de l'intelligence dépend de l'application et du contexte. La Figure 2.6 présente un exemple où la correction des échantillons est exécutée dans plusieurs niveaux du réseau. Dans le scénario 2.6 (a), le calculateur central copie les TEDS de calibration, qui sont stockés dans le TIM, et il effectue la correction des données provenant du TIM. Ceci est rentable pour les petits réseaux. Toutefois, le déploiement d'un tel scénario dans des réseaux avec de nombreux capteurs et calculateurs pourrait complexifier le processus de correction. Dans 2.6 (b), la correction est effectuée dans le NCAP. Celui-ci corrige les données après avoir copié les TEDS de calibration. Ce scénario simplifie la gestion des capteurs et élimine

les copies multiples des TEDS. Dans le scénario 2.6 (c), c'est le TIM qui exécute la correction. Ce scénario est utile pour les réseaux avec un grand nombre de transducteurs.

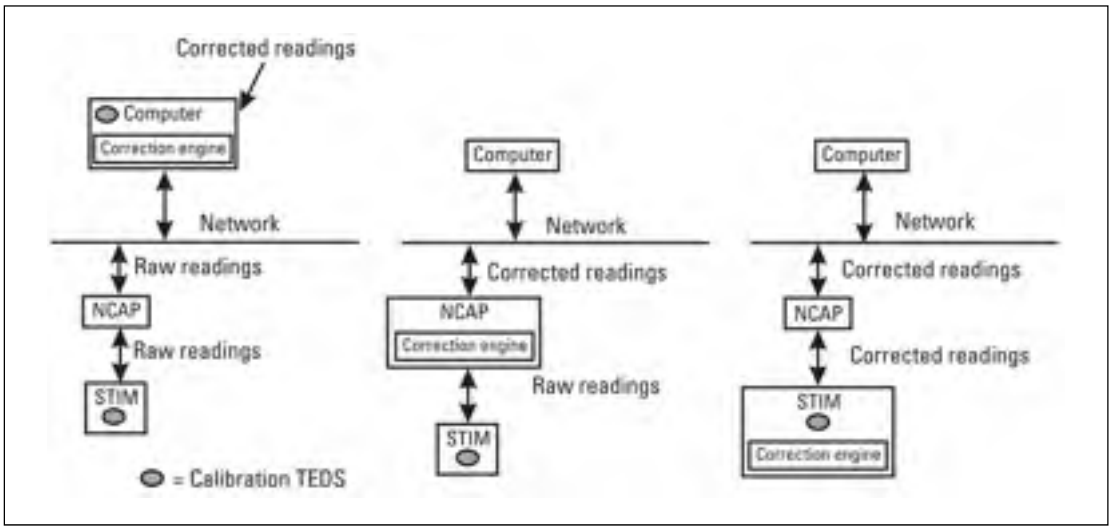

Figure 2.6 Algorithme de correction exécuté dans(a) Calculateur, (b) NCAP, et (c) TIM
Tirée de Frank (2000, p.288)

2.7 Projet AVIO 402

2.7.1 Architecture du réseau

Un des objectifs les plus importants du projet AVIO 402 est de développer un réseau avionique permettant de relier les capteurs et les actuateurs aux calculateurs de contrôle de vol et à l'IMA dans un aéronef. Le réseau principal de l'architecture proposée est basé sur le protocole AFDX. Le bus de terrain ARINC 825 a été adopté pour connecter les différents transducteurs au réseau principal AFDX. L'architecture de base du système est présentée à la Figure 2.7. Les interfaces qui relient les transducteurs aux bus ARINC 825 et celles qui interconnectent les deux réseaux AFDX et ARINC 825 sont basées sur la norme IEEE 1451.0. Une architecture complète du réseau a été proposée par José-Philippe Tremblay dans le cadre de ses travaux de doctorat (*Voir* Figure 2.8).

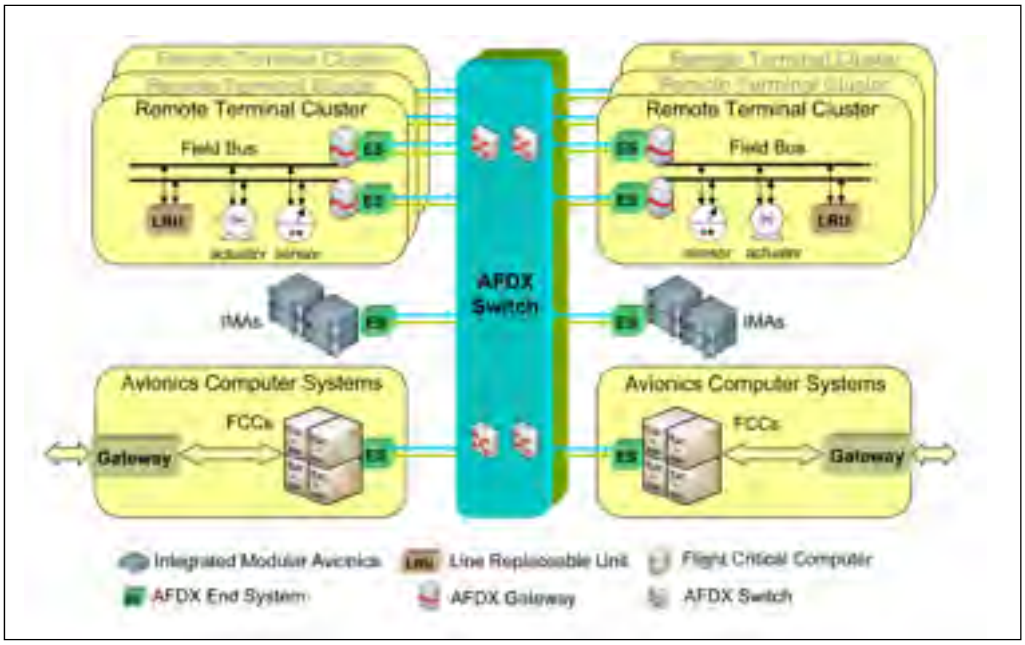

Figure 2.7 Aperçu du réseau proposé
Tirée de la documentation interne du projet AVIO 402

Deux réseaux secondaires de type ARINC 825 permettent de connecter un certain nombre de TIM assurant la gestion des données et le contrôle des transducteurs. Chaque TIM pourrait interfacer un ou plusieurs capteurs et/ou actuateurs. Il est également capable de supporter une large gamme de technologies de transducteurs allant des capteurs basés sur les vieilles technologies RVDT et LVDT jusqu'aux nouvelles générations de transducteur MEMS, et ce grâce à une architecture générique et flexible. Le nombre de TIM utilisé pour chaque système dépend de la criticité de l'application en question. Le bus ARINC 825 de chaque réseau secondaire est doublé afin d'augmenter la fiabilité et la bande passante du réseau. Les NCAP jouent le rôle de passerelles entre les réseaux ARINC 825 et AFDX et sont responsables de la gestion des flux de données transitant entre les deux réseaux. Deux interfaces de communication permettent aux TIM et aux NCAP de se connecter aux deux bus ARINC 825.

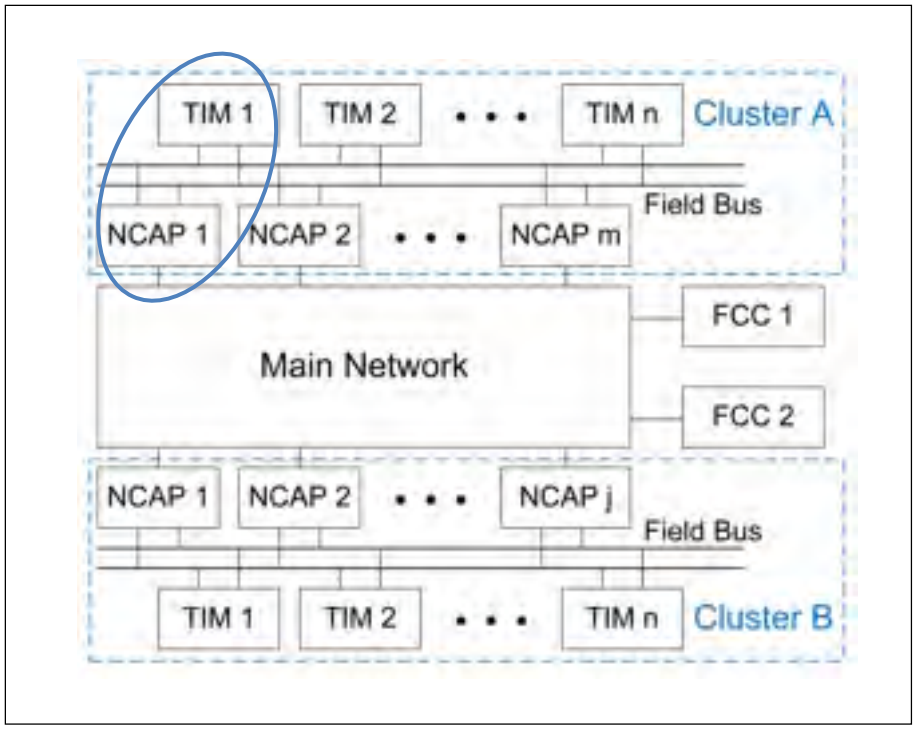

Figure 2.8 Architecture du réseau basée sur la norme IEEE1451
Tirée de la documentation interne du projet AVIO 402

Le nombre de NCAP utilisé dans chaque réseau secondaire n'est pas figé et peut changer selon les requis en matière de fiabilité et de fonctionnalité. Les TIM sont accessibles à partir de n'importe quel NCAP étant donné que le bus est partagé entre tous les sous-systèmes (NCAP et TIM). Le réseau ARINC 825 fournit une bande passante de 1 Mbit/s. Le réseau AFDX, quant à lui, garantit une bande passante de 100 Mbit/s, largement supérieure à celle du bus ARINC 825, et ce afin de pouvoir multiplexer les données provenant de plusieurs réseaux secondaires.

2.7.2 Encapsuleur du réseau

L'encapsuleur du réseau est l'ensemble formé par: un TIM, un bus ARINC 825 double et un NCAP (*Voir* Figure 2.8). Cette partie de l'architecture constitue la brique de base du réseau ARINC 825. Le développement de l'encapsuleur facilite la mise en place d'un réseau plus complexe incluant plusieurs TIM et plusieurs NCAP. Une architecture générique du NCAP,

telle que conçue par José Philippe Tremblay [40], est présentée dans la Figure 2.9. L'architecture comprend *N* modules de communication où chacun d'entre eux est dédié à un bus de terrain. Le NCAP dispose de *M* modules de service qui offrent un certain nombre de services pour les données qui les traversent. Les services les plus pertinents du NCAP sont la préparation et le formatage des données reçues par le bus ARINC 825 pour qu'elles soient transmises sur le réseau AFDX.

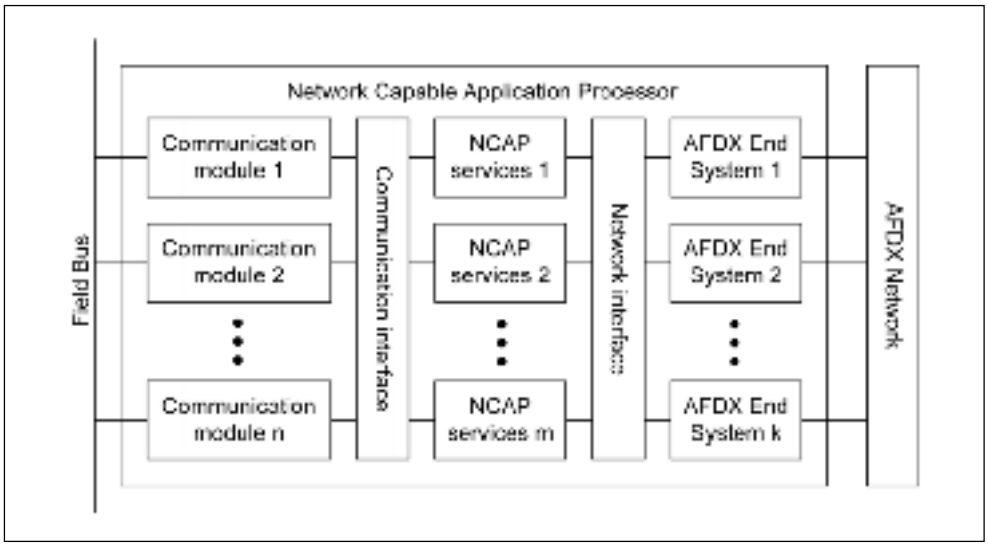

Figure 2.9 Architecture du NCAP
Tirée de la documentation interne du projet AVIO 402

Par ailleurs, *K* modules "End System" AFDX permettent de gérer le flux de données circulant sur le réseau AFDX. Deux "crossbar" insérés entre les modules du NCAP permettent de commuter les paquets de données vers n'importe quel module de service et n'importe quelle interface de sortie dépendamment de leurs destinations. En cas de pannes de l'un des modules ou de l'un des bus, les "crossbars" permettent de rediriger le flux de données vers les modules opérationnels disponibles. Ceci permet de minimiser la latence des données et d'éviter la perte des paquets pour les systèmes avioniques critiques. Plusieurs configurations du NCAP sont possibles en jouant sur les paramètres *N*, *M* et *K*. Celles-ci sont determinées en fonction de la fiabilité et la bande passante recquises. La configuration du NCAP developpée dans le cadre du projet AVIO 402 est obtenue en posant *N*=2 (*Voir* Figure

2.9) et en choisissant l'ARINC 825 comme bus de terrain. Dans le même cadre, une architecture générique du TIM a été proposée par José-Philippe Tremblay. Celle-ci est, dans l'ensemble, similaire à celle du NCAP présentée dans la Figure 2.9. Étant donnée que le deveploppement et l'implémentation du TIM font l'objet de ce mémoire, celui-ci sera étudié en détail dans les chapitres 3, 4 et 5. Par analogie avec le scénario (c) de la figure 2.6, il a été décidé que pour le developpement de l'encapsuleur du réseau, la totalité de l'intelligence sera allouée aux TIM et que le NCAP se limitera à gérer les flux de données entre le réseau principal AFDX et le réseau secondaire ARINC 825.

2.7.3 Prototype de l'encapsuleur

Le but de la tâche 2 du projet AVIO 402 est de développer un prototye de l'encapsuleur du réseau présenté dans la section précedente. L'encapsuleur en question est composé d'un NCAP et d'un TIM connecté à plusieurs capteurs. La communication entre le TIM et le NCAP est assurée via un bus ARINC 825 double. Le prototype servira comme preuve de concept de la technologie des transducteurs intelligents et constituera un banc d'essai pour la validation d'une multitude d'algorithmes et de fonctionnalités des systèmes avioniques critiques. La plateforme de developpement qui a été choisie pour l'implémentation du TIM et du NCAP est le FPGA. La carte de développement utilisée est la carte SP605 de Xilinx dotée de la puce FPGA Spartan-6 XC6SLX45T. En ce qui concerne le NCAP, il a été jugé qu'une implémentation logicielle est plus adéquate pour les "End Systems" et que le reste du NCAP allait être implémenté en matériel (*Voir* Figure 2.10). Ainsi, une première version du NCAP reposait sur une implémentation "soft" du microprocesseur Microblaze pour l'exécution des logiciels du "End System". Le reste des modules du NCAP, à savoir les interfaces ARINC 825, les modules de service et les "crossbars" devraient être implémentés en matériel sur le même FPGA. Cependant, étant donnée la complexité du prototype et les problèmes liés à l'interaction entre les composantes matérielles et logicielles du "End System", une deuxième version mise à jour du NCAP a été proposée par Davide Trentin dans le cadre de ses travaux de maîtrise.

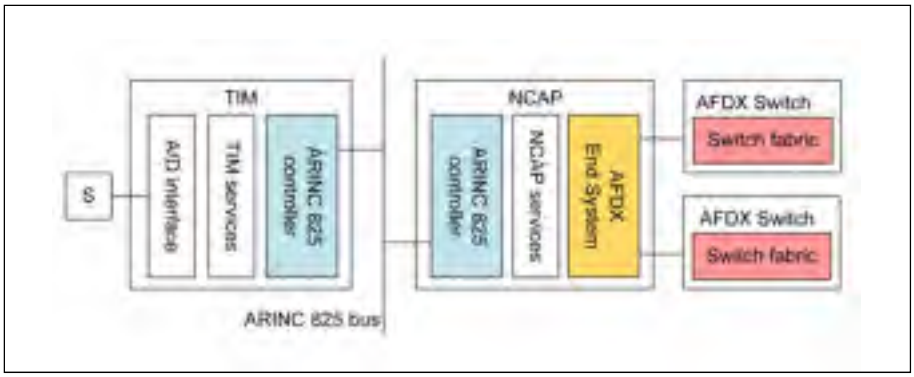

Figure 2.10 Architecture du prototype
Tirée de la documentation interne du projet AVIO 402

Tel qu'illustré dans la Figure 2.11, la nouvelle architecture du prototype du NCAP est composée de deux parties : une carte FPGA SP605 et un PC. Le FPGA est dédié à l'implémentation des contrôleurs ARINC 825 et de l'interface de communication avec le PC. Le PC, doté d'un système d'exploitation Linux, permet d'implémenter la pile protocolaire de l'AFDX ainsi que les applications du "End System". La communication entre la carte FPGA et le PC est assurée via les protocoles USB-UART ou PCI express. Dans un premier temps, le bus USB-UART a été sélectionné pour sa portabilité et sa simplicité. Celui-ci assure une communication duplex avec un débit maximal de 1 Mbit/s. J'étais responsable du développement de l'interface USB-UART du coté du FPGA. L'interface a été implémentée et testée avec succès. Par la suite, afin de pouvoir supporter un débit plus important, il a été décidé que l'interface USB-UART serait remplacée par une interface PCIexpress. J'étais impliqué dans la première phase de développement de ce bus. Une première version de cette interface qui supportait une communication en boucle fermée entre le PC et le FPGA a été réalisée en collaboration avec Talal Zakani. Le reste de l'implémentation est en cours de développement par Féderico Montano.

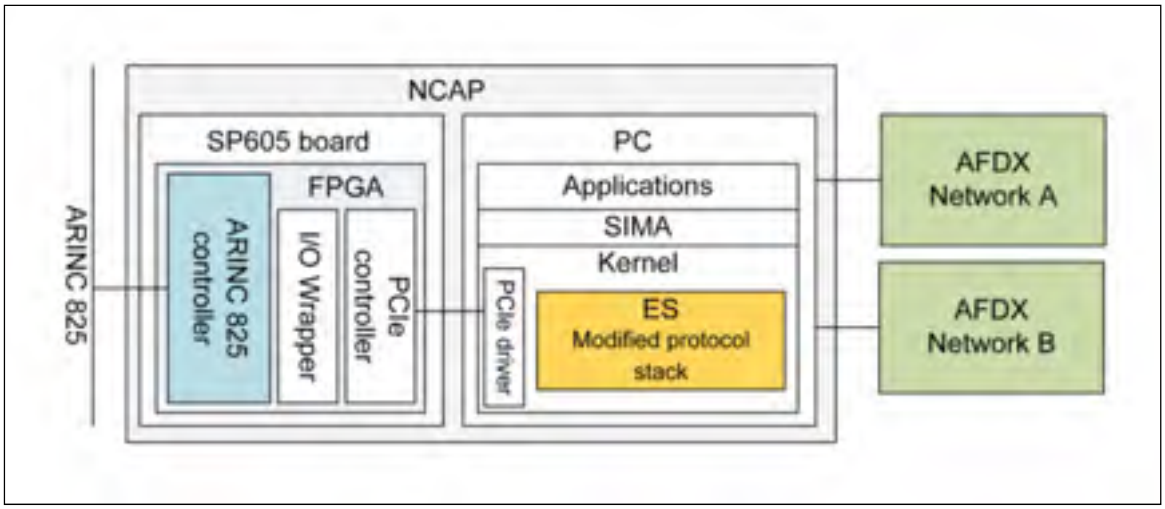

Figure 2.11 Architecture modifié du prototype du NCAP
Tirée de la documentation interne du projet AVIO 402

2.8 Conclusion

Ce chapitre a expliqué, en premier lieu, le besoin de normalisation des interfaces de transducteurs intelligents dans un contexte avionique. Par la suite, la famille des normes IEEE1451 a été présentée comme une solution viable pour interfacer les transducteurs intelligents dans les nouvelles générations d'aéronefs. Par ailleurs, les différents aspects liés aux fonctions et services des interfaces de transducteurs intelligents basées sur la norme IEEE 1451.0 ont été exposés. Finalement, une description détaillée de l'architecture du réseau découlant de l'IEEE 1451.0 développée dans le cadre du projet AVIO 402 a été présentée. Dans le même contexte, le prototype de l'encapsuleur générique du réseau a été discuté.

CHAPITRE 3

INTERFACE DE TRANSDUCTEURS INTELLIGENTS TOLÉRANTE AUX PANNES BASÉE SUR IEEE 1451.0

3.1 Introduction

Ce chapitre étudie les différents aspects liés à la tolérance aux pannes pour la conception d'interfaces de transducteurs intelligents. La démarche suivie pour sélectionner les techniques de tolérance aux pannes les plus appropriées est également discutée. Une section entière est dédiée à la présentation de l'architecture de l'interface de transducteurs basée sur IEEE 1451.0. Finalement, le chapitre dresse une comparaison entre quelques architectures d'interfaces de transducteurs intelligents en termes de tolérance aux pannes.

3.2 Besoin en tolérance aux pannes locale

Les interfaces de transducteurs intelligents conçues pour les systèmes avioniques critiques doivent répondre à de strictes exigences en matière de fiabilité et de sûreté imposées par la réglementation et les normes. Les normes RTCA DO-254 "Design Assurance Guidance For Airborne Electronic Hardware" et ARP 4761 "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment" définissent cinq niveaux de criticité et assignent à chaque niveau une probabilité maximale de défaillance par heure de vol (*Voir* Tableau 3.1). Les systèmes avioniques critiques sont généralement classés parmi les trois niveaux ayant les probabilités de défaillance les moins élevées, soit les niveaux A, B et C. La redondance au niveau système est souvent la solution utilisée pour concevoir des systèmes fiables. Toutefois, cette solution peut conduire à une hausse de la complexité, du poids et de la consommation d'énergie dans un aéronef. Selon [4], des nœuds de transducteurs incluant des mécanismes de détection, de diagnostic et de recouvrement de pannes permettent d'assurer une tolérance aux pannes d'un système CDVE avec un minimum d'unités matérielles (nœuds de transducteurs), ce qui permet de réduire la complexité et le poids du système. Le gain en termes de nombre d'unités matérielles est

obtenu grâce à l'adoption du principe de la redondance active. Celui-ci est également utilisé dans le cadre de ce projet afin de définir les mécanismes de tolérances aux pannes au niveau du FPGA du TIM. Le principe de la redondance active est détaillé dans 3.3.4.2.
Par ailleurs, l'auteur de [25, p12] affirme que le coût de gestion des pannes complexes dans les couches supérieures du système (calculateurs centraux) est beaucoup plus important que celui de leur gestion dans les couches inférieures (transducteurs).

Tout ceci étant dit, des interfaces de transducteurs dotées de mécanismes de détection et de diagnostic de pannes sont nécessaires dans la conception de la prochaine génération des systèmes avioniques critiques. Ces techniques permettent d'atteindre une tolérance aux pannes en utilisant le minimum de matériel (transducteurs et leurs interfaces) et de favoriser une gestion efficace de la redondance.

Tableau 3.1 Niveaux de criticité et probabilités de défaillances maximales associées
Tirée de RTCA DO-254 (2000, p.14)

Classification de danger	**Niveau de criticité**	**Probabilité maximale de défaillance par heure de vol**
Catastrophic	A	10^{-9}
Hazardous	B	10^{-7}
Major	C	10^{-5}
Minor	D	--
No Effect	E	--

3.3 Tolérance aux pannes

3.3.1 Terminologie

Les définitions suivantes ont été extraites de [27, p9] :

- une panne ("fault ") est une défectuosité physique, une imperfection, ou un défaut qui se produit dans un composant matériel ou logiciel. Exemples de pannes: court-circuit entre deux interconnexions adjacentes, une broche cassée, un bug logiciel, etc. ;
- une erreur ("error") est une déviation par rapport à l'exactitude et la précision dans le calcul, qui se produit suite à une panne. Les erreurs sont généralement associées à des valeurs incorrectes dans l'état du système. Exemples d'erreurs: une valeur erronée calculée par un circuit ou un programme, une information erronée reçue lors de la transmission de données ;
- une défaillance ("failure") est causée par une erreur et se produit lorsque le service délivré s'écarte de ce qui est considéré comme correct.

3.3.2 Définition de la tolérance aux pannes

Selon [28, p23], la tolérance aux pannes implique les actions suivantes:

Détection de panne: le processus visant à déterminer qu'une panne s'est produite.

Diagnostic de panne: le processus par lequel la cause d'une panne est déterminée.

Confinement de panne: le processus qui empêche la propagation des pannes à partir de leurs origines vers le reste du système.

Masquage de panne: le processus qui permet d'assurer que seules les valeurs correctes soient transmises à l'extérieur du système en dépit de pannes.

Recouvrement de panne: si une erreur se produit, il pourrait être nécessaire pour le système de fournir une réponse pour compenser la sortie du composant défectueux.

Réparation de panne: le processus par lequel les pannes sont enlevées du système.

En se basant sur cette définition, si les interfaces de transducteurs sont capables d'appliquer une partie ou la totalité des actions susmentionnées, alors il est possible de les qualifier de « tolérantes aux pannes ».

3.3.3 Origine des pannes

L'interface de transducteurs développée dans le cadre de ce mémoire est un système critique qui devrait fonctionner d'une manière fiable et sûre. Étant donnée l'environnement dans lequel ce type de composant est installé, ces interfaces sont susceptibles de subir des pannes et d'effectuer des calculs erronés. En effet, plusieurs facteurs environnementaux tels que les vibrations locales, l'humidité, la température, la décharge électrostatique, les chutes de tension et l'interférence électromagnétique, ont tendance à induire un stress sur le composant, ce qui peut causer une détérioration graduelle ou une défaillance brusque [27, p10].

Les rayonnements cosmiques constituent également une source de pannes pour les systèmes électroniques dans un avion. En effet, le taux de défaillance des composants électroniques causé par ce type de rayonnement à l'altitude de vol des aéronefs est 100 fois plus élevé que celui mesuré au niveau de la mer [27, p12]. La sévérité de ce type de pannes vient du fait qu'elles peuvent causer des "Single Event Upset" (SEU) qui altèrent le contenu de la mémoire du programme dans un microprocesseur ou de la mémoire de configuration d'un FPGA.

Par ailleurs, la miniaturisation des transistors accroit le taux de pannes des circuits intégrés. Ceci constitue un vrai défi pour le développement des prochaines générations de systèmes avioniques [29]. Les défectuosités qui surviennent lors de la fabrication des circuits intégrés et les erreurs de conception et d'implémentation peuvent aussi être à l'origine des défaillances des systèmes avioniques, si elles ne sont pas détectées lors des différents tests appliqués aux circuits et aux systèmes.

Les pannes découlant des facteurs susmentionnés sont soit permanentes ou transitoires:

Panne permanente : Une panne permanente reste active jusqu'à ce qu'une action corrective soit prise. Ces pannes sont généralement provoquées par des défectuosités physiques dans le matériel, tel qu'un court-circuit, une interconnexion brisée ou un bit dans la mémoire dont le contenu est figé (à 0 ou à 1) ;

Panne transitoire : Une panne transitoire reste active pendant une courte durée. Les pannes transitoires sont souvent détectées par les erreurs qui découlent de leur propagation. Les causes de ce type de pannes sont souvent environnementales.

3.3.4 Techniques de la tolérance aux pannes

3.3.4.1 Tolérance aux pannes par couche pour les systèmes distribués

Les auteurs de [25] [30] soulignent la nécessité de mettre en place des mécanismes de tolérance aux pannes dans plusieurs couches d'un système distribué. Ces mécanismes fonctionnent d'une manière indépendante les uns des autres. Si une couche n'arrive pas à détecter une panne, les mécanismes implémentés dans les autres couches essayeront de combler cet échec. Ainsi, chaque couche constitue une ligne de défense à part entière qui vise à détecter, diagnostiquer et empêcher les pannes de se propager dans les autres parties du système. L'auteur de [25] définit trois couches de tolérance aux pannes :

Couche matérielle ("Hardware-layer"): Cette couche fournit des mécanismes de base de la tolérance aux pannes implémentés en matériel. (Exemples: la détection des instructions invalides et des accès mémoire erronés dans un microprocesseur, la protection de la mémoire par des bits de parité ou par des codes de contrôle cycliques (CRC)) ;

Couche nœud ("Node-layer"): La tolérance aux pannes au niveau de cette couche est assurée via l'utilisation de composants matériels ou logiciels supplémentaires dans chacune des nœuds du système. (Exemple: l'exécution duplex ou triplex de versions différentes d'un programme) ;

Couche système ("System-layer"): Cette couche vise à tolérer les défaillances des nœuds à travers: 1) l'utilisation de nœuds redondants et 2) la gestion de la redondance par une tierce unité (tel qu'un calculateur central).

3.3.4.2 Redondance modulaire

La tolérance aux pannes du TIM se base principalement sur les techniques de la couche matérielle et de la couche nœud. Dans le cadre de ce mémoire, ces deux couches sont considérées comme étant une seule couche vu que tout le TIM est implémenté en matériel (FPGA). Par ailleurs, ce chapitre ne traite pas les techniques de tolérance aux pannes de la couche système.

La redondance modulaire au niveau de la couche matérielle est la technique de tolérance aux pannes qui a été sélectionnée pour le développement du TIM étant donné qu'elle permet de tolérer les pannes permanentes et transitoires [31]. Les techniques hors ligne (BIST) ont été écartées vu qu'elles ne ciblent principalement que les pannes permanentes. De plus, les techniques de redondance temporelle qui se basent sur l'exécution répétitive des tâches par le même module sont jugées inefficaces étant données qu'elles ne servent principalement qu'à détecter des pannes transitoires [31]. Les techniques de reconfiguration périodiques des FPGA ont été à leur tour écartées à cause de la lenteur du processus de recouvrement des pannes [32].
Il existe deux types de redondance modulaire: passive et active.

Redondance passive

La redondance passive permet de masquer les pannes plutôt que de les détecter. Grâce au masquage, seules les valeurs correctes sont transmises à la sortie du système. La redondance N-modulaire ("N-Modular Redundancy", NMR) est la forme générique de la redondance passive, où N est le nombre de modules redondants (*Voir* Figure 3.1). Dans ce type de

redondance, N modules de fonctionnalité identique effectuent simultanément le même calcul et un vote à majorité détermine le résultat valide.

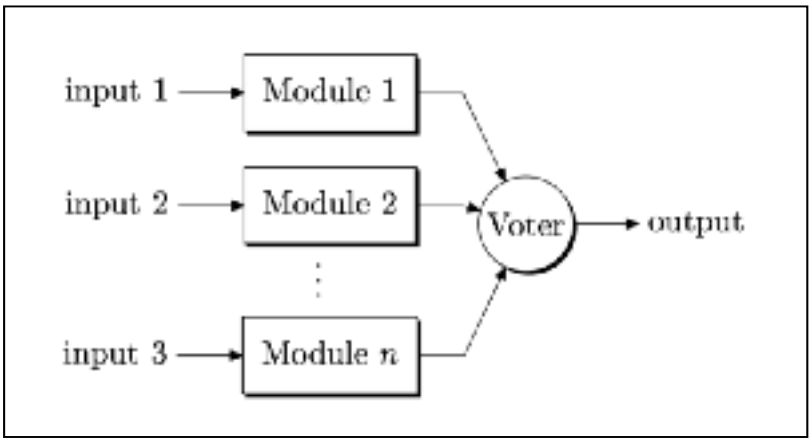

Figure 3.1 Redondance N-modulaire
Tirée de Durbova (2007, p.55)

La redondance modulaire triple ("Triple Modular Redundancy",TMR) est un cas particulier de la redondance NMR (N=3).Un système TMR ne peut supporter qu'un seul module défectueux. La défaillance d'un deuxième module peut provoquer des résultats erronés à la sortie du voteur et cause une défaillance non-sûre du système ("Fail Unsafe"). Le nombre de module minimal requis pour que ce type de redondance permette de tolérer une panne est 3. La technique TMR a été largement utilisée dans la conception de FPGA tolérants aux pannes [33] [34] [35] [36]. La principale motivation derrière ces travaux est la conception de FPGA capables de tolérer les pannes causées par des SEU.

Redondance active

Ce type de redondance assure une tolérance aux pannes grâce à la détection des erreurs et le recouvrement du système à un état opérationnel. Les techniques de redondance active sont utilisées dans les applications qui peuvent tolérer des arrêts de service courts et occasionnels. La forme de base de la redondance active est la technique Duplication avec Comparaison (*Voir* Figure 3.2): deux modules identiques effectuent le même calcul simultanément et leurs

sorties sont par la suite comparés par un module comparateur. Si les deux modules sont en désaccord, un signal d'erreur est généré.

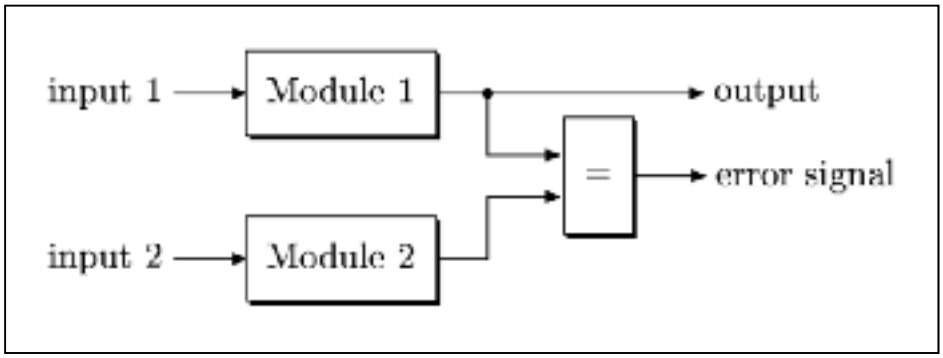

Figure 3.2 Duplication avec comparaison
Tirée de Durbova (2007, p.56)

Le comparateur permet de détecter les pannes mais n'est pas capable de déterminer le module fautif. Dans le cas où une panne permanente est détectée, le comparateur signale l'existence de la panne et le système cesse de générer des données, permettant ainsi une défaillance sûre.

La Redondance avec Rechange est la forme générique de la redondance modulaire active. Cette technique est basée sur l'utilisation de N modules redondants pour le traitement des données. Chaque module inclut une fonction de détection de panne FD (Fault Detection) (*Voir* Figure 3.3). Il existe deux variantes de cette technique. Dans la première variante, un seul module est opérationnel et fournit la sortie du système. Les N-1 modules restants sont en veille et servent de pièces de rechange. En cas de panne permanente du module opérationnel, le commutateur s'occupe du remplacement du module fautif par un module de rechange. Dans la deuxième variante, les N modules disponibles exécutent des tâches de manière indépendante les uns des autres. La charge de traitement du système est partagée entre les N modules. En cas de panne permanente de l'un des modules, les tâches dont le module fautif était responsable seront exécutées par d'autres modules. Dans ce cas, les performances de calcul du système se dégradent, mais elles devraient être maintenues en dessus d'un seuil bien déterminé. Ce mode de dégradation est appelé dégradation gracieuse ("Graceful

Degradation"). Afin de pouvoir tolérer une panne, tout système basé sur la technique de la redondance avec rechange devrait inclure au moins deux modules incorporant la fonction FD. Ceci représente un gain en ressources matérielles par rapport à la redondance passive

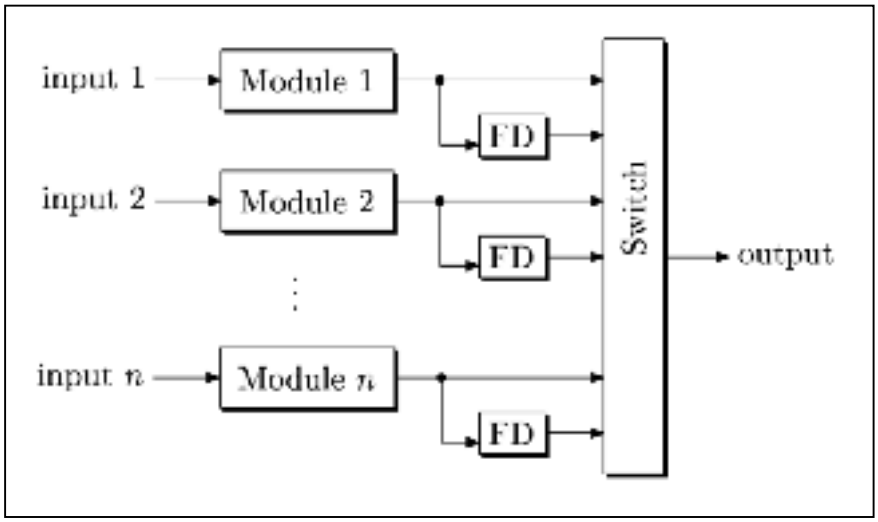

Figure 3.3 Redondance avec rechange
Tirée de Durbova (2007, p.57)

(3 modules pour la redondance TMR). Les auteurs de [37] proposent des architectures de processeurs multi-cœurs (CPU) tolérants aux pannes pour des applications automobiles critiques. Les architectures proposées dans ce travail sont basées sur la technique Duplication avec Comparaison. Par ailleurs, au cours de la dernière décennie, le fabricant de semi-conducteur Texas Instrument a développé une famille de microcontrôleurs "TMS570 Safety Microcontrollers" pour des applications critiques qui fonctionne selon la technique Duplication avec Comparaison [38]. En effet, ceux-ci implémentent deux cœurs CPU exécutant simultanément les mêmes tâches ainsi qu'un comparateur pour la comparaison des sorties des CPU. En cas de panne permanente, la paire de cœurs cesse de fonctionner, permettant ainsi une défaillance sûre du microcontrôleur ("Fail-Safe"). En 2012, le fabricant de semi-conducteur Freescale a annoncé le développement d'une nouvelle génération de microcontrôleurs multi-cœurs tolérants aux pannes pour des applications automobiles critiques [39]. Ceux-ci sont basés sur la technique Redondance avec Rechange et comportent deux paires de cœurs CPU (4 CPU). Chacune des paires fonctionne selon la technique

Duplication avec Comparaison. Cette architecture est caractérisée par une capacité de calcul deux fois plus importante que celle des microcontrôleurs TMS570. En cas de panne permanente de l'une des paires, la paire restante prend la relève tout en assurant un minimum de performance, ce qui permet d'étendre la disponibilité du microcontrôleur.
La redondance active et plus précisément la redondance avec rechange est la technique de tolérance aux pannes qui a été sélectionnée pour la conception du TIM en raison de sa capacité:1) d'assurer une tolérance aux pannes avec de ressources matérielles minimales, 2) d'assurer des défaillances sûres, et 3) d'augmenter les performances de traitement de l'interface. Le même principe de tolérance aux pannes s'applique au niveau de la couche système, ce qui permet de concevoir des systèmes avec un minimum d'unités matérielles.

3.4 Architecture du TIM

3.4.1 Première version

Une première version du TIM a été proposée par José-Philippe Tremblay. Celle-ci présente une architecture générique basée sur la norme IEEE 1451.0 (*Voir* Figure 3.4). Le TIM contient *M* canaux de transducteurs. Chaque canal est composé: d'un transducteur et d'une interface de mesures (TMI) pour la gestion des mesures. Par ailleurs, les *N* modules de service constituent les principaux modules du TIM. Ils sont responsables, entre autres: de l'exécution des commandes provenant des calculateurs centraux, de l'asservissement des actuateurs, du traitement et de l'encapsulation des mesures brutes transmises par les capteurs, et de la supervision des transducteurs ainsi que des différents modules du TIM. L'exécution des fonctions d'actuation des systèmes CDVE distribués présentées à la section 1.4 est également assurée par les modules de service. Les TEDS sont des ressources partagées entre les différents modules de service. Le transfert de données sur les bus de terrain est assuré via *P* modules de communication dédiés. D'une manière similaire au NCAP, deux "crossbar" insérés entre les modules du TIM permettent de commuter les paquets de données vers les transducteurs, les modules de service et les interfaces de communication en fonction de la destination des messages et de la disponibilité des modules. En cas de pannes de l'un des

modules de service, les "crossbar" redirigent le flux de données vers les modules disponibles restants. De plus, un mécanisme de recouvrement permet de faire passer le trafic par les bus disponibles en cas de défaillance de l'un des bus. L'ensemble des mécanismes de recouvrement proposés permet de minimiser la latence des données et d'éviter la perte des paquets. Par ailleurs, la redondance modulaire fournie par le TIM permet d'éliminer plusieurs points de défaillance unique. Plusieurs configurations du TIM sont possibles en jouant sur les paramètres (*M*, *N* et *P*). Le nombre de modules TMI (*M*) dépend du nombre de transducteurs par TIM qui, à son tour, dépend de la nature et de la criticité de l'application avionique. Le nombre de modules de communication (*P*) ainsi que le nombre de modules de service (*N*) dépendent du niveau de fiabilité et de la bande passante visés.

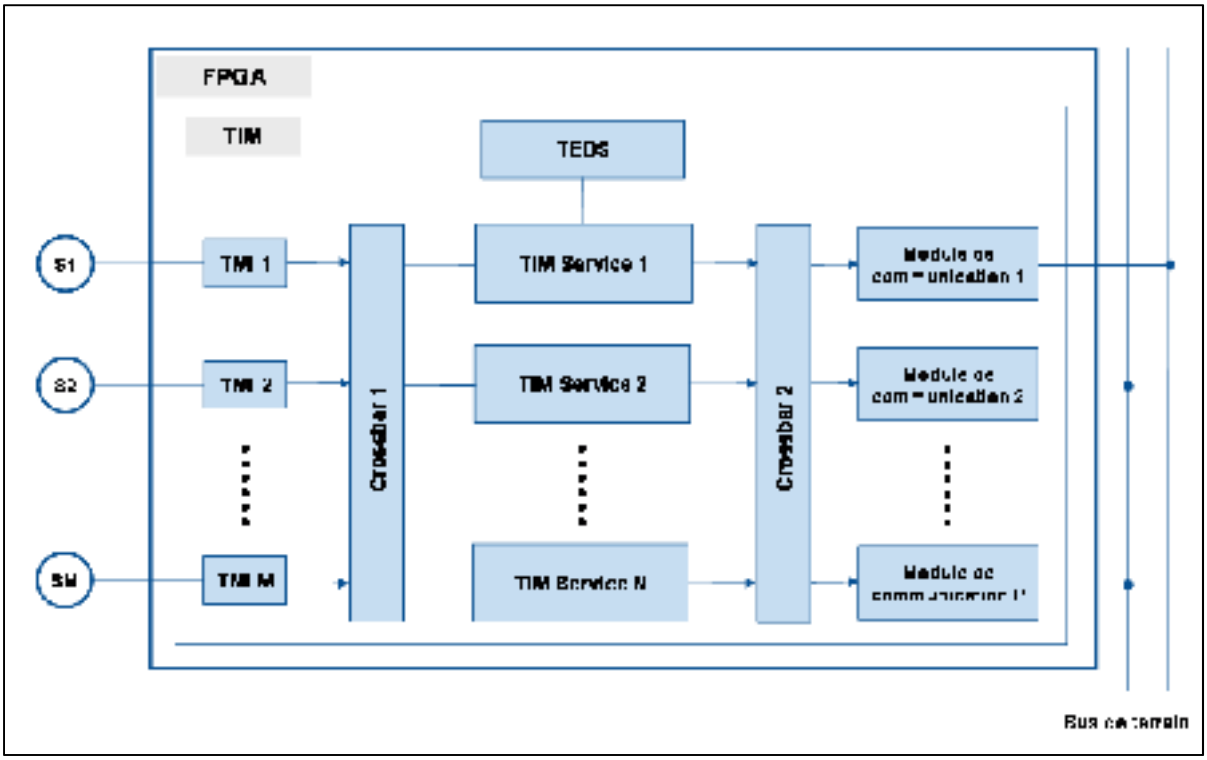

Figure 3.4 Architecture du TIM (première version)
Tirée de la documentation interne du projet AVIO 402

3.4.2 Deuxième version

Aux premiers stades du projet AVIO 402, les techniques de détection et de recouvrement de pannes n'étaient pas été encore définies. L'une de mes principales tâches depuis que j'ai rejoint le projet était d'investiguer la tolérance aux pannes du TIM et de proposer les techniques de détection de recouvrement de pannes les plus appropriées. En se basant sur les techniques de la redondance active, quelques modifications ont été apportées à la première version du TIM sans toucher à sa structure de base ni aux fonctions qu'il fournit (*Voir* figure 3.5). La deuxième version du TIM intègre plusieurs modules de service regroupés en paires redondantes P_i ($1 \leq i \leq N$) fonctionnant selon la technique de redondance avec comparaison. Ces paires sont désormais désignées par Paires de Service. Les modules de service de chacune des paires effectuent simultanément le traitement des échantillons et/ou des commandes. Un comparateur (Comp) permet de valider les résultats obtenus avant d'être transmis. Afin d'éviter les défaillances résultant d'effets de cause commune, les signaux à comparer sont légèrement retardés. Les paires sont complètement indépendantes et favorisent une défaillance sûre (Fail-Safe) en cas de pannes permanentes. Les bus internes ainsi que les TEDS sont protégés via des techniques de détection d'erreurs telles que des bits de parité ou des CRC.

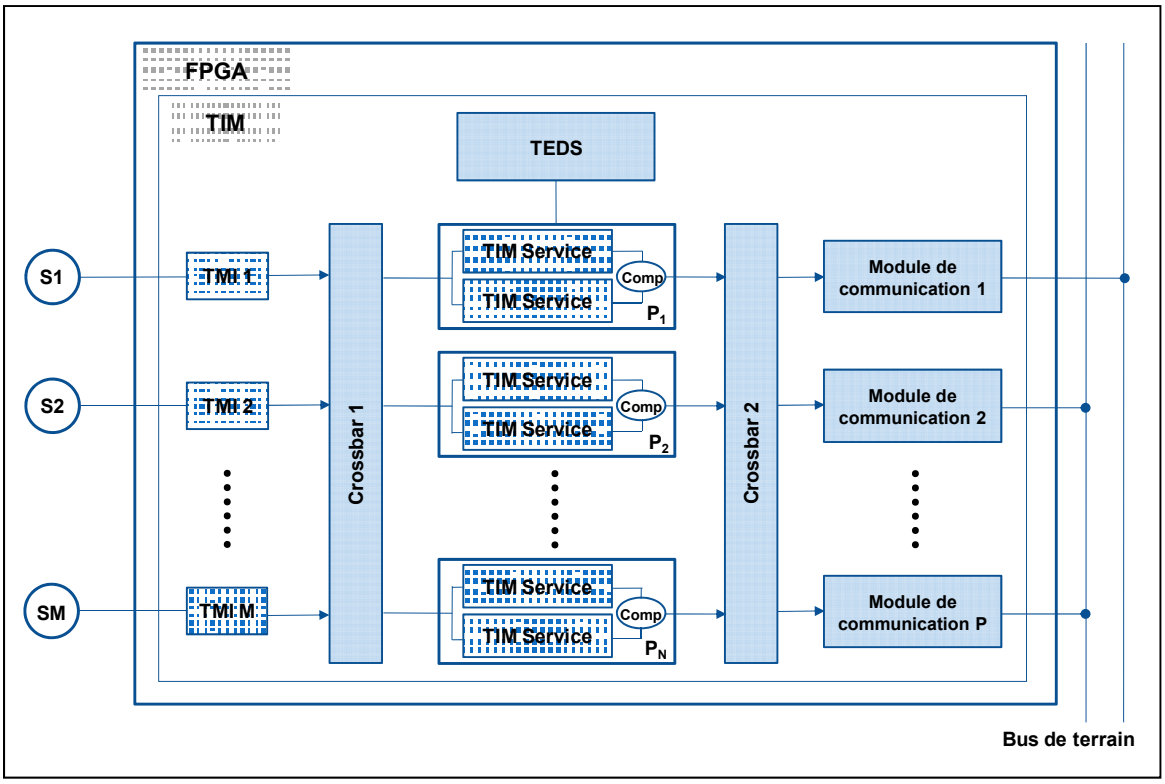

Figure 3.5 Architecture du TIM (deuxième version)

La deuxième version du TIM supporte deux types de configuration. Dans la première configuration, une seule paire de module de service traite les données. Les autres paires sont en veille et elles servent comme modules de rechange en cas de panne de la paire opérationnelle. La deuxième configuration est caractérisée par le fait que toutes les paires de service sont fonctionnelles. Celles-ci partagent les fonctions de traitement et chacune d'entre elles gère un ensemble de transducteurs. En cas de panne permanente de l'une des paires, deux scénarios sont possibles: 1) la paire fautive cesse de fonctionner et un mécanisme de recouvrement assigne les fonctions perdues aux autres paires, 2) la paire fautive cesse de fonctionner et aucun recouvrement n'est effectué. Dans ces deux scénarios, le TIM entre dans un mode de performance dégradée.

3.5 Étude comparative

Cette section dresse une comparaison, en termes de tolérances aux pannes, entre d'une part, les interfaces de transducteurs intelligents présentées dans chapitre1 et d'autre part, l'interface basée sur l'IEEE 1451.0 développée dans le cadre de ce mémoire.

3.5.1 Interface COM/MON

Nous utilisons ici l'interface présentée au premier chapitre dans la section 1.4.2 (*Voir* Figure 3.6). La tolérance aux pannes de l'interface est assurée par la technique COM/MON, une forme de redondance active.

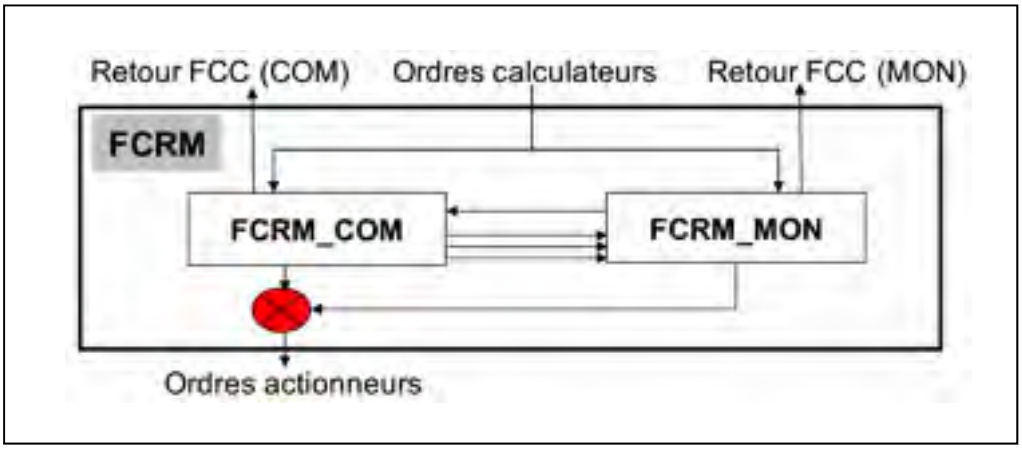

Figure 3.6 Architecture COM/MON
Tirée de Sghairi (2010, p.64)

En effet, une unité COM effectue le traitement des commandes/données alors qu'une autre unité MON surveille les tâches exécutées par l'unité COM. En cas de différence entre les sorties des deux unités, l'unité MON bloque la sortie de l'interface et signale l'existence d'une panne aux différents systèmes de l'avion. Ce mécanisme favorise une défaillance sûre en cas de panne de l'interface. La principale technique de tolérance aux pannes proposée dans cette interface est le traitement temporel double des commandes/données. Selon [3], les fonctions des unités COM et MON sont implémentées en logiciel sur des microcontrôleurs.

Ceux-ci n'incorporent aucun type de redondance modulaire matérielle et aucun recouvrement n'est possible en cas de panne permanente.

3.5.2 Interface à redondance temporelle

Il s'agit de l'interface présentée au premier chapitre dans la section 1.4.3 (*Voir* Figure 3.7) Dans cette architecture, une unité de calcul (microprocesseur) gère les transducteurs liés à l'interface. La principale technique de tolérance aux pannes proposée dans cette interface est le traitement temporel double des commandes/données.

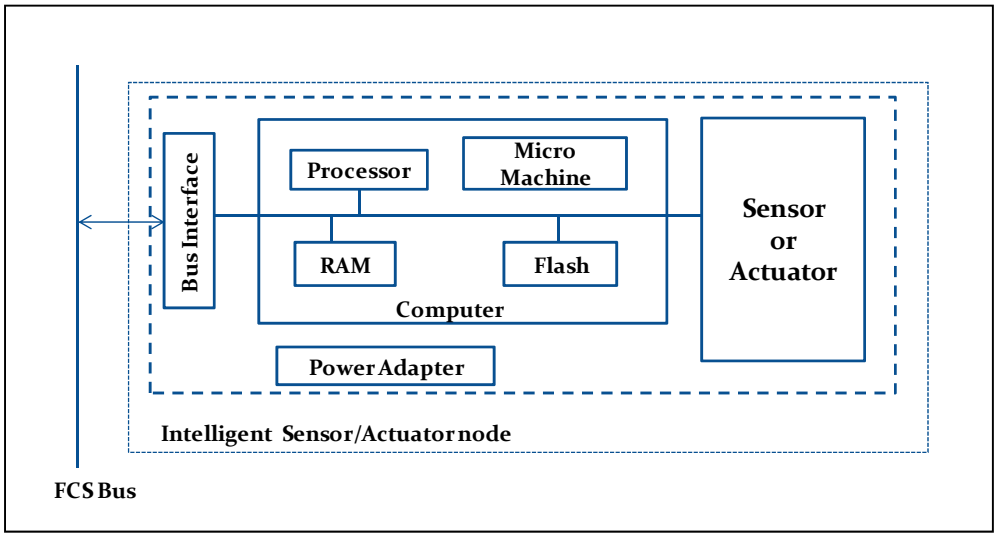

Figure 3.7 Architecture basée sur la redondance temporelle
Tirée de Forsberg (2003, p.50)

3.5.3 Tableau comparatif

Le tableau 3.2 dresse une comparaison entre les interfaces susmentionnées

Tableau 3.2 Tableau comparatif des interfaces

	Interface COM/MON	**Interface basée sur la redondance temporelle**	**Interface basée sur l'IEEE 1451**
Détection de pannes	- Exécution double de la tâche par les voies COM et MON et comparaison entre les résultats de calcul effectuée par l'unité MON	- Exécution double de la tâche par la même unité matérielle (microcontrôleur)	- Exécution double de la tâche par des modules redondants implémentés au niveau microarchitectural d'un FPGA
Avantages	- Détection des pannes permanentes et transitoires	- Détection des pannes transitoires	- Détection des pannes permanentes et Transitoires - Le recouvrement des pannes transitoires et permanentes est possible
Inconvénients	- L'unité matérielle supplémentaire augmente le taux de défaillances de l'interface - Aucun recouvrement n'est possible en cas de panne permanente	- Pannes permanentes non détectées -Aucun recouvrement n'est possible en cas de panne permanente	- Incapacité de tolérer certains types de pannes dans les ressources partagées (Ex : panne dans une source d'horloge ou d'alimentation)

3.6 Conclusion

Ce chapitre a expliqué, en premier lieu, le besoin d'incorporer des mécanismes de tolérance aux pannes dans les interfaces de transducteurs intelligents conçues pour des applications

avioniques critiques. La démarche qui a été suivie pour sélectionner les techniques de tolérance aux pannes les plus appropriées a été également discutée. Par ailleurs, le chapitre a présenté l'architecture du TIM qui a été adoptée. Finalement, le chapitre a dressé une comparaison, en termes de tolérance aux pannes, entre l'architecture du TIM et les interfaces présentées au premier chapitre. La fiabilité et la sûreté du TIM seront étudiées au quatrième chapitre.

CHAPITRE 4

ÉTUDE DE FIABILITÉ ET DE SÛRETÉ

4.1 Introduction

Ce chapitre dresse une analyse de fiabilité et de sûreté des interfaces de transducteur suivantes: 1) l'interface basée sur IEEE1451.0 présentée au troisième chapitre, 2) l'interface COM/MON. L'interface basée sur la redondance temporelle présentée dans le chapitre précédent n'est pas étudiée en raison de son incapacité de tolérer des pannes permanentes.

4.2 Définitions

- la fiabilité R(t) d'un système à l'instant t est la probabilité qu'il fonctionne sans défaillance dans l'intervalle de temps [0, t], sachant qu'il fonctionnait correctement à l'instant t=0 [27];
- la sûreté S(t) d'un système est la probabilité que, soit il effectue ses fonctions correctement ou il met fin à ses activités d'une manière sûre [27].

Du point de vue sûreté, il existe deux modes de défaillance : 1) défaillance sûre, 2) défaillance non-sûre.

4.3 Fiabilité et sûreté du TIM

Une étude préliminaire de la fiabilité du TIM a été présentée dans [40]. Dans ce travail, il a été supposé que toutes les erreurs sont détectées et manipulées correctement. En réalité, les pannes peuvent échapper aux mécanismes de détection d'erreurs implémentés. La présente section vise à perfectionner cette étude en introduisant une distinction entre les défaillances sûres et non-sûres. Afin d'investiguer le scénario le plus pessimiste, l'analyse suppose que la première panne (transitoire ou permanente) survenant dans le TIM est une panne permanente.

Ceci permet de déterminer et d'évaluer les bornes inférieures de la sûreté et de la fiabilité de l'interface. L'étude repose sur la modélisation avec les chaines de Markov. Celles-ci constituent un outil mathématique robuste couramment utilisé dans les évaluations de fiabilité et de sûreté des systèmes avioniques [4][41]. Nous définissons la couverture de détection et de recouvrement de pannes permanentes comme la probabilité conditionnelle C = P (panne détectée et recouverte | panne survenue).

Les paramètres suivants sont utilisés pour la modélisation du système:

- λ_{core} : le taux de défaillance d'une paire ;
- λ_{com} : le taux de défaillance des ressources partagées dans le TIM: TEDS, crossbar, etc. ;
- λ_{chip} : le taux de défaillance résultant d'effets de cause commune, telles que l'alimentation électrique, l'arbre d'horloge et le substrat de silicium ;
- C_{core} : la couverture de pannes des paires ;
- C_{com} : la couverture de pannes des ressources partagées.

Nous supposons que les pannes permanentes résultant d'effets de cause commune ne sont pas détectables par le TIM et qu'il n'est pas possible de faire un recouvrement suite à une panne dans les ressources partagées. Seules les pannes au niveau des paires de service pourront être recouvertes.

4.3.1 Premier scénario

Dans ce scénario, le TIM dispose de *N* paires de service qui fonctionnent en parallèle (*Voir* Figure 3.5). Chaque paire traite les données d'un certain nombre de transducteurs. En cas de panne permanente de l'une des paires, les fonctions anciennement exécutées par la paire fautive sont assurées par le reste des paires. La Figure 4.1 montre le diagramme de transition d'un TIM avec *N* paires de service. Dans ce scénario, nous supposons que le taux de défaillance des paires de service reste inchangé pour toutes les transitions du système. Nous supposons aussi qu'à un instant *t*, une seule panne pourra se produire.

La probabilité que le système se trouve dans l'un des *N+2* états est donnée par la résolution de l'équation de transition d'états suivante:

$$\frac{d}{dt}P(t) = M * P(t) \tag{4.1}$$

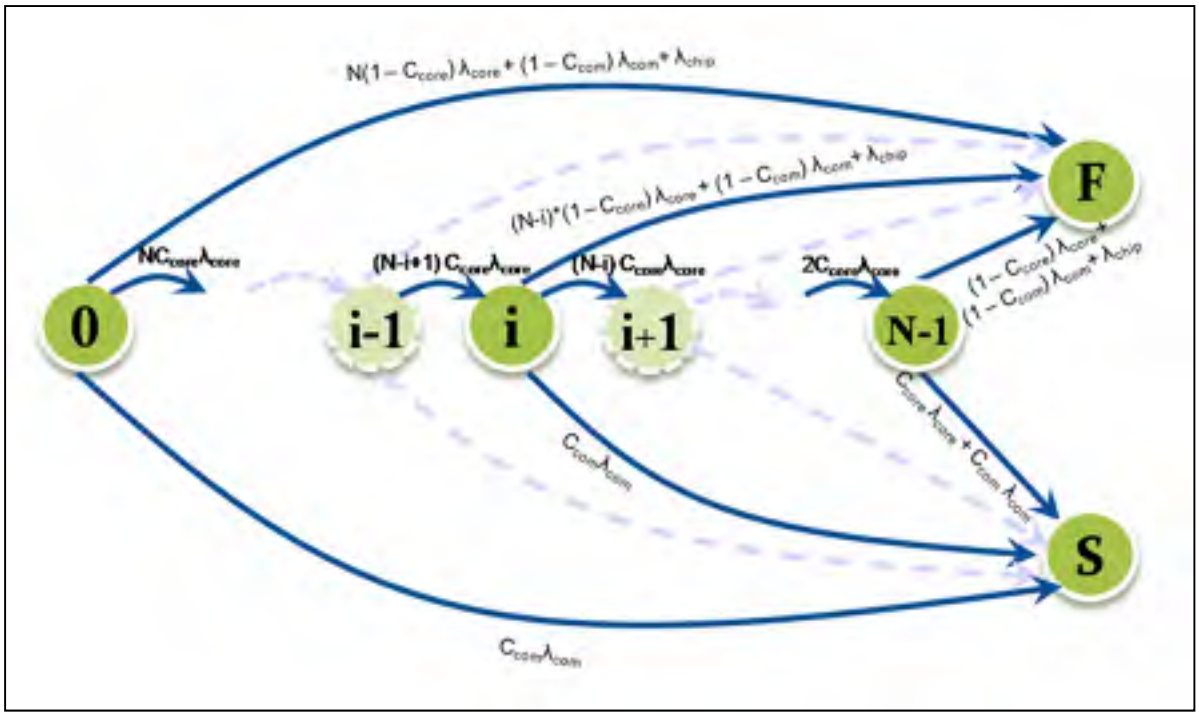

Figure 4.1 Diagramme de transition du TIM (premier scénario)

Où $P(t)$ est un vecteur de taille *N+2* dont le $i^{\text{ème}}$ élément représente la probabilité $P_i(t)$ que le système se trouve dans l'état *i* à l'instant *t*. *M* est la matrice de transition associée au système.

Pour *N=2*, la matrice de transition *M* est comme suit:

$$M = \begin{pmatrix} -(2\lambda_{core} + \lambda_{com} + \lambda_{chip}) & -(\lambda_{core} + \lambda_{com} + \lambda_{chip}) & 0 & 0 \\ 2C_{core}\lambda_{core} & 0 & 0 & 0 \\ C_{com}\lambda_{com} & C_{core}\lambda_{core} + C_{com}\lambda_{com} & 0 & 0 \\ 2(1 - C_{core})\lambda_{core} + (1 - C_{com})\lambda_{com} + \lambda_{chip} & (1 - C_{core})\lambda_{core} + (1 - C_{com})\lambda_{com} + \lambda_{chip} & 0 & 0 \end{pmatrix}$$

Le diagramme de transition représenté par la Figure 4.1 présente *N+2* états :

- état 0 : *N* paires fonctionnelles ;
- état *i* : *N-i* paires fonctionnelles ;
- état *N-1* : une seule paire fonctionnelle ;
- état *S* : état de défaillance sûre ;
- état *F* : état de défaillance non-sûre.

La transition entre l'état *i* et l'état *i+1* ($0 \leq \mathrm{i} \leq N-2$) dépend :

- du taux de défaillance d'une paire $(\mathrm{N}-\mathrm{i})\lambda_{core}$, et de la probabilité C_{core} qu'étant donnée l'existence d'une panne, le système parvient à la détecter et à faire un recouvrement.

La transition entre l'état *i* et l'état *S* dépend :

- du taux de défaillance des ressources partagées λ_{com}, et de la probabilité C_{com} qu'étant donnée l'existence d'une panne, le système parvient à la détecter.

La transition entre l'état *N-1* et l'état *S* dépend :

- du taux de défaillance d'une paire de service λ_{core} et de la probabilité C_{core} qu'étant donnée l'existence d'une panne, le système parvient à la détecter, et ;
- du taux de défaillance des ressources partagées λ_{com} et de la probabilité C_{com} qu'étant donnée l'existence d'une panne, le système parvient à la détecter.

La transition entre l'état i ($0 \leq \mathrm{i} \leq N-1$) et l'état *F* dépend :

- du taux de défaillance d'une paire de service $(\mathrm{N}-\mathrm{i})\lambda_{core}$ et de la probabilité$(1-C_{core})$ qu'étant donnée l'existence d'une panne, le système ne parvient pas à la détecter ;
- du taux de défaillance des ressources partagées λ_{com} et de la probabilité $(1-C_{com})$ qu'étant donnée l'existence d'une panne, le système ne parvient pas à la détecter, et ;
- du taux de défaillance résultant d'effets de cause commune λ_{chip}.

La fiabilité du TIM est donnée par la probabilité qu'il se trouve dans l'un des états *i* ($0 \leq \mathrm{i} \leq N-1$). La sûreté du TIM est donnée par la probabilité que le système se trouve dans un état *i* ($0 \leq \mathrm{i} \leq N-1$) ou dans l'état *S*.

Le système d'équations différentielles (4.1) a été résolu avec MATLAB. Les expressions analytiques exactes de fiabilité et de sûreté ont été élaborées et évaluées sur l'intervalle de temps [0, 200000 heures de vol]. Cet intervalle représente la durée de vie typique des équipements avioniques [29][41]. Les taux de défaillances et les probabilités de couverture utilisées sont donnés par le Tableau 4.1. Ces valeurs ont été fixées en se basant sur des travaux antérieurs [4] [25] [41].

Tableau 4.1 Les taux de défaillance et les probabilités de couverture utilisés;
hv = heure de vol

Paramètre	**Valeur**
λ_{core}	10^{-6}/ hv
λ_{com}	10^{-6}/ hv
λ_{chip}	10^{-6}/ hv
C_{core}	0.999
C_{com}	0.999

Résultats numériques

L'analyse de fiabilité du TIM (*Voir* Figure 4.2) montre qu'en incorporant des paires de service redondantes, la fiabilité du TIM augmente. L'introduction d'une deuxième paire de service améliore considérablement la fiabilité du TIM. Toutefois, l'amélioration due à l'ajout d'un nombre de paires supérieure à 2, est marginale. La tendance montre une stagnation de la fiabilité avec l'ajout d'un nombre important de paires.

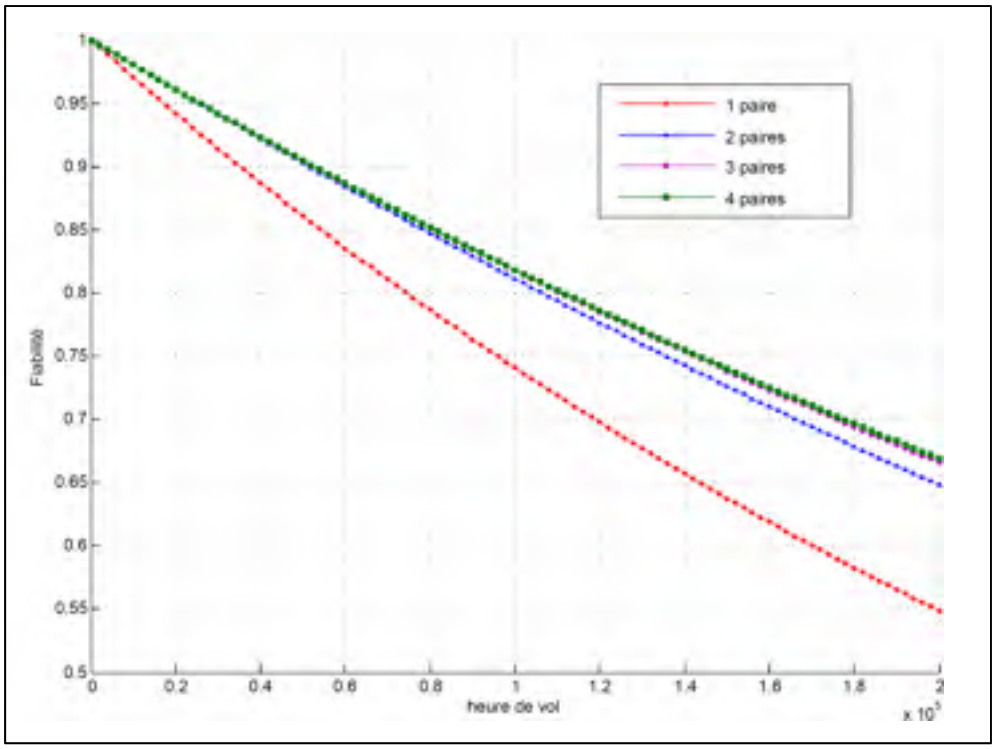

Figure 4.2 Fiabilité du TIM (premier scénario)

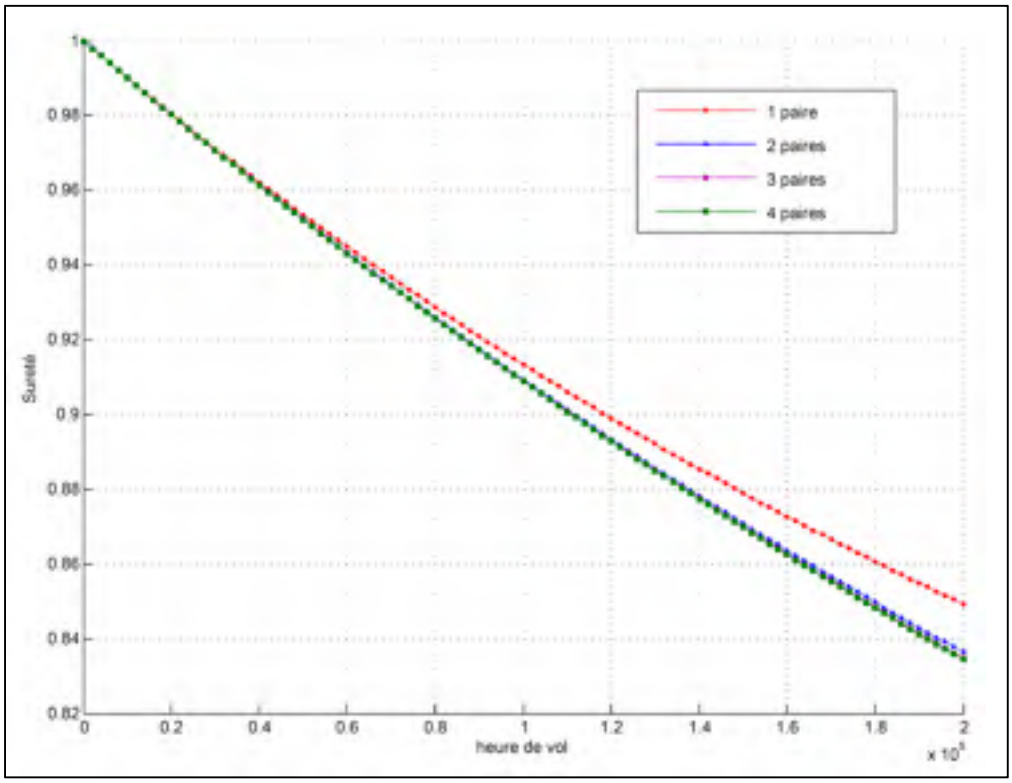

Figure 4.3 Sûreté du TIM (premier scénario)

La Figure 4.3 montre que les courbes de sûreté du TIM à 2, 3 et 4 paires sont pratiquement confondues. La sûreté du TIM à une seule paire est meilleure que celle d'un TIM à plusieurs paires sur l'intervalle [50000, 200000 hv].

Dans l'étude élaborée précédemment, il a été supposé que le taux de défaillance des paires ne dépend pas de la charge de calcul qui leur est affectée. Cependant, dans la réalité, le taux de défaillance d'une paire est influencé par sa charge de calcul. En cas de panne d'une paire, le TIM attribue plus de tâches aux paires survivantes, ce qui peut augmenter leur taux de défaillance [42]. Par ailleurs, il a été démontré que, dans un contexte de processeur multi-cœurs avec recouvrement de pannes permanentes, l'ampleur de l'augmentation du taux de défaillance d'un cœur est beaucoup plus faible que celle de l'augmentation de sa charge de traitement [42, p.43].

Nous considérons maintenant un TIM avec deux paires. Celles-ci sont initialement fonctionnelles et disposent de taux de défaillance égaux $\lambda_{core,1}$. Si une panne se produit dans l'une des paires, la paire restante prend la relève. Dans ce cas, la charge de la paire fonctionnelle est doublée et son taux de défaillance devient $\lambda_{core,2}$. Par analogie avec [42], nous supposons que $\lambda_{core,1} \leq \lambda_{core,2} \leq 2 * \lambda_{core,1}$.
La Figure 4.4 trace les courbes de fiabilité du TIM avec deux paires pour plusieurs valeurs de taux de défaillance $\lambda_{core,2}$.

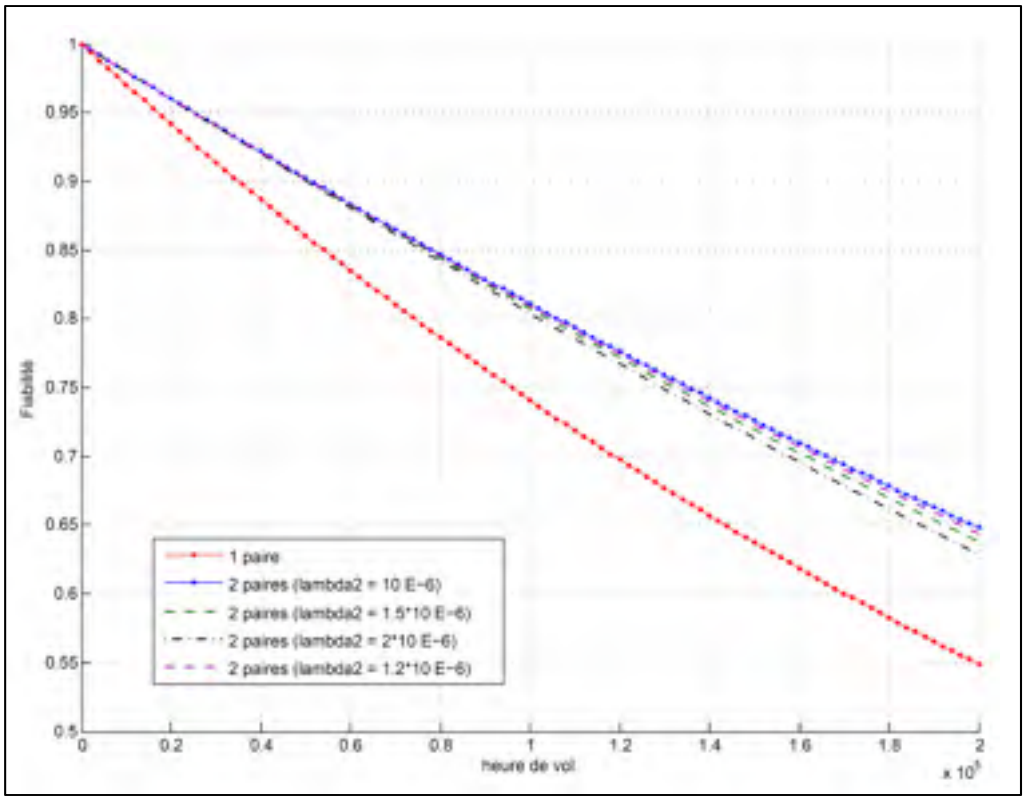

Figure 4.4 Fiabilité du TIM pour plusieurs valeurs de $\lambda_{core,2}$

D'après la Figure 4.4, nous pouvons constater qu'en augmentant le taux de défaillance $\lambda_{core,2}$, la fiabilité du TIM se détériore. Toutefois, cette détérioration est négligeable. Par ailleurs, les courbes de sûreté pour $\lambda_{core,2} = \lambda_{core,1}$, $\lambda_{core,2} = 1.2 * \lambda_{core,1}$, $\lambda_{core,2} = 1.5 * \lambda_{core,1}$, $\lambda_{core,2} = 2 * \lambda_{core,1}$ sont pratiquement confondues (*Voir* Figure 4.5) .

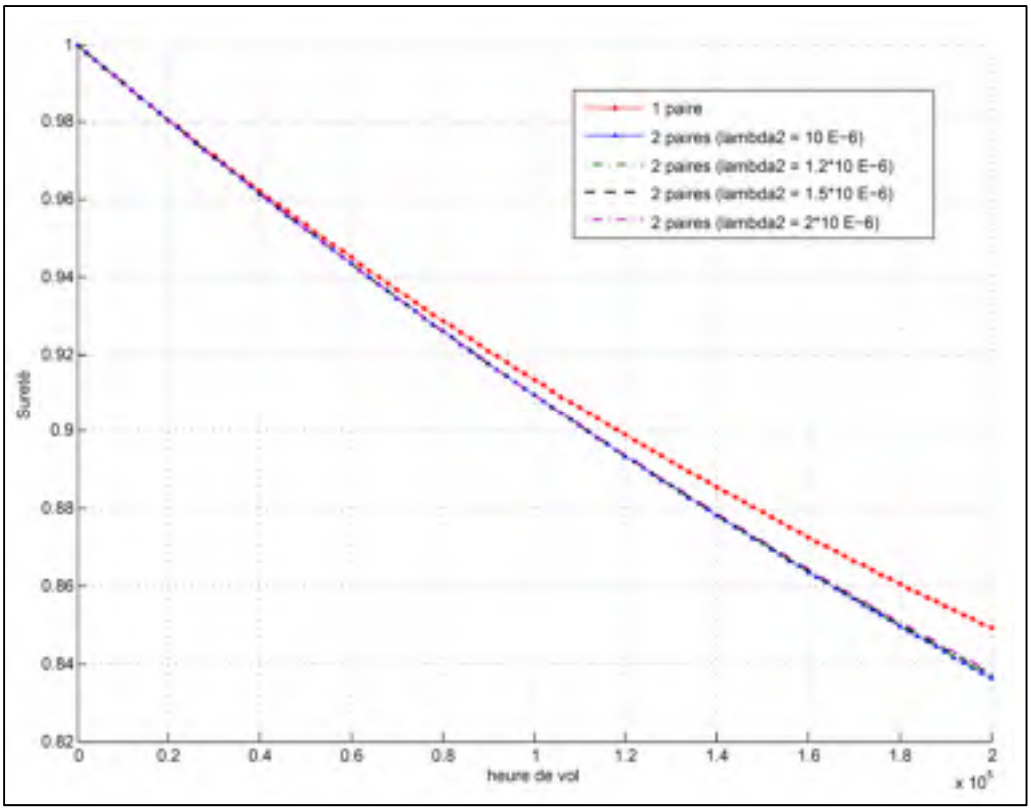

Figure 4.5 Sûreté du TIM pour plusieurs valeurs de $\lambda_{core,2}$

4.3.2 Deuxième scénario

Dans ce scénario, une seule paire est fonctionnelle et les N-1 paires restantes sont des paires de rechange. La transition d'un état à un autre dépend de l'apparition et de la détection d'une panne dans la paire en cours d'utilisation. Ce scénario est idéaliste étant donné que la paire en cours d'utilisation est considérée comme la seule paire susceptible de tomber en panne. En réalité les paires de rechange peuvent tomber en pannes et ne plus être disponible au moment de recouvrement d'une panne de la paire en cours d'utilisation. Le diagramme de transition du deuxième scénario est semblable à celui du premier scénario sauf que la transition de l'état i à l'état $i+1$ ($0 \leq i \leq N-1$) dépend uniquement du terme $C_{core}\lambda_{core}$ (*Voir* Figure 4.6).

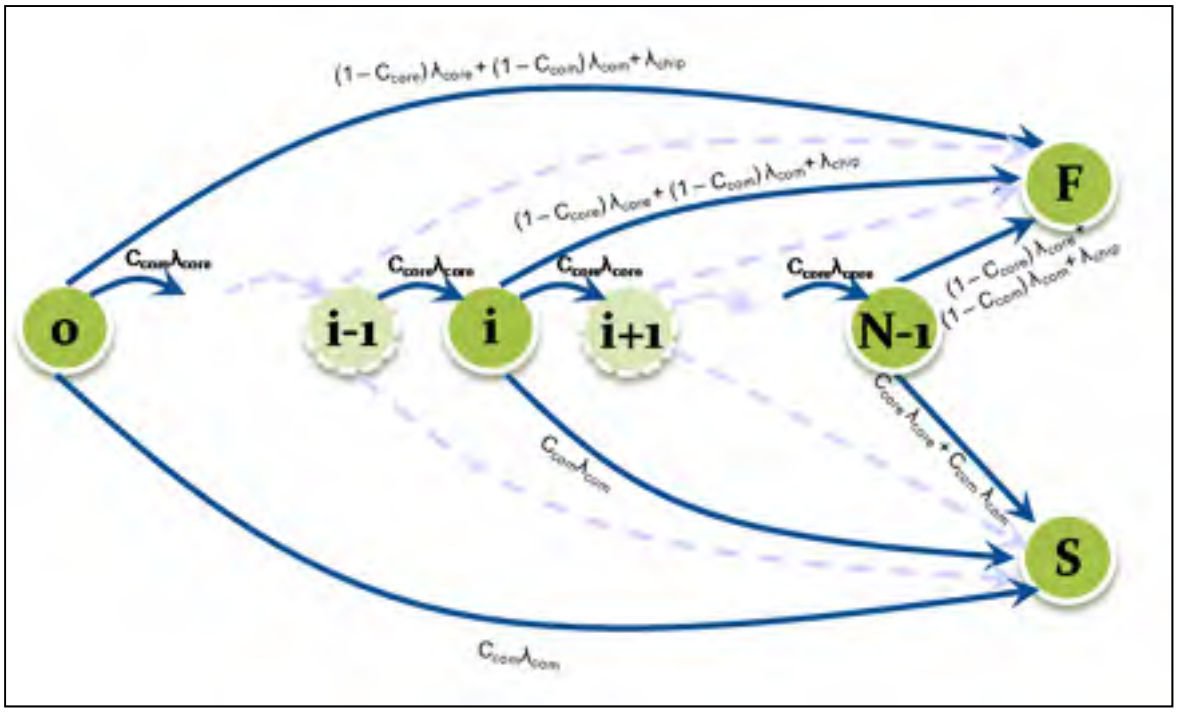

Figure 4.6 Diagramme de transition du TIM (Deuxième scénario)

La matrice de transition du système pour *N=2* est comme suit :

$$M = \begin{pmatrix} -(\lambda_{core} + \lambda_{com} + \lambda_{chip}) & -(\lambda_{core} + \lambda_{com} + \lambda_{chip}) & 0 & 0 \\ C_{core}\lambda_{core} & C_{core}\lambda_{core} & 0 & 0 \\ C_{com}\lambda_{com} & C_{com}\lambda_{com} & 0 & 0 \\ (1 - C_{core})\lambda_{core} + (1 - C_{com})\lambda_{com} + \lambda_{chip} & (1 - C_{core})\lambda_{core} + (1 - C_{com})\lambda_{com} + \lambda_{chip} & 0 & 0 \end{pmatrix}$$

Résultats numériques

La Figure 4.7 montre que, tout comme dans le premier scénario, l'incorporation d'une deuxième paire améliore significativement la fiabilité du TIM (courbe bleue dans la Figure 4.7). Il est aussi remarquable que l'amélioration de la fiabilité due à l'utilisation d'un nombre de paires supérieur à 2, est négligeable. Du point de vue sûreté, les courbes de sûreté des différentes configurations du TIM présentent un comportement comparable à celles du premier scénario (*Voir* Figure 4.8).

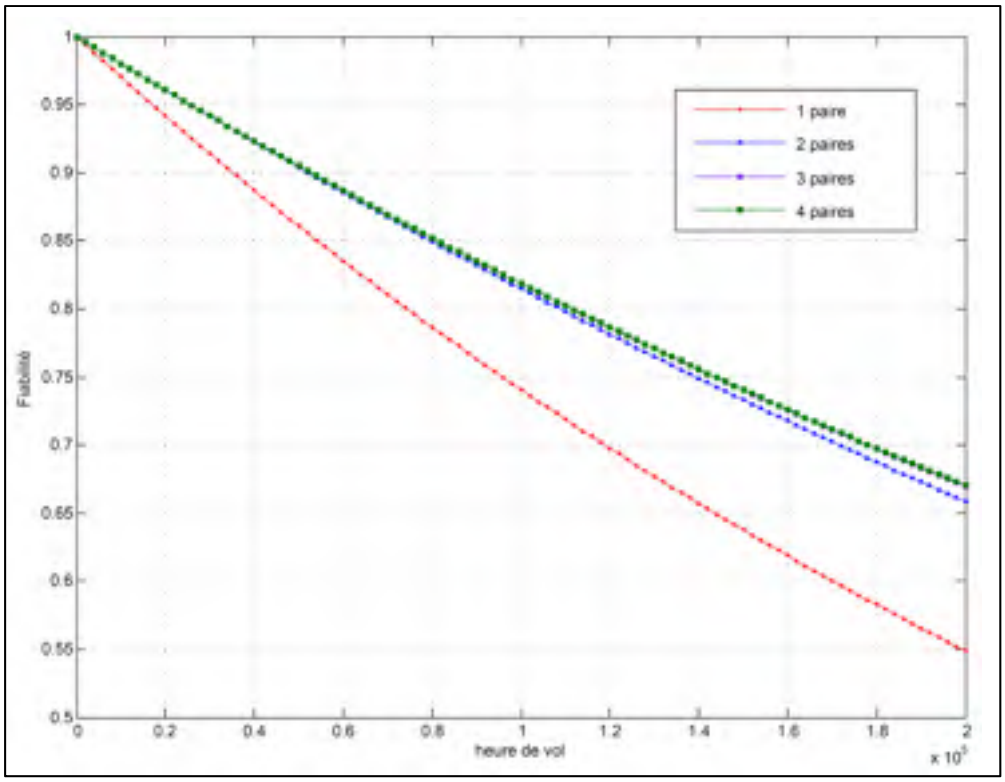

Figure 4.7 Fiabilité du TIM (deuxième scénario)

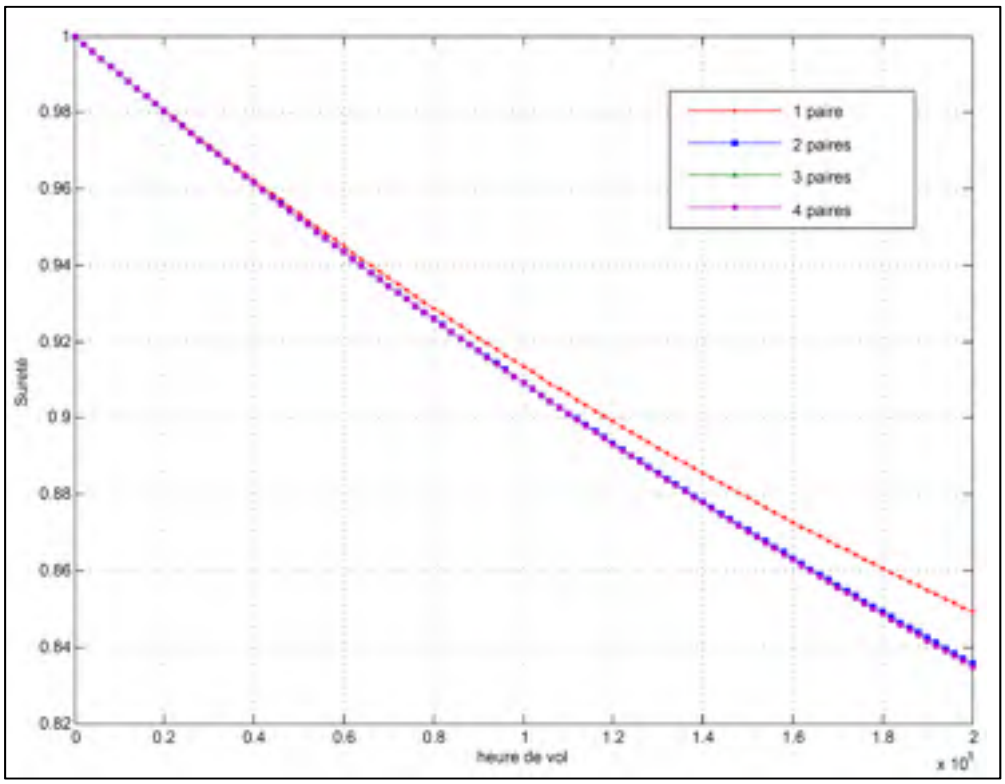

Figure 4.8 Sûreté du TIM (deuxième scénario)

4.3.3 Comparaison entre scénario 1 et scénario 2

La Figure 4.9 dresse une comparaison entre les deux scénarios en termes de fiabilité. Les deux scénarios présentent des probabilités de fiabilité très comparables. Un léger avantage est remarquable pour le scénario 2 sur l'intervalle [100000, 200000 hv]. La Figure 4.10 montre que les deux scénarios présentent les mêmes performances en termes de sûreté.

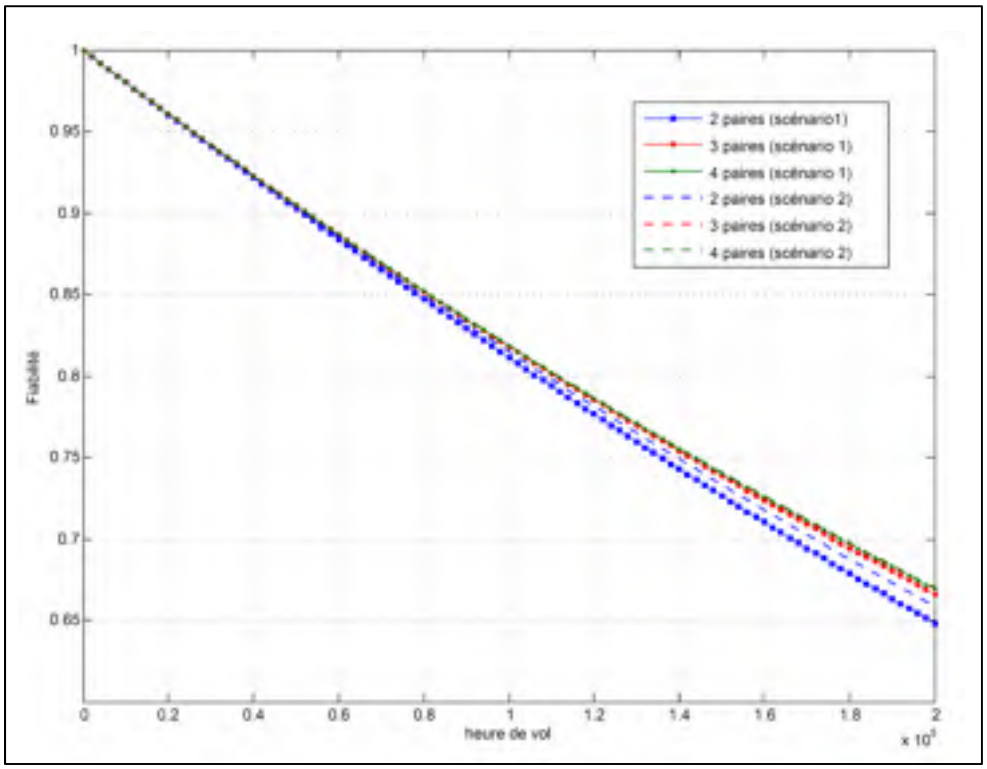

Figure 4.9 Comparaison entre le premier scénario et
le deuxième scénario en termes de fiabilité

En se basant sur ces résultats, le premier scénario semble être beaucoup plus avantageux pour l'implémentation du TIM vu qu'il permet d'améliorer significativement les performances de calcul du TIM tout en achevant des niveaux de fiabilité et de sûreté très similaires à ceux du deuxième scénario. Il semble aussi qu'en ayant deux paires, le TIM représenterait un compromis intéressant entre performance de calcul, fiabilité et sûreté.

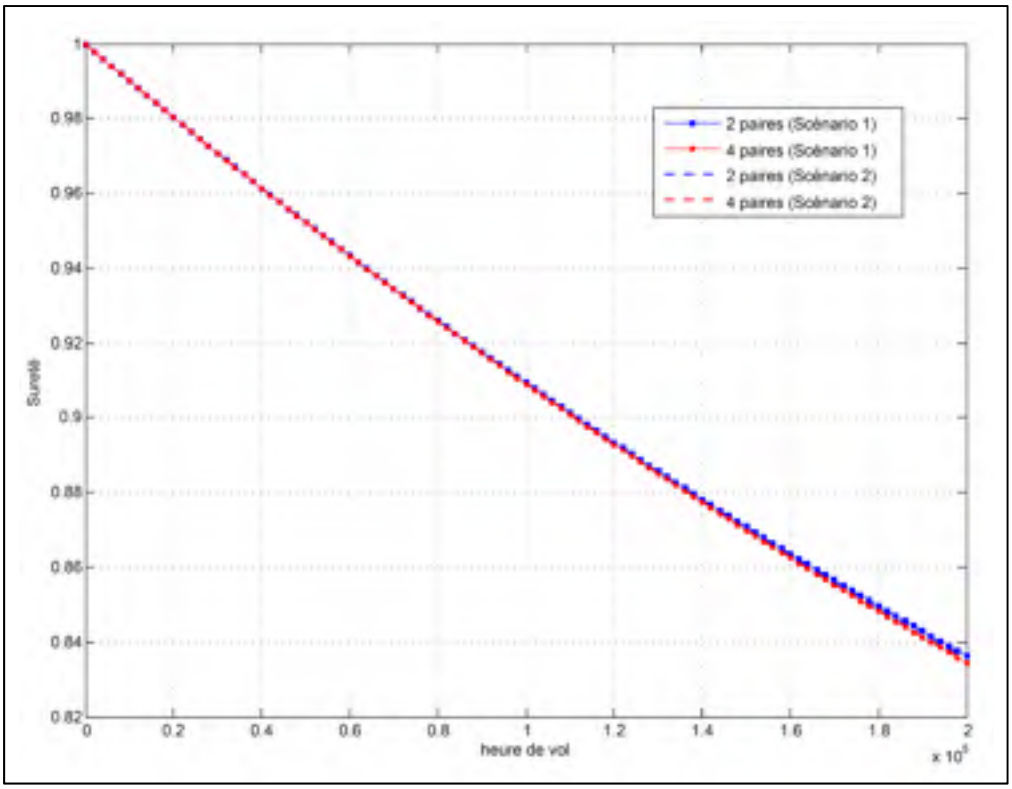

Figure 4.10 Comparaison entre le premier scénario et
le deuxième scénario en termes de sûreté

Impact de la couverture des pannes

Les figures 4.11 et 4.12 présentent l'impact de la couverture des pannes sur la fiabilité et la sûreté du TIM. Afin d'étudier cet impact nous supposons que les coefficients de couverture C_{core} et C_{com} sont égaux, c-à-d. $C_{core} = C_{com} = C$. Les courbes obtenues sont relatif à un TIM avec deux paires pour plusieurs valeurs de couverture C (0.9, 0.99, 0.999, 0.9999). Les figures 4.11 et 4.12 montrent qu'en augmentant la couverture des pannes, la fiabilité et la sûreté du TIM augmentent. En faisant passer la couverture de 0.99 à 0.999 et à 0.9999, la fiabilité s'améliore d'une manière marginale. Une augmentation semblable de la couverture entraîne une légère amélioration de la sûreté de l'interface. L'amélioration de la couverture des pannes pourrait induire des coûts de développement supplémentaires. Ceci étant dit, il est nécessaire d'établir un compromis entre les niveaux de fiabilité et de sûreté visés et le coût de développement.

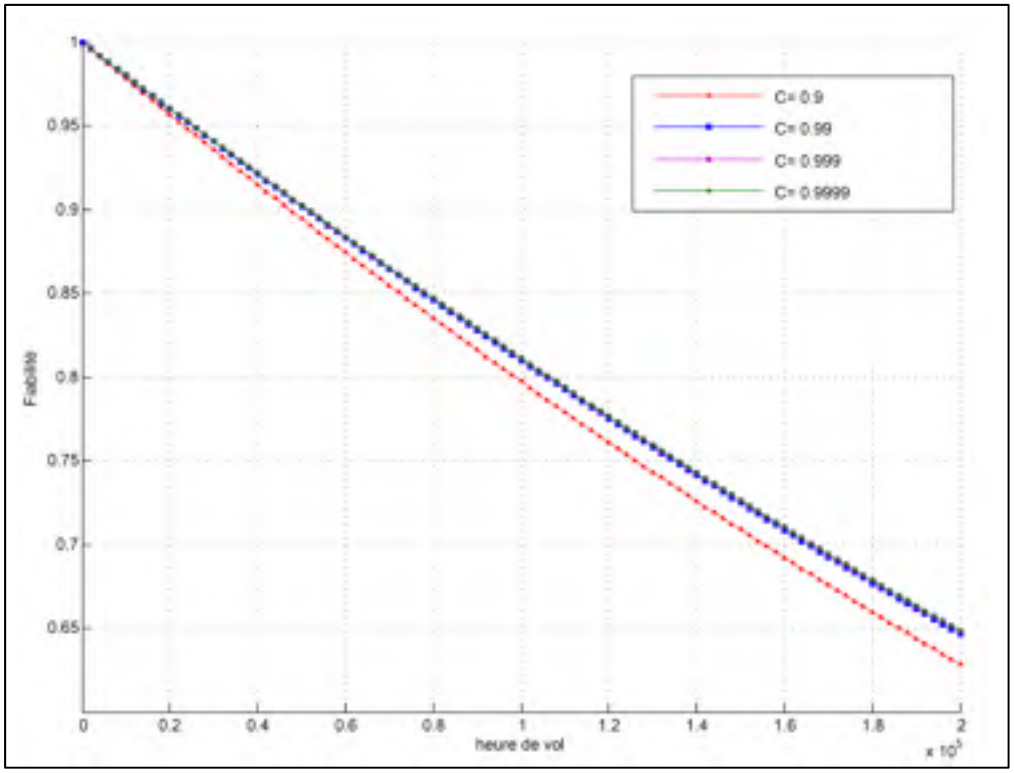

Figure 4.11 Impact de la couverture des pannes sur la fiabilité

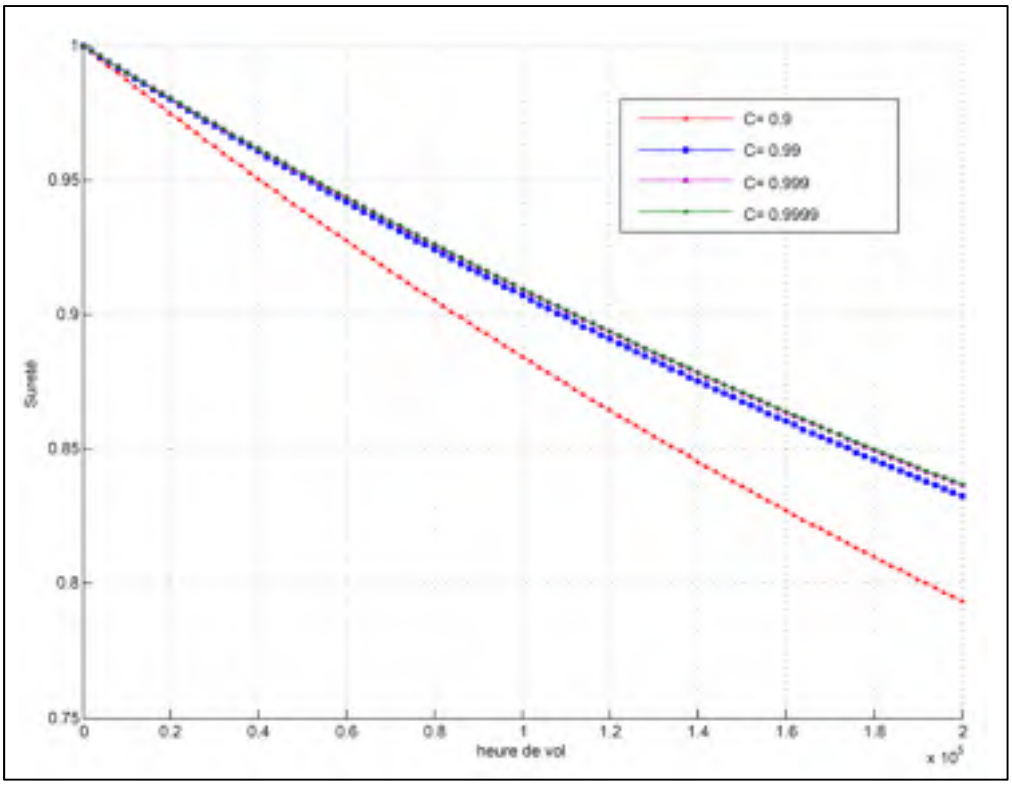

Figure 4.12 Impact de la couverture des pannes sur la sûreté

4.4 Étude de fiabilité et de sûreté de l'interface COM/MON

La Figure 4.13 montre le diagramme de transition de l'architecture COM/MON. Comme il a été mentionné au chapitre 3, la détection de pannes dans ce type d'interface est basée sur la comparaison des sorties des deux voies COM et MON. Pour que l'analyse de l'interface COM/MON soit effectuée sur les mêmes bases que celles de l'interface proposée, nous supposons que les voies COM et MON soient implémentées en matériel (FPGA) selon la norme IEEE 1451.0. De plus, nous supposons que les deux voies disposent d'une configuration simplex (*Voir* Figure 4.14).

Soit C la couverture des pannes de l'architecture COM/MON.

Le taux de défaillance de la logique dans chacune des voies est évalué comme suit :

$$\lambda = \lambda_{core} + \lambda_{com}$$

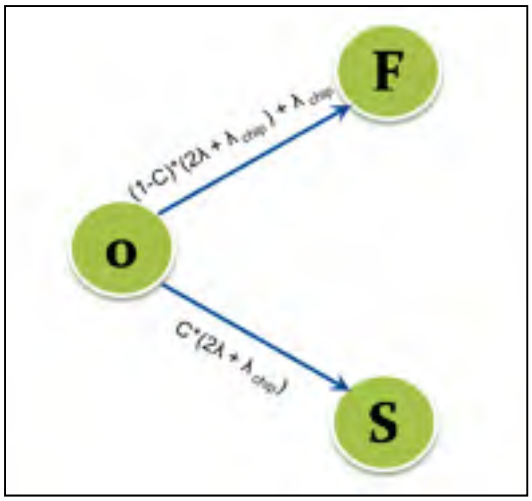

Figure 4.13 Diagramme de transition de l'architecture COM/MON

La transition entre l'état 0 et l'état S dépend :

- du taux de défaillance $2\lambda + \lambda_{chip}$ et de la probabilité C qu'étant donnée l'existence d'une panne, le système parvient à la détecter.

La transition entre l'état 0 et l'état F dépend :

- du taux de défaillance $2\lambda + \lambda_{chip}$ et de la probabilité $(1 - C)$ qu'étant donnée l'existence d'une panne, le système ne parvient pas à la détecter ;
- du taux de défaillance λ_{chip} associé à la voie MON.

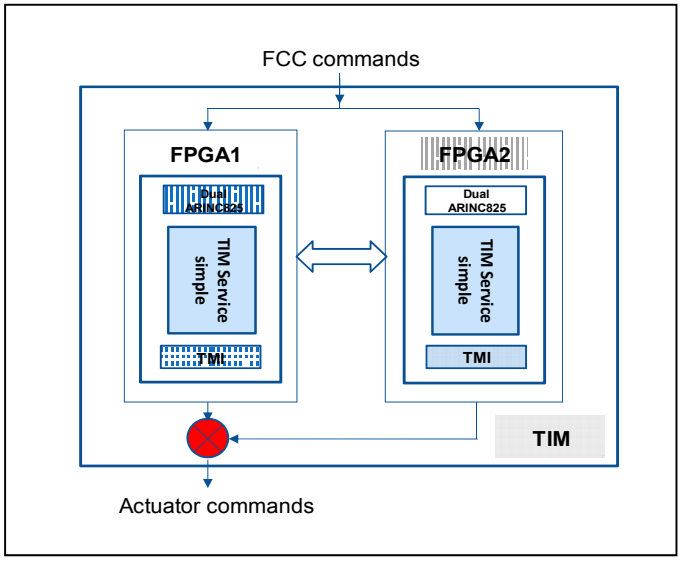

Figure 4.14 Interface COM/MON basée sur IEEE1451.0 (configuration simplex)

La figure 4.15 présente une comparaison entre l'interface COM/MON et l'interface proposée dans le cadre de ce mémoire en termes de fiabilité. Celle-ci montre qu'une configuration redondante de notre interface (deux paires ou plus) présente un avantage intéressant par rapport à l'interface COM/MON. Ces performances sont dues à l'augmentation du taux de défaillance causée par l'utilisation de deux FPGA en série dans le cas de l'interface COM/MON. Par ailleurs, la figure 4.16 montre que l'interface COM/MON est l'interface la plus sûre. L'écart entre la courbe de sûreté de l'interface COM/MON et celles des différentes configurations du TIM s'accentue graduellement sur l'intervalle [50000, 200000 hv].

Le choix de l'architecture de l'interface à utiliser dépend des exigences et de la nature de l'application avionique. Pour un système CDVE, la sûreté des interfaces d'actuateurs est extrêmement importante. L'architecture COM/MON constitue un bon candidat pour concevoir des interfaces d'actuateurs intelligents très sûres. Selon les résultats obtenus, il semble que l'architecture COM/MON pourrait être la plus appropriée pour les interfaces d'actuateurs des systèmes CDVE. En effet, les interfaces d'actionneurs intelligents constituent le dernier niveau du système qui est responsable de l'actionnement des surfaces de contrôle. De plus, si une panne ne peut être détectée localement, il est complexe de remettre l'interface à un état sûr.

D'autre part, les résultats obtenus montrent que l'interface proposée dans le cadre de ce mémoire est capable d'assurer des niveaux de sûreté et de fiabilité satisfaisants. Notre interface semble être plus avantageuse pour le développement d'interfaces de capteurs intelligents. En fait, dans le pire des scénarios caractérisé par une défaillance non sûre de l'interface, les pannes pourraient encore être détectés par les niveaux supérieurs du système, (c-à-d. NCAP, FCC, etc ..) et les données pourraient être extraites des interfaces de capteurs redondantes.

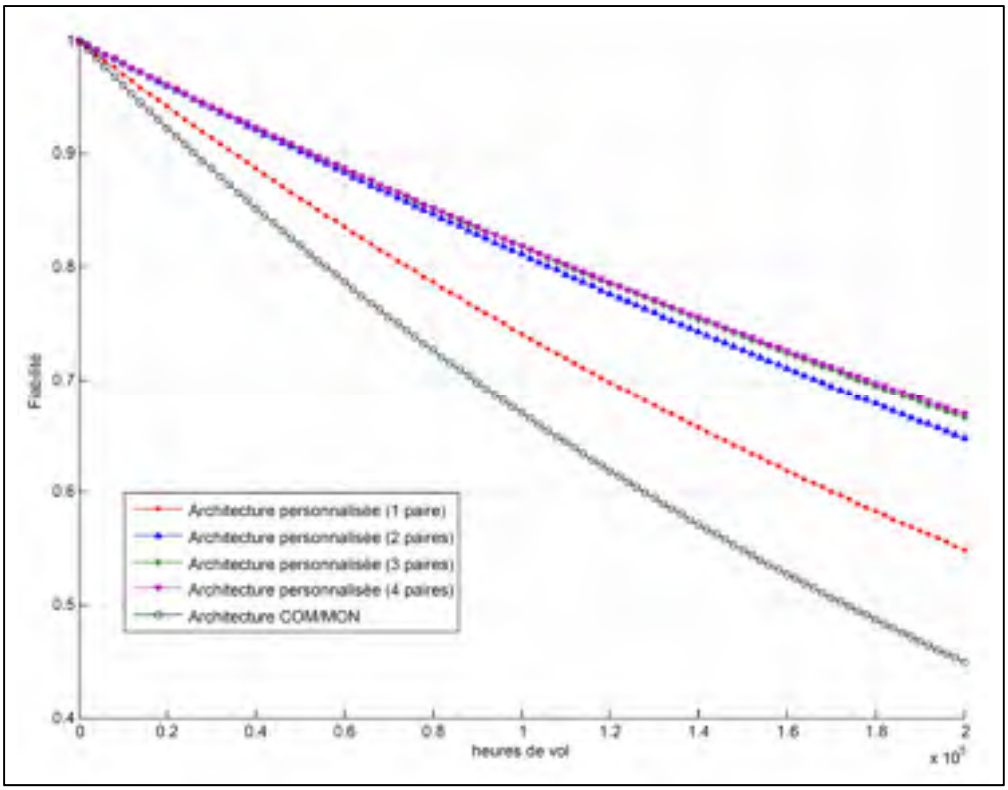

Figure 4.15 Fiabilité de l'interface COM/MON

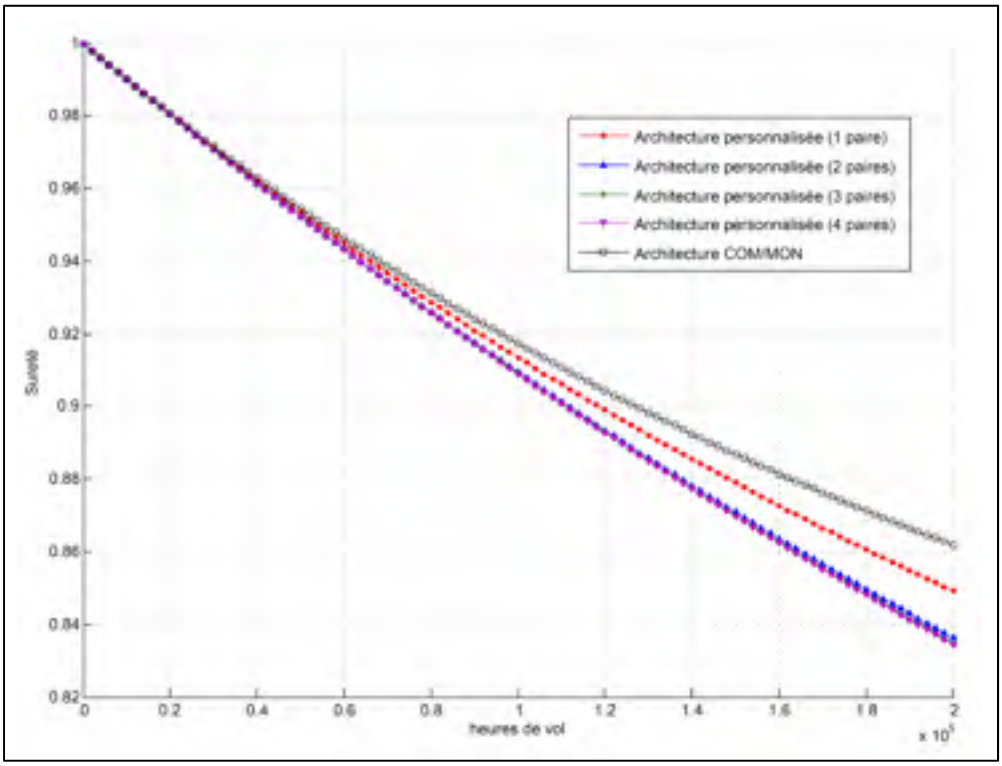

Figure 4.16 Sûreté de l'interface COM/MON

4.5 Application: interfaces de capteurs du système CDVE dans le cockpit

Cette section traite l'exemple de TIM utilisés pour interfacer les capteurs du cockpit mesurant les intentions des pilotes dans un aéronef. Compte tenu de la criticité de cette application, la défaillance du TIM est considérée comme hasardeuse (Hazardous). Selon les niveaux de criticité définies par la norme DO-254, la probabilité de défaillance maximale par heure de vol devrait être inférieure à 10^{-7}. Pour une durée de vol moyenne de 10 heures, la probabilité de défaillance d'un TIM avec deux paires de service est $2 * 10^{-5}$ (*Voir* Tableau 4.2), qui est largement supérieure à 10^{-7}. Afin de satisfaire le niveau de fiabilité imposé par DO-254, la redondance matérielle au niveau système est nécessaire. Dans une configuration duplex du système, où deux TIM sont installés pour interfacer un ensemble de capteurs, la probabilité de défaillance du système est égale à $(2 * 10^{-5}) * (2 * 10^{-5}) = 4 * 10^{-10}$, qui est largement inférieure à 10^{-7}. Une telle configuration permet de satisfaire le niveau de fiabilité requis pour le TIM.

Tableau 4.2 Probabilité de défaillance du TIM en fonction du nombre de paires de service

Nombre de paires	Probabilité de défaillance sûre	Probabilité de défaillance non-sûre	Probabilité de défaillance totale
1	2.0000e-05	9.9999e-06	3.0000e-05
2	1.0000e-05	9.9999e-06	2.0000e-05
3	9.9999e-06	9.9999e-06	2.0000e-05
4	9.9999e-06	9.9999e-06	2.0000e-05

Les figures 4.17 et 4.18 présentent deux scénarios de configuration d'un système CDVE utilisant des TIM redondants pour interfacer les capteurs du cockpit. La différence entre les deux configurations réside dans le nombre de capteurs, le nombre de TIM utilisés et le type des capteurs connectés à l'interface. Ceci permet de mettre en relief la généricité de l'architecture du TIM proposée.

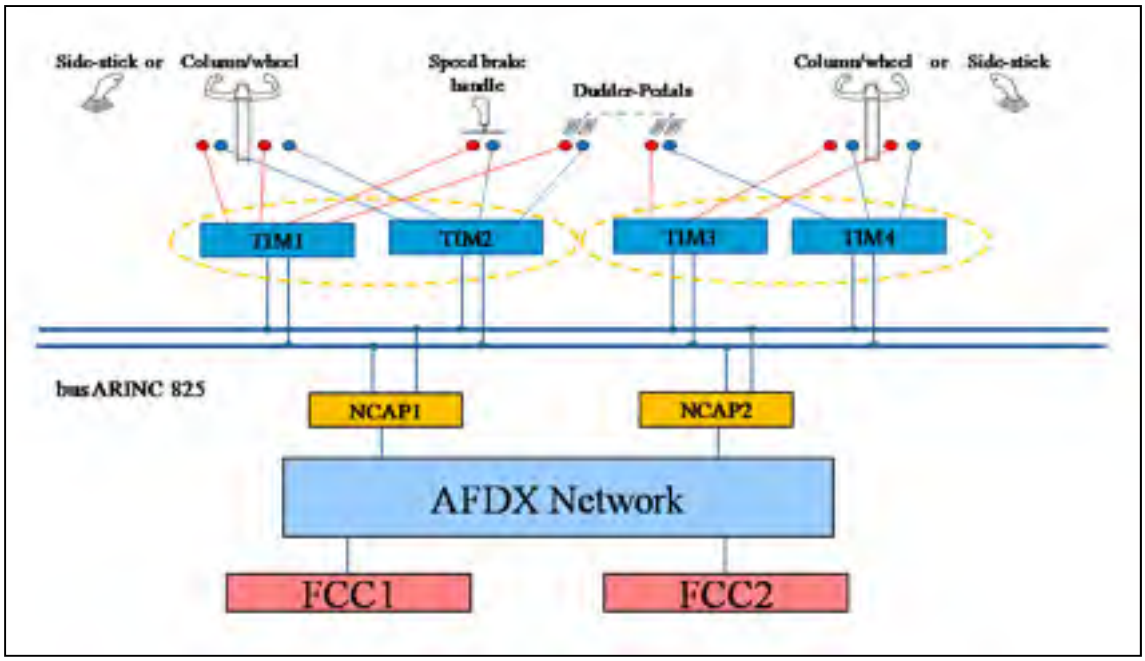

Figure 4.17 Système CDVE du côté du cockpit (première configuration)

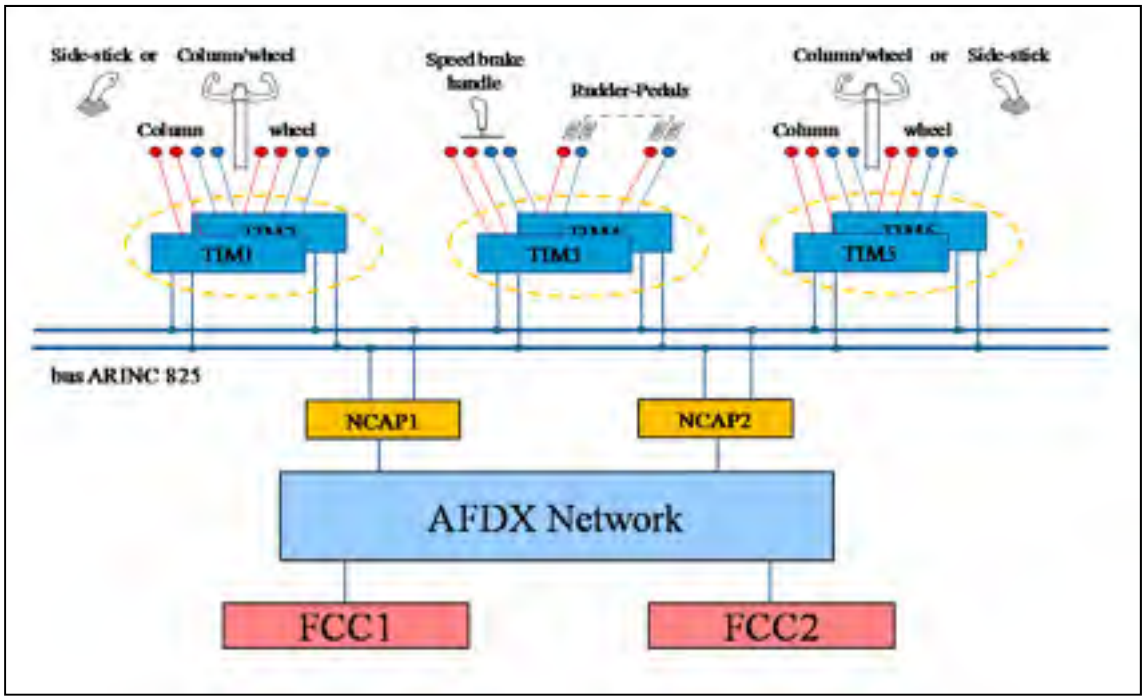

Figure 4.18 Système CDVE du côté du cockpit (deuxième configuration)

Analyse FMEA

Le Tableau 4.3 établit une étude FMEA (Failure Mode and Effect Analysis) des TIM utilisés pour interfacer les capteurs du cockpit dans un système CDVE. L'analyse FMEA représente une étape fondamentale dans le développement des systèmes avioniques critiques et est exigée par les normes telles que DO-254. Les modules de communication ARINC 825 ne sont pas couverts par cette analyse. Cette analyse expose les différents modes de pannes transitoires et permanentes de l'interface et met en relief les moyens disponibles pour les détecter et les recouvrir.

Tableau 4.3 Analyse FMEA d'un TIM

Module	Mode de panne	Détection de panne dans le TIM	Détection de panne dans le FCC	Recouvrement
Paire de service	- Donnée omise (transitoire)	- Exécution double	- Indiqué dans le message envoyé à travers le bus - Pas de nouvelles données	- Valeurs provenant d'un capteur redondant (FCC)
	- Donnée omise (permanente)	- Exécution double	- Indiqué dans le message envoyé à travers le bus - Pas de nouvelles données	- Recouvrement avec une paire redondante (TIM) - Valeurs provenant d'un capteur redondant (FCC)
	- Donnée erronée (transitoire)	- Exécution double	- Indiqué dans le message envoyé à travers le bus	- Valeurs provenant d'un capteur redondant
	- Donnée erronée (permanente)	- Exécution double	- Indiqué dans le message envoyé à travers le bus	- Recouvrement avec une paire redondante (TIM) - Valeurs provenant d'un capteur redondant (FCC)

Module	Mode de panne	Détection de panne dans le TIM	Détection de panne dans le FCC	Recouvrement
Crossbars	- Routage erroné des données (permanente/ transitoire)	- Vérification des champs d'adresse source et destination des messages	- Vérification des champs d'adresse source et destination dans les messages	- Valeurs provenant d'un capteur redondant
TEDS	- Donnée erronée lors de la lecture	- Vérification du contrôle de redondance cyclique (CRC) ou bit de parité	- Comparaison avec une donnée envoyée par un TIM redondant	- Valeurs provenant d'un TIM redondant
TMI	- Mêmes modes de panne que les modules de service	- Validation par les modules de service	- Comparaison avec une donnée envoyée par un capteur redondant	- Valeurs provenant d'un capteur redondant

4.6 Conclusion

Ce chapitre a été dédié à l'étude de fiabilité et de sûreté du TIM. Cette étude a permis de déterminer le mode de fonctionnement et le nombre de paires de service les plus appropriées pour l'implémentation du TIM. Le chapitre a également établi une étude comparative, en termes de fiabilité et sûreté, entre l'architecture du TIM présentée dans le chapitre 3 et l'architecture COM/MON. Finalement, l'exemple des interfaces de capteurs faisant partie du système CDVE du cockpit d'un avion moderne a été exposé.

CHAPITRE 5

IMPLÉMENTATION ET RÉSULTATS

5.1 Introduction

Ce chapitre expose les différents aspects liés à l'implémentation du TIM sur une plateforme FPGA. La vérification de l'interface est également présentée. Une attention particulière est dédiée à la présentation du prototype de l'encapsuleur de réseau développé dans le cadre du projet AVIO 402. Finalement, les processus de test et de validation de certaines caractéristiques de l'interface sont détaillés.

5.2 Spécifications et exigences

Les différentes fonctions et services du TIM ont été présentées au chapitre 2. Afin de mieux expliquer le fonctionnement de l'interface, la présente section vient exposer les choix qui ont été faits lors de l'implémentation du TIM ainsi que les exigences imposées dans le contexte du projet AVIO 402.

5.2.1 Spécifications du TIM

L'architecture du TIM repose dans son ensemble sur la norme IEEE 1451.0. Celle-ci définit une bonne partie des fonctions et caractéristiques du TIM. Toutefois, plusieurs points n'ont pas été spécifiés et ont été laissés ouverts aux concepteurs. Ainsi un certain nombre de spécifications liées à l'implémentation de l'interface devront être précisées. Celles-ci incluent principalement le partage des ressources et la latence des messages.

L'implémentation du TIM présentée dans ce chapitre est basée sur l'architecture présentée au chapitre 3. Le TIM implémenté servira à interfacer des capteurs de température de type COTS.

Partage des ressources

Les modules de service sont les éléments centraux du TIM regroupés en paires. Afin d'améliorer les performances de calcul du TIM, toutes les paires devraient fonctionner simultanément et partager équitablement le traitement des échantillons des capteurs. Ceci permet de minimiser le délai de traitement des échantillons. Le nombre de capteurs gérés par chacune des paires dépend du nombre total de capteurs et du nombre de paires de service disponibles. La Figure 5.1 montre l'architecture d'un TIM avec quatre paires de service et huit capteurs. Dans cet exemple, chaque paire de service gère deux capteurs. Le "crossbar1"» se charge de la commutation des données provenant des capteurs vers les paires de service correspondantes. Par ailleurs, comme il a été mentionné au chapitre 3, la duplication du contrôleur ARINC 825 ne sert pas uniquement à hausser la fiabilité mais aussi à augmenter la bande passante du TIM. En effet chacun des deux modules de communication gère le trafic provenant de la moitié des capteurs (la moitié des paires de service). Le "crossbar2" permet de d'acheminer les messages provenant des modules de service vers les contrôleurs ARINC 825 correspondants. Le chemin parcouru par les échantillons depuis les capteurs jusqu'au réseau est expliqué dans la figure 5.1 (traits rouges). Les commandes initiées par les calculateurs centraux ou les NCAP vers les canaux de capteurs parcourent les chemins inverses.

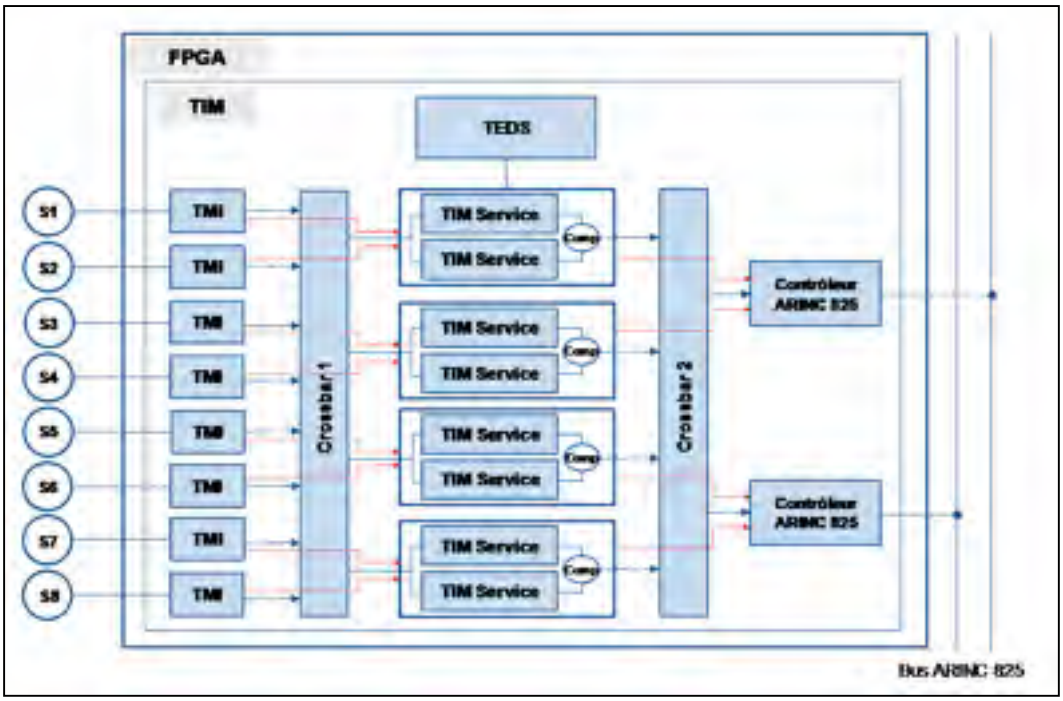

Figure 5.1 Architecture du TIM

L'assignation de chemins bien définis pour les données à travers le TIM permet d'améliorer le déterminisme de l'interface. En cas de panne permanente de l'une des paires des service, les données dont la paire fautive était responsable devraient emprunter un autre chemin. Le mécanisme de recouvrement sera étudié plus en détails dans la section 5.3. Rappelons que les TEDS sont des ressources mémoires partagées entre tous les modules de service dans le TIM. À titre d'exemple, dans la Figure 5.1, chacun des huit modules de service détiennent un accès aux TEDS. Un mécanisme implémenté veille à ce que les accès aux TEDS ne se produisent pas simultanément. Celui-ci est également détaillé dans la section 5.3

Délais des messages

D'après les spécifications du projet AVIO 402, le délai séparant l'acquisition d'une donnée par l'interface de mesures du TIM et sa réception par un NCAP devrait être inférieur à 2 ms. En d'autres termes, une fois les données reçues par le TMI, celles-ci devraient être traitées,

envoyées sur le bus ARINC 825 et finalement reçues par le NCAP dans un laps de temps ne dépassant pas 2 ms.

5.3 Développement du prototype

Tel qu'il a été mentionné tôt dans ce mémoire, la plateforme cible pour l'implémentation du TIM est un FPGA. Celle-ci a été choisie pour sa flexibilité et sa portabilité. En effet, les modifications peuvent être facilement apportées sur l'architecture au cours de la phase du prototypage et même après la fin de la phase du développement. La carte de développement choisie est la SP605 de Xilinx basée sur le FPGA Spartan-6 XC6SLX45T [43]. La carte fournit, entre autres, des entrées/sortie génériques, un connecteur UART, un connecteur JTAG, un connecteur Ethernet PHY, une mémoire DDR3, un connecteur PCI Express et un connecteur FMC LPC (Low Pin Count). Une carte d'extension ISMNET LPC [44] branchée sur le connecteur FMC a été utilisée pour fournir les deux connecteurs du bus CAN requis pour la communication ARINC 825 redondante. Les deux cartes SP605 et ISMNET LPC constituent une plateforme de développement peu chère et facile à utiliser pour l'implémentation du prototype de l'encapsuleur du réseau composé d'un TIM et d'un NCAP. Afin de prouver la compatibilité de l'architecture du TIM avec les transducteurs de type COTS, des capteurs de température AD 7415 d'Analog Devices [45] ont été utilisés. Ces capteurs génèrent des échantillons codés sur 10 bits qui seront acheminés vers la carte FPGA incorporant le TIM via le bus I2C [46]. Au moment de la rédaction du présent chapitre, des travaux de développement en cours visent à intégrer des capteurs RVDT avec le TIM. Un des principaux objectifs de cette phase est de prouver la généricité du design du TIM.

Les différents modules du TIM ont été développés conformément aux spécifications de la norme IEEE1451.0 décrites au deuxième chapitre. Afin de simplifier l'implémentation du mécanisme de transfert de données entre les différents modules du TIM, les interfaces de réception de données de chacun des modules ont été dotées de FIFO qui servent à stocker les données/commandes avant d'être traitées par le module en question. Les sections suivantes décrivent l'implémentation des modules du TIM. Seuls les contrôleurs ARINC 825 ne sont

pas décrits étant donné qu'ils ne font pas partie du travail achevé au cours de ce projet de maîtrise.

5.3.1 Interface de mesures du transducteur (TMI)

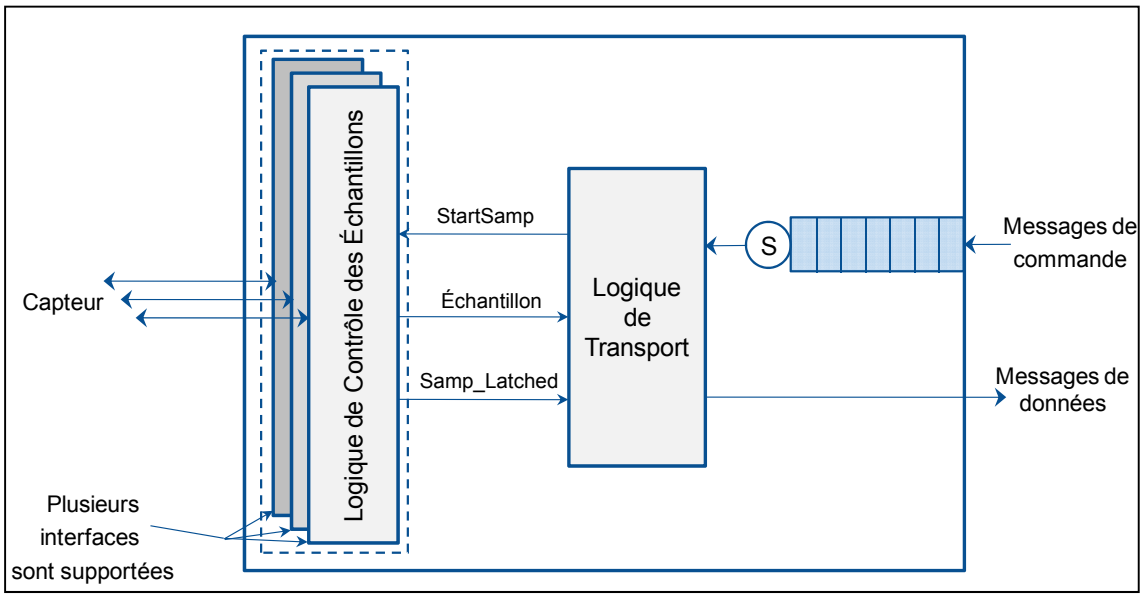

Figure 5.2 Architecture du TMI

Cette interface héberge la logique du canal de transducteur expliquée à la section 2.2.5.1. Toutefois, celle-ci n'inclut pas la logique du transducteur responsable de la conversion analogique numérique du signal. Le mode d'échantillonnage choisi est l'échantillonnage libre afin de permettre à l'interface de prendre des mesures de température d'une manière continue et périodique. Ce choix est basé sur le besoin de maintenir une disponibilité continue de données pour les applications avioniques critiques telles que les systèmes CDVE ou les systèmes de données de l'air. L'échantillonnage du capteur est effectué à toutes les 10 ms. La figure 5.2 montre le diagramme de bloc du TMI. La *Logique de Transport* implémente la machine à états du canal de transducteur (*Voir* Figure 2.3) ainsi que la machine à état de déclenchement du transducteur (*Voir* Figure 2.4).

Le module *Logique de Contrôle des Échantillons* gère l'ensemble des actions nécessaires à l'acquisition d'un échantillon. Dans le cas du prototype implémenté, ce module joue le rôle

d'un contrôleur I2C permettant d'initier une communication avec le capteur de température et de lire l'échantillon mesuré. I2C est un bus de données série synchrone, bidirectionnel et half-duplex. La version implémentée de ce protocole de communication assure des débits de transfert de données de quelques centaines de bit par seconde.

Après avoir reçu la commande `Enable_TranscuerChannel` de la part des modules de service, la L*ogique de Transport* entre dans l'état *Transducer Operating* et commence l'acquisition des échantillons en lançant le signal `StartSamp`. Suite à la réception de ce signal, le module *Logique de Contrôle des Échantillons* lance une communication avec le capteur sur le bus I2C pour acquérir une mesure.

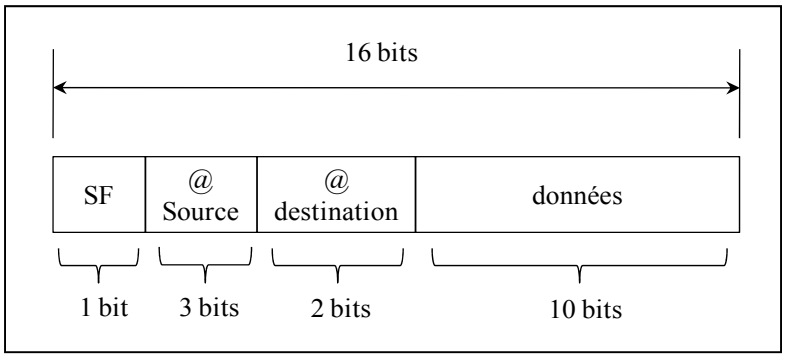

Figure 5.3 Structure modifiée du message

Le signal `Samp_Latched` indique la réception d'un échantillon. Les échantillons reçus sont codés sur 10 bits. Le module *Logique de Transport* encapsule les données et les transmet sous forme de message codé sur 16 bits. L'ensemble des actions susmentionnées est effectué périodiquement toutes les 10 ms. La figure 5.3 montre la structure du message transmis par le TMI vers les modules de service. Celle-ci présente des modifications par rapport à celle présentée dans le tableau 2.2. Des champs d'adresses sources et destination ont été ajoutés étant donné leur importance dans la gestion des sources de données par les calculateurs centraux. Le bit SF sert à indiquer les problèmes liés à l'acquisition d'une donnée (Exemple: échec de lecture de la mesure via le bus I2C).

5.3.2 Paire de service

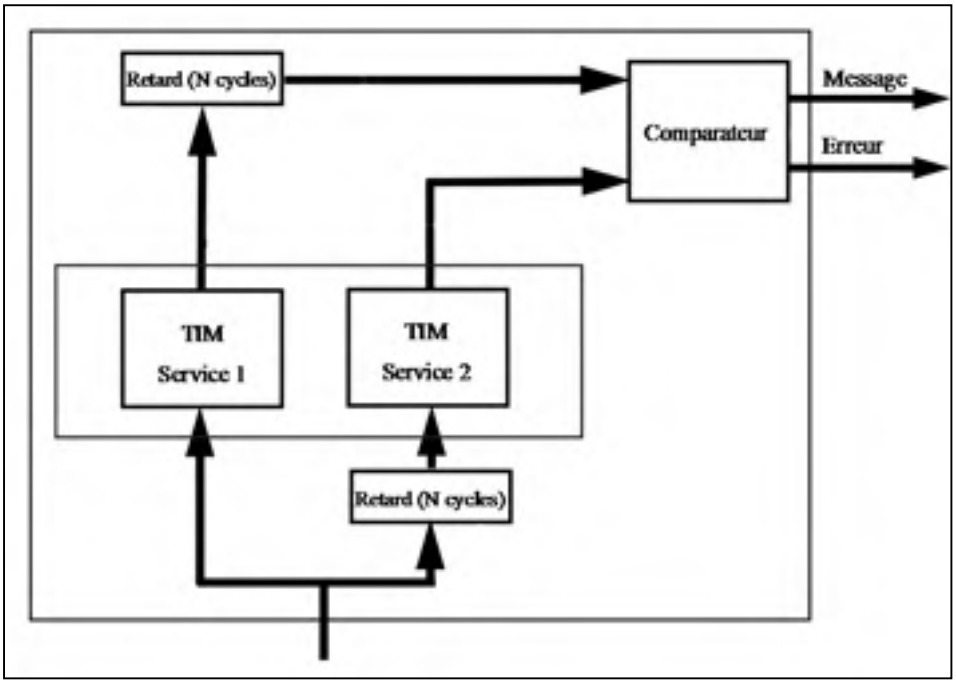

Figure 5.4 Architecture d'une paire de service

Les deux modules de service de chacune des paires de service traitent les échantillons reçus en parallèle. Les signaux de sortie des deux modules de service sont comparés par un module comparateur. Pour éviter les pannes résultantes d'effets de cause commune, les deux signaux à comparer sont retardés de manières différentes, comme indiqué à la Figure 5.4. Le retard introduit est un multiple de la période de l'horloge.

Par ailleurs, et dans le but d'éviter les pannes résultantes d'effets de cause commune qui peuvent affecter simultanément plusieurs paires de service, un décalage temporel a été également introduit entre les instants de traitement des différentes paires de service. La Figure 5.5 montre le décalage introduit dans un TIM à 4 paires de service. t1, t2, t3 et t4 représentent les instants de début de traitement des 4 paires de service. Le décalage temporel $t_{i+1} - t_i$ $(1 \leq i \leq 3)$ est un multiple de la période de l'horloge.

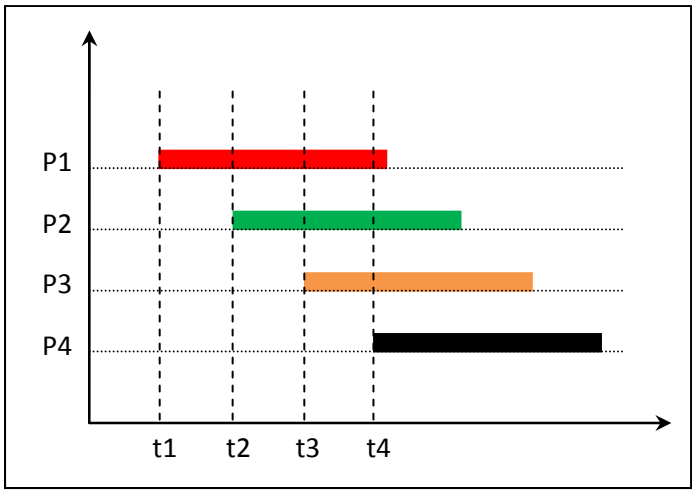

Figure 5.5 Décalage temporel introduit dans un TIM à 4 paires de service

Le principal service fourni par ces modules est la correction des échantillons. Après la mise sous tension du TIM, les modules de service entament la phase d'initialisation. Durant cette phase, chaque module de service copie les coefficients de correction de tous les capteurs qui sont stockés dans les TEDS de corrections. Le modèle de correction utilisé est un modèle linéaire (*Voir* équation 2.1) qui définit la fonction de transfert du capteur sur l'intervalle [10, 40] degré Celsius. Chaque capteur i est caractérisé par un couple de coefficient (α_i, β_i) (*Voir* figure 5.6) Les modules de service de TIM sont également en mesure de mettre à jour les valeurs suite à la réception et le traitement de la commande associée. Afin d'éviter les accès simultanés au TEDS, les lectures des coefficients par les modules de service sont réparties dans le temps. Une fois la phase d'initialisation terminée, les deux modules de service passent à l'état *TIM Active* et envoient la commande `Enable_TranscuerChannel` aux

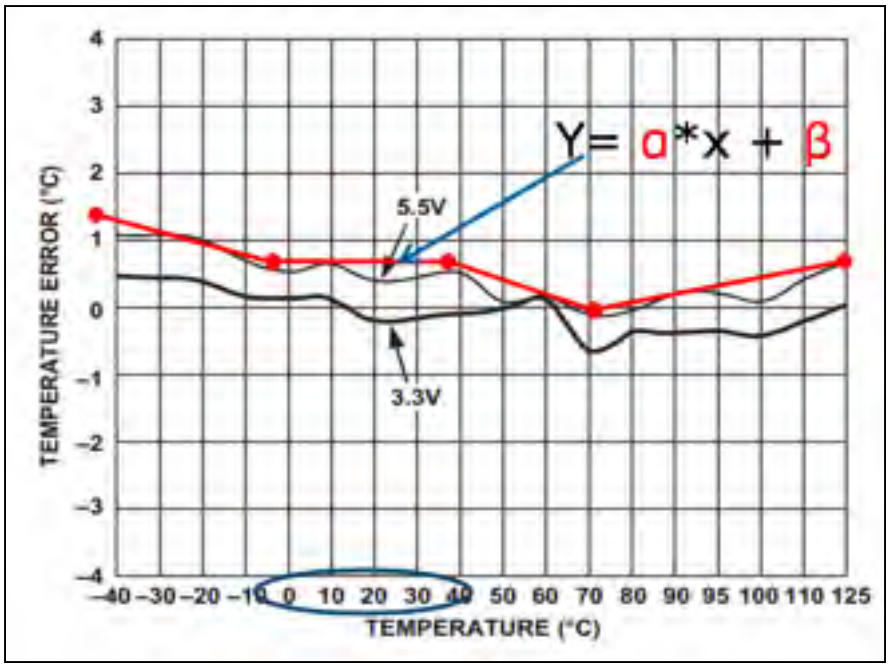

Figure 5.6 Modèle de correction des capteurs AD 7415
Tirée de Analog Devices (2010, p.15)

capteurs correspondants afin de déclencher l'échantillonnage. Dans chacune des paires de service, les échantillons envoyés par le TMI sont stockés dans une FIFO dans chacun des deux modules de service. Suite à la réception d'un échantillon, les deux modules de service examinent le champ d'adresse source du message reçu afin de déterminer les coefficients de correction correspondants. Par la suite, les modules de service procèdent au calcul de l'échantillon corrigé en utilisant les coefficients déterminés. Les résultats de correction sont codés sur 10 bits. Ceux-ci sont envoyés vers le comparateur pour une comparaison bit par bit. En cas de différence entre les deux résultats, le drapeau SF du message est mis à 1 et la valeur de l'échantillon est forcée à zéro. Le comparateur signale également l'existence la différence de calcul à travers le signal d'erreur `Error` qui sera émis vers les deux "crossbar". Dans le cas où les résultats calculés sont identiques, le drapeau SF est mis à 0. À la fin du processus de comparaison, le message est acheminé vers le contrôleur ARINC 825.

Gestion des pannes

Si le comparateur détecte une panne pour la première fois, celle-ci est considérée comme transitoire. Dans ce cas, la paire de service fautive est initialisée et reprend ses fonctions par la suite. Si une deuxième panne est détectée subséquemment, celle-ci est considérée comme permanente et la paire fautive cesse de délivrer des échantillons. Les transducteurs dont la paire fautive était responsable sont affiliés à une autre paire. Une table d'affiliation Transducteur-paire de service définit les recouvrements possibles en cas de pannes permanentes de l'une des paires de service.
Dans la version actuelle implémentée du TIM, les tables d'affiliation sont stockées au niveau des "crossbar". Toutefois, celles-ci peuvent être stockées dans des TEDS spécifiques afin de faciliter la reconfiguration du TIM.

5.3.3 Crossbar1

Le module "Crossbar1" est responsable de la commutation et l'ordonnancement des paquets provenant des capteurs vers les modules de service correspondants. Il est également responsable de l'acheminement des commandes envoyées par les paires de service vers les canaux de transducteurs correspondants. Les messages envoyés par les capteurs sont stockées dans des *FIFO_TX*. Ceux-ci devraient signaler à l'*Ordonnanceur* l'existence d'un message prêt à être transmis via le signal `request`, et ce automatiquement après la réception d'un échantillon. Les demandes de commutation sont effectuées par les *FIFO_TX* au même temps étant donné que l'acquisition des échantillons est synchrone. Un mécanisme d'ordonnancement à la ronde (*roundrobin*) permet à l'*Ordonnanceur* de multiplexer les messages qui seront émis vers la même paire de service. Le signal `Selection` permet au module *MUX* de sélectionner les messages à acheminer vers les paires de service. Les *FIFO_RX* reçoivent et stockent les messages provenant des paires de service. Ceux-ci seront commutés par l'*Ordonnanceur* après l'analyse du champ `Numéro_du_canal` de transducteur et en se basant sur une table de routage prédéfinie. La figure 5.7 montre un "Crossbar1" à quatre *FIFO_TX* et deux *FIFO_RX*. Le rôle du "crossbar1" ne se limite pas à

la commutation des messages. En effet, celui-ci assure également le recouvrement des pannes des paires de service conformément à la table d'affiliation Transducteur-Paire de service définie précédemment. L'*Ordonnanceur* se sert du signal `Error` pour localiser les paires de service fautives et pour rediriger le flux de données vers les paires disponibles.

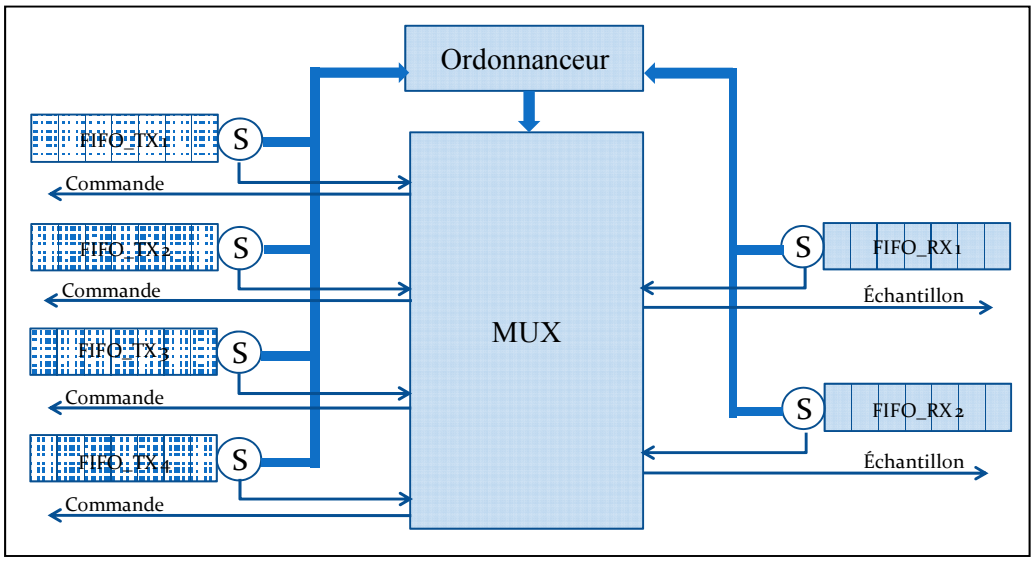

Figure 5.7 Structure du "crossbar".

5.3.4 Crossbar2

Ce module assure la transition des messages entre les paires de service et les contrôleurs ARINC 825. Il fournit pratiquement les mêmes fonctions que "Crossbar1". Un mécanisme d'ordonnancement à la ronde permet au "Crossbar2" de multiplexer les messages qui seront émis vers le même contrôleur ARINC825 (*Voir* Figure 5.1). Les messages de commandes envoyés par les modules de communication sont transmis vers les paires de service sur la base des adresses destinations des messages.

5.4 Résultats de synthèse

Cette section décrit les résultats de synthèse du système précédemment présenté. L'entrée du design du TIM a été réalisée en VHDL. Le TIM a ensuite été synthétisé pour le FPGA XC6SLX45T de Xilinx en utilisant la plateforme de développement logicielle ISE (version 12.4) de Xilinx. Le code des modules du TIM a été entièrement développé à partir du zéro et aucune propriété intellectuelle précompilée n'a été utilisée. Pour cette raison, le code du TIM pourrait être utilisé sur n'importe quel type de FPGA sans avoir besoin de le modifier. Toutefois, le code synthétisé n'a été essayé que sur le FPGA XC6SLX45T de Xilinx.

5.4.1 Taille du système

La quantité de ressources utilisée pour l'implémentation du TIM dépend du nombre de transducteurs interconnectés aux TIM et du nombre de paires de service implémentées étant donné que le nombre d'instanciation des autres modules est figé. Comme prévu, les paires de service occupent la plus importante quantité de ressources matérielles du FPGA. Toutefois, celles-ci constituent une petite fraction des ressources disponibles. Le reste de module du TIM est peu volumineux comparé aux paires de service. La taille de chacune des paires de service augmente en fonction du nombre de services fournis par ce module. La version finale de ces modules développées dans le cadre de ces travaux constitue l'ossature de base pour le

Tableau 5.1 Consommation de ressources

Module	Slice registres	Slice LUT	Instanciation
TMI	170	62	M
Crossbar1	506	721	1
Paire de service	1810	1531	N
Crossbar2	402	522	1
ARINC825	557	629	2
TEDS	80	52	1

contrôle de transducteurs et pourraient facilement être développées davantage en implémentant des fonctions plus complexes telles que des algorithmes de redondance analytique, des fonctions de traitement de signal complexes, etc. Avec l'implémentation de ce type de fonctions, il est prévu que la taille des paires de service sera considérablement augmentée. Le tableau 5.1 montre les quantités de ressources occupées par deux configurations du TIM: 1) 4 TMI et 2 paires de service, 2) 8 TMI et 4 paires de service. Il est remarquable qu'en doublant le nombre de TMI et de paires de service, les ressources occupées ont plus que doublé (250% pour les registres et 217% pour les LUT) (*Voir* tableau 5.1). Ceci est du non seulement au doublement du nombre de modules TMI et de paires de service, mais aussi à l'augmentation de la taille des "Crossbar1" et "Crossbar2" (causé par une augmentation du nombre de FIFO et à l'expansion de la logique d'ordonnancement). L'utilisation de FIFO pour l'interfaçage des modules fait croître les ressources utilisées en termes de *slice registers*. La profondeur des FIFO influence la quantité de registres utilisée pour gérer les adresses de lecture et d'écriture, ainsi que la complexité de la logique adjacente. Les FIFO implémentées permettent de stocker 8 messages de largeur 64 bits chacun.

Le coût d'incorporation de mécanismes de tolérances aux pannes est minime étant donnée l'importance de la quantité de ressources encore disponible dans le FPGA (*Voir* 2ème et 3ème colonne du tableau 5.2).

Tableau 5.2 Augmentation de ressources

	4 TMI, 2 paires de service	**8 TMI, 4 paires de service**	**Pourcentage d'augmentation des ressources**
# Slice registers	4288 (7%)	10699 (19%)	250%
# Slice LUT	4208 (15%)	9173 (33%)	218%

5.4.2 Synchronisation

Le tableau 5.3 montre les résultats de la synchronisation du système. Il est remarquable que le nombre de paires de service et de capteurs utilisées affecte la fréquence maximale et la période minimale pour le fonctionnement du TIM. Ceci est principalement dû aux tâches d'ordonnancement additionnelles réalisées par "Crossbar1" et "Crossbar2" quand plus de FIFO sont présents. La fréquence d'horloge maximale permise pour un TIM à 4 capteurs et deux paires de service est de 129 MHz. Celle-ci passe à 89 MHZ pour un TIM à 8 capteurs et 4 paires de service. Bien qu'il soit capable de supporter d'importantes fréquences d'horloge, le TIM a été implémenté et testé avec une fréquence d'horloge de 8 MHz. En fait, étant donné que les contrôleurs ARINC 825 ne supporte qu'une horloge de 8 MHz et afin d'éviter l'utilisation de plusieurs fréquences d'horloge, il a été décidé que la fréquence d'horloge de tout le TIM soit 8 MHz. La source de l'horloge utilisée provient d'un oscillateur à 27 MHz installé sur la plateforme SP605. Un diviseur de fréquence permet de ramener la fréquence de l'oscillateur (27 MHz) à la fréquence demandée (8 MHz). Tous les modules du TIM à part les contrôleurs ARINC 825 sont capables de fonctionner sous des fréquences d'horloge plus importantes sans avoir recours à des modifications du code VHDL. Les contrôleurs ARINC825 sont conçus pour fonctionner avec une fréquence d'horloge de 8 MHz afin d'assurer un débit maximal de 1 Mbit/sec. Pour que ceux-ci puissent fonctionner avec une fréquence d'horloge plus élevée tout en assurant le même débit de transfert, il faudrait apporter des modifications à leur code VHDL.

Tableau 5.3 Résultats de synchronisation

	4 TMI, 2 paires de service	**8 TMI, 4 paires de service**	**Variation en pourcentage**
Période minimale	7,75 ns	11,21ns	145%
Fréquence maximale	129,02 MHz	89,19 MHz	70%

5.5 Vérification

Afin de valider les fonctionnalités du système, chaque module du TIM a été vérifié individuellement. Une vérification du TIM au complet a été par la suite accomplie. La vérification a été achevée via des simulations effectuées sur le simulateur ISIM et via l'analyse des signaux acquises par l'outil Chipscope à partir du FPGA. Les bancs de test établis ont permis de vérifier le comportement de chaque module et de valider l'ensemble des mécanismes de tolérance aux pannes implémentés. Le TIM vérifié inclut 4 paires de service et est configuré pour interfacer 8 capteurs. L'ensemble de tests vise à 1) valider le fonctionnement du TIM dans un environnement sans pannes, 2) évaluer le comportement des différents modules en présence de pannes, et 3) valider les techniques de détection et de recouvrement de pannes conçues. Le tableau 5.4 décrit quelques tests parmi les plus intéressants qui ont été accomplis.

Tableau 5.4 Liste des tests

Test	**Module**	**Objectifs**
1	TMI	Vérifier l'acquisition d'un échantillon via le bus I2C
2	TIM au complet	Vérifier le comportement du TIM en présence de pannes
3	Paire de service	Vérifier la correction des échantillons
4	Paire de service	Vérifier le chargement des coefficients de corrections à partir des TEDS
5	Paire de service	Vérifier la détection et l'isolation des pannes
6	Crossbar1	Vérifier le recouvrement des pannes
7	Crossbar 2	Vérifier l'ordonnancement des messages

Afin de faciliter la vérification, les flux de données ont été générés par des émulations de capteurs implémentés dans les modules TMI. Les échantillons sont générés périodiquement toutes les 10 ms. La figure 5.8 couvre le test n°1. Celle-ci montre l'acquisition d'une donnée de capteur via un bus I2C. Par ailleurs, la figure 5.9 couvre le test n°2. Celle-ci met en relief le comportement du TIM et plus spécifiquement celui des "Crossbar1" et "Crossbar2" dans le cas où 1, 2 ou 3 paires de service sont en pannes. Les résultats du reste des tests sont fournis à l'Annexe I.

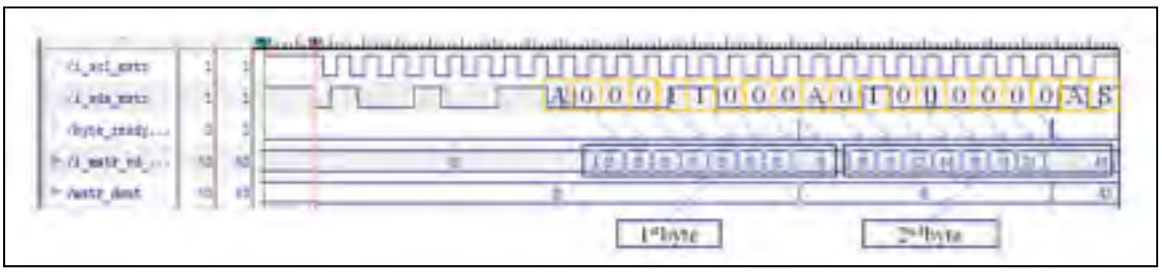

Figure 5.8 Acquisition de mesure

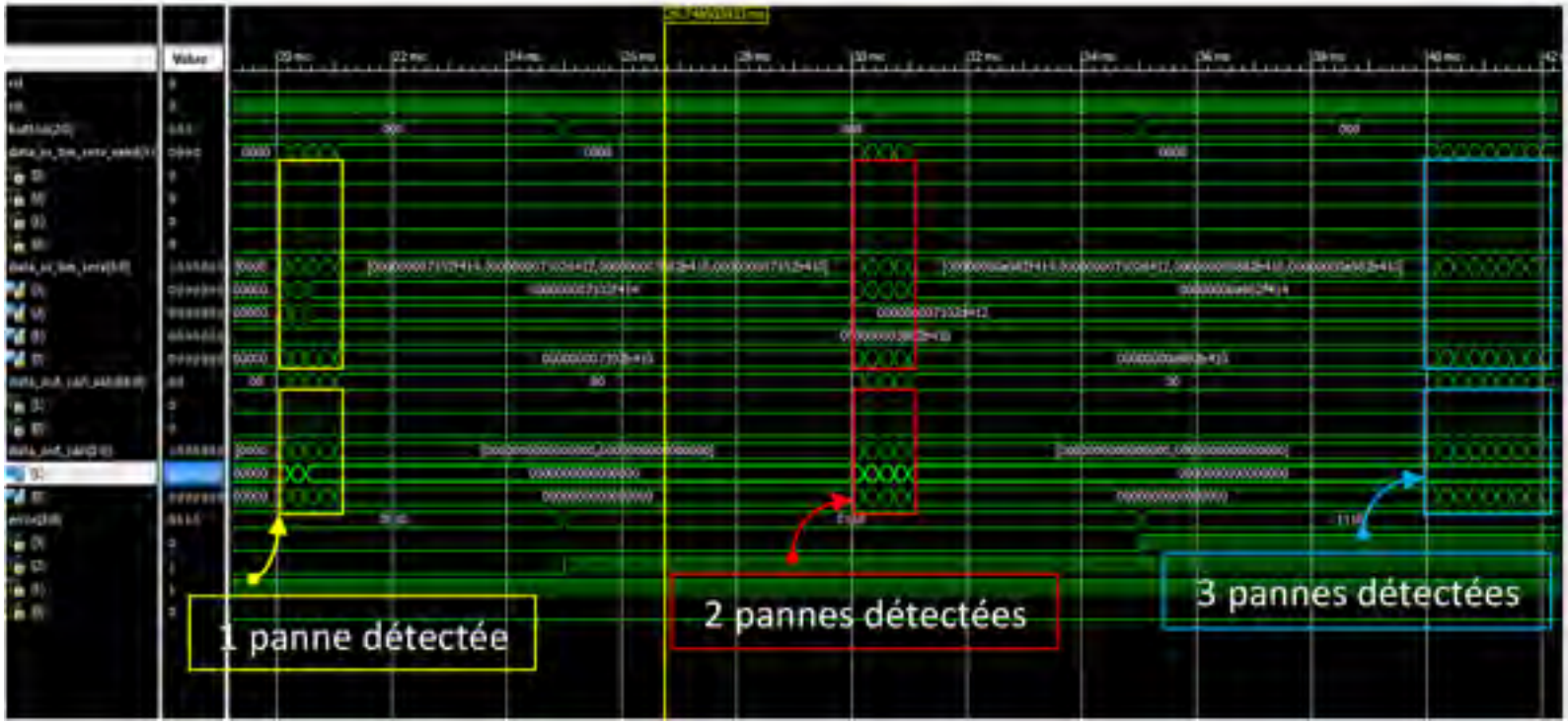

Figure 5.9 Comportement du TIM en présence de pannes

5.6 Validation via le prototype de l'encapsuleur

Cette section décrit la procédure suivie pour valider certaines spécifications et caractéristiques du TIM, et ce via des tests appliqués sur le prototype de l'encapsuleur réseau développé. Le prototype mis en œuvre a été principalement utilisé pour valider les mécanismes de tolérance aux pannes proposées ainsi que les caractéristiques temps réel du TIM. Le prototype constitue aussi une preuve de concept de la technologie des capteurs intelligents et fournit un banc d'essai pour la validation d'une variété d'algorithmes et de fonctionnalités liée à l'utilisation de cette technologie dans des applications avioniques critiques. Le prototype de l'encapsuleur implémenté est composé d'un TIM et d'un NCAP (*Voir* Figure 2.11). L'encapsuleur jouera le rôle d'interface réseau permettant le transfert des données de capteurs aux calculateurs centraux avioniques à travers un "End System" AFDX. Comme le montre la figure 5.10, le prototype est composé de deux plateformes SP605, une carte de distribution de voltage/courant et d'un capteur de température AD7415. Une des deux cartes FPGA est dédiée au TIM et l'autre à la partie implémentée en matériel du NCAP.

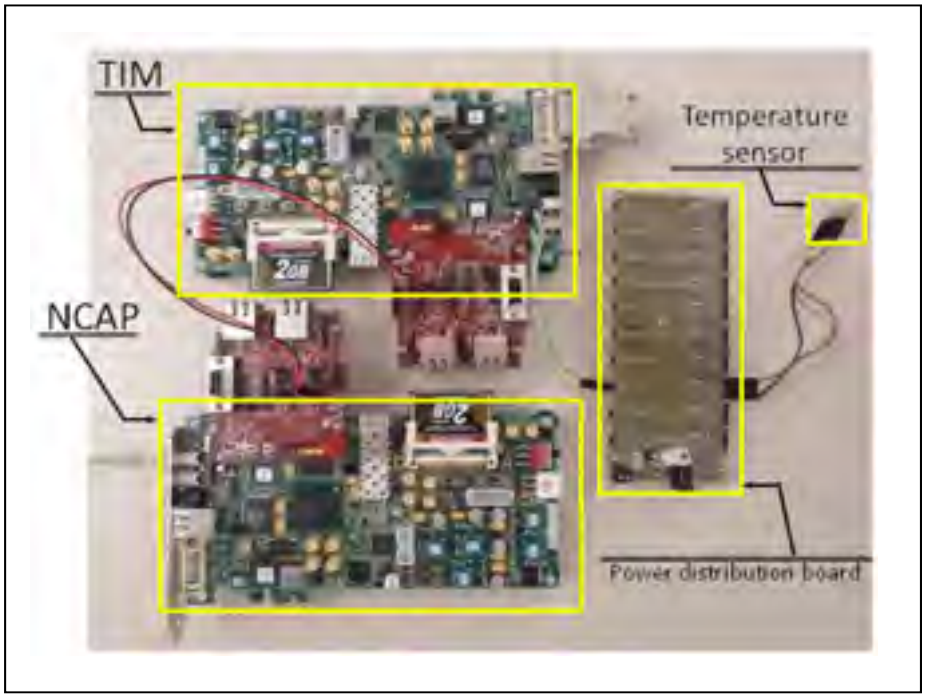

Figure 5.10 Configuration simple de l'encapsuleur basée sur un seul capteur

Le prototype montré à la figure 5.10 n'inclut pas le "End System" AFDX faisant partie du NCAP. La communication entre TIM et NCAP est assurée par un bus ARINC 825 double. La figure 5.10 illustre une configuration simple du prototype basé sur un seul capteur. Toutefois, le prototype est capable de supporter plusieurs capteurs (jusqu'à 10). Le prototype mis en place sert à émuler une interface de capteur de température qui fait partie d'un système de données de l'air et qui fournit des informations de température critiques aux différents systèmes de l'avion. Comme mentionné plus haut, et afin d'évaluer les performances d'un scénario complexe, la configuration du TIM testée intègre 4 paires TIM-service (P1, P2, P3, P4) et 8 capteurs de température (S1, S2, S3, S4, S5, S6, S7, S8). L'acheminement des messages est conforme à celui mentionné dans la figure 5.1. La partie implémentée en matériel de l'NCAP, présentée dans la figure 5.11, est composée d'un contrôleur ARINC 825 double, d'un module NCAP-service et d'une passerelle UART, tous configurés en série. Le FPGA de l'NCAP communique avec le "End System" AFDX par le biais d'un contrôleur intégré UART-USB. Celui-ci est géré par la passerelle UART. Les principales fonctionnalités du module *NCAP service* sont liées au contrôle de la passerelle UART, au formatage des données et à la mesure de la latence à travers des mécanismes qui sont expliqués plus tard. En ce qui concerne l'implémentation de l'NCAP, j'étais la personne responsable du développement de la passerelle UART, tandis que le reste des modules ont été implémentés par José-Philippe Tremblay.

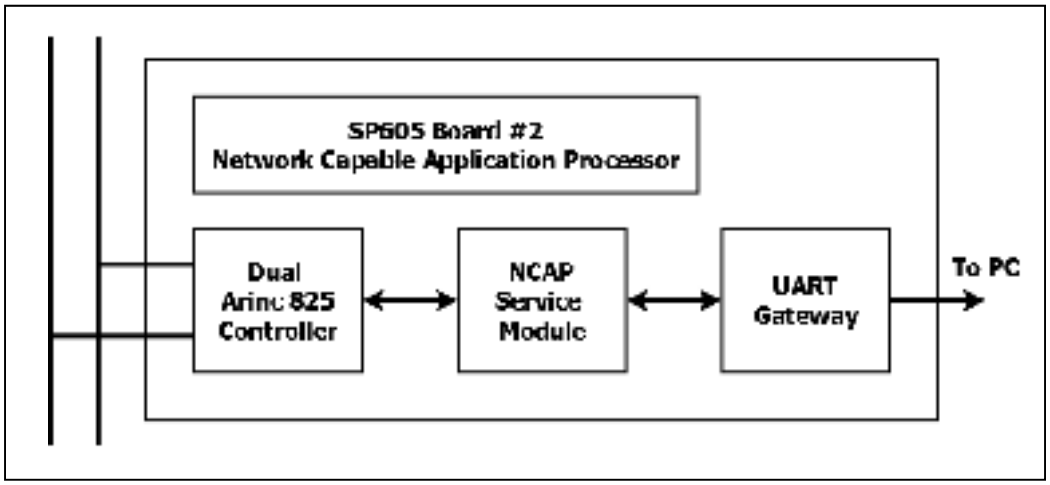

Figure 5.11 Partie implémentée en matériel du NCAP

Le FPGA du NCAP n'intègre pas des techniques de tolérance aux pannes. Cependant, des techniques de détection et de diagnostic de pannes similaires à ceux implémentés dans le TIM, pourraient y être implémentées. Le "End System" AFDX est développé en utilisant une plate-forme PC. Celui-ci implémente un logiciel permettant le débogage de l'encapsuleur et la validation des mécanismes de tolérance aux pannes.

Procédure de test et résultats

Le prototype a été utilisé pour valider les fonctionnalités de tolérance aux pannes et des caractéristiques temps réel suivantes:

- la capacité du TIM de détecter et de recouvrir les pannes permanentes qui surviennent au niveau des paires de service ;
- la latence des messages, séparant l'instant d'acquisition de l'échantillon par le TIM et l'instant de sa réception par le NCAP, devrait être inférieure à 2 ms.

Afin de valider les caractéristiques mentionnées ci-dessus, une stratégie de test a été définie. Celle-ci consiste à émuler la présence d'une panne permanente par l'injection d'un résultat de traitement fautif dans le module de comparaison de chacune des paires de service. Dans le but de simplifier la procédure d'injection de pannes, le TIM a été configuré pour que toutes les pannes détectées soient considérées comme permanentes. L'objectif principal du test est l'évaluation de l'impact de l'apparition de pannes permanentes sur la latence des messages. Le test mesure le temps qui sépare l'acquisition des données par l'interface de mesures du capteur (TMI) au sein du TIM et sa réception par les contrôleurs ARINC825 du NCAP, et ce en présence de pannes au niveau d'une ou plusieurs paires de service. Pour ce faire, des champs de latence *Latence_du_TIM* et *Latence_du_NCAP* ont été ajoutés à la structure initiale du message (*Voir* figure 5.12). Le champ *Latence_du_TIM* représente un horodatage qui sert à calculer la durée depuis l'acquisition de l'échantillon jusqu'au son arrivée aux contrôleurs ARINC 825. La *Latence_du_NCAP* représente la latence due au transfert des messages via le bus ARINC825. Le mécanisme permettant la mesure de cette latence est

expliqué en détail dans [47]. Lorsque le message est reçu par le "End system", une application calcule la latence totale en additionnant les deux mesures de latence (TIM et NCAP) et affiche la latence de chaque capteur en milliseconde.

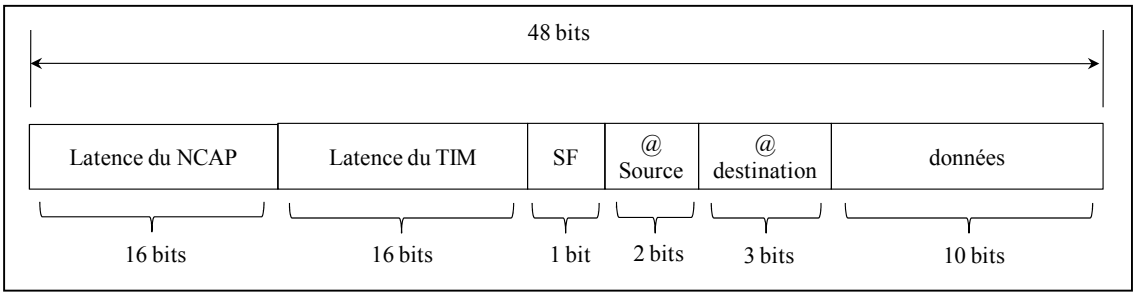

Figure 5.12 Format du message incluant les champs pour le calcul de la latence

Le recouvrement des pannes est effectué en se basant sur la table d'affiliation de transducteur-paire de service défini à la section 5.3.2. Le Tableau 5.5 montre la partie de la table d'affiliation qui couvre l'ensemble des scénarios d'injection de pannes réalisés. En d'autres termes, le tableau 5.5 illustre l'ensemble de capteurs qui sont traitées par chacune des paires de service, et ce en fonction des paires en panne.

Tableau 5.5 Table d'affiliation Transducteur- paire de service

Numéro de paire	**Fonctionnement normal**	**P1 en panne**	**P1 et P2 en panne**	**P1, P2 et P3 en panne**
Paire N° 1	S1, S2	-	-	-
Paire N° 2	S3, S4	S1, S2, S3, S4	S1, S2, S3, S4	-
Paire N° 3	S5, S6	S5, S6	-	-
Paire N° 4	S7, S8	S7, S8	S5, S6, S7, S8	S1, S2, S3, S4 S5, S6, S7, S8

Durant les tests, le mécanisme de détection de pannes a été validé par la réception du drapeau SF soulevée dans les messages reçus suite à l'injection d'une panne. Le tableau 5.6 montre les latences totales mesurées en fonction du nombre de paires fautives. La latence totale

réelle est calculée en compensant la partie de la latence mesurée causée par la transmission des bits relatifs aux champs *Latence_du_TIM* et *Latence_du_NCAP* dans la trame ARINC 825 (30 % des bits d'une trame ARINC 825 est dédié au transfert de ces champs). La latence totale réelle est ainsi toujours inférieure à la latence totale mesurée :

$$\text{Latence totale réelle} < \text{Latence totale mesurée} \tag{5.1}$$

Les mesures de latence dans le tableau 5.6 montrent que les pannes permanentes injectées dans les paires de service n'ont pas d'influence majeure. En fait, la latence de 2 ms des messages est toujours respectée. Selon le tableau 5.6, les latences mesurées en mode de fonctionnement normal varient entre 0,4 ms et 0,7 ms. Dans le cas où une seule paire est en panne (P1), la latence mesurée pour les capteurs S1, S2, S3 et S4 augmente légèrement. Cela est dû à l'augmentation de la charge de calcul de la paire P2. En présence d'une seconde panne (P3), les capteurs S5, S6, S7 et S8 sont à leur tour soumis à une légère augmentation de la latence similaire à celle observée dans le cas d'une seule paire en panne. Dans le pire des cas, caractérisé par 3 pannes permanents et un transfert de données via un seul bus (qui est équivaut à une perte d'un bus), la latence mesurée est de 2,1 ms. L'augmentation de la latence observée dans ce dernier scénario est principalement due au transfert de données sur un seul bus ARINC 825.Nous estimons qu'en compensant le délai causé par le transfert des bits relatifs à la mesure de latence (*Latence_du_TIM* et *Latence_du_NCAP*), la latence réelle est inférieure à 2 ms. Il peut être conclu que l'augmentation de la charge de calcul des paires de service due au remplacement des paires fautives n'a pas d'influence significative sur la latence des messages. De plus les résultats montrent que dans un mode de fonctionnement normal la contrainte de 2 ms est toujours satisfaite et qu'une marge de latence (plus de 1 ms) est disponible, ce qui permet d'implémenter des fonctions supplémentaires dans les paires de service.

Les résultats découlant du développement du TIM ont été publié dans un article [48] de conférence à DASC 2013 (IEEE Digital Avionics Systems Conference)

Tableau 5.6 Latences totales mesurées des messages

Capteur	Fonctionnement normal (ms)	1 paire en panne (ms)	2 paires en pannes (ms)	3 paires en panne (ms)
S1	0.4	0.4	0.4	0.4
S2	0.5	0.6	0.6	0.6
S3	0.6	0.9	0.9	0.9
S4	0.7	1.1	1.1	1.1
S5	0.4	0.4	0.4	1.4
S6	0.5	0.5	0.6	1.6
S7	0.6	0.6	0.9	1.9
S8	0.7	0.7	1.1	2.1

5.7 Conclusion

Ce chapitre a exposé, en premier lieu, les spécifications fonctionnelles liées à l'implémentation du TIM. Par la suite, les détails de l'implémentation de l'interface sur une plateforme FPGA ont été présentés. Par ailleurs, le chapitre a mis en relief le prototype de l'encapsuleur de réseau développé. Finalement, les résultats obtenus concernant la validation des mécanismes de tolérance aux pannes et des aspects temps réel de l'interface ont été exposés.

CONCLUSION

Dans ce mémoire, une interface de transducteurs intelligents tolérante aux pannes a été proposée. L'architecture de l'interface proposée est basée sur une redondance modulaire permettant la détection, l'isolation et le recouvrement des pannes. L'interface est également capable de communiquer avec plusieurs transducteurs et inclut un contrôleur ARINC825 double. La conception de cette interface constitue la première contribution de ce mémoire.

La deuxième contribution porte sur l'analyse des performances du système proposé en termes de fiabilité et de sûreté à l'aide des chaines de Markov. Ainsi, les performances de l'interface ont été comparées à celles d'autres types d'interfaces (COM/MON et redondance temporelle). Les résultats obtenus ont permis de montrer que l'insertion de paires de service redondantes améliorait la fiabilité de l'interface d'une manière considérable tout en assurant des niveaux de sûreté acceptables. Les mécanismes de tolérance aux pannes proposés favorisent un fonctionnement sûr en dépit de défaillances observées conduisant à un mode de fonctionnement correct avec des performances dégradées. Cette analyse de performance a été complémentée par une analyse FMEA, qui a exposé les différents modes de pannes de l'interface ainsi que les moyens de détection et de recouvrement possibles quand elles surviennent. Cette analyse constitue une base à partir de laquelle des améliorations architecturales pourraient être proposées.

La dernière contribution de ce mémoire porte sur l'implémentation de l'interface proposée sur une plateforme FPGA et son intégration dans le prototype d'un encapsuleur de réseau, qui achemine les données depuis les capteurs connectés à l'interface vers un "End System" AFDX. Le prototype a permis de valider des mécanismes de tolérance aux pannes via une technique d'injection de pannes. D'autre part, la latence des messages a été également validée. Les résultats obtenus confirment que dans un mode de fonctionnement normal ou en présence de pannes, la contrainte de latence qui était fixée au départ est toujours respectée. Le prototype développé a servi comme preuve de concept de la technologie des transducteurs intelligents et constitue un excellent banc de test pour la validation d'algorithmes et de

fonctionnalités liés à l'utilisation de cette technologie dans des applications avioniques critiques.

RECOMMANDATIONS

Les travaux futurs pourraient porter sur plusieurs sujets. En fait, le prototype implémenté pourrait être intégré dans un réseau AFDX. Ceci permettrait, entre autres, d'évaluer la latence de bout en bout des messages circulant depuis les capteurs jusqu'aux calculateurs des avions FCC. Par ailleurs, des fonctions de contrôle en boucle locale pour des actuateurs pourront être implémentées, testées et validées. Finalement, l'implémentation de l'interface COM/MON présentée dans ce mémoire pourrait être réalisée. Une telle interface pourrait facilement être développée étant donné les similitudes entre l'architecture des voies COM et MON et celle de l'interface proposée dans ce mémoire. De plus, les performances temps réel et de tolérance aux pannes pourraient être comparées à celles de notre interface.

ANNEXE I

RÉSULTATS DES TESTS

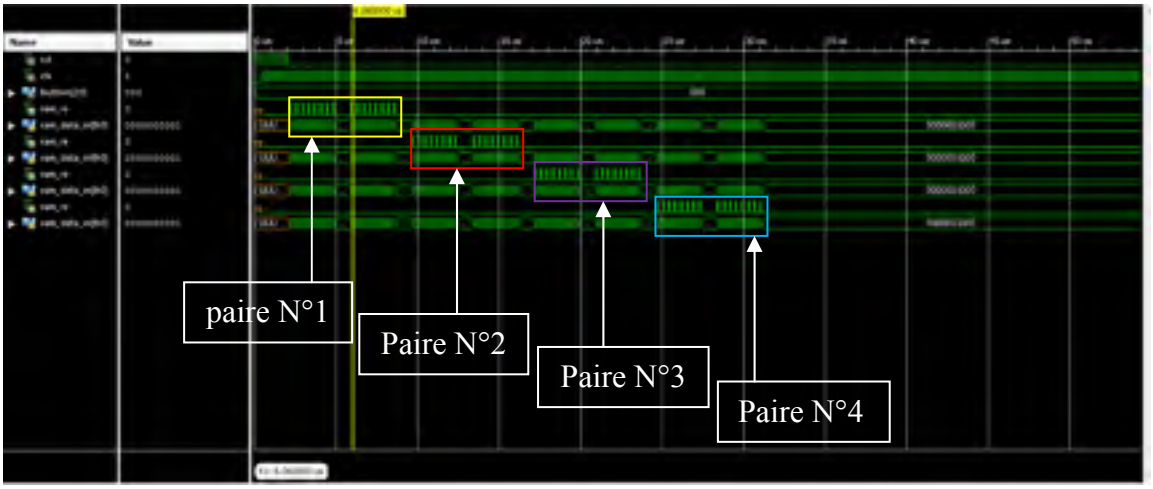

Figure-A I-1 Chargement des coefficients de correction (Test 3)

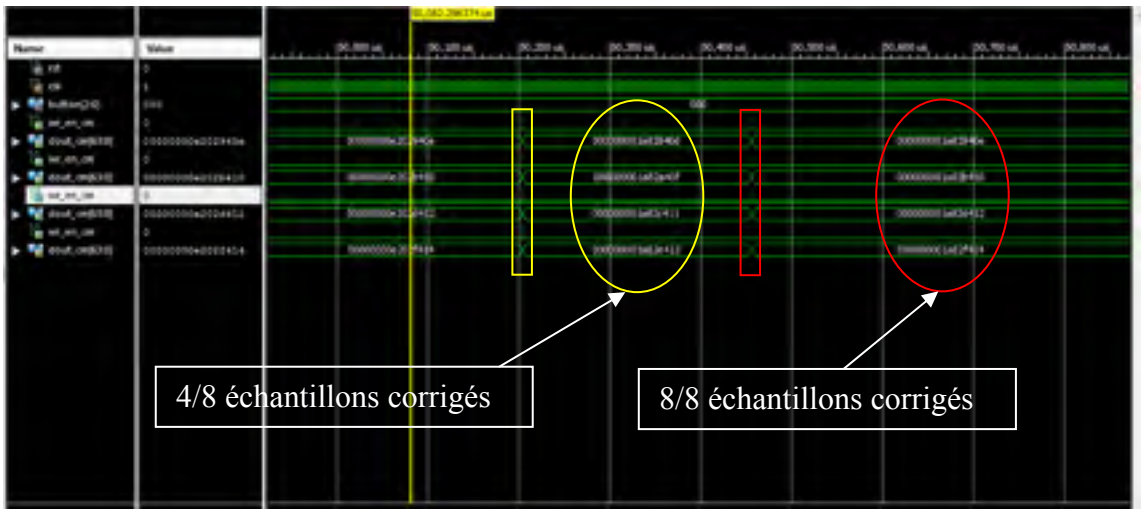

Figure-A I-2 Correction des coefficients (Test 4)

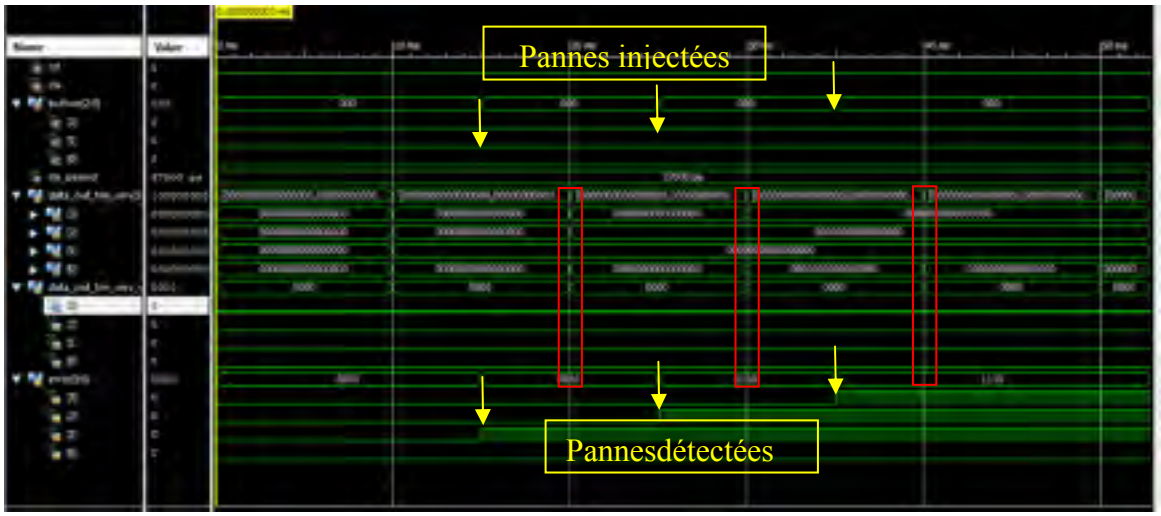

Figure-A I-3 Injection et détection de pannes (Test5)

Figure-A I-4 Recouvrement d'une panne par "crossbar2" (Test5 et Test 6)

Figure-A I-5 Recouvrement de deux pannes par "crossbar2" (Test 5 et Test 6)

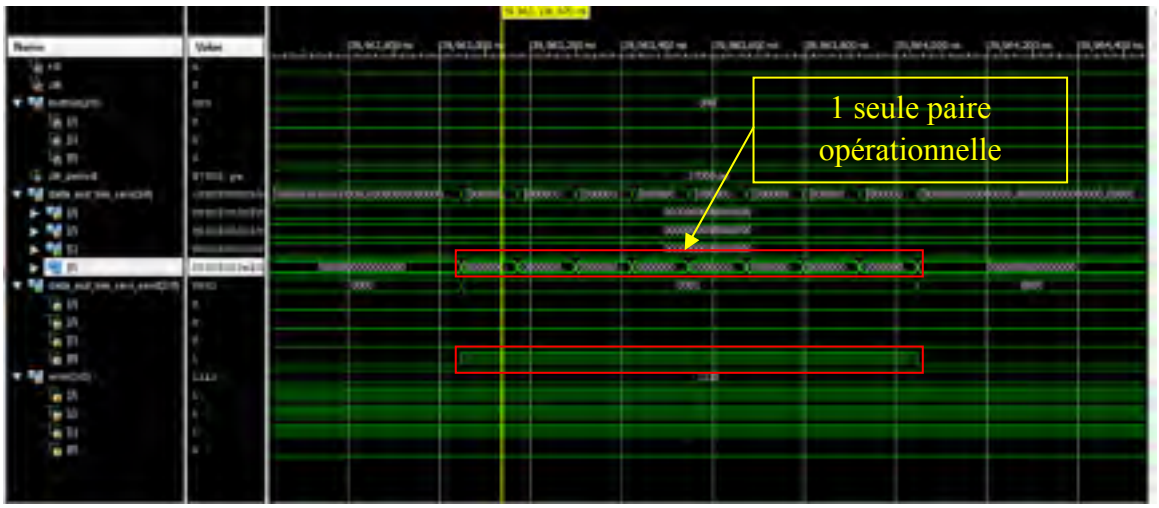

Figure-A I-6 Recouvrement de trois pannes par "crossbar2" (Test5 et Test 6)

Ordonnancement du Corssbar2 (Test 7)

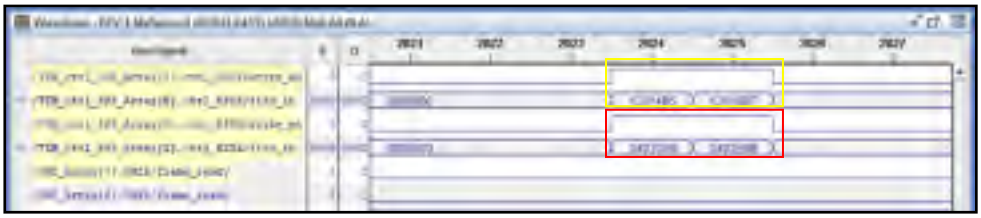

Figure-A I-7 Mode de fonctionnement normal

Figure-A I-8 Une paire de service en panne

Figure-A I-9 Deux paires de service en panne

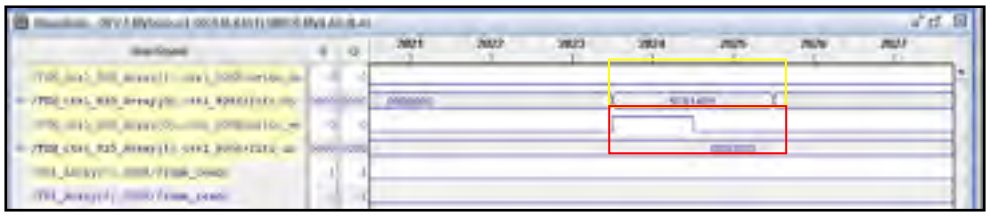

Figure-A I-10 Trois paires de service en panne

BIBLIOGRAPHIE

[1] Frank R. Understanding smart sensors. Artech House, 2013.

[2] Favennec J-M. "Smart sensors in industry." Journal of Physics E: Scientific Instruments 20, no. 9 (1987): 1087.

[3] Sghairi M. "Architectures innovantes de systèmes de commandes de vol." PhD diss., Institut National Polytechnique de Toulouse-INPT, 2010.

[4] Forsberg K. "Design principles of fly-by-wire architectures." (2003).

[5] InvenSense. "MPU-6000/6050: World's First Integrated 3-Axis Gyro, 3-Axis Accel and 9-Axis MotionFusion." *http://invensense.com/kr/mems/gyro/mpu6000.html*

[6] InvenSense. "MPU-6000 and MPU-6050 Product Specification Revision 3.3"(2012).

[7] Zhang J-G., Pervez A., and Sharma A. B. "Avionics data buses: An overview."*Aerospace and Electronic Systems Magazine, IEEE* 18, no. 2 (2003): 18-22.

[8] Zhang J-G., Ni Y-D., and Sharma A. B. "Data buses take flight."*Circuits and Devices Magazine, IEEE* 18, no. 4 (2002): 18-31.

[9] Goodrich. "Air Data Handbook" (2002).

[10] Honeywell. "Air Data Product Family" (2003).

[11] *http://www.tttech.com/en/markets/aerospace/projects/embraer-legacy/*

[12] *http://www.tttech.com/en/markets/aerospace/projects/bombardier-cseries/*

[13] Alena R., Ossenfort J., Laws K., Goforth A., and Figueroa F. "Communications for integrated modular avionics." In *Aerospace Conference, 2007 IEEE*, pp. 1-18. IEEE, 2007.

[14] Goupil P. "AIRBUS state of the art and practices on FDI and FTC in flight control system." *Control Engineering Practice* 19, no. 6 (2011): 524-539.

[15] Briere D., and Traverse P. "AIRBUS A320/A330/A340 electrical flight controls-a family of fault-tolerant systems." In *Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers., The Twenty-Third International Symposium on*, pp. 616-623. IEEE, 1993.

[16] Yeh Y. C. "Triple-triple redundant 777 primary flight computer." In *Aerospace Applications Conference, 1996. Proceedings., 1996 IEEE*, vol. 1, pp. 293-307. IEEE, 1996.

[17] Godo E. L. "Flight control system with remote electronics." In *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, vol. 2, pp. 13B1-1. IEEE, 2002.

[18] *http://ec.europa.eu/research/transport/projects/items/scarlett_en.htm*

[19] SCARLETT Project. "SCARLETT at a Glance DEVELOPED PRODUCTS." In *Aerodays, 2011*.

[20] Bernard S., and Garcia J-P. "Braking Systems with New IMA Generation." Training 2013: 12-12.

[21] Mats E. "Avionic Architectures Trends and challenges."

[22] Eccles L. H. "The need for smart transducers: an aerospace test and evaluation perspective." *Instrumentation & Measurement Magazine, IEEE* 11, no. 2 (2008): 23-28.

[23] *http://ieee1451.nist.gov/introduction.htm*

[24] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," IEEE, Piscataway, NJ, Sept. 2007.

[25] Barbosa R. Layered Fault Tolerance for Distributed Embedded Systems. Chalmers University of Technology, 2008.

[26] RTCA. *DO-254 Design Assurance Guidance For Airborne Electronics Hardware. 2000.*

[27] Dubrova E. Fault Tolerant Design: An introduction. Kluwer Academic Publishers (2007).

[28] Heimerdinger W., and Weinstock C. *A conceptual framework for system fault tolerance*. No. CMU/SEI-92-TR-33. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1992.

[29] Abella J., Cazorla F., Quiñones E., Gizopoulos D., Grasset A., Yehia S., Philippe B., Riccardo M., and Guillem B. "Towards improved survivability in safety-critical systems." In *On-Line Testing Symposium (IOLTS), 2011 IEEE 17th International*, pp. 240-245. IEEE, 2011.

[30] Kopetz H. *The Fault Hypothesis for the Time-Triggered Architecture*. TU Wien (2004).

[31] Gizopoulos D., Psarakis M., Adve S., Ramachandran P., Hari S. K. S., Sorin D., Meixner A., Biswas A., and Vera X. "Architectures for online error detection and recovery in multicore processors." In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, pp. 1-6. IEEE, 2011.

[32] Stott E., Sedcole P., and Cheung P. "Fault tolerant methods for reliability in FPGAs." In Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on, pp. 415-420. IEEE, 2008.

[33] D'Angelo S., Metra C., Pastore S., Pogutz A., and Giacomo R. S. "Fault-tolerant voting mechanism and recovery scheme for TMR FPGA-based systems." In *Defect and Fault Tolerance in VLSI Systems, 1998. Proceedings., 1998 IEEE International Symposium on*, pp. 233-240. IEEE, 1998.

[34] D'Angelo S., Metra C., and Sechi G. "Transient and permanent fault diagnosis for fpga-based tmr systems." In *Defect and Fault Tolerance in VLSI Systems, 1999. DFT'99. International Symposium on*, pp. 330-338. IEEE, 1999.

[35] Mojoli G. A., Salvi D., Sami M. G., Sechi G. R., and Stefanelli R. "KITE: a behavioural approach to fault-tolerance in FPGA-based systems." In *Defect and Fault Tolerance in VLSI Systems, 1996. Proceedings., 1996 IEEE International Symposium on*, pp. 327-334. IEEE, 1996.

[36] Carmichael C. "Triple module redundancy design techniques for Virtex FPGAs." *Xilinx Application Note XAPP197* 1 (2001).

[37] Baleani M., Ferrari A., Mangeruca L., Sangiovanni-Vincentelli A., Peri M., and Pezzini S. "Fault-tolerant platforms for automotive safety-critical applications." In *Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems*, pp. 170-177. ACM, 2003.

[38] TMS570LS3137 16/32-Bit RISC Flash Microcontroller Datasheet, 2012.

[39] *http://media.freescale.com/phoenix.zhtml?c=196520&p=irolnewsArticle&ID=1757908&highlight=*

[40] Tremblay J-P., Savaria Y., Zhu G., Thibeault C., and Bouanen S. "A System Architecture for Smart Sensors Integration in Avionics Applications." *SAE International Journal of Aerospace* 5, no. 1 (2012): 189-195.

[41] Peshave M., Bastani F., and Yen I-L. "High-Assurance Reconfigurable Multicore Processor Based Systems." In *High-Assurance Systems Engineering (HASE), 2011 IEEE 13th International Symposium on*, pp. 220-226. IEEE, 2011.

[42] Lin H., "Lifetime Reliability of Multi-core Systems: Modeling and Applications." PhD diss., Chinese University of Hong Kong, 2011.

[43] Xilinx, UG526 SP605 Hardware User Guide, July 18, 2011.

[44] Avnet, ISM Networking FMC Module User Guide, June 29, 2010.

[45] Analog Devices, AD7414/AD7415 temperature sensors datasheet in AD7414/AD7415, 2010.

[46] UM10204 I2C-bus specification and user manual Rev. 5-9 October 2012.

[47] Tremblay J-P., Savaria Y., Thibeault C., Bouanen S., and Zhu G. A Hardware Prototype for Integration, Test, and Validation of Avionic Networks, *IEEE Digital Avionics Systems Conference* (2013).

[48] Bouanen S., Thibeault C., Savaria Y., Tremblay J-P., and Zhu G. Fault Tolerant Smart Transducer Interfaces for Safety-Critical Avionics Applications, *IEEE Digital Avionics Systems Conference* (2013).