# Convertisseurs d'électronique de puissance et systèmes numériques en aéronautique : application au radar météo

Cédric Milleret

**Université Joseph Fourrier**

***N° attribué par la bibliothèque***

**/__/__/__/__/__/__/__/__/__/__/**

**DOCTORAT**
**Spécialité : "Génie Electrique"**

Effectué au **Laboratoire de Génie Electrique de Grenoble (G2Elab – LEG)**
UMR 5269
Dans le cadre de l'école doctorale "**E**lectronique, **E**lectrotechnique, **A**utomatique, **T**élécommunication, **S**ignal".

Et

Effectué à **Thales Systèmes Aéroportés Brest**
EWS/I2M - DDT MPS - Alimentations
Dans le cadre de projets civils en cours

Présenté et soutenu par

**Cédric MILLERET**

Le 5 février 2009.

**Titre :**

# Convertisseurs d'électronique de puissance et systèmes numériques en aéronautique : application au radar météo

## JURY

| | | |
|---|---|---|
| M. Philippe LEMOIGNE | Président | Laboratoire L2EP, Lille |
| M. Bernard MULTON | Rapporteur | Laboratoire SATIE, ENS Cachan |
| M. Gérard ROJAT | Rapporteur | Laboratoire Ampère, UCB Lyon |
| M. James ROUDET | Directeur de thèse | Laboratoire G2Elab, Grenoble |
| M. Pierre-Olivier JEANNIN | Encadrant universitaire | Laboratoire G2Elab, Grenoble |
| M. Emmanuel TOUTAIN | Encadrant universitaire | Laboratoire G2Elab, Grenoble |
| M. Arnaud MAHE | Encadrant industriel | Thales Systèmes Aéroportés, Brest |
| M. Hervé STEPHAN | Encadrant industriel | Thales Systèmes Aéroportés, Brest |
| M. Jean-Paul ARTIS | Responsable projet | Thales Systèmes Aéroportés, Brest |

**Titre :** Convertisseurs d'électronique de puissance et systèmes numériques en aéronautique, application au radar météo.

**Résumé :**

En aéronautique civile, les systèmes électriques qui composent l'avion sont de plus en plus nombreux, de par l'accroissement des fonctionnalités, des besoins des usagers (confort), mais aussi par le besoin d'améliorer les performances des actionneurs électromécaniques et hydrauliques, voire de les remplacer par du tout électrique. Ainsi, les systèmes d'électroniques de puissance sont les premiers dispositifs électriques que voient les actionneurs. Ces dispositifs sont interfacés avec les interfaces de pilotage par des systèmes de contrôles-commandes.

Les normes aéronautiques imposent que tous les dispositifs utilisés dans un avion répondent à des contraintes sévères quant à leur fiabilité et leur sécurité de fonctionnement. Dans le cas précis des convertisseurs statiques qui pilotent les actionneurs, on est confronté aux normes de fonctionnement des réseaux électriques, aux normes C.E.M., aux normes environnementales (au sens du contexte du dispositif), et aux normes liés aux systèmes de contrôle.

Le marché de l'aéronautique, bien que très spécifique, est très concurrentiel au niveau mondial. Les multiples objectifs de matériels très performants, très fiables, très sûrs, mais les moins chers possibles sont donc de véritables problématiques.

Dans le cadre des convertisseurs statiques, un moyen pour réduire les coûts est l'utilisation de DSP (Digital Signal Processors) pour maximiser l'intégration de la commande et pour réduire les coûts, mais ce composant n'est pas conçu pour le milieu aéronautique.

L'ensemble de cette étude porte sur la conception d'un nouveau type de radar marqué par plusieurs ruptures technologiques. Le prototype réalisé comporte un convertisseur statique piloté par un DSP, mais ayant des fonctionnalités qui vont largement au-delà du convertisseur basique d'électronique de puissance. L'aspect « système » est largement pris en compte. Le convertisseur mis en œuvre est un dispositif autonome de pilotage de moteurs.

**Mots clefs :**

Aéronautique, radar, onduleur triphasé, normes DO178B, sûreté de fonctionnement, DSP, Matlab®/Simulink®, Code Composer Studio®, Génération de code, machine synchrone SPMSM, indexage au démarrage, C.E.M., asservissements.

**Title:** Power Electronics Static Converters and Digital Control for aeronautical applications, control of weather radar.

**Abstract:**

In civil aeronautics, the electric systems which compose the aircraft are increasingly numerous, from the increase in the functionalities, the needs for the users (comfort), but also by the need to improve the performances of the electromechanical and hydraulic actuators, to even replace them by electric whole. Thus, the power electronics systems are the first electric devices which the actuators see. These devices are interfaced with the interfaces of piloting by systems of control.

The aeronautical standards impose that all the devices used on an aircraft answer severe stresses as for their reliability and their safety. In the precise case of the static inverters which control the actuators, one is confronted with the standards of operation of the electrical supply networks, with standards E.M.C., the environmental standards (within the meaning of the context of the device), and with the standards related to the systems of control.

The market of aeronautics, although very specific, is very competing on a world level. Multiple objectives of very powerful materials, highly reliable, very sure, but the least expensive possible, are thus of true problems.

Within the framework of the static inverters, a means to reduce the costs is the use of DSP (DIGITAL Signal Processors) to maximize the integration of the control and to reduce the costs, but this component is not designed for the aeronautical medium.

The whole of this study relates to the design of a new type of radar marked by several technological ruptures. The developed prototype comprises a static inverter controlled by a DSP, but having functionalities which go largely beyond the basic converter of power electronics. The aspect "system" is largely taken into account. The converter implemented is an autonomous device of servomotors piloting.

**Key words:**

Aeronautics, radar, three-phase inverter, standards DO178B, safety, DSP, Matlab®/Simulink®, Code Composer Studio®, code generation, synchronous machine SPMSM, start indexing, E.M.C., servo controls.

# Remerciements.

Je tiens avant toute chose à remercier Emmanuel Toutain du G2Elab™ qui a été le précurseur dans l'initiative de l'introduction de la commande numérique dans l'équipe électronique de puissance, et celui qui m'a permis de me lancer dans ce créneau. Son implication dans le système de recherche universitaire le rend à la fois proche des recherches amont, et son vécu et ses relations avec le monde de l'industrie lui donne une vision différente vis à vis des besoins des industriels. Tout fruit d'une recherche a pour but de mûrir en étant appliqué au plus de domaines possibles. Avec cette vision, j'ai travaillé efficacement avec son soutien.

Je remercie également Arnaud Mahe de Thales™ qui m'a donné « carte blanche » vis à vis de mes idées et m'a donner les moyens pour les tester. Sa vision technique et son côté manager industriel m'a rappelé plusieurs fois le but et l'applicabilité de mes idées : une bonne idée est une idée qui peut se mettre en oeuvre sans compromettre les coûts, qui soit durable et évolutive. La performance pure vient ensuite. Je regrette qu'il n'aie pu se donner corps et âme durant ma thèse avec son départ anticipé de l'entreprise.

Je remercie aussi Hervé Stéphan de m'avoir donné l'opportunité de travailler à Thales™ et donné des moyens pour un tel projet. Sa vision des projets et ses interactions avec les laboratoires permettent que ce genre de projets et les doctorants se rencontrent.

Je remercie aussi Pierre-Olivier Jeannin du G2Elab™ qui m'a suivi depuis plusieurs années, sur plusieurs projets, qui est à l'écoute des innovations et qui me permet d'y participer. Sa tâche n'a pas toujours été très gratifiante de par le contexte industriel du projet. Son implication au niveau du laboratoire contribue à l'évolution de la commande numérique dans les systèmes d'électronique de puissance.

Je fais bien sûr un clin d'oeil aux personnes que j'ai côtoyées, qui ont été des collègues agréables au G2Elab et à TAS, puisque la thèse m'a impliqué comme un membre d'une équipe normale. Les techniciens, ingénieurs et administratifs du G2Elab et TAS se reconnaîtront. Je ne citerai personne au risque d'en omettre car ils sont très nombreux.

Je remercie aussi Jean-Paul Artis et Maurice Callac de Thales™ avec leur équipe « météo » d'avoir travaillé avec moi, alors que j'étais un électron libre.

Je remercie naturellement James Roudet et l'équipe électronique de puissance pour m'avoir fait confiance au sein du laboratoire G2Elab.

Je remercie les rapporteurs Bernard Multon et Gérard Rojat pour le temps conséquent qu'ils ont consacré à la correction de ce manuscrit traitant de plusieurs domaines et Philippe Lemoigne pour avoir présider le jury sur ce sujet.

# Sommaire.

# Acronymes.

ADC : Analog to Digital Converter

ALARP : As Low As Reasonably Practicable

ALU : Unité Arithmétique et Logique

AMDEC : Analyse des Modes de Défaillances, des Effets et de leurs Criticités

ASV : Asservissements

BIOS : Built In Operating System

BREL : Boeing Radiation Effects Laboratory

CCS : Code Composer Studio

CEI : (ou IEC) Commission électrotechnique Internationale

CEM : Compatibilité Electro – Magnétique

CGS : convention Centimètre / Gramme / Seconde

CIPM : International Committee for Weights and Measures

COTS : Commercial Off-The-Shelf

CPLD : Complex Programmable Logic Device

CPU : Central Processing Unit

DRAM : Dynamic Random Access Memory

DDRAM : Dual Dynamic Random Access Memory

DSP : Digital Signal Processor

DMA : Direct Memory Access

eCAN : enhanced CAN Network

EPLD : Electrically Programmable Logic Device

EV : EVent manager

FAA : Federal Aviation Administration

F.E.M. : Force ElectroMotrice

FMECA : Failure Mode, Effects and Criticality Analysis

FPGA : Field Programmable Gate Array

GPT : Global Purpose Timer

I/O : Input / Output

INFORM : INdirect Flux detection by On-line Reactance Measurement

IPM : Interior Permanent Magnet

ISO : International Organization for Standardization

ISR : Interrupt Service Routine

I.T. : InterrupT

MAIN : programme principal

McBSP : Multiple Channel Buffered Serial Protocol

MOS : Metal Oxyde Semiconductor ( + FET : Field Effect Transistor)

MTBF : Mean Time Before Failure

NMI : Non Masquable Interrupt

NOP : No Operation

PFC : Power Factor Correction

PIE : Peripheral Interrupt Enable

PLL : Phase Locked Loop

PMSM : Permanent Magnet Synchronous Machine

PWM : Pulse Width Modulation

QdS : Qualité de Service

RAM : Random Access Memory

RAZ : Remise A Zero

RMA : Rate Monotonic Analysis

ROM : Read Only Memory

RTCA : Radio Technical Commission for Aeronautics

RTOS : Real-Time Operating System

RTS : Real Time Software

RTSU : Receiver Transmitter Servo Unit

SCI : Serial Communication Interface

SDRAM : Synchronous Dynamic Random Access Memory

SEE : Single Event Effect

SETR : Système Embarqué Temps Réel

SEU : Single Event Upset

SI : Système International d'unités (ISU)

SPI : Serial Protocol Interface

SPM : Surface Permanent Magnet

SRAM : Static Random Access Memory

μC : MicroContrôleur

WCET : Worst Case Execution Time

WNR : Weapons Neutron Research (Facility at Los Alamos National Lab)

# Préambule

L'objet de cette thèse est le résultat d'une motivation industrielle et d'une motivation personnelle quant au sujet.

En effet, ce doctorat et sa concrétisation m'ont permis d'ouvrir les portes de la spécialisation en industrie, et non celles de l'enseignement ou de la recherche. Cette thèse a été axée dans la recherche applicative directement aux problèmes industriels. La différence entre la recherche scientifique et la recherche industrielle est pour moi une différence fondamentale quant au but même du sujet et de son contexte.

La Rt&D (Recherche Technologique et Développement) est et a été mon créneau. Cette thèse propose des solutions, des méthodes pour faire du design en électronique de puissance vis à vis des convertisseurs statiques avec des systèmes numériques qui s'opposent aux règles ancestrales du contexte aéronautique. Mes propositions ont été testées sur des cas concrets de projets en cours à TAS (Thales Systèmes Aéroportés™).

Le sujet de thèse ne porte pas sur un point scientifique dur en particulier, mais sur un système qui soulève de multiples problèmes, corrélés, qui n'ont pas été résolus dans une étude interne à l'entreprise. De même, le sujet n'était pas clair au début, et le sujet de thèse s'est construit dans les premiers mois compte tenu des premières recherches bibliographiques et des interventions avec les gens du métier. C'est donc une thèse « système » où l'on va aborder plusieurs points difficiles sur un sujet, dont l'application est définie : un convertisseur statique de pilotage d'un radar. Le radar est un support d'application adéquat pour cette thèse puisqu'il pose de nombreux problèmes, et les résultats obtenus seront réutilisables pour d'autres projets de convertisseurs statiques en aéronautique. On ne tend pas encore vers un système « sur étagère » puisque le système est assez intégré, mais ce n'est pas forcément difficile de décliner le système présenté avec des fonctions allégées et dissociées. C'est pour cette raison que le système présenté permet de montrer un ensemble de résultats et de performances obtenus afin de se faire une idée du potentiel sous-jacent.

# Introduction

Cette thèse a pour but de montrer des techniques de développement pour la commande numérique de convertisseurs statiques, avec un environnement de fonctionnement aéronautique. L'évolution permanente des dispositifs d'électronique de puissance, notamment les structures à semi-conducteurs et les composants passifs, permet de réduire les masses et les coûts. C'est un double objectif : réduire les coûts de fabrication, mais aussi les coûts d'exploitation, car un ensemble de convertisseurs plus performants permet d'économiser du carburant. Le terme « techniques de développement » doit être décomposé en différents aspects, car le numérique pose des problèmes de conception selon la cible et demande des compétences particulières, mais le numérique pose des problèmes de maîtrise et d'interaction de tous les paramètres (I/O) vis-à-vis de l'intégration très importante. On a aussi un problème de fiabilité vis-à-vis des erreurs de conception, et des problèmes de fiabilité vis-à-vis des perturbations des éléments et évènements extérieurs.

Dans un avion, un grand nombre de systèmes sont présents : les onduleurs des moteurs de servitudes (pompes), les PFC en tête pour avoir une bonne exploitation des alternateurs des réacteurs, les convertisseurs multiples greffés au réseau de bord pour alimenter les appareils du cockpit, les convertisseurs pour alimenter les appareils de confort, les convertisseurs qui alimentent l'instrumentation et le système de guidage, et les convertisseurs des servomécanismes. Chaque convertisseur a une application spécifique, et même si les gammes de puissance et de taille sont identiques, ils ne sont pas interchangeables. Chaque application est soumise à des normes différentes selon le niveau de criticité. Les normes sont cumulables et l'on peut facilement se trouver dans des contextes de développement très difficiles.

On peut différencier par exemple :

- Les convertisseurs critiques de servomécanismes : le train d'atterrissage se doit de ne jamais être défaillant, donc son système est redondé, protégé, robuste. L'onduleur qui pilote le moteur a donc ces caractéristiques. Comme on ne peut pas envisager une quelconque panne, son niveau de criticité sera le plus élevé.
- Le système de détection radar : le système qui fait balayer le radar est du domaine des servomécanismes. Le radar vise loin et on rafraîchit les données sur une période assez grande (sauf cas militaire). Un mode de défaillance peur être toléré si le réarmement du système est rapide. Son niveau de criticité sera donc moyen. Par contre, on va superposer

d'autres normes quant aux interconnexions avec d'autres systèmes. Par exemple, les servomécanismes ne devront pas perturber le radar vis-à-vis de la C.E.M.

Les systèmes actuels d'électronique de puissance sont souvent analogiques, de par leur développement ancien, mais surtout à cause de la robustesse de ces technologies. Sur un onduleur en contrôle moteur, une boucle de régulation analogique (en courant) a le mérite d'être robuste et insensible aux perturbations telles que les agressions C.E.M. de l'environnement. Cependant, la place qu'elle occupe est loin d'être négligeable et son évolution par rapport à un actionneur différent implique des modifications coûteuse et longues (câblages, mise au point). Les régulations possibles sont limitées aux gains et intégrateurs. Des correcteurs plus complexes deviennent alors très délicats en mise au point, en reproductibilité, en implémentation. L'analogique impose des limitations en performance et manque de compacité.

Un système numérique peut remplacer une boucle analogique, mais on change alors radicalement de mode opératoire, entre la consigne et la commande. Le calcul implique des sources d'erreurs, des formats variables et variés, des possibilités de bogues, mais on accède alors à des correcteurs très complexes. Une régulation numérique est insensible à la température, elle est reproductible, et une mise à jour est plutôt facile.

La performance est de moindre importance par rapport à la fiabilité, mais les techniques utilisées pour concevoir et contrôler un convertisseur statique demandent une interaction forte entre le logiciel et le matériel. Un code pointu pour chercher la performance peut et va poser des problèmes de certification, car le programme qui définit l'application n'est pas forcément « surveillé » et possible « à surveiller » par une couche logicielle d'un autre niveau.

Une difficulté existe quant à la sûreté de fonctionnement [LAP89] et la robustesse du programme numérique, mais le réel problème entre un système analogique et un système numérique en aéronautique apparaît en altitude. En effet, autant le composant analogique (macroscopique) est insensible aux particules traversant l'atmosphère, autant le composant numérique (microscopique) qui sera percuté et/ou traversé par une particule non filtrée par la couche atmosphérique, sera soit perturbé, soit dégradé. Cette particule, appelée S.E.U (Single Event Upset), est un problème sérieux contre lequel aucun blindage raisonnable autour du composant ne peut le protéger. A la surface du sol, la densité de particules est très faible et l'énergie très atténuée. En altitude, ces deux derniers critères sont tous à faits différents.

Dans le cadre des logiciels qui pilotent des systèmes embarqués (convertisseurs statiques), le logiciel qui contrôle les entrées / sorties matérielles est d'un niveau de type « driver -

algorithme » : il s'agit d'une fusion entre la couche « application » qui donne les lois, les algorithmes de traitement du signal, mais aussi la couche « logiciel d'exploitation » qui coordonne les ressources, et la couche « driver » qui interface les périphériques avec l'application. On voit bien que le développement est plus proche du driver évolué, que du système d'exploitation allégé car les contrôleurs embarqués sont définis par leurs « drivers ».

La question est donc de savoir comment intégrer des techniques logicielles pour développer un système matériel avec un logiciel de premier niveau, lorsque le critère est la sûreté de fonctionnement, la fiabilité, et la portabilité des techniques sur de multiples cibles.

Afin de pouvoir réaliser de tels convertisseurs, j'ai travaillé 9 mois en relation entre le laboratoire et l'entreprise sur les normes pour extraire les besoins et les attentes d'un développement logiciel et matériel affectés par de nombreuses normes (aéronautique), mais aussi par différentes techniques. Ensuite, j'ai passé 12 mois à temps complet dans l'entreprise sur un projet Level C pour mettre en application une des techniques et être impliqué dans un processus de développement pour un produit dont l'application impose des normes très strictes. Ce projet est le support des mes travaux.

Aujourd'hui, le marché de l'aéronautique est partagé par plusieurs grands groupes, où Thales™ se positionne dans plusieurs domaines. Thales™ est aussi bien impliqué dans les équipements de confort des avions, que des équipements de sécurité, des équipements de navigation et bien d'autres fonctions dans l'avion et sa périphérie. Thales™ propose des solutions complètes pour le cockpit, mais auquel il manque un élément dans le package, le radar de pointe. Celui-ci est fourni principalement par Honeywell™, Bendix™ et Rockwell-Collins™. Ce radar est situé dans le radôme de tous les avions d'une taille « commerciale », c'est-à-dire des avions constituant la majorité du trafic aérien. C'est un radar à balayage mécanique et électronique ; sa fonction est de dresser en temps réel une cartographie météorologique le long du plan de vol de l'avion, afin de montrer aux pilotes les obstacles. On parle bien sûr des nuages, mais c'est surtout l'invisible à l'homme qui est le plus recherché, tels que les vents (latéraux, cisaillants), les trous d'air, les gouttes d'eau. Thales™ souhaite donc développer son propre équipement, mais pas avec un simple équivalent. Mon travail s'oriente vers une antenne disposant d'une rupture technologique au niveau des servomécanismes, qui elle-même introduit une rupture au niveau du système électrique. Les problèmes liés aux hyperfréquences et aux algorithmes de détection sont traités dans d'autres thèses. Notre problématique est l'intégration de la commande numérique des servomécanismes. Cette nouvelle conception de radars, permettant un gain significatif sur le coût du produit, tant en fabrication qu'en entretien, est rendue possible par les progrès technologiques

des composants de commande, sur leur intégration et sur leurs performances, ce qui n'était pas encore possible quelques années plus tôt.

La thèse fait le lien entre des techniques de sécurité de fonctionnement existantes, mais appliquées dans d'autres disciplines, que l'on étend au cas du convertisseur statique d'électronique de puissance ; puis la thèse permet de conduire l'étude sur le contrôle compact de l'antenne aéroportée avec les technologies de commande modernes.

# Chapitre 1 : contexte et contraintes sur l'électronique embarquée

# I. Les normes et les contraintes

L'usage d'un DSP (pVIII) dans un dispositif aéronautique introduit à juste titre la problématique de la sûreté de fonctionnement (et QdS [BAB05]) pour un système embarqué. On rappelle qu'un système embarqué (peu importe sa taille : avion ou chronomètre pour les J.O.) est un système limité, à la fois par ses ressources matérielles, par sa puissance, par sa source d'énergie, et aussi par ses capacités de traitement. La sûreté de fonctionnement et les dispositifs pour y parvenir seront donc aussi limités.

Au même titre que le FPGA qui est soumis à la norme firmware DO-254 [DO254], le DSP est soumis à la norme logicielle DO-178B [DO178]. Cette norme est une vaste procédure pour valider toutes les étapes, de la conception à l'industrialisation. La criticité 'C' en ce qui nous concerne est un « palier » de la norme qui durcit les contraintes au fur et à mesure que l'on s'approche de la criticité « A », la plus sévère (ex : train d'atterrissage). Les criticités B et C autorisent quelques souplesses dans le fonctionnement du système, telle qu'une possibilité de redémarrage en cas de problème.

Le logiciel est soumis au respect de la norme DO-178B, ainsi qu'aux règles de codages [SCS04], et aux sécurités Hardware et Software tel que le monitoring [FOR90] et la surveillance du code.

Le DSP est aussi un composant dont le fonctionnement est différent du FPGA puisque le DSP est intrinsèquement conçu pour réagir sur évènements. Il existe plusieurs façons de contrôler et d'interagir avec ses ressources. Chacune a des avantages et des inconvénients, et on peut se demander à juste titre pourquoi choisir une méthode plutôt qu'une autre. Les DSP dédiés à l'électronique de puissance et aux contrôles moteurs sont des variantes des microcontrôleurs, donc des systèmes à faible fréquence de fonctionnement, ayant beaucoup de périphériques, mais dont le cœur est un DSP de traitement du signal. La thèse montre et met en application le contrôle des ressources sur évènements, puisque le DSP est intrinsèquement conçu pour ce type de contrôle. Les techniques habituelles par séquenceurs sont à chaque fois utilisées pour les commandes car le recul sur ces techniques conditionne le choix du contrôle du composant. En proposant une commande sur évènements, donc en marge des méthodes habituelles, on doit augmenter les performances et donner des degrés de libertés supérieurs aux concepteurs. Ainsi, la démarche de conception et de programmation est typique, et on verra comment montrer qu'un fonctionnement évènementiel est déterministe, donc utilisable en aéronautique.

La sûreté de fonctionnement passe aussi par la robustesse aux S.E.U. (Single Event Upset) qui est une forte contrainte en altitude, puisque les systèmes doivent présenter une véritable robustesse face aux particules qui ne peuvent pas être stoppées. Il faut donc établir des parades pour contourner les effets qu'elles provoquent.

## *A. Rappels sur la sûreté de fonctionnement :*

### 1. La décomposition de la sûreté de fonctionnement

Quand on parle de sûreté de fonctionnement [TCHIN], le nombre de termes utilisés peut introduire une certaine ambiguïté (hors termes anglais), notamment au niveau de la sécurité et de la fiabilité. La sûreté de fonctionnement (voir Fig. 1) est bien la base de toutes nos interrogations, associée à ces divers termes.

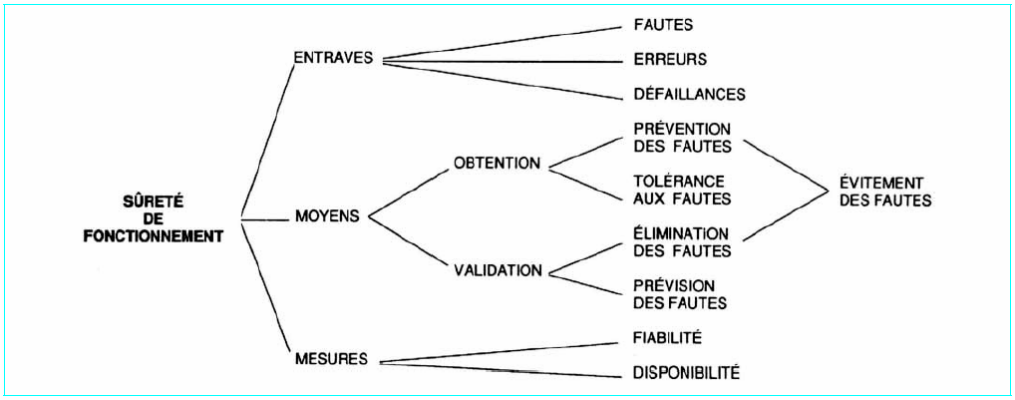



Fig. 1 : Arborescence de la sûreté de fonctionnement [TCHIN]

Cette arborescence se précise ensuite pour chaque thème. Prenons le cas des fautes (voir Fig. 2) :

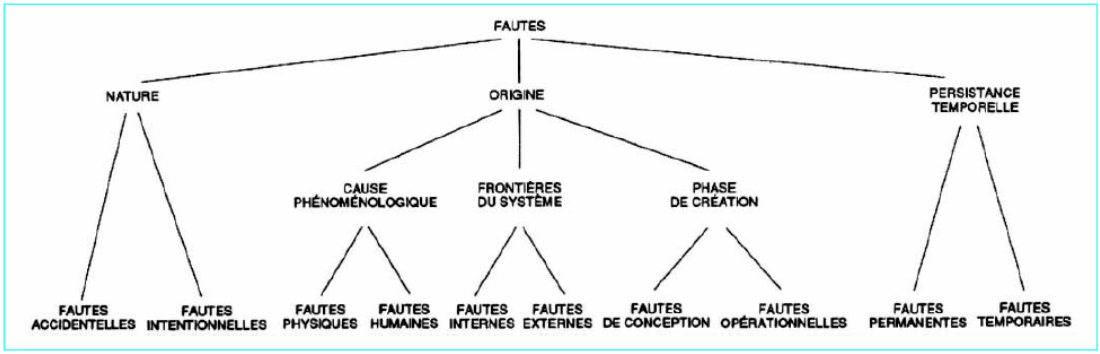



Fig. 2 : Arborescence des fautes de la sûreté de fonctionnement [TCHIN]

On peut alors appliquer ce type d'arborescence à chaque item de la sûreté de fonctionnement. Il est donc indispensable de mener une analyse rigoureuse sur le système si l'on veut étayer et justifier notre design. On peut synthétiser les principales fautes selon la Fig. 3 :

**Tableau 1 – Classes de fautes résultant des combinaisons selon les différents points de vue**

| Nature | | Origine | | | | | | Persistance temporelle | | Appellation usuelle |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Cause phénoménologique | | Frontières système | | Phase de création | | | | |
| Fautes acciden-telles | Fautes intention-nelles | Fautes physiques | Fautes humaines | Fautes internes | Fautes externes | Fautes de conception | Fautes opération-nelles | Fautes perma-nentes | Fautes temporaires | |
| ◆ | | ◆ | | ◆ | | | ◆ | ◆ | | Fautes physiques |
| ◆ | | ◆ | | ◆ | | | ◆ | | ◆ | Fautes intermittentes |
| ◆ | | ◆ | | | ◆ | | ◆ | | ◆ | Fautes transitoires |
| ◆ | | | ◆ | ◆ | | ◆ | | ◆ | | Fautes de conception |
| ◆ | | | ◆ | ◆ | | ◆ | | | ◆ | Fautes intermittentes |
| ◆ | | | ◆ | | ◆ | | ◆ | | ◆ | Fautes d'interac-tion |
| | ◆ | | ◆ | ◆ | | ◆ | | ◆ | | Chevaux de Troie, bombes logiques |
| | ◆ | | ◆ | | ◆ | | ◆ | | ◆ | Intrusions |

Fig. 3 : Synthèse des fautes de la sûreté de fonctionnement [TCHIN]

D'après ce tableau, on peut voir que les fautes de conception ne sont qu'une partie du problème. Il faut vraiment décomposer l'analyse pour isoler et résoudre un maximum de fautes possibles.

De plus, la sûreté de fonctionnement ne se résume pas à un fonctionnement sans bogues, mais implique la rigueur, de la conception à l'industrialisation, pour éviter les bogues (on parle ici du logiciel). Par l'analyse de la sûreté de fonctionnement, on extrait 3 thématiques essentielles (voir Fig. 4):

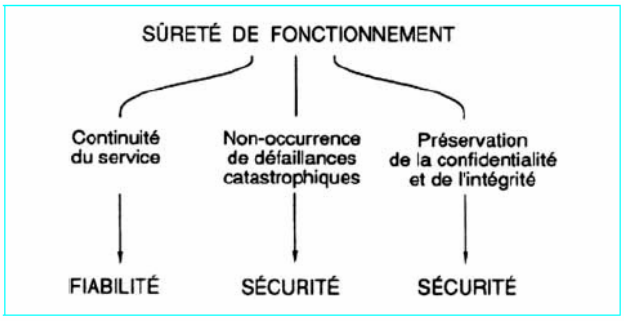



Fig. 4 : Thèmes de la sûreté de fonctionnement [TCHIN]

La fiabilité est une problématique essentielle en industrie, car elle est le gage de profits. Généralement, la fiabilité est obtenue (maximisée) pendant le processus d'industrialisation. A notre niveau, la fiabilité ne doit pas être un frein aux innovations, tant qu'elle peut être atteignable en termes de coûts et de moyens. La sécurité, concernant l'intégrité et les non défaillances, est le

thème qui nous intéresse : tous les choix techniques doivent garantir la sécurité que ce soit au niveau du prototype ou au niveau du produit final.

## 2. Les risques et la classification

La sûreté de fonctionnement précédemment décrite est très bien appréhendée en aéronautique. En effet, ce thème a toujours fait l'objet d'une grande attention, et les termes ont un sens bien plus défini que ceux que l'on a l'habitude de manipuler. Il convient de faire un bref point sur les termes qui nous intéressent. En aéronautique, les échelles de risques [GUI03] sont durcies par rapport aux échelles de risques des systèmes courants. On en donne la description au §I.A.2.a).

### a) Définitions

**Système à sécurité critique** : transfert de responsabilités de l'humain au dispositif dans un système où la sécurité des biens et des personnes est mise en jeu. On voit alors apparaître le terme de sûreté de fonctionnement, défini par Laprie (1989-1992) [LAP89] :

« *La propriété du système qui permet de placer une confiance justifiée dans le service qu'il délivre.* »

Deux termes distincts en ressortent :

- Sécurité-confidentialité : « security »
- Fiabilité, disponibilité, sécurité-innocuité : « safety »

Comme le risque zéro est illusoire, le concepteur intègre la notion du risque résiduel dans le développement, la fabrication et l'utilisation. Les effets non désirés sont définis par la notion de dommage.

**Les dommages** (normes IEC 60300-3-9, 1995) : c'est la gravité d'un incident

- catastrophique : décès
- majeur : blessures permanentes
- mineur : traitement médical
- minime : soins
- négligeable : incident sans traitement

La gravité d'un dommage est la sévérité. Le dommage est aussi qualifié par son occurrence :

**L'occurrence** générale : c'est l'occurrence usuelle pour bon nombre d'applications

- fréquente : >1 /par an
- probable : 1-$10^{-1}$ /par an
- occasionnelle : $10^{-1}$ – $10^{-2}$ /par an
- rare : $10^{-2}$ – $10^{-4}$ /par an
- improbable : $10^{-4}$ – $10^{-6}$ /par an
- invraisemblable : <$10^{-6}$ (tous les 1 000 000 d'années)

**L'occurrence en aéronautique** :

- fréquente : >$10^{-3}$ /par an
- probable : $10^{-3}$-$10^{-4}$ /par an
- occasionnelle : $10^{-4}$ – $10^{-5}$ /par an
- rare : $10^{-5}$ – $10^{-7}$ /par an
- improbable : $10^{-7}$ – $10^{-9}$ /par an
- invraisemblable : <$10^{-9}$ (catastrophe naturelle)

On voit donc que l'aéronautique est un milieu où l'échelle des risques est révisée en faveur des utilisateurs. Compte tenu de ces chiffres éloquents, on n'a pas de mal à comprendre pourquoi l'aéronautique fait partie des moyens de transports les plus sûrs, et pourquoi les coûts sont si élevés. Ainsi, le risque en aéronautique prend tout son sens dans le dimensionnement et la conception d'un système, puisque le risque est la combinaison entre la gravité et l'occurrence d'un dommage. On peut le qualifier dans le tableau suivant (occurrence générale) :

| Fréquence d'occurrence | Fréquence indicative (par année) | Gravité du dommage | | | | |
|---|---|---|---|---|---|---|
| | | 1 Catastrophique | 2 Majeure | 3 Mineure | 4 Minime | 5 Négligeable |
| Fréquente | >1 | H | H | H | H | I |
| Probable | 1-$10^{-1}$ | H | H | H | I | I |
| Occasionnelle | $10^{-1}$-$10^{-2}$ | H | H | I | I | L |
| Rare | $10^{-2}$-$10^{-3}$ | H | I | I | L | T |
| Improbable | $10^{-4}$-$10^{-6}$ | I | I | L | T | T |
| Invraisemblable | <$10^{-6}$ | I | L | T | T | T |

Légende des risques :
H : fort
I : intermédiaire
L : faible
T : insignifiant

Fig. 5 : Tableau de classification des risques, fonction de la gravité et de l'occurrence [GUI03]

On notera que le risque est déjà faible pour des cas invraisemblables, improbable et rare pour des occurrences générales, donc le cas occasionnel en aéronautique qui est un cas invraisemblable pour les usages commun, ce qui nous donne une vision très fermée des marges

de manœuvres sur un quelconque système. On trouve cette approche sous d'autres noms, telle que la représentation ALARP (voir Fig. 6):

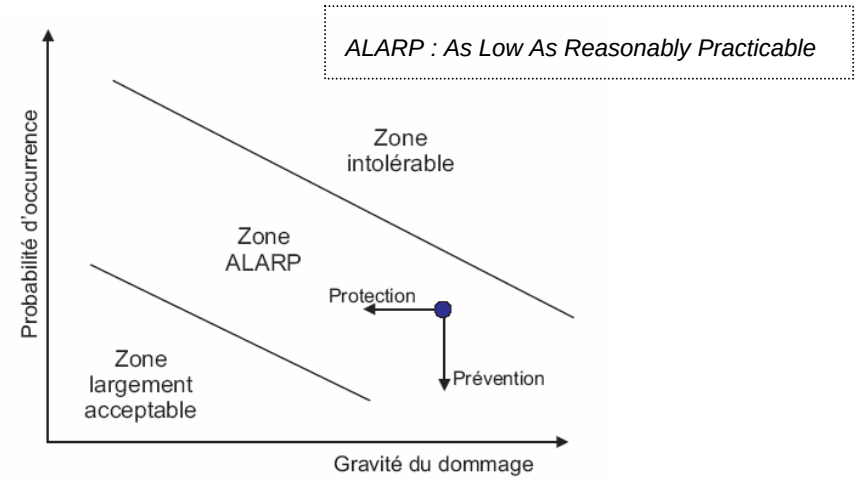



Fig. 6 : Représentation ALARP des risques [GUI03]

D'après cette représentation, la prévention doit augmenter si la probabilité du risque augmente. De la même manière, le degré de protection augmente en corrélation avec la gravité du dommage.

## b) Exemple sur l'application du radar

Sur un système tel que le radar, on a par exemple pour la classification précédemment illustrée :

- le contrôle moteur : la gravité du dommage est liée à une casse électrique du matériel. La protection devra être d'autant plus forte que le dommage causé pourra être important. La probabilité de ce risque est liée à l'arrêt du programme et au temps de redémarrage du système en cas d'anomalie. Si c'est dû à un SEU (développé au §II.A), la probabilité est a priori faible. Le contrôle moteur est une tâche autonome, indépendante et prioritaire, qui n'a qu'une seule entrée, elle-même vérifiée. Seuls les fonctionnements nominaux sont acceptés par des verrous logiciels. Il semble que la probabilité soit un critère moins prépondérant que la gravité.
- l'asservissement des trajectoires : le risque de défaillance est limité sur l'aspect gravité par le contrôle moteur. En effet, une mauvaise consigne sera ignorée, et n'aura pas d'impact sur la carte électronique. Par contre, la mécanique pourra être chahutée par de mauvaises trajectoires (vers les butées). Là aussi, la gravité est modérée puisse

qu'un mécanisme de protection est prévu. La probabilité de ce mauvais fonctionnement est sans doute l'aspect prépondérant. Comme il existe plusieurs façons d'aboutir à un comportement anormal, il est plus difficile de mettre en place tous les mécanismes de protections, et d'envisager toutes les sources d'anomalies. Le critère de probabilité démarre donc de plus bas.

## c) L'intégration des règles et des normes

A quel moment le risque est-il acceptable ?

*« Un risque acceptable (ISO/CEI) est défini comme un risque accepté dans un contexte donné, basé sur des valeurs courantes de notre société. »*

L'acceptabilité concerne le risque et non la gravité du dommage ou sa probabilité d'occurrences considérées séparément. L'acceptabilité est aussi fonction du contexte : est-ce un système précurseur avec de nouvelles technologies ou l'état d'un savoir, ou alors un système avec une mise en œuvre rodée ?

Les valeurs courantes sont une comparaison avec quelque chose d'approuvé de tous : on accepte le risque de mourir en prenant l'avion si la probabilité de ce décès par cette cause est identique voire inférieure à la probabilité de décès induit par un séisme ou une crise cardiaque.
Peut-on prendre un risque élevé si un dénouement fatal est inéluctable dans l'état actuel du contexte ?

On parle de sécurité lors de l'absence de risque inacceptable, c'est l'équilibre entre l'idéal de la sécurité absolue, et les exigences à remplir. Le danger est caractérisé par un transfert d'énergie ou dû à un ensemble de conditions couplées, induisant une accumulation conduisant à un accident, donc capable de provoquer un dommage.

- Phénomène dangereux : source potentielle de dommage ;
- Situation dangereuse : exposition aux phénomènes dangereux ;
- Evènement dommageable : évènement déclencheur qui fait passer de la situation au dommage ;
- accident : évènement non désiré et occurrence non prévue résultant en un dommage ;
- incident : évènement qui ne conduit pas à des pertes, mais qui a le potentiel de créer des dommages en d'autres circonstances.

L'analyse du risque consiste en la production d'informations en vue de la certification. La certification est possible si et seulement si l'analyse est considérée tout au long du processus de

mise en place du système (développement et fabrication). L'analyse est continue et itérative. On rejoint alors les thématiques exposées dans les normes, telle que la DO-178B. C'est pour cela que lorsque l'on veut appliquer une norme pour un projet, il faut tout au long de la conception prendre en compte l'analyse des risques : on doit pouvoir la justifier pour l'industrialisation du prototype. L'étude du radar météo a été faite pendant la thèse. Bien que la thèse soit une étude scientifique, le travail fait dans ce cadre peut être largement remis en question si l'on ne passe pas beaucoup de temps à s'impliquer dans la rigueur du développement, qui représente un temps significatif dans la thèse.

## d) La classification du risque

L'analyse du risque doit identifier les phénomènes dangereux par :

- la technique « forward » : on part des phénomènes dangereux et on en déduit les situations dommageables
- la technique « backward » : on part des dommages pour retrouver les situations et les phénomènes dangereux

L'erreur humaine doit être intégrée dans les phénomènes dangereux car en avionique l'erreur humaine est responsable de 70% des accidents. Dans le cas du développement du radar météo, c'est une approche « backward » qui a été adopté de façon naturelle. En effet, on s'est posé la question des dommages possibles suite à un mauvais fonctionnement, pour retrouver les manières existantes pouvant conduire à ces dits dommages. D'une certaine manière, la méthode « forward » est dans notre manière de penser, puisse qu'à chaque étape du développement, on anticipe sur les phénomènes dangereux afin d'éviter les situations dommageables, telle qu'une défaillance de l'exécution du programme, qui conduit à un blocage du système. Il ne faut donc pas trop se formaliser sur ces points « méthodiques » qui nous dispersent, puisqu'on remarque que finalement, avec la méthode AMDEC, on revient sur ces points.

L'analyse des risques va mettre en avant la défaillance éventuelle de composants du système. Un sous système logiciel doit avoir un niveau d'intégrité (Software Integrity Level) correspondant à un niveau de criticité des fonctions que le sous système doit réaliser. La norme Logicielle DO178B / ED-12 rev. B donne cette indication :

> *« Un logiciel dont un comportement anormal contribuerait à la défaillance d'une fonction du système, et qui conduirait à une défaillance catastrophique pour l'avion, est de niveau A le plus élevé (pour un effet nul, le niveau E est attribué) »*

Le but de cette classification est d'éliminer les mauvais logiciels et non de les analyser. La zone ALARP (Fig. 6) est un indicateur de prévention et de protection. En zone tolérable, on prend un risque si l'on peut en retirer un bénéfice. Plus les risques augmentent, plus les coûts pour y pallier augmentent. On travaille donc toujours avec ces deux objectifs :

- Prévention : réduire la probabilité d'occurrence
- Protection : réduire la gravité du dommage

Comme on a pu le répéter plus haut, une bonne analyse AMDEC synthétise un bon nombre de points. La méthode AMDEC (Analyse des Modes de Défaillance et de leurs Effets et de leurs Criticité) est une analyse point par point de toutes les fonctions du système. On ne prend en compte qu'un des modes de défaillance (parmi : coupure électrique, dépassement mémoire, pointeur fou, etc.) à la fois.

| Composant | Modes de défaillance | Cause | A. Effet local<br>B. Effet sur le système | Evaluation du risque | | | A. Moyens de détection possibles<br>B. Solutions |
|---|---|---|---|---|---|---|---|
| | | | | Occurrence | Sévérité | Risque | |
| Processeur du contrôleur de robot | Figé<br>*Plus de signe d'activité* | Interblocage du programme ou du système d'exploitation | A. Envoie commande constante<br>B. Mouvement bloqué en un point | F | 1 | H | A. Système externe type watchdog<br>B. Réinitialisation et Alerte utilisateur |

Fig. 7 : Exemple de la méthode AMDEC sur une fonction [AMDEC]

Dans ce tableau (Fig. 7), on rajoute autant de lignes qu'il est nécessaire pour lister tous les composants et les fonctions susceptibles d'avoir un mode de défaillance, donc tout ce qui existe dans le projet ! Compte tenu des résultats que l'on aura dans l'évaluation du risque, on pourra mettre en œuvre les dispositifs adaptés pour retrouver les tolérances définies au début du projet par tous les acteurs. On pourra par exemple prendre les décisions suivantes :

- La redondance structurelle : multiplier les composants matériels. Les données sont comparées pour conduire à écarter un composant dont la défaillance est détectée. Si la défaillance est critique, la redondance est une technique de sécurité. Cette technique est celle de la « triplication » dans le cas du FPGA. Notre architecture n'est pas vraiment orientée dans ce sens, comme cela a été expliqué au §I.
- La redondance fonctionnelle : elle permet d'effectuer une fonction avec une conception différente (autres capteurs, autres processeurs, autres connections). Cette technique est celle retenue au niveau du RTSU (pVIII) qui est interchangeable. C'est une redondance fonctionnelle sur un système complet. C'est une redondance adaptée au système, et qui est très coûteuse.

- On peut brider les performances si un évènement extérieur est détecté, et pouvant mettre en doute le fonctionnement nominal. A priori, dégrader le mode de fonctionnement n'est pas envisageable car le radar météo doit fonctionner en mode opérationnel quel que soit le contexte.

L'analyse débouche sur le constat suivant : il faut séparer le système qui gère les aspects fonctionnels et le système dédié à la surveillance, sans dépendance entre eux, sans complexifier le système, et avec une maintenance simple. Les RTSU interchangeables à distance, dont l'antenne est surveillée par un dispositif extérieur, est le résultat d'une telle analyse. Ainsi, on est confronté à nos propres contraintes qui brident sévèrement nos marges de manœuvres en ce qui concerne l'indexage du moteur (voir Chap3. §I.BI.B), puisque le RTSU est interchangeable.

## e) La criticité

La méthode AMDEC introduit aussi l'indice de criticité, qui est fonction des 3 facteurs essentiels (voir Fig. 8), dont le troisième est la détection. Ce facteur est décrit par les différents contrôles ou tests, actuels ou envisagés, permettant de détecter l'apparition du mode de défaillance ou de l'effet. Son nombre (arbitraire selon la méthode) est d'autant plus grand qu'il n'est pas détecté. On comprend bien que l'indice de criticité sera fort si la détection est médiocre, puisque la non anticipation va augmenter l'effet de la défaillance.

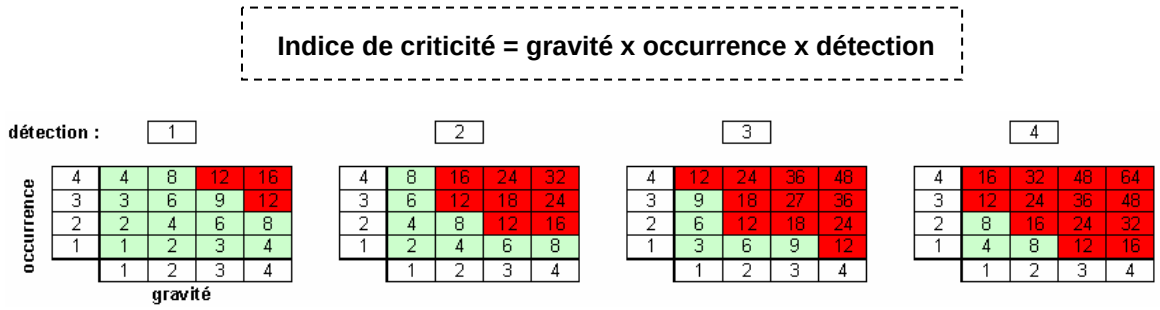



Fig. 8 : Représentation de l'indice de criticité pour un exemple d'un maximum admis de 11 sur 64

L'exemple en Fig. 8 illustre l'indice de criticité lorsque chaque paramètre est compris entre 0 et 4, 4 étant mauvais. On peut donc trouver cet indice entre 0 et 64. En industrie, on peut modifier les indices à notre guise. Chez Renault par exemple, chaque paramètre varie de 0 à 10, pour une criticité maximale de 100 pour les éléments porteurs. Dans ce cas (ici) courant, on décline les paramètres comme :

| Gravité : | Occurrence : | détection : (optionnelle selon les cas) |
|---|---|---|
| - 0 : sans danger | - 0 : nulle | - 0 : directe et immédiate |
| - 1 : mineure | - 1 : extrêmement rare | - 1 : indirecte mais rapide |
| - 2 : marginale | - 2 : occasionnelle | - 2 : (in)directe et peu différée |
| - 3 : critique | - 3 : moyenne | - 3 : tardive |
| - 4 : catastrophique | - 4 : fréquente | - 4 : aucune |

Dans une application ferroviaire, on aurait :

- 0 : incident mineur n'entraînant aucune conséquence sur les personnes,
- 1 : incident mineur pouvant entraîner quelques blessés légers, pas de grave, pas de morts,
- 2 : incident majeur/limité pouvant entraîner quelques blessés graves et légers, pas de morts,
- 3 : accident grave pouvant entraîner quelques morts et un certain nombre de blessés graves,
- 4 : accident catastrophique pouvant entraîner de nombreux morts et blessés.

Il faut donc fixer les critères au début du projet. Dans le cas du radar, les dénominations précédentes ne sont pas adaptées.

Les exigences de la certification se justifient donc par la traçabilité des choix, l'intégration de l'analyse logicielle, l'intégration des facteurs humains, l'approche multi domaine, et la terminologie non ambiguë.

## *B. La DO-178B*

### 1. La genèse de la norme et son évolution

La norme logicielle DO-178 est une norme plutôt ancienne (1982) rédigée par le comité RTCA (pVIII) afin de réglementer les nouveaux dispositifs logiciels, revue en 1985 (révision A) suite à un retour d'expérience. La révision B, la dernière à ce jour, date déjà de 1992. Cette mise à jour a été demandée en 1989 par la FAA (pVIII) suite aux évolutions rapides des technologies et aux différentes interprétations des industriels et des autorités de certifications (qui sont des problèmes récurrents). Cette norme est apparue avec les premières utilisations des logiciels embarqués en aéronautique, et n'est pas vraiment adaptée aux cibles « DSP » modernes. Le DSP est pourtant aussi ancien que la norme, mais c'est un composant « nouveau » en aéronautique compte tenu de sa maturité à ce jour. Cela s'explique sans doute par la montée en puissance récente de ces composants qui les rendent attractifs. Cependant, les DSP existent déjà en

aéronautique, mais sous des « versions d'utilisation » processeur où les programmes exécutés sont calqués sur les calculateurs, c'est-à-dire suivant un séquenceur ou un RTOS (pVIII).

La nouveauté de notre application est l'utilisation d'un DSP optimisé pour l'électronique de puissance, où la fréquence d'horloge est faible en comparaison des DSP de traitements numériques, et les périphériques sont reliés à de l'électronique de puissance qui demande un contrôle évènementiel. On s'écarte donc du DSP qui traite des données qui lui-même communique avec d'autres composants qui traitent des données. La sûreté de fonctionnement a ici un impact direct sur les composants électroniques, c'est donc une approche software/hardware.

Cette norme est globalement trop générale pour que l'on puisse appliquer une quelconque règle au DSP en question, mais elle est aussi souvent contradictoire avec l'utilisation de celui-ci. En effet, la norme ne pouvait pas anticiper toutes les éventualités des progrès des microprocesseurs / microcontrôleurs. La révision 'C' de cette norme est prévue pour fin 2008, des industriels tel que Thales™ participent à son élaboration, mais la question directe des interruptions (évènements) n'a pas été abordée d'après les sources internes.

## 2. Exemples significatifs de la norme appliqués au projet

L'étude de la documentation RTCA DO-178B (1992) nous donne un support qui régit le développement logiciel. Elle n'est pas forcément détaillée sur le plan technique, mais elle relie les techniques, comment les mettre en œuvre, les tester et les appliquer. Elle a été écrite pour donner au concepteur les lignes directrices à suivre pour son développement logiciel.

La norme traite des aspects : (liste non exhaustive)

- de compatibilité avec la cible (interruptions),
- de boucles infinies,
- de temps critiques,
- d'entrée arrivante déterminée,
- de transitions d'états,
- de saut d'interruption,
- de code mort,
- d'utilisation des interruptions,
- des marges,
- des méthodes alternatives.

En fait, à partir du moment où l'on peut prouver et justifier ses choix, analyser les temps critiques, définir les transitions, faire apparaître les marges, il ne semble pas y avoir de contre-indication à utiliser les interruptions. Beaucoup de critères sont communs entre le fonctionnement en ISR et le séquenceur tel que le code mort (code inactif et présent), mais d'autres changent le raisonnement de fond :

- les boucles infinies : en mode I.T., les boucles infinies sont remplacées par le fonctionnement asynchrone où l'on démarre une tâche par I.T. et l'on recommence avec une autre I.T. synchronisée avec le Hardware. Avec un séquenceur, la boucle infinie est indispensable pour recommencer un cycle.
- la compatibilité avec la cible : lorsque la cible fonctionne sous I.T. dédiées, c'est que son fonctionnement est prévu, donc les mécanismes associés des évènements sont supportés.
- Les temps critiques : en interruptions, les temps critiques apparaissent clairement car le Hardware impose les contraintes du système, donc on peut voir si la réponse est respectée, alors qu'avec un séquenceur, le « scan » du système est indépendant des changements de l'électronique (cet exemple illustre notre cas de convertisseur statique). L'exemple précédent illustre les enchaînements.

Voyons quelques points significatifs de la norme, qui ne sont pas des règles pour standardiser et justifier les développements (traçabilité, vérifications, outils, patchs, process, analyses…).

- **§2.3.1 : le partitionnage :** le logiciel est implanté dans des cibles distinctes, pour isoler les fonctions et les rendre indépendantes. Chaque partie à son propre niveau de sécurité. Les critères sont :

  →*Le DSP est un composant où cohabitent des périphériques matériels distincts (fonctions). Considère-t-on un DSP/microcontrôleur comme une fonction générale, ou un ensemble de fonctions ?*

  - Les ressources matérielles : processeurs, mémoires, I/O, bases de temps, interruptions,

  →*Tout est partagé*

  - Les interactions des commandes : accès externes,

  →*Chaque périphérique dispose de ses accès propres*

  - Les données partagées : piles et registres,

    →*La pile est commune, les registres sont exclusifs aux fonctions, mais sont tous accessibles par le programme.*

  - Les mécanismes de protection et modes de défauts.

→*Les mécanismes de protections sont adaptés à chacun, mais le code est exécuté sur des ressources communes.*

- **§2.3.3 : la supervision :** c'est la surveillance d'une fonction.

→*La fonction, est-ce la fonction globale du DSP, ou chaque fonction qu'il remplit ?*

  - La supervision a le degré de sécurité le plus élevé du système,

    →*Le dispositif de supervision est donc soit purement logiciel « software » (DO-178B Level B/C) ou soit logiciel câblé « Firmware », type EPLD, (DO-254 Level B/C)*
  - Les défauts doivent être détectés quel que soit les conditions,

    →*Si le dispositif est logiciel, il est indépendant des ressources disponibles (priorités), des états de fonctionnements (démarrages, I.T., standby, sauts…)*
  - Les mécanismes de protections ne sont pas affectés par la panne qui cause le défaut.

    →*Le dispositif ne peut pas être interrompu par un bogue logiciel, c'est-à-dire que le dispositif de supervision n'exécute pas son code avec le code fonctionnel : problème des ressources du DSP, sauf watchdog.*

- **§6.3.3 : analyse de l'architecture du logiciel :** cela doit confirmer que l'architecture atteint les objectifs :

→*L'étape d'analyse détecte et reporte les erreurs introduites durant le développement.*

  - (a) La compatibilité avec les critères de hauts niveaux, comme les fonctions proches de l'intégrité du système (techniques de partitionnement),

    →*On revient au problème du partitionnement de la fonction 'DSP' ou des fonctions du DSP*
  - (b) L'uniformité (entre les flux de données et de contrôles),

    →*Les formats soft/hard et inversement, ou entres fonctions doivent être clairement identifiés*
  - (c) La compatibilité avec la cible, comme les initialisations, les opérations asynchrones, les synchronisations et les interruptions.

    →*Le logiciel est développé pour la cible. On n'est pas dans le cas du logiciel générique qui peut être exécuté sur n'importe quelle cible.*

- **§6.3.4 : analyse des codes sources :** les codes sources répondent à toutes les règles de codage

→*Les règles sont : propres au langage, aux spécifications, à la cible, aux marges…*

  - (a) Tous codes implémentent des fonctions documentées,

    →*Explications par documentation, et dans code source*
  - (c) Tout état ou toute structure est vérifiable, sans que le code soit altéré par le test,

→*En interruption, on peut suivre l'exécution dans les temps morts. Il faut néanmoins veiller aux données locales, non visibles par les outils de développements.*

- (f) La conformité et la précision : utilisation de la pile, exactitude et conformité des dépassements et des résolutions des calculs en virgule fixe, des ressources, des temps critiques, des sauts d'exceptions, l'utilisation de variables ou constantes non initialisées, non utilisées, corruption de données suite à un conflit de tâches ou d'interruptions.

  →*Ces points ne semblent a priori pas contraignants. C'est tout à fait vérifiable.*

- **§6.4.2.2 : robustesse :** le logiciel doit avoir la capacité de répondre à une entrée ou une condition anormale.

→*Chaque fonction dispose de dispositifs de vérifications*

- (a) On exerce des valeurs invalides sur les variables réelles et entières,

  →*Tests concernant les données des correcteurs notamment. Cela peut être anticipé en simulation.*

- (b) On initialise le système dans des conditions anormales,

  →*Tests avec une mécanique aléatoire*

- (c) On doit déterminer des modes de défauts sur une donnée entrante (en provenance d'un système extérieur),

  →*Analyse des trames qui sont les seules à pouvoir créer des fonctionnements erronés.*

- (d) Les boucles à compteurs finis sont robustes aux valeurs hors limites,

  →*Concerne toutes les bases de temps (I.T. et compteurs logiciel)*

- (e) Avoir un mécanisme de protection des trames trop longues,

  →*Cela fait parti de l'analyse des trames*

- (f) Pour les fonctions dépendantes du temps (filtres, retards, intégrateurs), il faut être robuste aux dépassements arithmétiques,

  →*Ces tests sont anticipés en simulation (bloc Simulink® - Texas Instruments™)*

- (g) Pour les transitions d'état, transitions non autorisées ?

  →*Machines à états finis ?*

- **§11.7 : Standards de conception logicielle :** règles pour le développement de l'architecture

→*Cela regroupe les règles, méthodes et outils*

- (c) conditions imposées sur les méthodes de conception permises : utilisation des interruptions et les architectures à évènements, tâches dynamiques, réentrencité, variables globales, sauts d'exceptions et leurs logiques,

*→L'usage du DSP C2x requiert une manipulation de variables globales, donc on a la contrainte que n'importe quelle variable est accessible par chaque fonction. On a l'avantage qu'il n'y a aucune création dynamique, hormis l'exécution du code en RAM, et la création des variables à l'initialisation. On ne fonctionne qu'en interruptions, mais elles ne sont pas réentrantes.*

- (e) Contraintes sur les exclusions de récursivité, les objets dynamiques, les alias, les expressions compactées.

  *→Le code est rendu plus lisible avec les alias. Ils sont tous définis dans des fichiers réservés.*

- **§11.9 : données requises du logiciel :** C'est la définition des critères de haut niveau :

  *→Ce sont des critères de conception.*

  - (a) Description des allocations du système au logiciel, en faisant attention aux critères de sûreté et les conditions potentielles de pannes,

    *→On borne l'utilisation de chaque périphérique, et son interaction avec le code exécuté.*
  - (b) Critères opérationnels et fonctionnels sous chaque mode d'opération,
  - (c) Critères de performances (précision et exactitude),

    *→Fréquence des I.T. et temps de traitement. Compromis avec la bande passante du système.*
  - (d) Critères temporels et contraintes,

    *→Relations avec les entrées/sorties, synchronismes. Exemple avec la synchronisation de l'ADC (pVIII)*
  - (e) Contraintes de taille mémoire,

    *→Taille du code stocké, et du code en cours d'exécution. Toutes les variables sont allouées lors de la conception. La taille mémoire est déterminée à chaque instant.*
  - (f) Interfaces Hard/Soft (protocoles, formats, Fréquences des entrées et des sorties),

    *→Les interfaces sont définies à la conception. Les interfaces logicielles/matérielles sont intégrées au DSP. Seuls les formats sont liés aux grandeurs extérieures.*
  - (g) Détection des pannes et des critères de surveillance pour la sécurité,

    *→Les entrées sont vérifiées avant tout traitement.*
  - (h) Critères de partitionnement.

    *→Tout est défini lors de la conception initiale.*

Comme on peut le constater, la DO-178B donne des indications à respecter. Cette norme est donc une norme de haut niveau pour la mise en place des méthodes, comment lier ces méthodes, lesquelles appliquer et à quel moment. La norme ABD100 d'Airbus™ est bien mieux adaptée pour les informations techniques. Des extraits sont étudiés au second chapitre.

# II. Sûreté de fonctionnement, hardware et software

Dans le thème de la sécurité de fonctionnement, on attend du système un fonctionnement qui n'engendre pas de dommages quels que soient les cas de figures. On parle aussi bien des dysfonctionnements engendrés par une architecture insuffisamment éprouvée, que d'un résultat de traitement avec des cas non prévus, ou bien d'un dysfonctionnement dû à une perturbation du système. La perturbation extérieure la plus redoutée en numérique aéronautique est le SEU, dont le détail est donné au §II.A.1.

## *A. Les SEU*

### 1. L'existence du SEU crée une difficulté actuelle

L'introduction de logiciel dans les convertisseurs statiques s'accompagne de nouveaux problèmes, dont fait partie la sensibilité aux particules provenant de l'espace, tels que les neutrons. En effet, on appelle SEU (Single Event Upset) l'effet indésirable sur les matériels informatiques dû à ces particules, car ils touchent essentiellement les mémoires, de part leurs technologies et leurs tailles. On met en relief ci-dessous les trois types récurrents :

- Un registre : c'est une mémoire statique dédiée à un usage unique, liée à un périphérique ou une mémoire système très rapide et incontrôlable par une autre ressource, car c'est elle la mémoire de plus haut niveau (ex : les paramètres de passage en pile). Les registres ont dans l'ensemble une petite taille et ont des répercussions directes sur le déroulement du programme, un ou plusieurs bits qui changent d'états compromettent fortement l'exécution d'un programme.
- Une RAM : c'est une mémoire dynamique où les données de calcul peuvent être erronées en changeant un ou plusieurs bits. En général, les données (au sens des variables) en RAM sont rafraîchies et écrasées de façon cyclique et à intervalles très courts. Par exemple, une donnée corrompue d'un asservissement sera perçue comme un « glitch », et sera filtrée par l'algorithme. L'impact d'une erreur est donc à relativiser. Par contre, une erreur d'un code exécuté en RAM sera d'une conséquence de toute autre nature car cela change les instructions, les adresses, etc.

- Une ROM : des données du programme altérées peuvent conduire à un bogue système selon le bit touché : un bit de variable peut changer sa valeur, cela peut être gênant ou non ; un bit de saut d'adresse conduit à un échec d'exécution.

Contrairement aux composants analogiques et numériques discrets qui ont une taille et une énergie de fonctionnement significative, les composants numériques « très intégrés » ont une énergie élémentaire (pour chaque cellule mémoire) proche des énergies des particules [DOD03]. Ainsi, une particule qui traverse une cellule mémoire peut changer l'état énergétique de cette dernière, donc altérer l'état mémoire. L'aéronautique est un secteur touché par ce phénomène car l'altitude est un facteur prépondérant sur la densité des SEU (voir Fig. 9).

Les SEU sont une catégorie de défauts des S.E.E. (Single Event Effect [MAJ95]) issus des particules redoutées car on ne peut pas les détecter, les canaliser ou les éviter. Ces particules ont été mises en évidence ces 20 dernières années suite à différents incidents. En 1988-1989, IBM™ a conduit des études sur des SRAM sur plusieurs avions, avant de se joindre à Boeing. Cet industriel a mis en place un laboratoire distinct nommé BREL [RLAB] sur ces thématiques, disposant d'un réseau de compétence sur différents sites, notamment le WNR à Los Alamos, sur les neutrons atmosphériques. Ainsi, il est démontré que les SEU sont corrélés avec le flux de neutrons atmosphérique [NOR93] [NOR95] [NOR96] [NOR97] [NOR98] [OBE93]. Un SEU provoque un changement d'état d'un bit. Les radiations qui en sont la source sont :

- concentrées dans la ceinture magnétique terrestre et affecte le passage des satellites,
- présentes dans l'espace via les rayons cosmiques galactiques, très énergétiques mais moins concentrés. L'effet est alors bien plus grave qu'un simple changement d'état,
- issues du soleil dont l'activité varie (heures, jours…). Il produit des électrons, des protons et d'autres particules moins énergétiques.

Les particules des radiations atmosphériques sont (voir Fig. 9):

- les photons,
- les particules chargées,
- les neutrons.

Ce sont les neutrons qui créent des SEU car il n'existe pas de blindage pour l'avion, et la quantité de protons et de pions n'est pas significative devant le nombre les neutrons. Le flux de neutrons est fonction de l'altitude et de la latitude. Les neutrons sont créés par l'interaction des rayons cosmiques avec l'$O_2$ et le $N_2$ dans l'air. Les densités sont :

- 1.4 Neutron/cm².sec à 60000 pieds (pic), soit 18.2 km d'altitude
- 0.4 Neutrons/cm².sec à 30000 pieds, soit 9.1 km d'altitude

- 0.004 Neutrons/cm².sec au sol

Le changement d'état dans le silicium n'est pas du à la ionisation directe due au neutron, mais plus à la réaction nucléaire dans le silicium. Les énergies <1MeV[1] sont sans effet pour le silicium. Pour l'aéronautique, les mesures laissent à penser que le spectre des énergies est compris entre 1MeV et 10MeV, en fonction de l'altitude et de la latitude. (On notera qu'on peut avoir des énergies >1000 MeV en dehors du spectre qui nous concerne).

A titre d'exemple, la normalisation est donnée pour une altitude de 40000 pieds et une latitude de 40°. On a alors une densité de 0.89 neutrons/cm².sec dans l'intervalle 1-10 MeV. Dans le pire cas de vol (Paris – Anchorage), on a 2.4 neutrons/cm².sec (8600 n/cm².h) dans la gamme 1-800 MeV d'après les sources Thales Avionics™/SE.

Pour les essais en laboratoire par exemple, la section efficace des neutrons atmosphériques est comparable à celle des protons pour des énergies >50 MeV.

Pour un A320 d'un vol d'une heure, avec un bombardement de 1.4 neutrons/cm².s (pour altitude d'un vol court), le bombardement total est de 5000 neutrons/cm² pour une énergie de 1MeV. Pour un A340 d'un vol de 8 heures, avec un bombardement de 2.4 neutrons/cm².s (pour l'altitude d'un vol long), le bombardement total est de 70000 neutrons/cm² pour une énergie de 1MeV. Pour la comparaison avec les résultats donnés en Fig. 12 provenant du laboratoire TRIUMF - UBC, on donne les énergies en MeV et les densités de bombardement pour les conditions des tests et des essais qui permettent de reproduire les effets en laboratoire.

[1] L'eV est une énergie cinétique acquise par un électron en passant dans une différence de potentiel de 1V dans le vide, soit 1eV = $1{,}6.10^{-19}$J. L'eV n'est pas une unité SI mais est acceptée par le CIPM. Dans la suite du document, on fait la distinction entre cette énergie et l'irradiation.

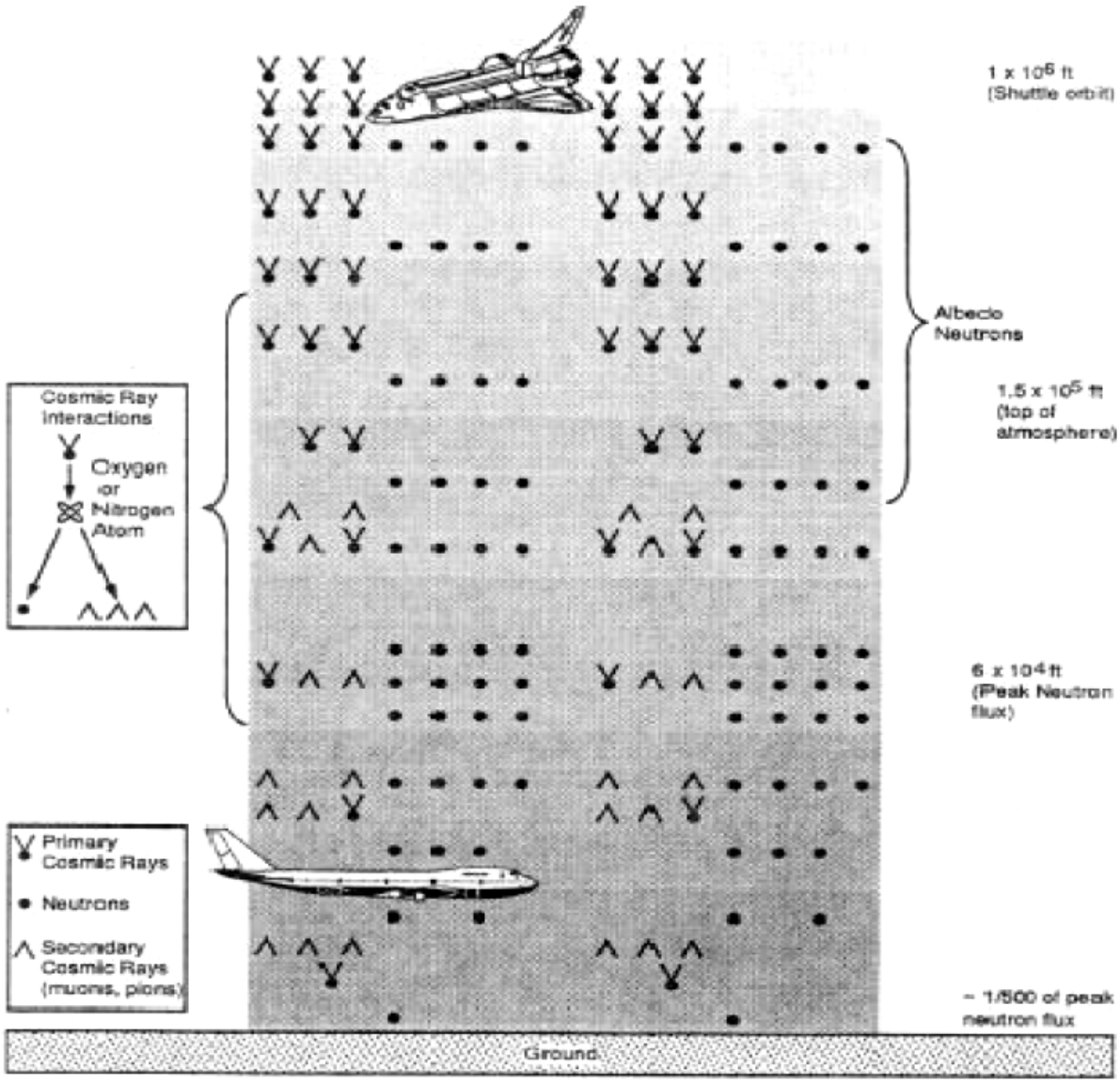



Fig. 9 : Densité de neutrons et altitude [ABD100]

L'aéronautique se situe là où les particules sont éclatées (voir Fig. 9), d'où la problématique des neutrons. Le vol de croisière d'un A340 est de 13000m, soit 43000 pieds, donc tout juste dans la tranche d'irradiation.

Les énergies aux pôles sont 6 fois supérieures aux énergies à l'équateur, à cause du champ magnétique terrestre qui protège mieux l'équateur. A la latitude de 60°, le flux est constant ; 40° sera la latitude de référence. On a donc des graphes corrélés aux altitudes et aux latitudes pour le flux de neutrons (voir Fig. 10 et Fig. 11).

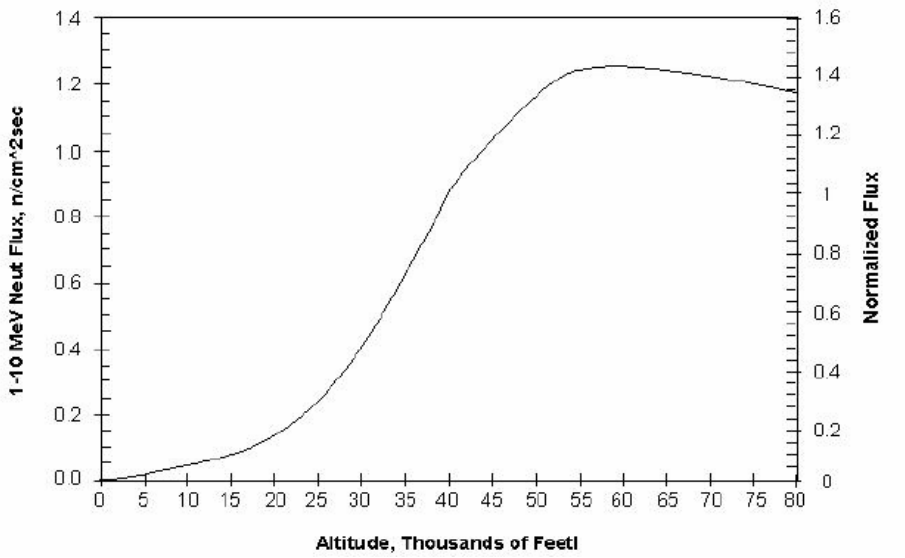



Fig. 10 : Energie des neutrons en fonction de l'altitude [ABD100]

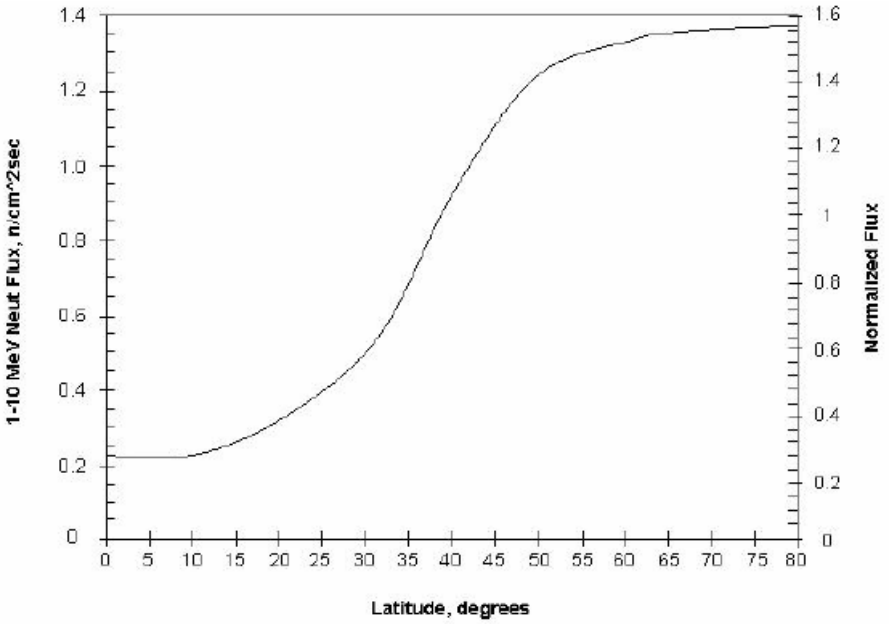



Fig. 11 : Energie des neutrons en fonction de la latitude [ABD100]

Les effets sur les mémoires sont particulièrement visibles et conséquents dans les systèmes embarqués des avions. Autant les anciennes technologies étaient robustes, autant les actuelles sont sensibles, et autant les prochaines seront problématiques. Le constat est clair, on ne pourra pas faire évoluer les systèmes numériques discrets des convertisseurs statiques actuels par une simple miniaturisation des fonctions de commande. Il est nécessaire de concevoir des systèmes numériques qui prennent en compte des aléas de fonctionnements imprévisibles, puisque l'on ne peut pas s'en protéger, avec un blindage par exemple.

## 2. Le SEU et le monde numérique

En ce qui concerne les commandes numériques par FPGA, Thales™, ses sous-traitants et ses concurrents ont l'expérience et le recul sur les techniques à mettre en œuvre, comme la « triplication » des fonctions sensibles, le checksum[2] des mémoires pour vérifier leur intégrité, les bits d'extension des trames et les protocoles spécifiques pour les détections et les corrections d'erreurs. C'est l'énorme avantage du FPGA qui permet de concevoir des commandes typiques, et dédiées aux systèmes aéronautiques. Cependant, le FPGA à un inconvénient, c'est son coût, aussi bien au niveau du composant qu'au niveau de sa mise en œuvre, tant en conception qu'en certification. De plus, le FPGA sera physiquement assez gros à partir du moment où l'on aura besoin d'une mémoire significative.

Avec le DSP, on peut répondre à la contrainte du coût, mais on n'a pas la souplesse de conception du FPGA. De même, la technologie du composant est assez différente. On est alors confronté à une autre problématique, la sensibilité de la mémoire n'est plus la même, et les moyens de rendre le système robuste seront différents. Pour cette partie, on se concentrera sur son fonctionnement en milieu hostile, c'est-à-dire dans un environnement où les SEU sont présents. Les DSP font déjà partie de nombreux systèmes de traitement du signal en aéronautique, mais souvent sous forme de cartes pour des traitements de signaux dédiés dans un calculateur, où les algorithmes se partagent du temps d'exécution. Les moyens pour rendre le système robuste sont multiples, comme les associations de composants, et des algorithmes spécifiques dédiés aux traitements des signaux. Notre travail se situant dans la commande « éloignée » des convertisseurs statiques, on demande au DSP d'être quasiment intégré à la commande rapprochée. La problématique « système » est tout à fait différente, car on a aussi besoin d'une commande minimaliste afin de pouvoir l'intégrer à la carte de puissance. Le format du calculateur (en rack) n'est pas du tout adapté. De même, si on fait une commande avec un DSP, c'est pour ne pas avoir d'autres composants. Il n'est pas envisager de coupler le DSP avec d'autres composants de performances équivalentes (type FPGA) pour assurer le fonctionnement de l'ensemble. C'est un problème d'encombrement, de complexité, mais aussi de coût. S'il avait fallu un FPGA, dans ce cas, autant en prendre un qui soit un peu plus gros si besoin, plutôt que d'avoir deux composants numériques ! La robustesse aux SEU envisagée est bien pour un DSP

[2] Le checksum (somme de contrôle) est un type de code correcteur (simple et rapide), concept de la « théorie des codes ». C'est un cas particulier de contrôle par redondance. Un « checksum » des plus connus et répandus est le bit de parité, ne permettant la détection que d'une seule erreur.

intégré à l'application des convertisseurs statiques, avec les contraintes d'un fonctionnement sur les évènements matériels du convertisseur.

## 3. Le DSP, un composant numérique typique

Contrairement au FPGA qui a des fonctions créées sur mesure par le concepteur (avec les moyens associés de « triplication », de checksum, etc.), le DSP a une architecture figée en ce qui concerne les fonctions numériques. En effet, on ne parle pas de la disposition sur le silicium des fonctions mémoires, I/O, etc. des deux composants, bien que cette disposition ne soit pas anodine. Ainsi, le DSP a une structure figée qui ne permet pas de tripler les fonctions, on ne peut pas rajouter de mécanismes sur le traitement des données (ex : checksum), sauf bien sûr de façon logicielle, mais après il faudrait continuer la boucle pour être sûr que ce mécanisme ne soit pas lui-même corrompu. Le DSP est conçu pour un usage spécifique, c'est pour cela que l'on distingue les DSP de traitements de signaux (ex : audio et vidéo), des DSP dédiés à des applications comme l'électronique de puissance. Quand on choisit un DSP, on ne peut pas choisir l'architecture la plus adaptée pour mettre en œuvre des mécanismes de sécurité de fonctionnement. Ainsi, l'usage des évènements dans notre famille de DSP est voulue et orientée par notre application. Les mécanismes à mettre en œuvre pour la robustesse aux SEU seront à explorer.

## 4. Le SEU, une présence inéluctable, une cohabitation exigée

La quantification des SEU permet de mettre en œuvre des techniques pour éprouver les composants, et plus directement de classer les composants en fonction de leurs susceptibilités. Les études de Thales Avionics™ mettent en avant la robustesse de la flash. La RAM est quant à elle sensible aux SEU, que ce soit de la RAM, SRAM, SDRAM, DRAM, DDRAM ou une autre version de mémoire vive. La RAM du DSP fait parti des RAM en général, donc a priori des mémoires sensibles. Tout le design du projet est orienté vers du code efficace, c'est-à-dire exécuté en RAM interne. Compte tenu de nos marges d'utilisation du processeur, il est possible de s'affranchir de la RAM pour le code critique en l'exécutant directement à partir de la mémoire flash. La RAM ne serait alors utilisée que pour les variables qui sont périodiquement écrasées et n'altérant pas le code fonctionnel. Le DSP a cet avantage sur le FPGA qu'il peut fonctionner à partir de la mémoire flash, alors que le FPGA charge la « RAM » à partir de la flash/ROM déportée pour le « boot ». Il reste le problème de la « pile » qui est de la RAM critique. Peut-on se suffire d'un redémarrage en cas d'anomalie ? Cela fait partie du level « C ».

Une caractérisation des effets sur un DSP est donnée dans la publication [HIE05]. Le test est réalisé sur un DSP TMS320 C6701 de version militaire (nuance du boîtier céramique), issue de la famille C6000 très performante. Le notre est un C2000, mais on reste dans la même gamme, les TMS320. On ne donnera en Fig. 12 que des résultats issus de la synthèse de la publication.

L'expérience a permis de mettre en avant la robustesse du DSP soumis à un faisceau de 120 MeV, avec une dose totale d'irradiation pendant le test de 16.6 Krad(SI)[3] sur la ligne TRIUMF PIF MDA Ontario. Le protocole de test simule un fonctionnement en orbite à 556km.
Voici les résultats :

| Elément testé | Erreurs |
|---|---|
| Registres internes | Aucune |
| ALU virgule fixe | Aucune |
| ALU virgule flottante | Aucune |
| DMA | Aucune |
| Echantillons dynamiques en RAM (on les change régulièrement), avec[4] erreur avant fin du test | $1.5 \times 10^{-2}$ upsets/day |
| Echantillons dynamiques en RAM (on les change régulièrement), sans erreur avant fin du test | $2.2 \times 10^{-2}$ upsets/day |
| Echantillons statiques en RAM (écris au début du test), avec erreur avant fin du test | $1.9 \times 10^{-2}$ upsets/day |
| Echantillons statiques en RAM (écris au début du test), sans erreur avant fin du test | $2.9 \times 10^{-2}$ upsets/day |

Fig. 12 : Synthèse des erreurs du DSP6701 en orbite dans l'espace

La publication conclue sur le fait que ce DSP est utilisable dans l'espace pour bon nombre d'applications, en veillant à mettre en œuvre des corrections automatiques sur la SRAM. Apparemment, seule la RAM est sensible aux SEU. On en déduit que les registres sont fiables, et c'est intéressant pour l'application du radar. La RAM du C6000 a été testé à au moins la fréquence du C2000. Il faudrait maintenant avoir les tests équivalents dans l'atmosphère, donc avec l'énergie adéquate des particules.

[3] Le Krad(SI) ou 'Krd(SI)' est la dose de radiation absorbée dans les unités hors CGS, dont son équivalent est le gray (Gy), 1Gy = 100rd = 1J/kg = 1 $m^2/s^2$.

[4] Les erreurs (avec ou sans) concernent le protocole de test. Le test peut être figé en fonction de bits erronés, donc on distingue les erreurs avec un test achevé et les erreurs avec un test avorté.

## *B. Le monitoring*

### 1. Le principe

Le monitoring est un concept bien connu en informatique, et en informatique industrielle. Il existe beaucoup d'ouvrages relatifs à la sécurité et la sûreté de fonctionnement logicielle. Le monitoring est sur le fond identique d'une application à l'autre, mais il est décliné en autant de manières de mise en pratique possible que d'applications à monitorer. En effet, le monitoring surveille et/ou analyse des données ou/et des comportements des applications [QIU00] [TIN03]. En ce sens, les dispositifs de monitoring sont spécifiques aux grandeurs associées.

La publication [PET02] « requirements-based monitors for real time systems » est une description haut niveau du principe de monitoring. La méthode s'applique assez bien à notre problématique de logiciel embarqué, qui est un programme interfacé directement avec du « matériel ». En soi, ce n'est pas une révolution, mais le DSP n'est pas souvent cité dans les exemples d'applications aéronautiques, et encore moins en termes de sûreté de fonctionnement et de normes ; ainsi, on va faire le point sur les techniques applicables.

En cas d'erreur, il se produira un effet non souhaité et non souhaitable. L'erreur va générer un effet qui se décline en plusieurs niveaux, qui chacun traduit une criticité. On trouvera l'erreur catastrophique, sévère, majeure et mineure. Dans tous les cas, l'erreur doit être détectable et détectée. Comme le projet s'inscrit dans une criticité 'C' de la DO178B, en cas de problème majeur ou plus du RTSU (carte servomécanisme pour l'objet de l'étude), on peut basculer d'un RTSU à l'autre afin d'isoler le matériel défectueux (principe de redondance). Ainsi, une défaillance majeure ou plus n'aura pas d'effet sur les systèmes connexes (notamment sur le PFC en tête et les moteurs). Le DSP est à la base un microcontrôleur, et est ainsi doté d'un watchdog. Cet appendice, intégré au silicium du DSP, est un circuit en étroite relation avec le cœur permettant de réaliser un « reset » général. Le watchdog est configuré au démarrage de la cible, et fonctionne de façon autonome en parallèle du DSP. Lors de la configuration, le concepteur choisit comment surveiller le programme. Si au bout d'un certain time-out (périodique), le watchdog n'est pas rafraîchi, on considère qu'une erreur est survenue, et le DSP redémarre. Pour minimiser l'impact d'une erreur, la fréquence de la vérification est la plus élevée possible. C'est donc un choix arbitraire du concepteur pour avoir un compromis entre de bonnes performances du système et des tests qui détectent toutes les erreurs à effets importants. D'après [PET02], on peut résumer tout système selon le schéma suivant :

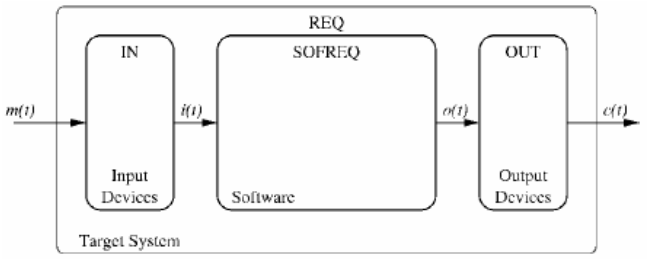



Fig. 13 : Système logiciel générique à monitorer [PET02]

Définitions des noms :

- m(t) : environnement monitoré ;
- c(t) : environnement contrôlé ;
- i(t) : entrée de la solution numérique (registres, ADC, I/O) ;
- o(t) : sortie de la solution numérique (registres, PWM (pVIII), I/O) ;
- input devices : capteurs, boutons ;
- ouput devices : actuateurs, amplificateurs, relais, affichage ;
- SOFREQ : software requirements, caractérise le jeu de fonctionnements acceptables du logiciel ;
- REQ : requirements, caractérise le fonctionnement du système. REQ(m, c) doit être vrai.

Le monitoring peut soit vérifier un fonctionnement pendant l'exécution du programme de la cible, soit utiliser un enregistrement du fonctionnement. Le monitoring est fonction du temps écoulé, et non du temps absolu. Dans tous les cas, le monitoring doit donner un retour de tous les fonctionnements passés (en particulier s'ils ont été enregistrés).

## 2. Sa mise en œuvre

Dans notre cas, le monitoring doit empêcher un effet indésirable sur le matériel, donc dans la surveillance « instantanée », où l'on est en phase avec l'exécution du code pour maximiser les temps de réactions. On peut ensuite faire la différence entre le monitoring logiciel et le monitoring système :

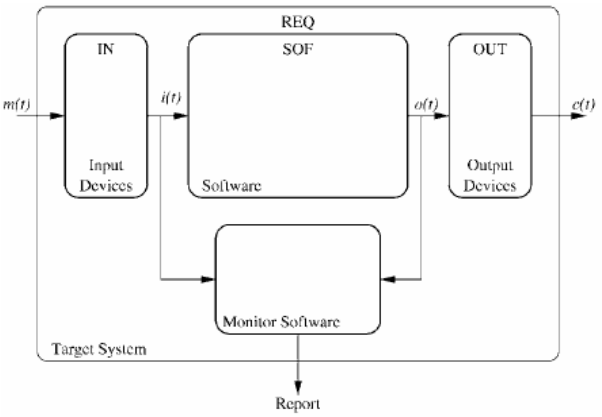



Fig. 14 : Monitoring logiciel [PET02]

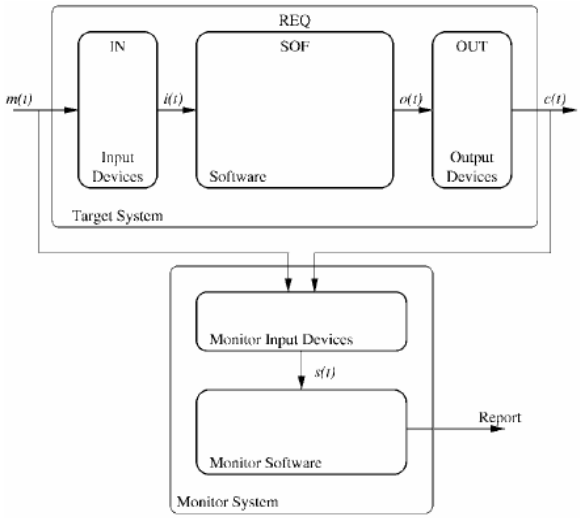



Fig. 15 : Monitoring système [PET02]

Dans notre cas, le « report » (ou le rapport d'erreur) peut être traité par l'extérieur (calculateur), ou directement par l'antenne. Le traitement « in-situ » est plutôt adapté aux erreurs majeures ou plus, qui peuvent engendrer une défaillance matérielle significative. Ainsi, on pourrait avoir un temps de réaction bien plus rapide. Le monitoring aurait aussi un pouvoir de décision.

- Le watchdog du DSP est un monitoring logiciel intégré, qui prend la décision immédiate de relancer le logiciel. Le temps de réponse est minimal. Il n'y a aucune mémorisation de l'erreur.
- Les C-BIT sont des monitoring logiciels, qui renvoient un rapport à une fréquence fixe, donc d'une certaine façon, c'est un monitoring avec enregistrement car $F_{CBIT} < F_{max\ logiciel}$, et c'est le calculateur qui prend la décision relative au défaut constaté. Le temps de réponse peut être grand.
- Le watchdog de l'EPLD (pVIII) est un monitoring logiciel, avec un mécanisme et un composant distinct du DSP. L'EPLD, qui assurent des fonctions logiques et d'interfaces (voir Fig. 18 et Fig. 25) sur le système, peut redémarrer le DSP si une condition n'est pas respectée. Le DSP et l'EPLD ont des données en commun, donc l'EPLD a accès aux données internes du DSP (celles qui sont prévues dans la conception).

Pour avoir un monitoring système, il faudrait avoir un dispositif de surveillance un peu équivalent au DSP, car la sortie est issue d'un traitement complexe des entrées ; donc pour prédire la sortie, il faudrait traiter des calculs équivalents au DSP. On revient donc à une sorte de redondance totale (pour avoir les mêmes performances du monitoring face au DSP), qui est contradictoire au besoin d'un système compact. Le monitoring « système » peut nécessiter des

capteurs supplémentaires, si par exemple des grandeurs relatives au bon fonctionnement ne sont pas directement traitées par le système principal.

Le monitoring tolère une certaine liberté de fonctionnement sur les données qu'il analyse. Plus les données scrutées peuvent donner des combinaisons complexes de fonctionnement, plus la liberté sera importante, et moins le monitoring sera efficace. En effet, un bon résultat peut être la combinaison de mauvaises données. Quand il y a trop de cas possibles, on ne peut plus justement distinguer tous les cas possibles.

## 3. Exemple de monitoring

On identifie/qualifie un monitoring selon la démarche présentée dans [PET02], dont l'illustration est donnée en Fig. 16 :

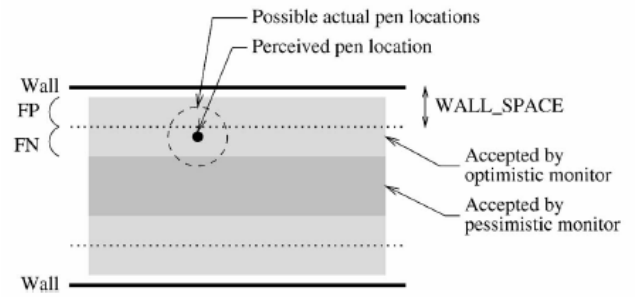



Fig. 16 : Bornes optimistes et pessimistes du monitoring [PET02]

Cette représentation illustre les bornes du monitoring par une analogie avec un stylo sur une table traçante (robot traceur de la publication [PET02]) : les marges sont volontairement exagérées.

- wall : limites de la feuille. Ici, c'est une limite mécanique, limites de fonctionnements.
- FP : précision de monitoring «possible false positive », entre la limite wall_space et wall.
- FN : précision de monitoring «possible false negative », entre la limite wall_space et la zone pessimiste.
- WALL_SPACE : marge de sécurité, fonction de l'erreur d'estimation de la position du stylo.
- Perceived pen location : position mesurée du stylo.
- Possible actual pen locations : positions possibles du stylo selon les erreurs de mesures.
- Accepted by optimistic monitor : comportement normal.

- Accepted by pessimistic monitor : comportement dans la limite basse tolérée qui n'introduit pas de défaillance.

Après avoir défini les bornes du monitoring, il faut définir sa réactivité. En effet, le monitoring est dimensionné par son temps de réponse qui est fonction du temps de traitement du système. Le chronogramme Fig. 17 illustre cet aspect :

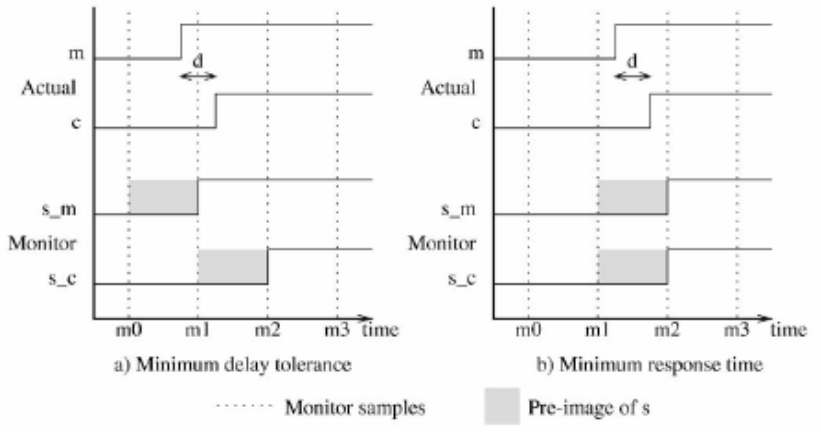



Fig. 17 : Précision des timings du monitoring [PET02]

On voit que l'échantillonnage du monitoring introduit un délai (cas a), qui est un temps où une défaillance peut s'opérer sans qu'il n'y ait aucune détection, ou alors si ce temps est trop grand, le monitoring ne voit pas le changement et détecte une fausse défaillance. Le temps « d » et le temps d'échantillonnage du monitoring sont corrélés. Le cas (b) montre un monitoring qui voit un changement simultané de la sortie sur l'entrée. Cela peut être considéré comme une fausse erreur puisque le temps de traitement « d » est masqué par l'échantillonnage du monitoring. Le monitoring doit tolérer des marges sur les exécutions, sans que ces marges soient pénalisantes pour la détection de défaillances à effets rapides. Les performances du monitoring peuvent donc être élevées.

## 4. Transposition de l'exemple à l'application du radar

Dans notre application, l'analogie du stylo n'est pas si éloignée dans le cas des butées mécaniques :

- wall : butées mécaniques de l'antenne.
- FP : précision de monitoring «possible false positive », entre la limite wall_space et wall.
- FN : précision de monitoring «possible false negative », entre la limite wall_space et la zone pessimiste.

- WALL_SPACE : marge fonction de la vitesse instantanée et des attitudes du porteur pour l'angle d'arrêt.
- Perceived location : position mesurée de l'antenne.
- Possible actual locations : positions possibles de l'antenne relatives au retard de traitement (la position évolue entre les échantillonnages).
- Accepted by optimistic monitor : comportement normal.
- Accepted by pessimistic monitor : comportement dans la limite basse tolérée qui n'introduit pas de défaillance, aucun effet sur la mécanique.

Et pour les temps de réponse :

- « d » est le retard introduit par l'asservissement, soit 1ms (échelle fixe des calculs), ou c'est le temps pour mesurer une réponse en vitesse par rapport à une consigne d'arrêt par exemple (et le temps est fonction de plein de paramètres).
- « m0, m1, m2… » sont les échantillonnages du C-BIT et l'algorithme de sécurité qui surveillent les butées

Les watchdog n'ont aucun effet sur la gestion des butées (comportement mécanique), mais uniquement sur le comportement électrique (erreur générale et sortie PWM fixe par exemple→court-circuit momentané).

Dans l'analyse plus concrète de notre système, on reprend la démarche précédente appliquée à la carte de servomécanismes et non au DSP seul car on s'intéresse au mode de défaillance général comme l'illustre le schéma en Fig. 18 :

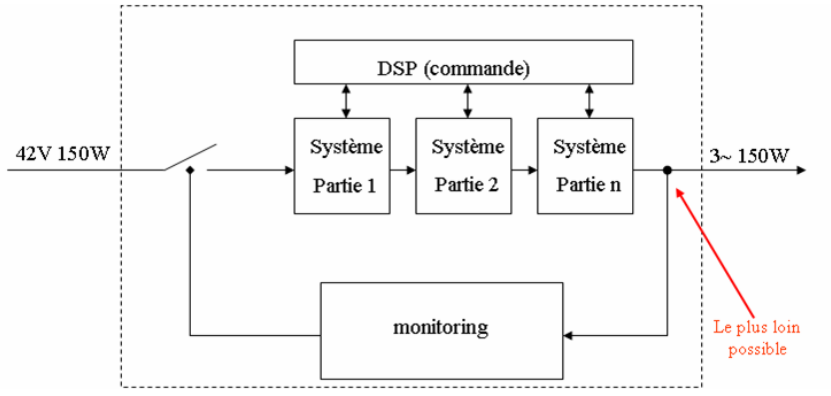



Fig. 18 : Monitoring général

Afin de détecter toutes les anomalies pouvant engendrer une défaillance majeure, donc les fonctionnements en dehors de la plage maximale de fonctionnement, on cherche à détecter ces anomalies aux extrémités du système puisque l'on est sur une stratégie de redondance du RTSU. Si on ne surveille pas suffisamment loin, on risque de laisser des défauts. Cela pose alors le problème de la réutilisation des capteurs existants, puisqu'il faut qu'ils soient en bout de chaîne.

La méthode de la surveillance logicielle est intéressante, mais si on procède à l'analyse ci-dessus, on se rend compte que l'on peut facilement tomber dans le piège où les interactions avec le système ne sont pas les plus judicieuses. En effet, si on utilise l'EPLD comme circuit logique de « buffer » des signaux du DSP, entre la commande des drivers (partie 2 sur Fig. 18 et Fig. 19) et les semi-conducteurs de puissance (partie 3 sur Fig. 18 et Fig. 19), où l'EPLD a un pouvoir de décision en fonction du fonctionnement du DSP, on insère l'EPLD dans la chaîne de traitement au lieu de le mettre en parallèle sur l'ensemble :

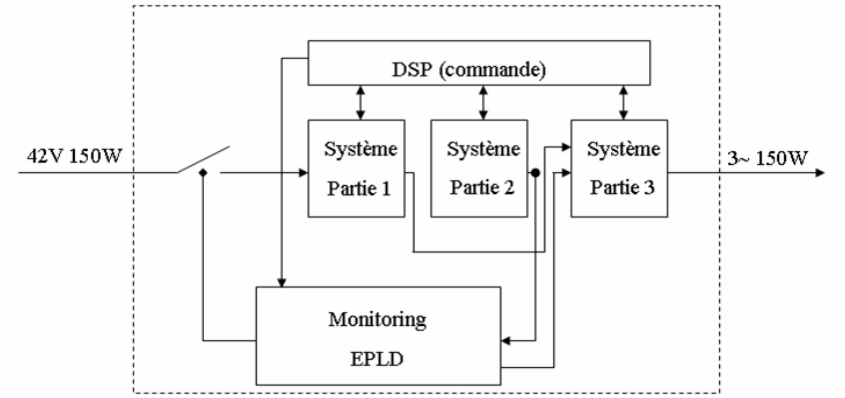



Fig. 19 : Monitoring entrelacé

Ainsi, par l'entremêlement des signaux, on n'a plus deux systèmes distincts, et la dernière partie de la chaîne n'est pas surveillée. On peut éventuellement détecter une anomalie sur l'entrée qui n'aurait pas de cause connue.

## *C. L'analyse du fonctionnement*

L'analyse du fonctionnement [BON03] fait partie des points incontournables en vue d'une certification. Un système aéronautique ne sera autorisé à être utilisé que si l'on peut prouver son fonctionnement, comment il a été développé, et comment ont été menés les tests et les méthodes pour garantir une conception comme prévu dans les règles de développements. Ainsi, nous allons discuter du fonctionnement évènementiel (par interruptions) qui est un point délicat pour la certification, car les autres fonctionnements des systèmes numériques sont déterministes, c'est-à-dire que l'exécution du programme est connu à chaque instant (ex : par un séquenceur), sans évènements. Il faut donc déployer un argumentaire et une analyse qui permettent de justifier d'un fonctionnement certain du système.

Comme il en était question dans la synthèse des risques, nous allons mettre en œuvre des méthodes éprouvées de l'industrie sur notre problématique de commande numérique.

## 1. Le fonctionnement évènementiel

Le fonctionnement en interruptions est par définition un fonctionnement qui est interrompu. Contrairement à un séquenceur cadencé par une horloge qui alloue du temps aux différentes fonctions (par différents procédés), un programme en interruptions est composé de routines d'interruptions indépendantes les unes des autres. Elles sont indépendantes dans leurs modes de déclenchement, d'arrêt, mais elles s'échangent ou se partagent des données, donc elles sont dépendantes les unes des autres en ce qui concerne le fonctionnement du système. Il faut donc bien dissocier les deux analyses [RIC03a] [SPU96]:

- l'analyse des évènements, des déclenchements, des arrêts et des priorités : c'est la gestion des ressources, les conflits matériels, les crashs du système, les blocages. Les temps d'exécutions de chaque fonction sont primordiaux, donc comment peut-on garantir l'exécution de toutes les fonctions dans un temps donné?
- l'analyse des flux : c'est la cohérence des données, le risque du bogue, les mauvaises utilisations. Il faut prendre en compte le fait du non séquencement, mais de l'enchaînement des fonctions (et non des tâches au sens du séquenceur). En effet, quel est l'impact sur les données manipulées si les interruptions sont glissantes les unes avec les autres ?

Ces deux analyses prennent en compte le caractère de l'évènement, c'est-à-dire si c'est un évènement périodique ou non, ce qui change fondamentalement le caractère de l'interruption. On rappel ces caractères temporels :

- Périodique : l'exécution de la tâche a lieu dans un intervalle régulier. On connaît le pire temps d'exécution de la tâche (temps maximal), sa période et son deadline.
- Apériodique : c'est une tâche exécutée aléatoirement. Son apparition est inconnue lors de la conception.
- Sporadique : elle arrive arbitrairement, mais avec un intervalle minimum entre deux occurrences. Elle a un WCET (pVIII) et un deadline (échéance stricte). Ses attributs sont connus lors de la conception.

Les différentes analyses sont sensibles au déterminisme, donc une interruption apériodique est prohibée. Il n'en est pas moins possible de continuer avec les autres. Avec une approche de type RMA (pVIII), on procède à l'analyse temporelle des pires temps de réponse (WCET) [RIC03b] pour que n'importe qu'elle tâche puisse s'exécuter dans son temps imparti. L'analyse est proche de ce qui a été dit plus tôt, lorsque l'on s'intéressait au fonctionnement du DSP. On rappelle les critères suivants et on étaye sur le fonctionnement DSP type C2x :

- restrictions sur le matériel :
  - durées d'exécution des instructions fixes ou bornées : c'est le mode de fonctionnement normal du DSP où toutes les instructions sont fixes ainsi que le mode d'adressage.
  - pas de mémoire cache : la mémoire du DSP est conçue de telle manière que la RAM est une ressource constante et unique pour les calculs. L'utilisation des registres internes de l'ALU est prédéterminée en câblé, donc cela ne constitue pas une mémoire cache (développé au second chapitre).
  - pas de pagination : la pagination du DSP est une pagination de conception, où l'on identifie les différentes zones mémoires, différente de celle évoquée ici qui est une pagination dynamique des données pour occuper la RAM au mieux. Cela n'est pas utilisé dans le DSP.
  - pas de swapping : il n'y a pas de phénomène de déplacement de données car rien n'est dynamique. Tous les espaces mémoires sont réservés et figés dès le démarrage.
  - pas de pipeline : le pipeline fait parti des attributs majeurs du DSP, mais c'est un mécanisme constant et permanent, dont l'utilisation est définie à la compilation. Le pipeline n'est pas dynamique en fonction des demande d'accès type DMA d'un PC.
- restrictions sur le logiciel
  - pas de récursivité : le code a un accès unique à chaque instant. Il n'a pas de multiples utilisations et de fonctions réentrantes.
  - pas de déclarations dynamiques : toutes les variables sont créées au démarrage, et les mémoires sont allouées à ce moment là.
  - pas de goto, break… : ce langage n'est pas compatible avec une exécution sans sauts conditionnels et sans sauts sans retour.
  - pas de float, long : le programme est réalisé en virgule fixe dans le format du DSP. Le « long » est utilisé car c'est un format prévu dans le DSP, donc la restriction ne s'applique pas.
  - appels explicites des procédures ou des fonctions : tous les appels aux fonctions sont fixes et constants, avec un mode d'adressage identique et complet. Il n'y a pas d'ambiguïté.
  - limitation des boucles (durée bornée ou nombre d'itérations borné) : toutes les itérations sont bornées selon les règles de codage en vigueur. Les durées sont

aussi bornées avec ces mêmes règles. Seul le code qui n'est pas en interruption est infini, donc la restriction est hors contexte.

A priori, bien que le DSP soit sans RTOS, l'utilisation des ressources est saine. On peut continuer l'analyse sur les interruptions.

Dans l'application, le DSP est le seul organe de contrôle de l'antenne (partie servomécanismes), donc il gère toutes les entrées/sorties (au sens fonctionnel). A ce titre, chaque entrée/sortie dispose de ses propres contraintes. On a différentes interfaces qui sont les amplificateurs, les communications, les capteurs. Chaque « famille » de fonction a ses propres périphériques, des mécanismes « hardware » sont mis en œuvre afin de bénéficier d'une capacité de traitement maximisée des informations et des signaux (d'où un processeur de traitement du signal). Ainsi, on va trouver dans le fonctionnement du DSP des bases de temps distinctes et des périphériques dédiés, indépendants les uns des autres.

Les périphériques n'ont pas tous la même importance, que l'on qualifie soit sur le plan fonctionnel (dans la chaîne fonctionnelle du système), soit sur le plan sécurité de fonctionnement (importance de la défaillance d'un périphérique). En effet, un périphérique peut être défaillant sans qu'il remette en cause l'intégrité des dispositifs, ou inversement une défaillance furtive peut avoir un impact significatif sur la fiabilité et la durée de vie du matériel.

Le DSP est un composant utilisable de multiples manières, soit avec un séquenceur dédié à l'application, soit avec un RTOS (donc un système d'exploitation pour les composants embarqués), soit en interruptions directes.

- le séquenceur/scheduler [LIU73] dédié à l'application est basé sur une conception simpliste d'un RTOS, où celui-ci est réduit à sa plus simple expression. On accède aux différents périphériques par scrutation des entrées, et les sorties sont rafraîchies au rythme du séquenceur. Le séquenceur est plutôt d'une fréquence élevée car les « tâches[5] » doivent s'enchaîner très vite pour donner l'illusion du temps réel ;
- le RTOS est un système de gestion temps réel des tâches, et embarqué dans la cible avec les tâches. Il est conçu sur mesure pour la cible (Texas Instrument à son propre RTOS : DSP BIOS (pVIII)). Ces RTOS ne sont généralement pas certifiés. Les systèmes certifiés sont plus accessibles pour des stations temps réel où la station a tellement d'usages multiples dans l'industrie qu'elle est réutilisable, comme le RTOS. La conception du système est faite en fonction du RTOS utilisé. Les performances temps réel sont alors

[5] Le terme « tâche » est employé abusivement dans ce document car c'est souvent d'une fonction ou d'une routine dont on parle. Cela permet cependant d'homogénéiser la lecture pour comprendre les différentes variantes.

très hautes. Le désavantage est que le RTOS consomme des ressources au détriment des tâches, et c'est difficile sur une cible dont les capacités sont limitées ;

- le fonctionnement en interruptions directes est ce qui se rapproche le plus du « temps réel », car les tâches disposent de 100% des ressources de la cible, mais la conception des « tâches » requiert une conception en prenant en compte les ressources matérielles. Il n'y a plus de distinctions entre les tâches, les drivers et le RTOS. Concrètement, les drivers et les tâches forment des routines d'interruptions, et le « RTOS » fictif est réalisé par les « flags hardware » et les évènements, dont l'architecture est laissée aux soins du développeur. C'est pour cela que chaque conception est unique et que le code n'est pas transportable.

On notera qu'il y a une différence assez significative entre un ordonnanceur/séquenceur et des interruptions directes. L'ordonnanceur temps réel va essayer de remplir au maximum le CPU pour faire un maximum de tâches afin de donner une illusion de temps réel. Avec des interruptions directes, on va essayer d'avoir la marge de temps (CPU non utilisé) la plus forte possible pour maintenir/garantir le temps réel [THI04]. Ceci requiert donc une analyse des temps, qui apparaît essentielle.

## 2. L'analyse temporelle (approche RMA)

L'analyse RMA (Rate Monotonic Analysis) est une méthode d'analyse temporelle des temps d'exécution. Cela peut être une méthode assez lourde à mettre en œuvre selon la complexité du système et du type de tâches utilisées. On retrouve l'utilisation des méthodes RMA dans les systèmes à ordonnancement [AZZ04] [KAL04], où cette méthode prend tout son sens avec les stratégies de partage du temps qui créent tout un éventail d'enchaînement des tâches. Cet enchaînement dynamique induit des temps d'exécutions différents, ce qui implique de déterminer le temps maximal. C'est une analyse temporelle où le but final est bien de connaître le pire cas, et non de donner toutes les stratégies possibles. Les méthodes d'analyses permettent de calculer le pire temps en fonction de la stratégie de l'ordonnanceur, de la date d'activation, de la pire durée d'exécution, du délai critique (temps maximal entre l'activation et la terminaison) et de la période.

### 3. L'analyse fonctionnelle (approche AMDEC)

L'analyse AMDEC (ou FMECA) est un outil [AMDEC] très générique à appliquer à n'importe quel système en vue de dresser un panorama sur la sûreté de fonctionnement du système. C'est une étude dont les axes significatifs sont choisis par le concepteur, tel que l'analyse de maintenance, des procédures, des défaillances, des points critiques.

Malgré son caractère très générique, la démarche générale permet d'adapter l'analyse à son projet, en choisissant bien les points pertinents. Ainsi, dans le cas du système à base du DSP, on s'intéresse à la fois au composant et à ses modes de défaillances, ainsi qu'à l'architecture du programme qui lui est dédié. L'analyse RMA, qui est très ciblé à la problématique du temps d'exécution, va être complété par l'analyse AMDEC dont le but est de vérifier la cohérence du système, ses interactions fonctionnelles et ses modes de fonctionnements pour soulever des modes de défaillances, leurs effets et leurs criticité ; le but étant de donner des solutions ou de borner les cas de fonctionnement. Cette analyse sera développée dans le second chapitre sur l'exemple de la gestion des données.

## III.Bilan des normes et analyses

L'usage des normes est essentiel et obligatoire pour fournir un travail cohérent et utilisable. Il faut cependant avoir un recul et une vue d'ensemble sur les normes pour en avoir une interprétation juste et efficace. Tout l'aspect documentaire et la rigueur du projet doivent être abordés en amont. Le travail de la thèse s'associe à cette démarche. Le caractère novateur, quant à l'utilisation du DSP, ne doit pas être entravé pour toutes ces normes qui à cause d'un usage « d'habitudes » peut masquer la réelle possibilité d'exploitation de composants qui ne sont pas dans les référentiels usuels et conventionnels des dispositifs (notamment lors des développements).

En ayant étudié la problématique de l'usage du DSP dans les systèmes embarqués temps réel (SETR) évoqués par les normes, on remarque qu'il n'est pas du tout prohibé d'utiliser des dispositifs à base de DSP, à partir du moment où l'on est capable de justifier n'importe quel choix et de garantir n'importe quel cas de fonctionnement. Ce n'est alors plus qu'une question de mise en œuvre des moyens de tests et de vérifications adaptés au dispositif en question. Les moyens abordés ici ont été mis en œuvre dans les chapitres suivants, même si tous les détails ne sont pas fournis afin de ne pas faire dériver les thématiques des chapitres suivants.

# Chapitre 2 : modélisation, simulation et conception de l'ensemble

# I. Le système

## A. La problématique de l'antenne

L'antenne aéroportée est un système en plusieurs parties dont les servomécanismes, les amplificateurs, les calculateurs et les interfaces utilisateurs. Cette thèse concerne les servomécanismes et les amplificateurs des moteurs. On notera que l'amplificateur hyperfréquence sera intégré à proximité des autres. L'antenne est un assemblage mécanique assez sophistiqué, situé à la pointe de l'avion (voir Fig. 20), derrière le radôme. Celle-ci est soumise aux contraintes environnementales car le radôme n'est pas une protection isolante.


Fig. 20 : Emplacement du radar aéroporté

Le radar remplit la fonction « météo », ce n'est pas un radar de vision comme ceux que l'on peut rencontrer dans les avions de combat. Il permet de détecter des conditions météorologiques, ainsi le niveau de criticité « C » suffit. C'est une antenne à balayage mécanique, donc deux axes de mouvements sont nécessaires pour diriger les ondes hyperfréquences. Il faut donc une certaine précision des mouvements pour que les mesures soient traitables.

## B. L'existant

Les systèmes actuels (voir Fig. 21) permettent d'obtenir des positionnements précis de l'antenne avec des motoréducteurs. Ce sont des actionneurs mécaniques complexes qui contiennent des engrenages. Ces derniers introduisent des jeux, de l'usure et cela aura un impact sur les résonances basses fréquences qui sont du même ordre de grandeur que nos bandes passantes. En plus de l'impact sur les asservissements, on a une contrainte sur la durée de vie de l'antenne, où l'on doit prendre en compte le MTBF. Les motoréducteurs ont aussi un volume et

un poids non négligeables. Le plateau de l'antenne est relié à l'amplificateur hyperfréquences via des guides d'ondes, constitués de parties fixes, et d'un joint tournant. Celui-ci est une pièce coûteuse qui demande un entretien (maintenance préventive et corrective).


Fig. 21 : Radar actuel (concurrent)

### C. La rupture technologique

Le radar de Thales™ (voir Fig. 23) propose une rupture technologique des servomécanismes, concernant l'électromécanique et les amplificateurs. En effet, on remplace les motoréducteurs par un entraînement direct (voir Fig. 22), c'est-à-dire une prise directe entre l'axe du moteur et l'axe de rotation de l'antenne, nommé par la suite « direct-drive ». On supprime ainsi les jeux et les usures liés aux engrenages. Ensuite, les amplificateurs qui étaient déportés sont intégrés à l'antenne. Comme les amplificateurs des moteurs et de la partie hyperfréquence sont portés par la mécanique mobile, on s'affranchit d'une connectique complexe et du joint hyperfréquence.

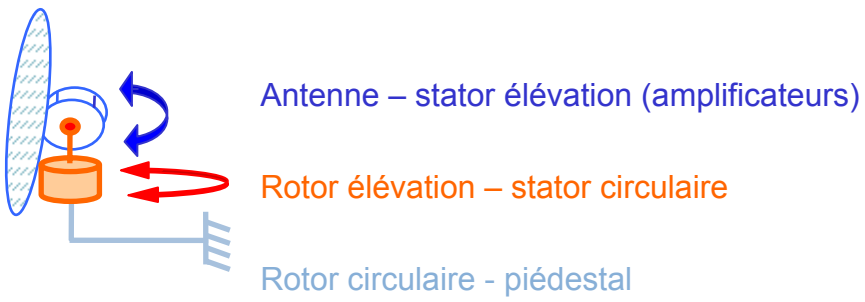



Fig. 22 : Cinématique du « direct-drive »

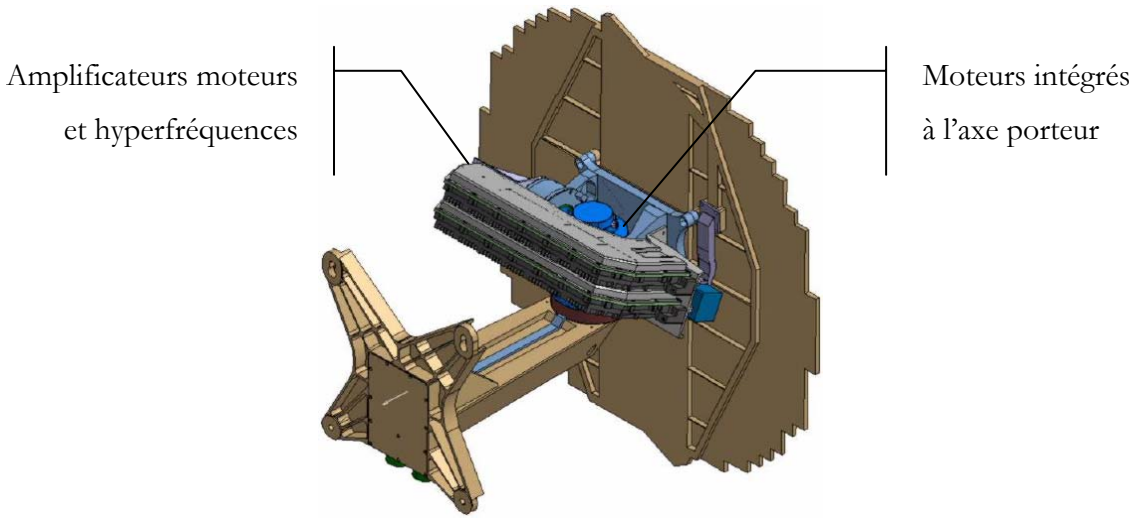



Fig. 23 : Radar « direct-drive » (image de simulation)

C'est à partir d'une stratégie mécanique destinée à augmenter la fiabilité et à diminuer le coût (d'entretien et de fabrication) qu'est apparue la problématique de l'intégration et de la conception de l'ensemble électronique. Pour parvenir à l'entraînement direct, il faut des actionneurs adaptés et une commande associée. Les moteurs sans balais « Brushless » sont une solution pour répondre à la problématique de la fiabilité. Un ensemble électromécanique a été conçu et dimensionné pour cette application : la mécanique a été construite autour d'un stator d'une machine synchrone, et des aimants permanents ont été déposés à la surface de l'axe qui est le pivot de la partie portée. Un pivot est dédié à la rotation circulaire et un pivot à la rotation en élévation. Les actionneurs sont des machines étroites, mais avec un diamètre important compte tenu des dix paires de pôles (pour la basse vitesse) et de leur couple (classe centaine de W, 3 N.m), qui permet de disposer d'un couple important pour l'entraînement direct. Le moteur circulaire a un débattement angulaire au maximum de ±90° tandis que le moteur élévation a un débattement angulaire au maximum de ±45°. Sachant que la précision de positionnement est de 0.25°, on remarque que le contrôle moteur va devoir présenter des performances assez élevées. On a la double contrainte entre une forte intégration de l'électronique dans le contrepoids de l'antenne, et une grosse capacité de contrôle – commande. Cela explique le choix de composants numériques dédiés.

Ensuite, il faut tenir compte des contraintes imposées par le contexte aéronautique. En effet, l'antenne est soumise aux vibrations, aux accélérations, aux variations de température, mais aussi à la C.E.M. (ex : [DO160]), à l'interchangeabilité, et aux normes. Les contraintes mécaniques et électriques font partie du dimensionnement du convertisseur et des asservissements, et les contraintes normatives [RCM] et fonctionnelles [REFSA] font partie des verrous techniques qui écartent certaines solutions techniques.

Les retours des positions sont obtenus par des capteurs angulaires absolus sur chaque axe, qui sont référencés au montage au 0° absolu du plateau antenne (visée droit devant). Ce sont les deux seuls capteurs ; on ne compte pas les capteurs sur les circuits imprimés. On utilise les deux capteurs de position pour les fonctions d'asservissements et pour les contrôles moteurs (se référer au dernier chapitre pour la problématique de l'indexage).

# II. Une étude de servomécanisme

## *A. Le découpage fonctionnel*

Les servomécanismes sont un ensemble constitué d'un système mécanique, d'un système électronique, et d'un système informatique [ABD01]. Vu de l'extérieur, la fonction « servo » (Fig. 24), qui est le système de pilotage de l'antenne, n'est constitué que de deux entrées que sont les capteurs de positions, et de deux sorties que sont les moteurs. La fonction est interfacée avec un calculateur pour recevoir les consignes de fonctionnement, et pour renvoyer des informations de fonctionnement.

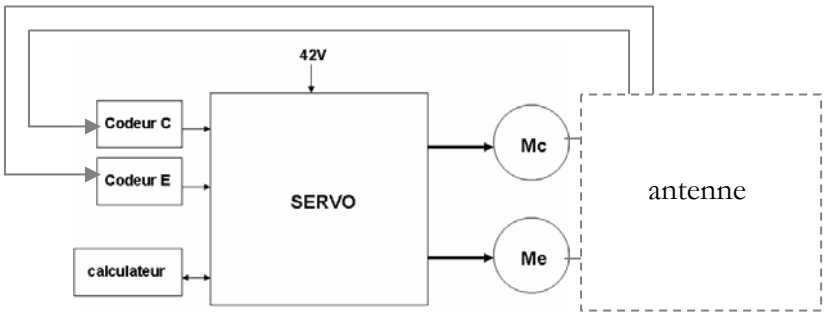



Fig. 24 : Fonction servomécanisme et I/O

La fonction « servo » est intégrée à l'antenne et fait partie d'un ensemble électronique (voir Fig. 25) dénommé RTSU, qui est le « coffret » (Unit) partagé entre les amplificateurs des moteurs (Servo) et les amplificateurs hyperfréquences (Receiver/Transmitter).

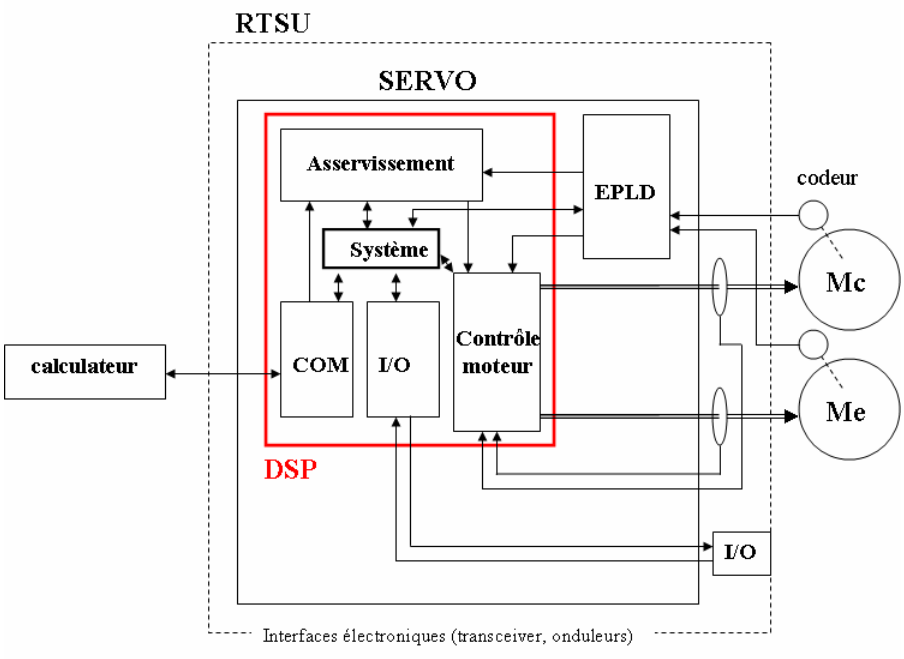



Fig. 25 : RTSU et Servo

On remarque l'EPLD aux côtés du DSP. Il remplit une double fonction : c'est à la fois une interface pour convertir les formats des capteurs, et c'est aussi un composant du monitoring. Le convertisseur statique ainsi étudié, pour la commande numérique, est étroitement lié aux servomécanismes que sont l'antenne avec ses mouvements mécaniques, ses asservissements et ses contraintes [MAR06] [OZE06] [MAU03].

## *B. Les boucles*

La fonction « servo » prend en charge à la fois le contrôle moteur des deux axes et leurs indexages, les asservissements des mouvements, les générateurs de consignes, le changement de repère, le monitoring, la communication et la gestion des butées. Chaque fonctionnalité a été simulée indépendamment. On a des machines à états finis, et des fonctions 'C' déterministes pour les parties transverses à la fonction principale que sont les asservissements mécaniques. Ce point a été abordé précédemment. La figure suivante (Fig. 26) montre les différentes fonctionnalités qui représentent l'architecture de conception dans le DSP :

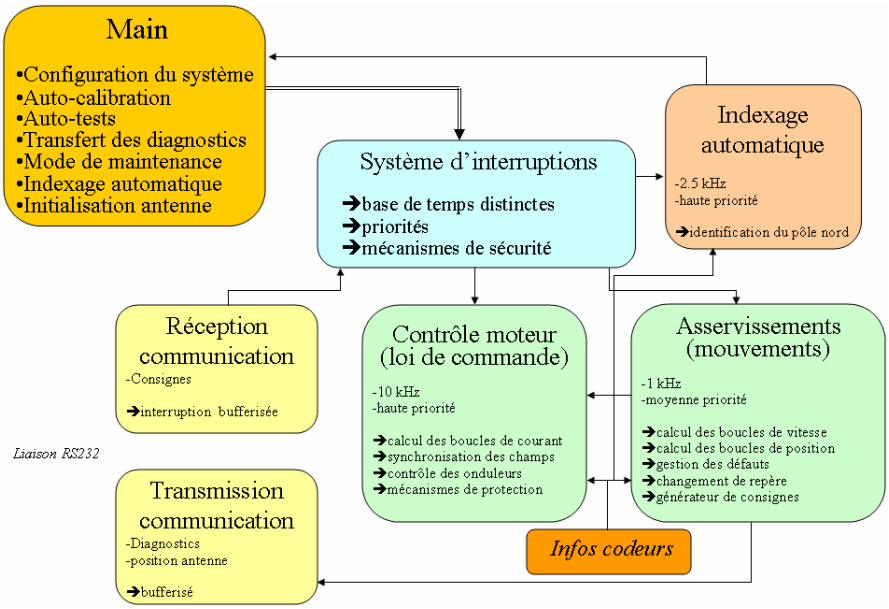



Fig. 26 : Organigramme du programme

La partie « automatique » est dimensionnante pour la commande, car elle fixe les bases de temps, les temps de calculs et les ressources nécessaires. La conception de l'architecture n'est donc pas fixée au début du design, mais découle des simulations abordées dans la prochaine partie. L'architecture (Fig. 26) ne fait pas clairement apparaître les entrelacements des fonctions, ainsi la représentation des boucles (Fig. 27) montre une imbrication des différents blocs fonctionnels.

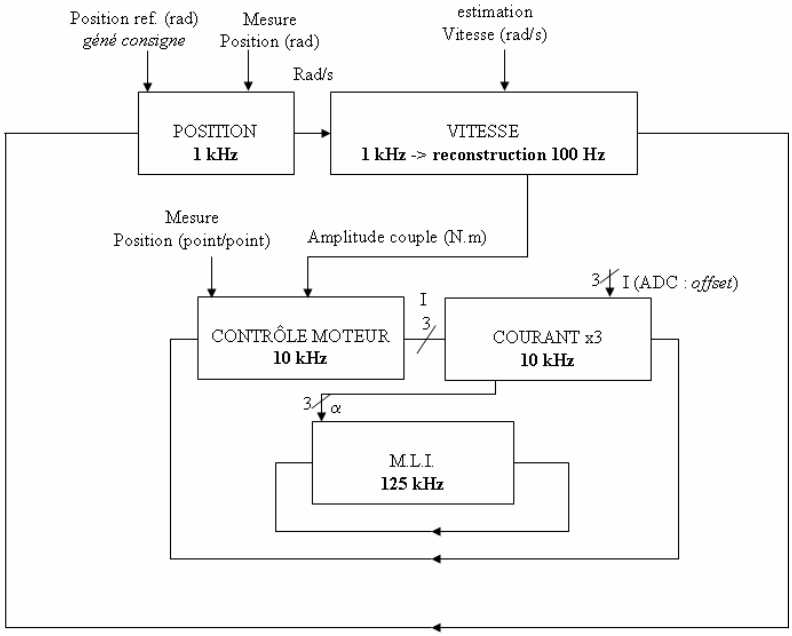



Fig. 27 : Boucles de la fonction principale

*Note : La vitesse est estimée (à 100Hz) à partir de l'échantillonnage de la position à 1 kHz, résultant d'une étude par simulation pour les besoins des asservissements.*

A partir du moment où l'on commence à fixer les bases de temps et l'architecture, on est dans une phase où l'étude des servomécanismes en simulation et la conception des programmes deviennent fortement couplées. Une étude Matlab®/Simulink® se prête bien à cet exercice puisque l'on peut simuler de grosses structures de façon itérative. On ne montre pas ici les fonctionnements transverses, notamment concernant l'indexage moteur dont le fonctionnement est différent au niveau des bases de temps, tel qu'il est montré dans l'architecture (Fig. 26).

Les boucles de courant sont échantillonnées à 10 kHz pour le contrôle des correcteurs de courant qui poursuivent les composantes sinusoïdales basses fréquences du contrôle moteur. En effet, la fréquence de découpage à 125 kHz n'est pas liée au 10 kHz (on ne travaille pas au niveau de la commutation). La fréquence de découpage est filtrée en sortie des onduleurs (avant les moteurs) afin que les moteurs ne soient pas excités par des fronts, mais par des sinus purs. Ce sont des filtres LC amortis, en série avec les inductances des moteurs. Ces filtres seront l'objet d'une problématique au dernier chapitre. Les moteurs et leurs connectiques sont situés juste derrière le plateau hyperfréquence, ce qui nécessite une certaine maîtrise des effets C.E.M. compte tenu de la très forte proximité entre les moteurs et le plateau rayonnant. On est donc contraint par la qualité des signaux, mais aussi par le contrôle moteur. C'est à partir des simulations qui intégraient aussi les filtres que l'on a obtenu toutes les bases de temps, relatives aux dimensionnements des multiples régulations.

# III.Une approche sous Matlab®/Simulink®

L'idée initiale au début du projet était de mettre en œuvre une architecture de simulation sous Matlab®/Simulink® qui permettait d'aboutir à une génération complète du code, et qui puisse être réutilisable pour les projets similaires. Le but était de pouvoir donner un outil de génération de programme des cibles DSP pour un ingénieur n'étant pas expert en informatique industrielle. Cela consistait à décrire et simuler un convertisseur statique sous Simulink® (la structure électronique, les lois de contrôles, les régulations), puis de laisser Matlab® créer le programme pour le DSP. Plusieurs problèmes ont été soulevés par cette étude, et il y a encore une étape complexe à franchir entre les perspectives de génération de code certifié, et l'outil certifiable qui demande des connaissances approfondies dans plusieurs domaines.

Ainsi, on propose une solution alternative, où l'on utilise Matlab®/Simulink® pour simuler les implémentations de la fonction principale, où l'aspect « automatique » est prépondérant, avec une importante étape de codage sous CCS™. Les fonctionnalités transverses sont abordées séparément avec d'autres outils.

Dans un premier temps, on valide le système avec une étude continue, puis on s'intéressera à la problématique des formats de la cible en échantillonné. On terminera par l'analyse du code automatique.

## *A. L'étude continue*

### 1. Le principe général

La simulation (Fig. 28) permet de dimensionner et de tester le fonctionnement des asservissements et du contrôle moteur de chaque axe à partir des modèles de la mécanique, des modèles des moteurs et des onduleurs. C'est une simulation « amont » qui est en quelque sorte l'étude de faisabilité du projet à partir des dimensionnements électromécaniques [MAH07]. On vérifie le dimensionnement des étages électroniques et mécaniques, ce qui permet d'évaluer la complexité des correcteurs et lois de commande à mettre en œuvre. C'est bien là qu'est le problème : il est très difficile d'évaluer la complexité des asservissements à mettre en œuvre, puisqu'en simulation, tout est possible, et il est difficile de se rendre compte si les lois de contrôle proposées pourront être implémentées. Cette difficulté sera abordée dans l'étude au format de la cible.

La simulation permet aussi de gérer les changements de repères qui permettent de donner des consignes dans le repère terrestre alors que l'avion évolue dans son propre repère. On prend donc en compte la fonction « servomécanismes » et les traitements associés dans un seul ensemble.

Les modèles mécaniques sont obtenus par les résultats de simulation de CATIA™ réalisés par le bureau d'étude qui a développé une antenne ayant une rupture technologique dans son architecture, soit une cinématique novatrice. Cette nouvelle architecture d'antenne donne des déformations typiques de l'antenne sous des accélérations instantanées, résultat de la rigidité du bras ; et on obtient des fréquences de résonances de la mécanique. Ces fréquences de résonance évoluent avec des géométries différentes de l'antenne (versions selon les avions). La prise en compte de modèles précis directement issus des simulations est un très gros avantage pour le dimensionnement et la mise au point. Les résultats sont transposés sous Matlab® pour être exploités sous Simulink®. La validation du système est faite avec le modèle continu.

Les blocs « 1 » et « 2 » (voir Fig. 28) représentent respectivement les axes circulaires et élévations. Leurs entrées proviennent du bloc « changement de repère » qui permet de donner les coordonnées du mouvement dans le repère sphérique alors que les consignes initiales sont dans le repère terrestre. Les sorties sont les visualisations des trajectoires et des états de fonctionnements (couples, courants…). Les blocs « 1 » et « 2 » ont leurs propres asservissements et modèles.

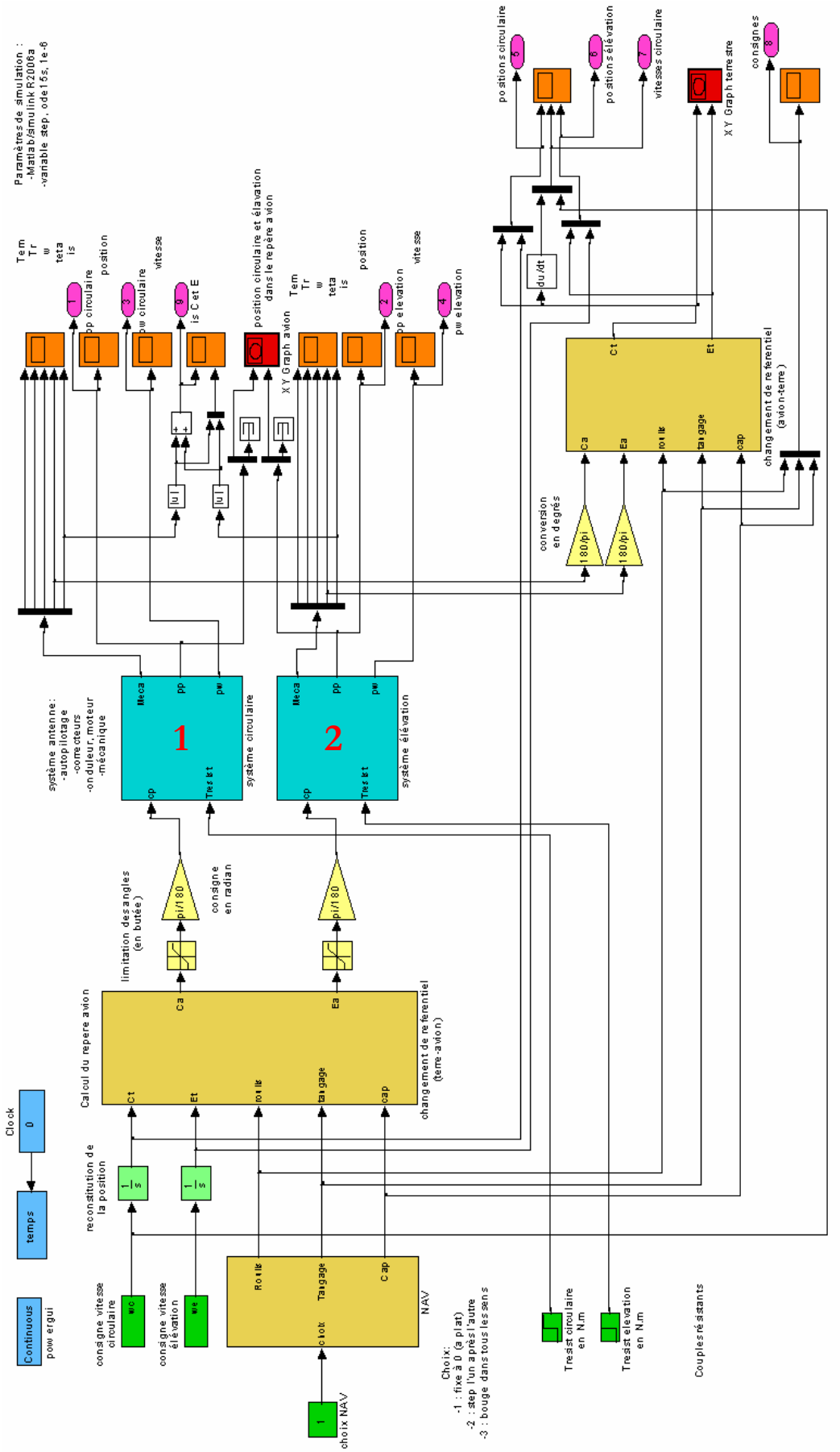



Fig. 28 : Système complet sous Simulink®

## 2. Le détail des axes circulaire et élévation

Le schéma Simulink® (Fig. 29) montre le détail de ces deux blocs principaux. Le schéma blocs de l'axe permet de distinguer les différents blocs fonctionnels tels que les asservissements et les modèles. Les modèles de la mécanique sont chargés par le bloc « machine synchrone ». Sur l'ensemble des blocs, ceux qui posent des problèmes vis-à-vis des dynamiques de calculs sont ceux des asservissements. A partir de cette description du système en continu, on peut intervenir sur chaque bloc problématique pour imposer des contraintes de conception localisées, afin d'avancer pas à pas. On donne dans la suite un exemple simple de passage d'un correcteur continu (Fig. 31) vers un correcteur échantillonné (Fig. 32) avec les contraintes d'implémentations.

Les contraintes d'implémentations des correcteurs échantillonnés sont un moyen de tester du code sans la cible. En effet, les outils Simulink® permettent d'intégrer des blocs Texas Instruments® qui sont des interfaces représentatives des librairies mathématiques en assembleur, des fonctions en C (ex : des structures de correcteurs) et des interfaces de la cible. Ainsi, la simulation prend en compte des contraintes échantillonnées qui sont celles de la cible. L'utilisation de ces blocs a aussi un autre avantage : on peut instrumenter les étapes de calcul au niveau des opérations élémentaires, ce qui dans la cible demanderait de s'introduire dans l'exécution du code de l'ALU. On pourrait alors être intrusif, perdre le temps réel, ou ne pas avoir accès aux données au moment souhaité. L'intégration des opérations mathématiques est un réel avantage. Les diverses configurations que l'on trouve dans les blocs Simulink® sont celles que l'on trouve aussi dans les librairies de la cible. Par exemple, les outils de virgule fixe permettent dans le cas d'une multiplication : soit de travailler directement avec le multiplieur, auquel cas le résultat est brut (rapide mais avec le risque de dépasser le format), soit d'utiliser le multiplieur avec un code dédié qui est un peu moins rapide, mais qui gère les dépassements et les arrondissements. Sans recherche de performances, il suffit de mettre les options au maximum, mais dans ce cas on pourra perdre inutilement beaucoup de cycles (coups d'horloge), alors qu'avec ce genre de simulations, on pourra déceler les calculs pouvant être accélérés.

Pour déceler les anomalies de fonctionnement des correcteurs échantillonnés, on peut lancer la simulation continue et hybride en parallèle (complète ou partielle), comparer des résultats, ou suivre toutes les étapes. C'est assez flexible. Dans le cas exposé au §III.B, on connaissait l'allure de la sortie préalablement étudiée en continue, et la simulation complète du système était incohérente.

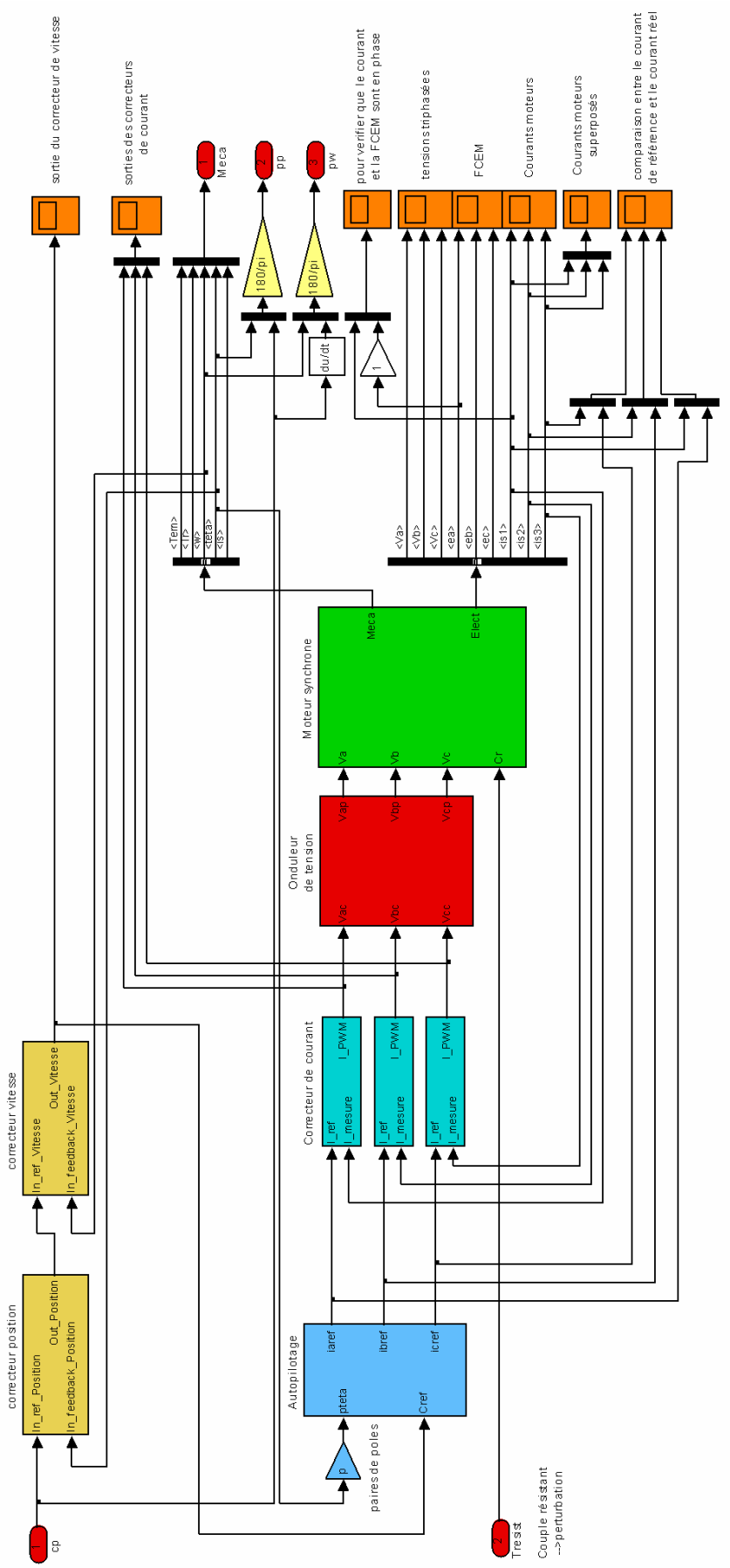



Fig. 29 : Bloc « 1 » ou « 2 »

## B. Etude échantillonnée au format du DSP

Ainsi, on remplace le correcteur continu (Fig. 30-a) par son équivalent échantillonné avec les blocs C2000© (Fig. 30-b) :


Fig. 30 : Correcteurs continu (a) et échantillonné (b)

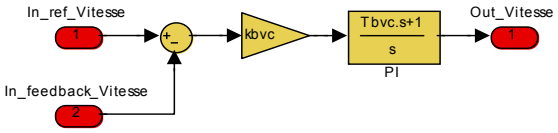



Fig. 31 : Détail du correcteur continu (Fig. 30-a)

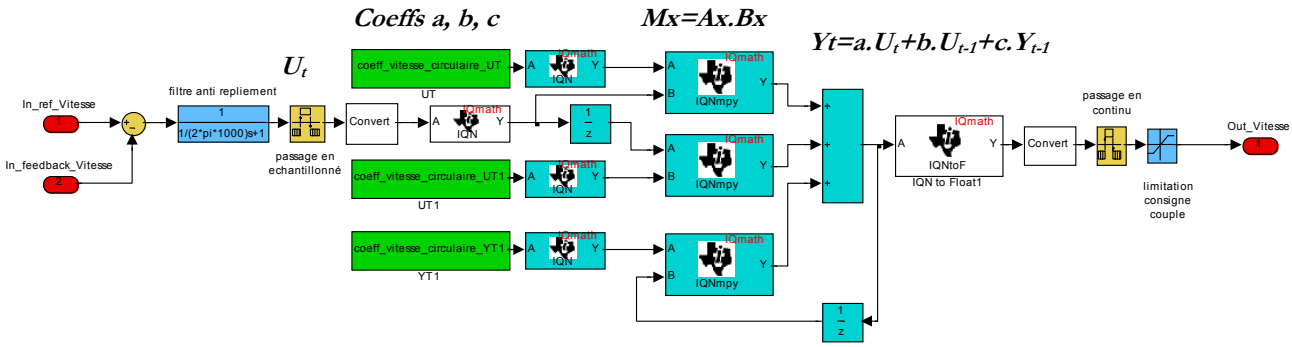



Fig. 32 : Détail du correcteur échantillonné (Fig. 30-b)

Le schéma (Fig. 33) intègre un bloc échantillonné avec les contraintes du DSP, puisque ce bloc repose sur des librairies de Texas Instruments™ [TEX] pour le composant choisi. Les correcteurs sont interchangeables car l'environnement Simulink® permet de gérer des bases de temps distinctes via des interfaces de transitions dédiées, qui permettent le passage entre des pas de calculs variables (pour le continu) et des pas fixes (pour l'échantillonné). On notera cependant que la simulation est beaucoup plus lourde à gérer pour le simulateur, et que les temps de calculs sont significativement augmentés. De plus, les précautions à prendre sont nombreuses (en ce qui concerne chaque transition et configuration de base temps et les changements de formats), et il faut des étapes intermédiaires pour tester les bons fonctionnements. C'est donc une étape délicate : c'est un point faible pour construire une simulation.

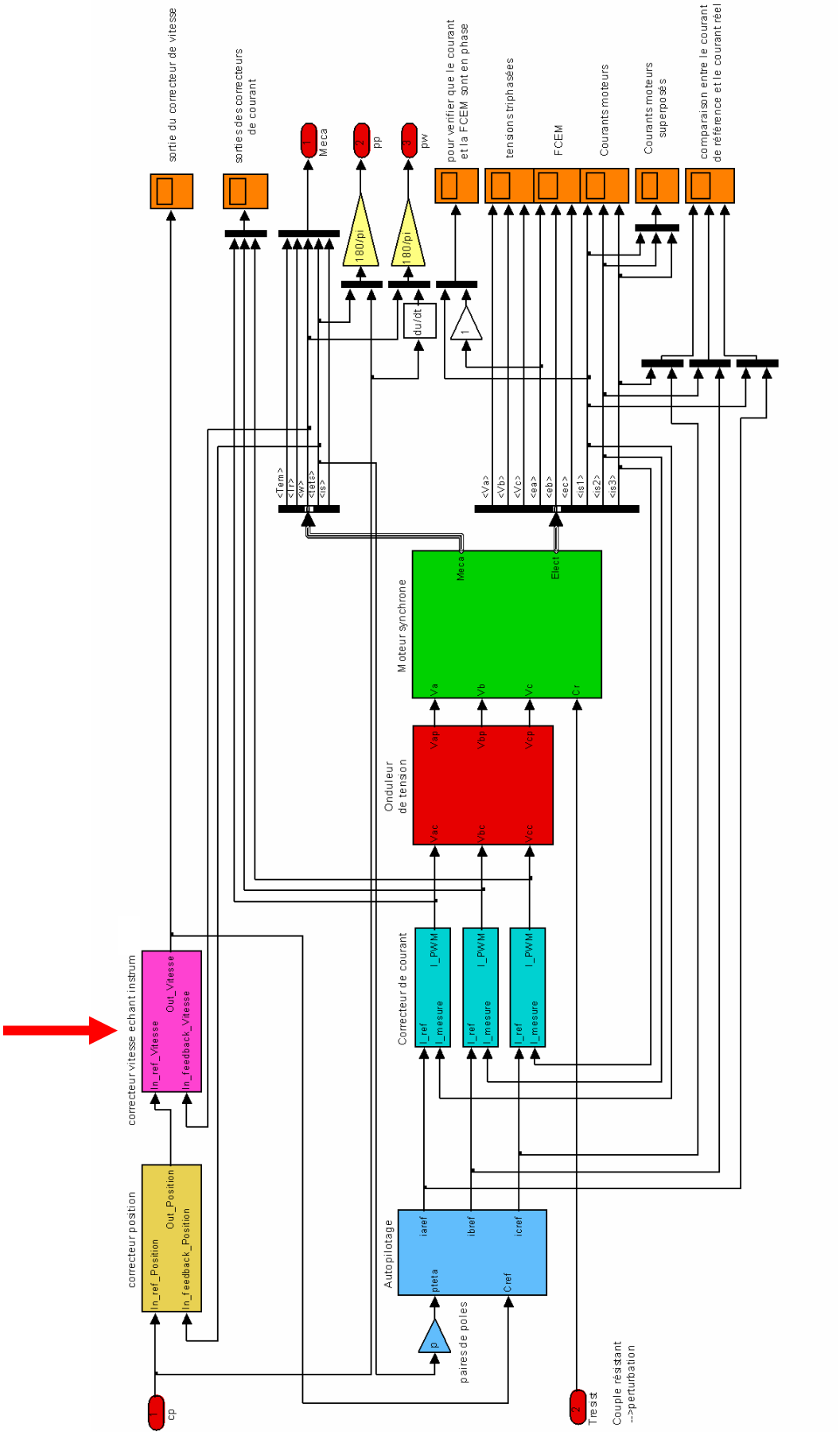



Fig. 33 : Correcteur de vitesse échantillonné dans la simulation continue

Le but d'évoluer d'une approche continue à une approche échantillonnée, au format souhaité, est de borner les dynamiques, choisir les résolutions pour que les calculs soient réalisables dans la cible. Cette étape peut tout à fait remettre en question l'étude continue, et la mise au point est itérative entre le dimensionnement en continu pour la rapidité de simulation, et la vérification en échantillonné pour confirmer ou infirmer l'implémentation.

Le relevé en Fig. 34 est le résultat de l'instrumentation du correcteur échantillonné précédemment évoqué, dans une première version issue de l'analyse continue. Le correcteur comporte trois branches (Fig. 32), que sont les multiplications d'une équation de récurrence, qui sont sommées pour obtenir la sortie. L'équation est de la forme : $Y_t = a.U_t + b.U_{t-1} + c.Y_{t-1}$ (voir Fig. 32)

Chaque branche de l'équation est visible sur le relevé en Fig. 34 (M1, M2, M3 où M1=a.$U_t$, M2=a.$U_{t-1}$, M3=a.$Y_{t-1}$). L'entrée (consigne) est excitée par une forme d'onde simple pour l'analyse. On ne le voit pas sur les schémas Simulink®, mais par des fonctions de Matlab®, on peut dérouler la simulation hybride avec les résultats de la simulation continue en parallèle. Cela permet de voir les différences de comportement entre les correcteurs continus et échantillonnés. Ainsi, lors de la comparaison, la sortie du correcteur a montré une défaillance (variation de la sortie inattendue), c'est pour cela que l'on investigue à l'intérieur du correcteur pour identifier la source d'erreur. On a M1 = a.$U_t$ avec a>0, or M1<0 avec a>0 et $U_t$ >0. C'est le même problème pour M2 et M3 qui conduit à une sortie $Y_t$ erronée.

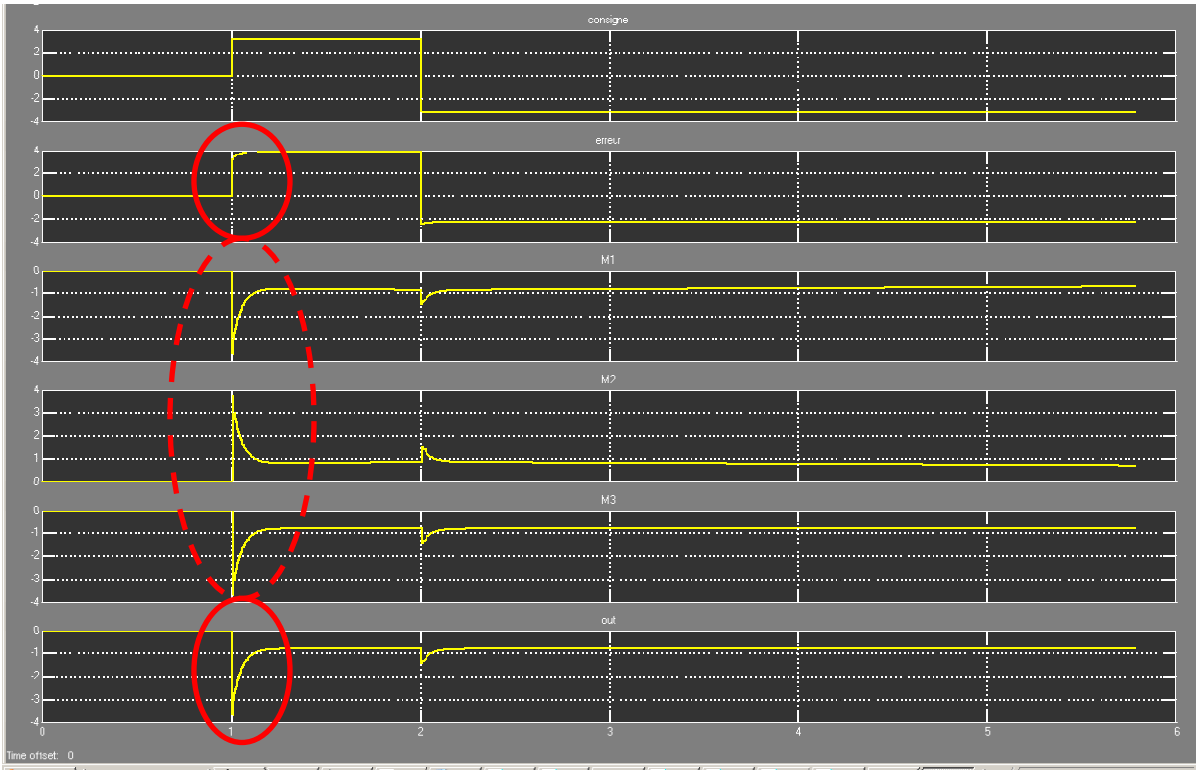



Fig. 34 : Calculs intermédiaires du correcteur dans sa première version

On remarque qu'avec les signaux forts (transition de l'erreur lors de l'échelon de consigne), on assiste à une incohérence entre l'erreur qui augmente après le glitch et la sortie qui diminue dans le négatif alors qu'elle aurait du augmenter dans le positif (voir traits pleins). C'est la conséquence d'un dépassement d'un calcul intermédiaire Mx (voir traits pointillés). Les dynamiques de calculs des simulations continues n'étaient pas bornées comme elles le sont ici. Dans tous les cas, le correcteur en question n'aura pas les mêmes performances entre une dynamique limitée (16/32 bits) et illimitée (au sens d'un nombre de bits considéré comme précision infinie).

Ensuite, on modifie les paramètres des blocs Texas Instruments™ pour réaliser cette même fonction « asservissements » en prenant en compte les cas à ne pas atteindre. Concrètement, les multiplications vont être bornées dans le « multiplieur », ce qui se traduit par un code « assembleur » différent au niveau du temps d'exécution car on va utiliser plus d'instructions. Il faut bien remarquer que dans la simulation on gère des problèmes d'implémentation qui sont « atomic », c'est-à-dire au niveau de l'ALU, les données manipulées entre le multiplieur, les accumulateurs via les instructions dédiées. On donne en Fig. 35 le relevé dans les mêmes conditions de test (qu'en Fig. 34) : On a bien pour M1 : M1>0 pour a>0 et $U_t$>0.

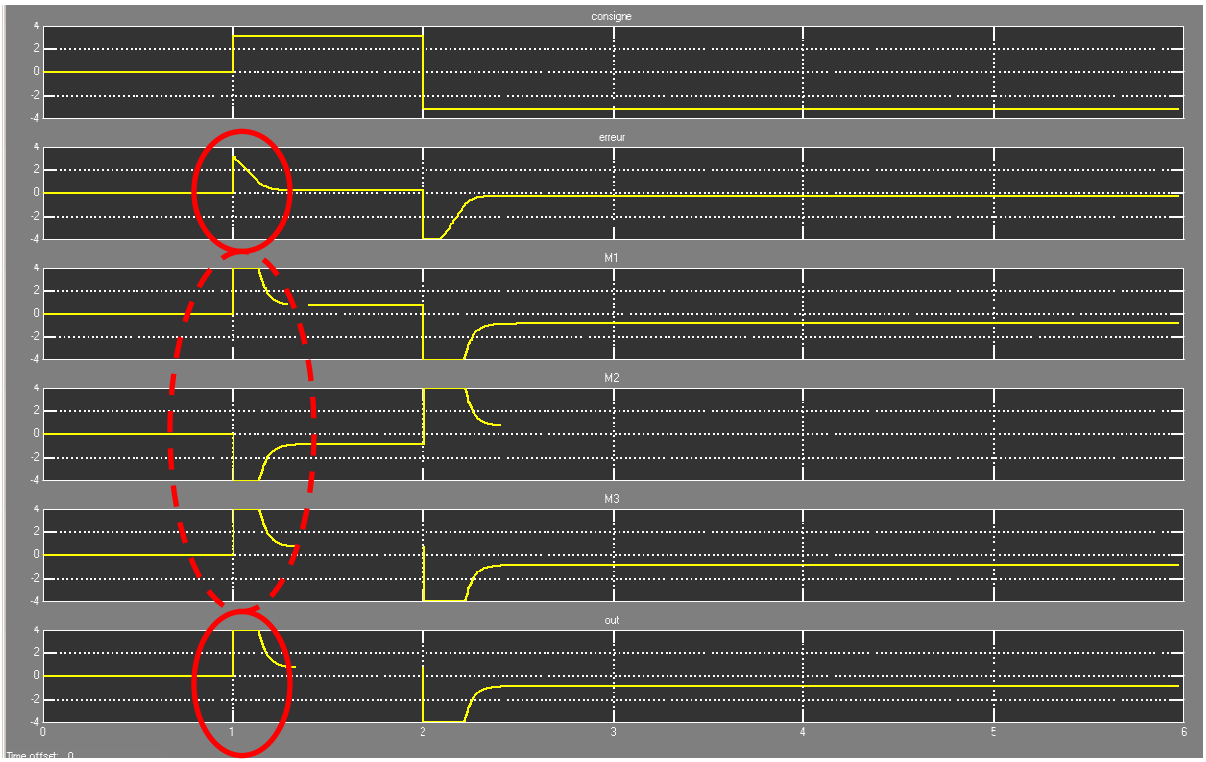



Fig. 35 : Calculs intermédiaires du correcteur dans sa seconde version

Dans cette nouvelle implémentation, les calculs intermédiaires Mx (voir traits en pointillés) sont saturés, et les calculs sont stables (voir traits pleins).

On peut alors voir directement sur la simulation système l'impact des calculs limités quant aux performances générales. Dans le cas où on aurait justement un fonctionnement dégradé, on intervient soit sur le réglage ou la structure des correcteurs, soit sur la manière de réaliser les calculs. Ces étapes ont été réalisées mais ne sont pas représentées, on n'a ici que l'illustration des résultats sur un cas simple.

On voit en Fig. 36 (par le cerclage et son zoom) la différence entre les correcteurs continus et échantillonnés de vitesse, avec la consigne et la réponse : la réponse n'est pas tout à fait la même entre la simulation continue et échantillonnée.

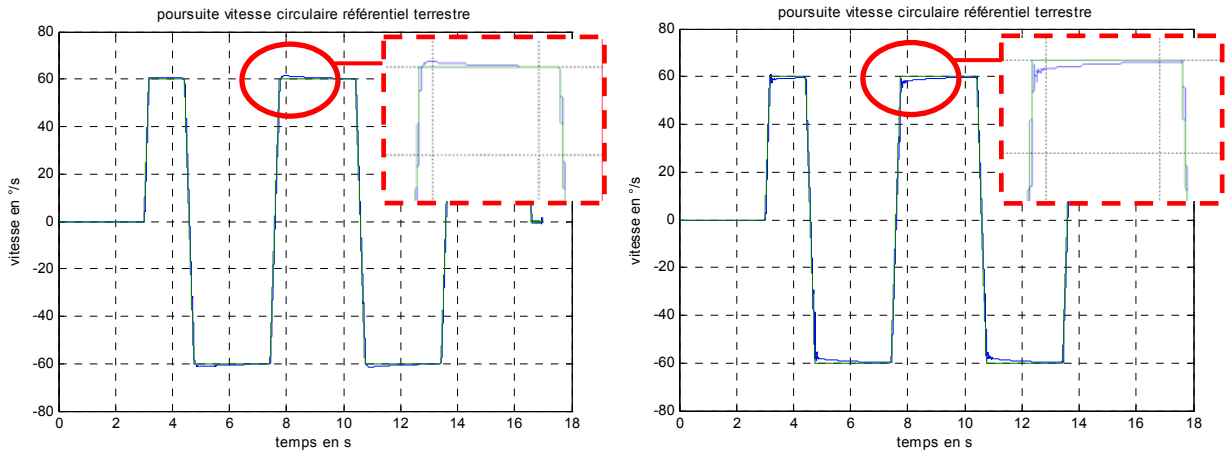



Fig. 36 : Poursuite de vitesse en continu (à gauche) et en échantillonné (à droite)

Les mises au point itératives (ajustement des correcteurs) ont permis de converger vers une solution numérisée dont le résultat correspond au cahier des charges, et qui s'approche du résultat en continu. Les limitations des calculs en échantillonné ont pour effet d'amortir un peu plus les variations comme on peut le voir sur la zone cerclée.

## *C. L'analyse du code*

L'atout initial de Matlab® que l'on souhaitait exploiter était la génération de code par Matlab® du projet. Plusieurs aspects du projet étaient délicats à implémenter, ce qui posait déjà quelques problèmes. De plus, le code généré ne devient efficace qu'à partir du moment où l'on connaît très bien la cible. Ce n'est alors plus adapté puisqu'il faut être expert dans les deux domaines et il faut en plus travailler avec les contraintes de plusieurs outils.

Ainsi, on fait ici la synthèse des points bloquants et/ou limitants sur la génération de code qui ne concernent que les implémentations des fonctions asservissements. Dans la génération de

code, il y a plusieurs points à aborder et à respecter pour que le code final soit en adéquation avec les règles de codages et les architectures logicielles. Pour la synthèse de l'analyse et pour diversifier les exemples, on ne parlera pas de la problématique du code fonctionnel, mais de l'architecture de conception d'un schéma Simulink® et des règles de codages aéronautiques.

On parle ici directement du fonctionnement sous interruptions, où l'on s'approche de la philosophie de développement avec le logiciel dédié à la cible (en C et en assembleur) pour exécuter les fonctions d'asservissements. Le but est de déclencher une routine contenant le code des asservissements à partir de l'acquisition des grandeurs analogiques. Pour la lisibilité de l'exercice (et pour ne s'intéresser qu'à la structure), on utilise un code fonctionnel très simple. L'exemple choisi est donné en Fig. 37 :

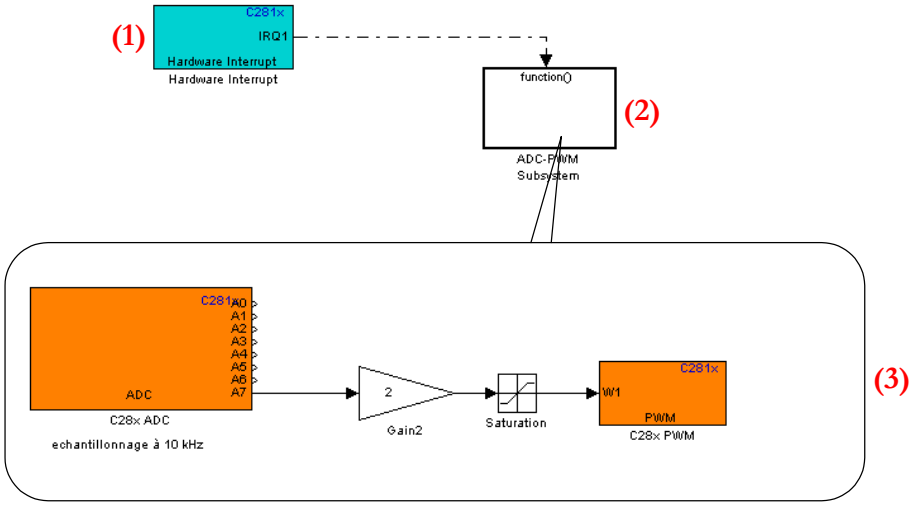



Fig. 37 : Exemple Simulink® pour la génération de code

On exécute une fonction (2) via le déclenchement d'une interruption (1). La fonction est détaillée en (3). Cette fonction est la routine, avec dans cet exemple un code fonctionnel qui est un gain suivi d'une saturation. L'interruption est celle qui concerne l'ADC (on ne le voit pas sur le schéma, c'est un paramètre). On a bien un évènement hardware qui lance une routine, comme on le ferait directement dans le logiciel de la cible. On examine le code généré présenté en Fig. 38, Fig. 40 et Fig. 41. Cette analyse met en relief la structuration du code, sa performance, sa lisibilité : on montre les fonctions générées, et ce que le code contient avec la génération de code des deux blocs Simulink® (1) et (2), le (3) n'étant que le contenu du (2).

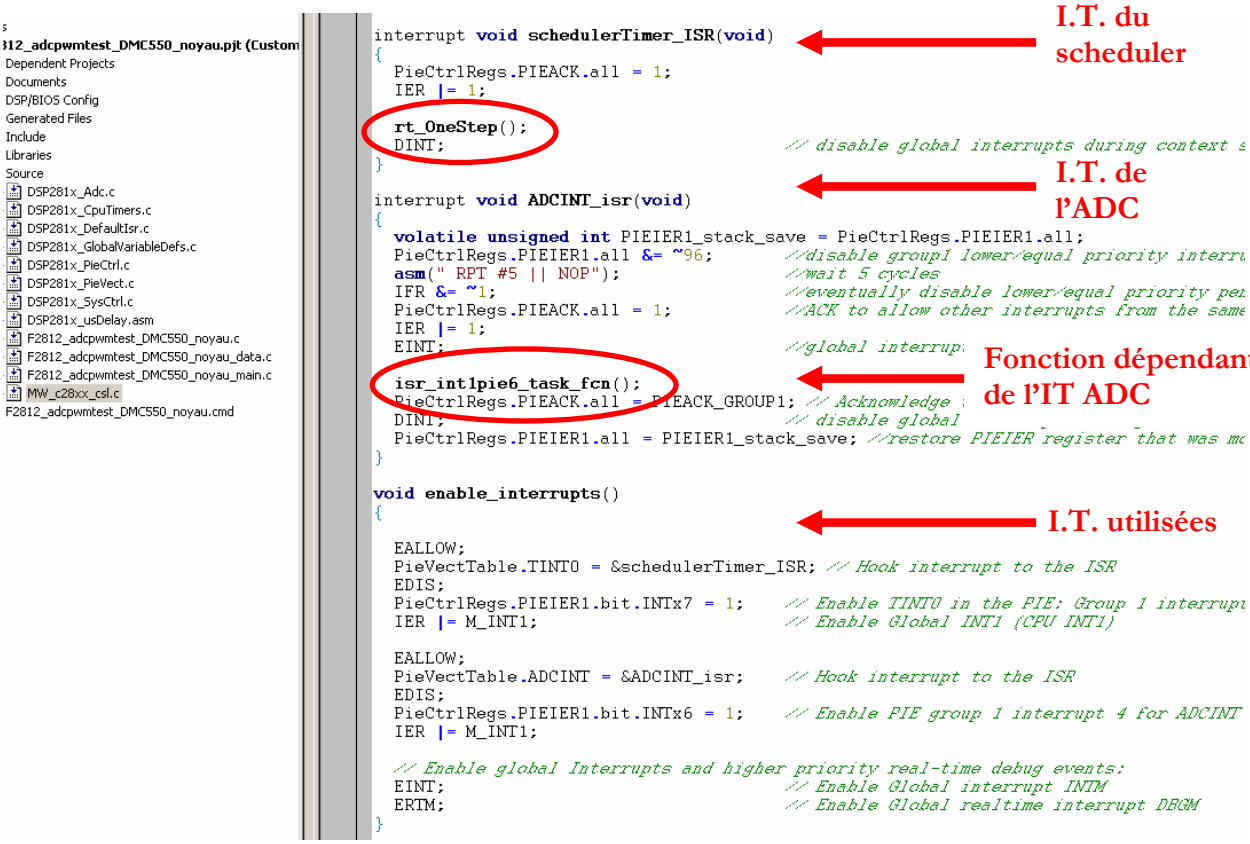



Fig. 38 : Système d'interruption du code généré

Bien que sous l'environnement de Simulink® on ait configuré l'interruption de l'ADC seule et directe, on se retrouve avec un séquenceur sur TINT0, qui relève du fonctionnement sans interruptions, comme illustré en Fig. 39.

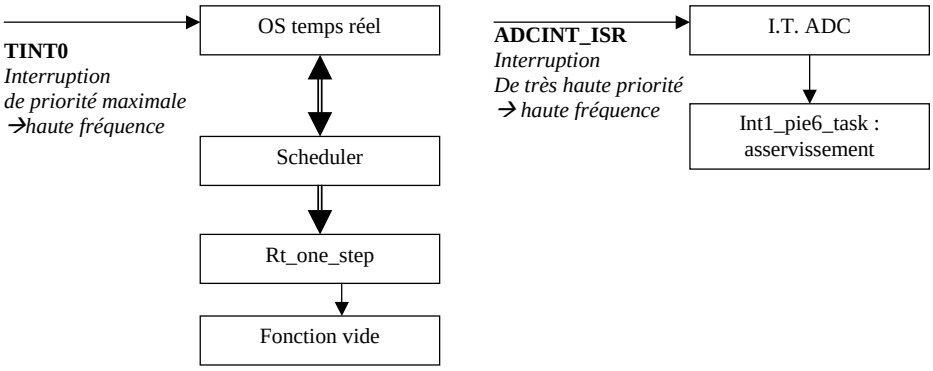



Fig. 39 : Synoptique du système d'interruption généré

De plus, la fonction gérée par le noyau fait appel à un mécanisme de flag qui va complexifier la démarche d'analyse et de certification, puisque le mécanisme est imbriqué avec le reste du programme. Le code est plus complexe qu'un codage en « C » dans l'environnement de la cible.

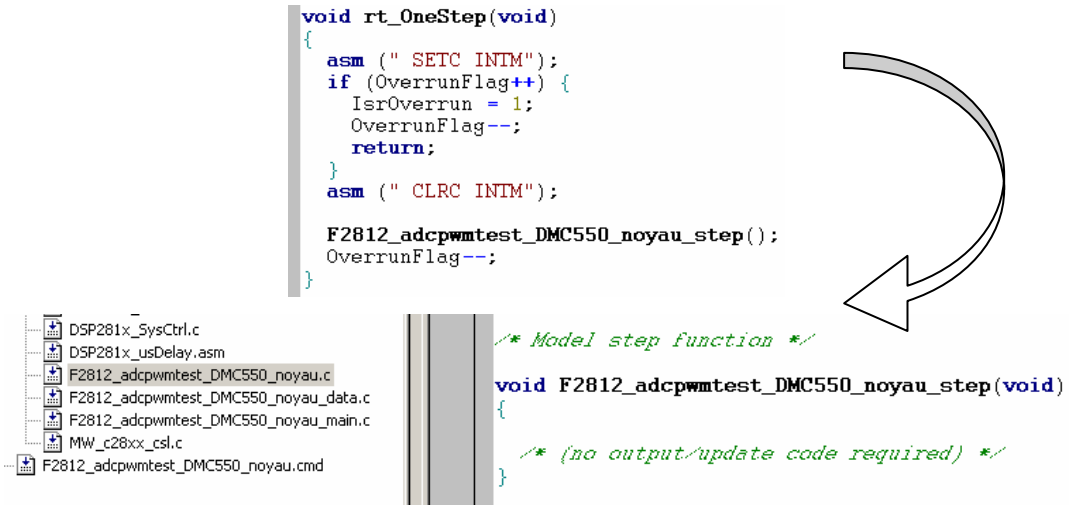



Fig. 40 : Fonction du « scheduler »

En ce qui concerne la gestion des interruptions, le résultat attendu n'est pas celui que l'on aurait codé. La stratégie n'est d'une part pas la même, mais on a aussi un code trop générique qui va limiter notre champ d'action en terme de souplesse, mais aussi d'évolution et de réserve de puissance de calcul. L'analyse se poursuit ensuite sur la routine d'interruption qui contient le code fonctionnel, que l'on peut voir en Fig. 41 :

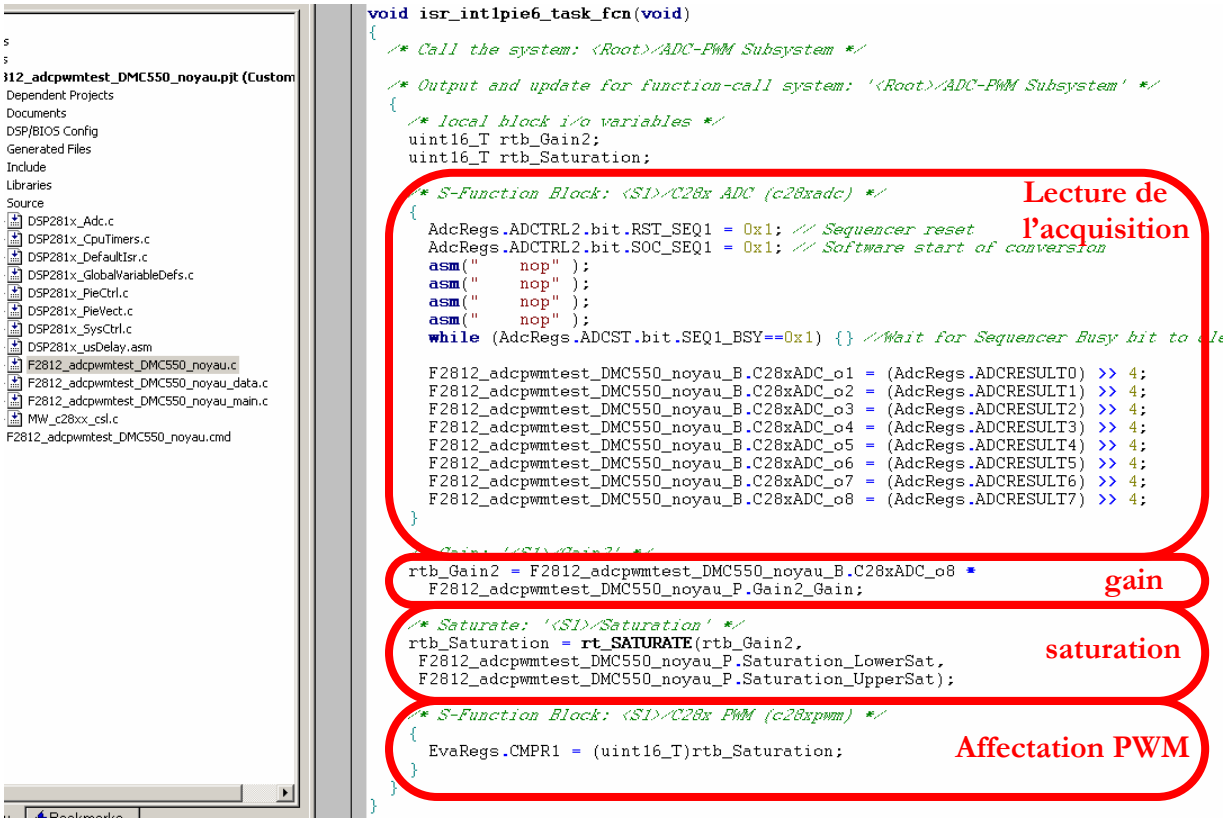



Fig. 41 : Routine d'interruption générée

Dans la routine d'interruption générée (voir Fig. 41), on a l'avantage que le code soit plutôt bien lisible, et on fait très vite le lien avec le schéma Simulink® malgré des lignes de code inutiles (voies ADC non utilisées). Par contre, la gestion de l'ADC n'est pas celle souhaitée, puisque le « Start Of Conversion » est lancé par l'activation du Flag après que la routine ait été lancé au lieu d'être gérée par le timer dédié. De plus, un « while » est utilisé pour atteindre la fin des conversions, et en aéronautique, d'après les normes dans ce type de programme, on ne peut pas avoir d'instructions bloquantes. Ces limitations ont été signalées, en plus d'autre problème avec les Timers, à MathWorks™ pour les prochaines révisions.
Dans le contexte actuel, il n'a pas été possible d'atteindre les objectifs avec le code automatique.

## *D. L'analyse normative des évènements*

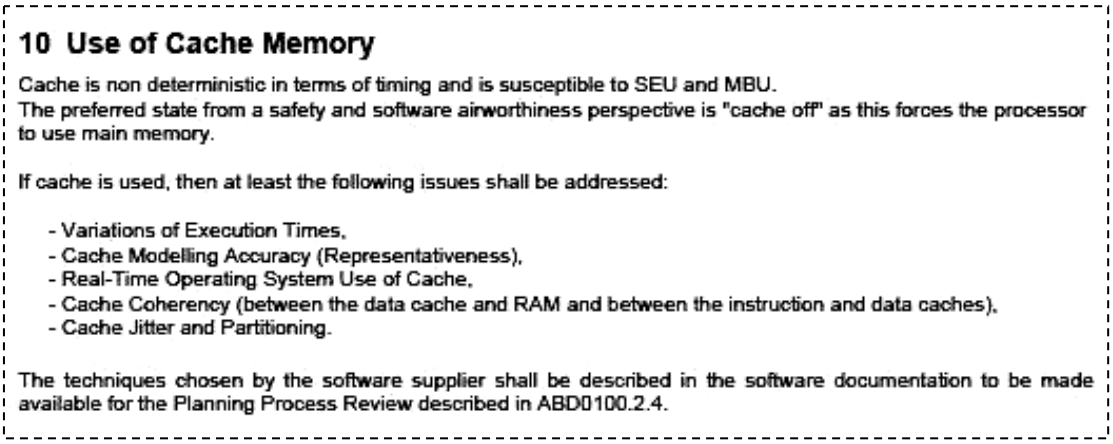
### 1. La prise en compte des normes

Nous venons de voir la génération automatique de code et ses actuelles limitations. Lorsque l'on intègre les contraintes apportées par les normes, le problème devient encore plus complexe. Pour la conception du système, on doit prendre en compte les contraintes d'industrialisation avant même de concevoir un prototype, sinon beaucoup de travail serait perdu.

On illustre ce travail amont par l'analyse de deux points importants de la norme d'AIRBUS™ [ABD100], directement liés à la commande embarquée que constitue le DSP. On rappelle qu'ici on ne dispose plus de calculateurs, mais d'un DSP qui rassemble les fonctions anciennement réalisées en analogique avec un FPGA en tête (avec par exemple une stratégie de triplication), et commandées par un calculateur déporté. La norme d'AIRBUS™ [ABD100] a l'avantage sur la norme DO178B d'être plus explicite sur les contraintes techniques, mais comme c'est une norme beaucoup plus complète, elle demande une étude plus approfondie pour la satisfaire.

La principale préoccupation pour le DSP est la mémoire RAM qui est unique et intégrée, si l'on reste dans une configuration où on utilise le DSP seul. Dans le cas d'une mémoire externalisée, la RAM est contrôlée par le DSP qui a une architecture unique et encore une fois intégrée. On n'a donc pas la souplesse du FPGA qui permet de construire des espaces mémoires en fonction des besoins et des contraintes.

**Extrait ABD100.1.10** au sujet du "non déterminisme" de la gestion de la mémoire cache :



**10 Use of Cache Memory**

Cache is non deterministic in terms of timing and is susceptible to SEU and MBU.
The preferred state from a safety and software airworthiness perspective is "cache off" as this forces the processor to use main memory.

If cache is used, then at least the following issues shall be addressed:

- Variations of Execution Times,
- Cache Modelling Accuracy (Representativeness),
- Real-Time Operating System Use of Cache,
- Cache Coherency (between the data cache and RAM and between the instruction and data caches),
- Cache Jitter and Partitioning.

The techniques chosen by the software supplier shall be described in the software documentation to be made available for the Planning Process Review described in ABD0100.2.4.

Fig. 42 : Extrait ABD100.1.10

Tout d'abord, il y a plusieurs façons d'interpréter le terme « cache ». Au sens général, **dans les systèmes informatiques**, un **système d'exploitation** contrôle les ressources matérielles :

- processeurs,
- bridges (interfaces électroniques numériques),
- mémoires RAM et ROM.

Toutes ces entités sont distinctes. Quand l'on parle de mémoire cache, c'est un type de mémoire très particulier du processeur, dont la caractéristique principale est la très grande vitesse de fonctionnement comparé au reste du système, mais dont la taille est très limitée. Cette mémoire est dédiée aux calculs à proximité d'une ressource matérielle très rapide sans passer par les accès mémoires usuels. On utilise surtout cette mémoire lorsque l'on fait de multiples accès sur une variable pendant un calcul. Cela revient à « travailler » localement sur une donnée avant de la « stocker » dans la mémoire « étendue », plus lente. La mémoire cache est une mémoire « localisée », la gestion de son utilisation impacte directement sur la stabilité du système. Comme elle est au cœur du processus, on doit garantir sa robustesse.

Le facteur d'échelle entre cette mémoire et les autres mémoires fait donc apparaître que cette mémoire est plus sensible (plus vite, plus petit !). On trouve donc la mémoire cache dans la norme, sûrement vue pour ce type de système, où l'architecture hardware est très éclatée.

Avant de transposer la norme sur le DSP, il semble nécessaire de poser le contexte de la norme, basée sur un système **microprocesseur et périphériques (≠DSP)**. Analysons chaque terme au sens général :

- **S.E.U.** : la RAM ultra rapide (localisée près des ressources critiques) est plus sensible car elle est plus petite que la RAM de grande capacité ; elle est à ce titre différente des autres mémoires ;
- **Déterminisme** : comme l'utilisation de la mémoire cache dépend d'un besoin de traitement accéléré, elle est par définition non déterministe. C'est le programme qui oriente les données. D'une certaine manière, on peut choisir la mémoire, mais les transferts sont accélérés temporairement. Ainsi, si deux traitements sont faits en parallèle, ou en pseudo-parallèle, le passage d'un calcul en mémoire cache va engendrer l'obtention d'un résultat plus rapide qu'un autre traitement, surtout si cela concerne une ressource hardware ;
- **Variation of Execution Times** : les temps d'exécution en mémoire cache et en RAM sont significativement différents, donc on a des niveaux de temps d'exécution. Les temps dépendent des ressources matérielles, donc a priori, c'est difficile de réguler ou contrôler les flux de données si on souhaite paralléliser ;
- **Cache modelling accuracy** : la possibilité de modéliser fidèlement la mémoire cache et son utilisation permet de simuler les comportements dynamiques. Cela implique de modéliser un comportement logiciel, sur un environnement matériel très précis ;
- **Real time operating system use of cache** : la mémoire cache est une ressource partagée par l'ensemble du système, toutes données confondues. En temps réel, on se doit de contrôler les flux, donc de bien contrôler la mémoire cache pour éviter les engorgements, et que le bus (d'entrées/sorties) des données puisse véhiculer les données traitées rapidement en mémoire cache. Le traitement temps réel implique donc une utilisation de la mémoire cache, mais aussi vis à vis des autres ressources ;
- **Cache coherency** : pour passer d'un système lent à un système rapide et vice-versa, il faut une certaine cohérence entre les flux de données et d'instructions. Dans un microprocesseur, le bus est commun, donc tout transite sur le même canal (structure Von Neuman). Un synchronisme et un mécanisme de fonctionnement sont nécessaires ;
- **Cache jitter and partitionning** : comme la mémoire cache est plus ou moins indépendante, les accès en mémoire cache et en « RAM » sont du fait dé-corrélés. La gigue entre les différences de timings est donc un point délicat. Les fréquences de fonctionnement sont très éloignées, et le mécanisme d'accès au bus est différent. Normalement, c'est matériel. Comme la mémoire cache est petite, (et généralement c'est elle que l'on veut augmenter en priorité sur la RAM pour les performances), on

ne peut pas stocker beaucoup de données. Faire du partitionnement, c'est ajouter une gestion « de pile » logicielle dans un mécanisme matériel, donc nécessairement, c'est ajouter des contraintes pour garantir l'intégrité des données.

Voyons les différences vis à vis du **DSP** :

Le DSP utilisé (dédié à l'électronique de puissance ; famille C2000©) n'est pas un microprocesseur avec un environnement matériel séparé, mais c'est un **microcontrôleur** où tout est intégré. Le composant intègre tous les périphériques dans une structure « harvard » (multi-bus : data, instructions, I/O) avec une seule et unique fréquence de traitement des données. Par contre, on trouve un pipeline typique pour le traitement des instructions qui consiste à palier une fréquence de fonctionnement bien plus faible par rapport à un microprocesseur, en parallélisant les mécanismes matériels internes aux gestions des instructions. Et c'est cela la puissance du DSP : petit composant (faible fréquence = faible consommation = bon rendement) et grosse capacité de calcul : c'est complètement différent sur le fond et sur la forme d'un microprocesseur.

De ce fait, la gestion mémoire est différente. Dans le DSP, l'adressage mémoire est direct car l'espace mémoire utilisé est inférieur aux capacités d'adressage maximales. Le « noyau » de calcul exploite directement les données en RAM. On peut dire que la mémoire cache et la RAM sont la même chose, donc que tout est « mémoire cache », ou que rien n'est mémoire cache. C'est bien là où l'interprétation pour faire basculer la conception. La mémoire cache peut-être aussi les registres « accumulateurs » de l'ALU. Dans ce cas, tous les registres sont de la mémoire cache. Afin de différencier les zones mémoires entre les données stockées, les paramètres, etc…la mémoire est sectorisée pour borner la « pile », et le reste. Sur le fond, c'est radicalement différent du microprocesseur (pour les registres).

Revoyons les différents points précédents concernant le DSP :

- **S.E.U.** : La RAM est commune. Tout est sensible de la même manière ;
- **Déterminisme** : avec un adressage direct du noyau, le déterminisme semble plus simple à comprendre et à démontrer. Les enchaînements sont séquentiels, donc un ensemble d'instructions pour manipuler les données sera traité à la chaîne, selon la séquence du programme prédéfini. Ensuite, c'est la gestion de la « pile » qui sera à vérifier pour le déterminisme. C'est le point délicat ;
- **Variation of Execution Times** : Les temps sont identiques compte tenu de l'unique fréquence pour la RAM interne. Il y a cependant un bémol car la RAM interne où l'on

manipule les données est un peu plus rapide que la Flash qui contient le code exécutable. De la même manière, la RAM externe est plus lente. Dans ce cas, on considère la RAM interne comme mémoire cache, sachant qu'elle peut être seule (sans RAM externe) ;

- **Cache modelling accuracy** : Est-on capable de modéliser un cœur DSP : on peut implémenter un cœur DSP dans un FPGA, mais est-ce certifié pour une comparaison ?
- **Real time operating system use of cache** : le fonctionnement en interruption veut aussi dire qu'il n'y a pas de système d'exploitation. L'utilisation des ressources est directe, via le code exécutable applicatif. Tout est déterminé par conception ;
- **Cache coherency** : là, il faut poser le problème (au sens d'exercice) du pipeline. Les bus sont parallélisés, donc il y a une différence entre les données et les instructions. Sachant que les instructions se suivent dans le mécanisme de pipeline, ce n'est pas trivial, mais c'est un mécanisme qui fait la force du DSP, on peut donc penser que le système est robuste puisque c'est la base de son fonctionnement ;
- **Cache jitter and partitionning** : la gigue n'est à évoquer qui si l'on assimile la RAM à la mémoire cache avec la flash qui contient le code exécutable, ou alors s'il l'on utilise aussi la RAM externe.

L'usage du DSP présente donc quelques inconvénients pour l'implémentation puisque l'on crée un système dédié, mais on a aussi des gros atouts qui vont dans le bon sens au vu de la certification, de part son homogénéité.

**Extrait ABD0100.1.10. §2** au sujet de la gestion des IT : gestion statique ?

For RTS off the shelf software, the COTS paragraph applies. In all other cases, the following recommendations shall be taken into account:

- minimise the number of Interrupts,
- minimise the interfaces between the RTS and the application software,
- prioritise Interrupts depending on the safety impact of the associated function(s), i.e. the more critical the function is, the higher the interrupt priority should be.

The Real Time Scheduler should:

- have a clear and visible scheduling mechanism,
- support the synchronisation between the different tasks,
- monitor for the completion of each task in a given time,
- have a well defined exception handling,
- in case of severe software problems, a hardware device (e.g. watchdog) shall force the system into a safe state.

To evaluate the proposed Real Time Scheduler, detailed information about this topic should be provided to the Purchaser.

**Software Level of the Real Time Scheduler:**

The Real Time Scheduler - if it is designed especially for a given application - will be handled by the ED12B/ DO 178B processes like all the other on board Software. But since the remaining software of the application is depending on the Real Time Scheduler, the highest SW level assigned to the application - or a part of the application - shall also be assigned to the Real Time Scheduler.

Fig. 43 : Extrait ABD100.1.10.§2

On est dans le cas typique du système hors étagère, car chaque programme est dédié à une application spécifique. Il n'y a pas de système d'exploitation, donc c'est l'application qui contrôle les ressources. Analysons chaque point :

- **minimise the number of interrupt** : ceci n'est pas défini par notre volonté, mais par les périphériques que l'on utilise sur le DSP (Convertisseur Analogique Numérique, boucles d'asservissements…), donc on n'a pas des interruptions au sens du système à microprocesseur qui donne du temps aux tâches, mais au sens microcontrôleur qui provoque des interruptions logicielles suite à un événement matériel. C'est la conception qui « impose » les choix des interruptions.
- **Minimise the interface between the RTS and the application software** : contrairement à un système avec microprocesseur, nous n'avons pas d'interface car il n'y a pas de RTS ; ou alors, on considère que l'application et le RTS ne font qu'un, auquel cas l'interface est maximisée puisqu'elle est complète !
- **Prioritise interrupts depending on the safety impact of the associated function(s), i.e. the more critical the function is, the highter the interrupt priority should be** : c'est le cas car on s'est appuyé sur la DO178B au début du développement pour faire les choix réglementaires. Tout ce qui touche au contrôle

moteur, donc à l'électronique de puissance est en priorité matérielle maximale. On s'est interdit tout mécanisme de priorité logiciel : tout est matériel. Ensuite, les priorités sont par niveaux dégressifs. On peut avoir une priorité identique pour la communication « normale » et « debug », mais on fonctionne avec une exclusion mutuelle et des « buffers », avec un temps de réponse inférieur au temps de cycle de communication, pour garantir le temps réel.

The real time scheduler :

- **have a clear and visible scheduling mechanism** : Il n'y a pas de mécanisme logiciel, et le mécanisme matériel est vraiment transparent. C'est un point qui va en faveur du DSP et qui est étayé dans les documentations du composant.
- **support the synchronisation between the different tasks** : les synchronisations sont dépendantes des évènements matériels, sachant que les tâches synchrones sont synchronisées au niveau matériel. On garantit la synchronisation sur celles qui le nécessitent, ainsi que le fonctionnement même si les tâches sont désynchronisées. Dans un microcontrôleur, l'asynchronisme est cependant le type de fonctionnement par défaut. Un synchronisme est un cas particulier.
- **Monitor for the completion of each task in a given time** : un des axes de codage a été de faire du code non extensif. A ce titre, les temps d'exécutions des tâches sont identiques quelles que soient les entrées. On peut aussi contrôler les durées sur un oscilloscope via des sorties dédiées (la norme demande des moyens de vérifications). Seuls les passages en pile sont masqués. Pour le moment, on extrapole et on vérifie les temps disponibles entre les tâches pour vérifier les sauts entre les tâches, (entre les interruptions).
- **Have a well defined exception handling** : il n'y a pas de RTS, donc pas d'exceptions.
- **In case of severe software problem, a hardware device (e.g. watchdog) shall force the system into a safe state** : c'est le principe même du microcontrôleur. Il y a un watchdog interne pour contrôler les données internes au DSP, et le CPLD intègre aussi un watchdog qui surveille le fonctionnement macroscopique du DSP. Leurs rôles sont de réinitialiser le DSP. Dans notre cas, le « safe mode » est obtenu par redondance du RTSU.

Les interruptions sont définies au démarrage et elles sont verrouillées. Aucune interruption n'est rajoutée dynamiquement car aucune tâche n'est rajoutée pendant le fonctionnement. Les

tâches fonctionnent toutes sans mécanismes de mise en sommeil. Les activations et désactivations des interruptions au démarrage pour les séquences d'indexages ne sont pas concernées par ces points puisque les interruptions manipulées au démarrage sont pilotées par une machine à états finis, donc avec un fonctionnement défini.

## 2. L'analyse évènementielle

### a) Les mécanismes

Dans notre architecture, on a trois interruptions pour les trois thèmes suivants :

- **Une interruption de communication :** les messages entrants sont réceptionnés par un « buffer » matériel (avec une file d'attente). Une interruption est générée lorsqu'un message complet est reçu. Il faut venir le lire avant que la file d'attente soit pleine (contrainte matérielle) pour ne pas perdre de messages, et aussi pour que l'information reçue ne soit pas traitée avec du retard[6] (contrainte logicielle). La fréquence des messages dépend de l'émetteur, donc on impose au DSP par l'extérieur le flux de données qui lui-même va imposer le nombre d'interruptions. Si le flux devient significatif vis-à-vis de la charge CPU (autres interruptions), on peut craindre des dysfonctionnements. Une rupture ou un ralentissement de la communication va/peut perturber les cycles de balayages (non-conformité par rapport aux scénarios de balayages définis par le calculateur).
- **Une interruption de contrôle des amplificateurs :** c'est l'interruption de contrôle moteur, mais le DSP pilote directement les amplificateurs qui sont les premiers dispositifs assujettis aux risques. La fréquence est fixe car elle est définie en interne. Les mesures sont rafraîchies à chaque interruption en ce qui concerne les entrées. Les sorties sont les signaux PWM. Les correcteurs de courants sont calculés et vérifiés à chaque interruption, et les sorties PWM sont rafraîchies. Le mécanisme de découpage (125 kHz) est matériel, donc on n'a pas de risque de stopper son évolution (en cas de « court-circuit » via l'impédance moteur), mais on a le risque de donner une consigne figée, qui peut être forte (courant maximal). Les sécurités limitent le courant maximal au courant imposé par le couple maximal.

[6] On parle du retard supplémentaire (non connu). Le retard fixe dû à la transmission (qui lui est incompressible) est considéré comme un retard connu et prévu.

- **Une interruption d'asservissements :** c'est une interruption qui n'a pas de conséquences sur un périphérique. En effet, c'est pour créer une base de temps bien spécifique pour les calculs des asservissements. Les entrées proviennent malgré tout de capteurs, mais c'est une simple lecture. Les sorties sont des données exploitées soit par d'autres fonctions, soit par l'interruption de contrôle moteur (qui a son propre mécanisme de protection).

Ces trois interruptions ont chacune leur particularité, quant à leurs priorités et leurs fonctions, mais la particularité sans doute la plus importante pour l'analyse type RMA est leur caractère temporel [ISO01]:

- l'interruption des amplificateurs : c'est une routine périodique fixe, à priorité statique préemptive. Elle est déclenchée par un EOS (End Of Conversion) dont un timer périodique et constant cadence les conversions de l'ADC ;
- l'interruption d'asservissements : c'est aussi une routine périodique fixe, à priorité statique préemptive. Elle est cadencée par un timer fixe et constant ;
- l'interruption de communication : c'est une routine sporadique, à priorité fixe préemptive (préemptive si on considère la tâche de fond). Elle est déclenchée par un évènement extérieur non constant, mais avec un intervalle borné.

On n'a pas la présence d'interruptions apériodiques, donc une analyse temporelle est possible (enfin rendue beaucoup plus simple). Comme les interruptions périodiques sont de plus hauts niveaux, on a en plus un déterminisme assez fort, et la routine sporadique peut consommer le temps restant, car elle reste une routine préemptive sur la « tâche » de fond (les NOP). Outre la conception du logiciel Radar Météo, dans les applications de contrôle-commande des amplificateurs (électronique de puissance), on aura une philosophie très similaire quant aux stratégies et interruptions utilisées. Les interruptions apériodiques ne sont a priori pas des interruptions courantes dans ce genre d'applications. On rappelle que le logiciel pilote deux onduleurs triphasés avec leurs moteurs, avec en plus des régulations pour la mécanique, ce qui est dans l'optique d'utilisation de ces DSP.

Voyons comment sont gérés les mécanismes d'interruptions dans les DSP C2000©. On s'appuie sur les documentations du TMS320F281x©.

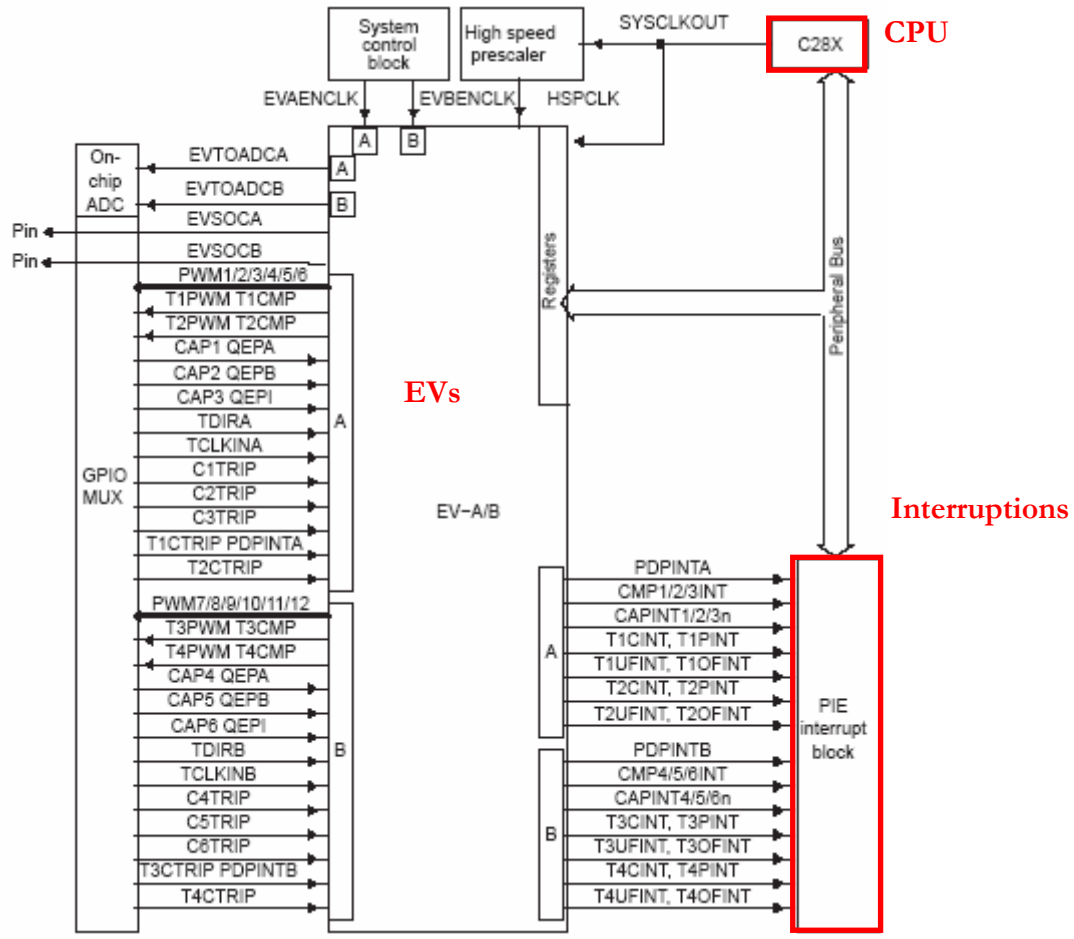



Fig. 44 : Architecture PIE, EV et CPU dans le F281x [TEX]

On voit ici (Fig. 44) que le DSP est intrinsèquement constitué d'organes indépendants. On y trouve le cœur de calcul CPU, dont le moteur C28x est un cœur de calcul en 32 bits virgule fixe avec les instructions et les bus typiques aux DSP cadencé à la fréquence issue de la PLL. Les « events managers » (EV) avec tous leurs périphériques disposent de leurs propres fréquences et mécanismes de contrôles. On a enfin le PIE qui est le routeur d'interruptions, dont les mécanismes hardware sont accessibles par le logiciel.

On peut voir que les accès (bus) permettent des interactions entre toutes les entités. On comprend bien ici que le moteur de calcul est une ressource partagée.

Le PIE dispose d'accès directs des EV, concernant les évènements hardware tel que les « top » des timers, ou d'ADC. Ceux-ci n'ont donc pas de temps de latence dû à l'utilisation du bus. C'est la même chose avec les interruptions sur le coeur, représentées Fig. 45.

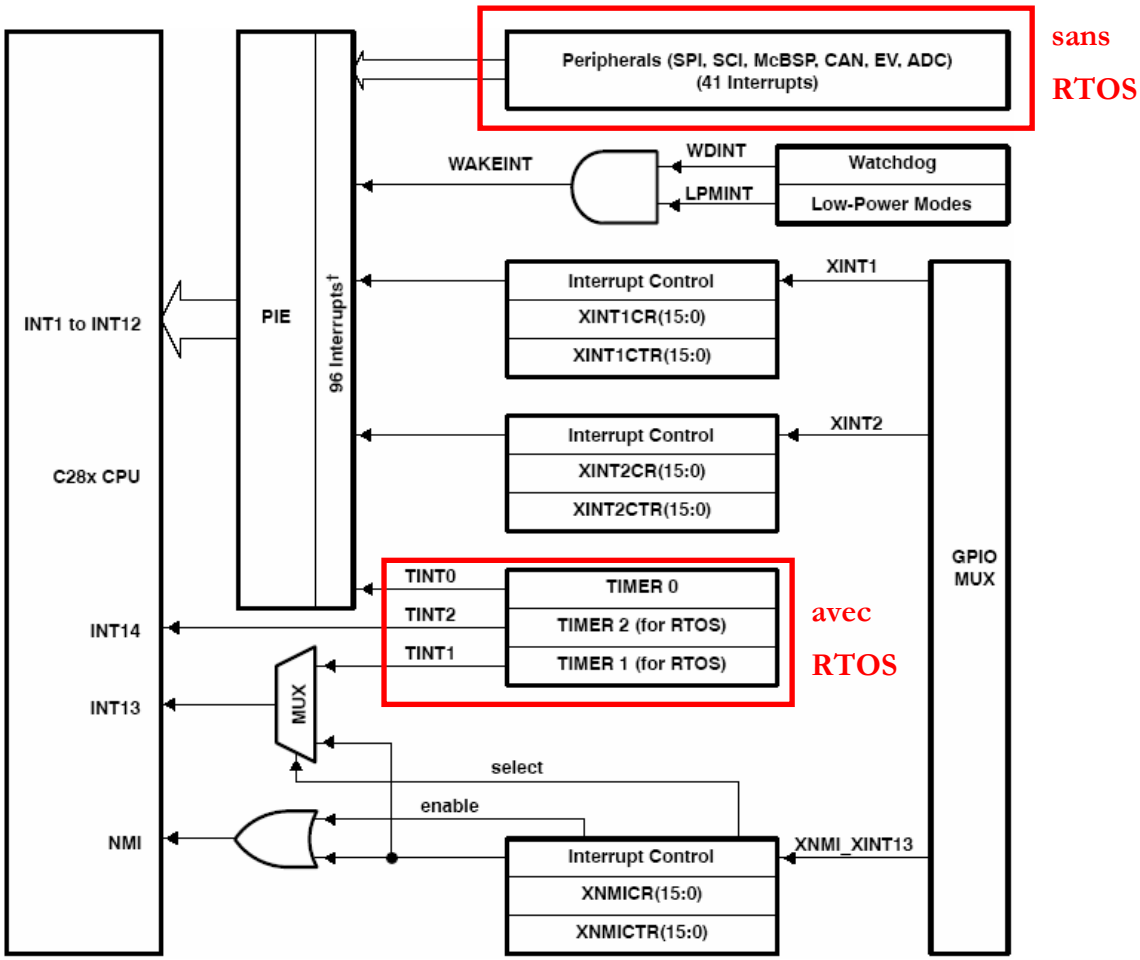



Fig. 45 : Interaction des mécanismes d'interruption dans le F281x [TEX]

Le DSP permet de travailler selon la manière RTOS pour les traitements à base d'un noyau temps réel, ou bien de travailler avec les interruptions des périphériques pour du contrôle moteur par exemple. On voit bien (voir Fig. 45) que c'est deux mécanismes physiquement différents, qui ne sont pas dimensionnés et structurés pour les mêmes applications. On notera aussi la présence de NMI qui est une interruption apériodique, mais pour des usages très particuliers avec la priorité la plus élevée. On a aussi le watchdog pour le monitoring qui est un périphérique distinct.

Le RTOS fonctionne directement avec des interruptions sur le CPU, avec un timer de priorité maximale sur le système, et deux timers auxiliaires.

Pour le fonctionnement type « microcontrôleur », on peut voir plusieurs « boîtes » pour les systèmes d'interruptions. La raison est simple, c'est que sans RTOS ou un équivalent, les interruptions sont générées par les différents périphériques (boîte 'peripherals') et sont gérées de manière matérielles (PIE) en ce qui concerne les déclenchements et les arrêts, mais aussi en ce qui concerne les préemptions [LEE98], les priorités, et les mécanismes de gestion des routines associées. Les autres entrées du PIE sont des entrées spéciales (périphériques non liés aux mécanismes des microcontrôleurs embarqués sur la puce). Le code doit donc prendre en compte

toutes ces contraintes, ce qui conduit à intégrer les « drivers » au code fonctionnel. Le détail des interruptions des périphériques est donné Fig. 46.

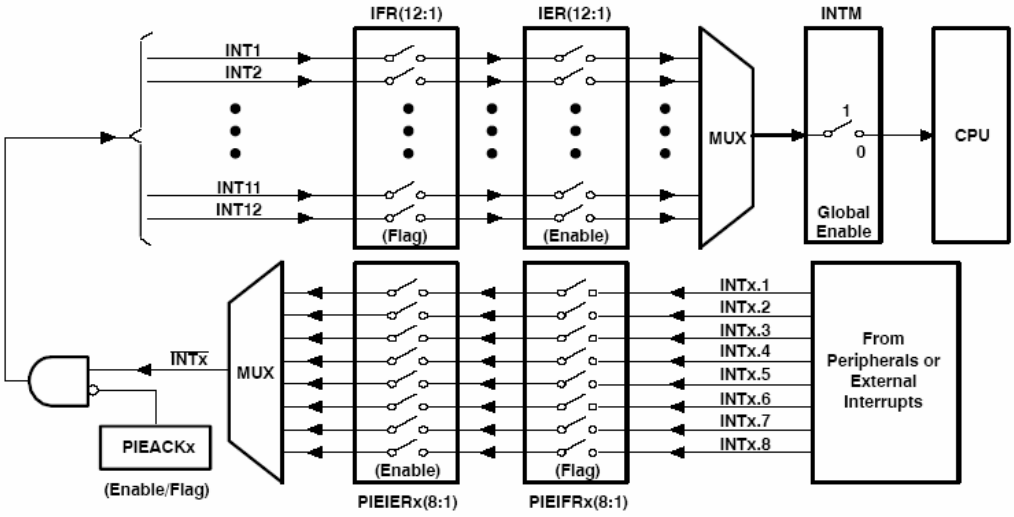



Fig. 46 : Mécanismes d'interruptions des périphériques, pour le F281x [TEX]

Ce schéma de principe (Fig. 46) représente le « routeur » d'interruptions. Chacune des interruptions possibles (prédéterminées par conception de la puce ou par reconfiguration) est validée ou non pour l'initialisation, mais aussi masquée/démasquée pendant le fonctionnement. On associe ce schéma à l'organisation des interruptions selon le tableau Fig. 47:

| CPU INTERRUPTS | PIE INTERRUPTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | INTx.8 | INTx.7 | INTx.6 | INTx.5 | INTx.4 | INTx.3 | INTx.2 | INTx.1 |
| INT1 | WAKEINT (LPM/WD) | TINT0 (TIMER 0) | ADCINT (ADC) | XINT2 | XINT1 | Reserved | PDPINTB (EV-B) | PDPINTA (EV-A) |
| INT2 | Reserved | T1OFINT (EV-A) | T1UFINT (EV-A) | T1CINT (EV-A) | T1PINT (EV-A) | CMP3INT (EV-A) | CMP2INT (EV-A) | CMP1INT (EV-A) |
| INT3 | Reserved | CAPINT3 (EV-A) | CAPINT2 (EV-A) | CAPINT1 (EV-A) | T2OFINT (EV-A) | T2UFINT (EV-A) | T2CINT (EV-A) | T2PINT (EV-A) |
| INT4 | Reserved | T3OFINT (EV-B) | T3UFINT (EV-B) | T3CINT (EV-B) | T3PINT (EV-B) | CMP6INT (EV-B) | CMP5INT (EV-B) | CMP4INT (EV-B) |
| INT5 | Reserved | CAPINT6 (EV-B) | CAPINT5 (EV-B) | CAPINT4 (EV-B) | T4OFINT (EV-B) | T4UFINT (EV-B) | T4CINT (EV-B) | T4PINT (EV-B) |
| INT6 | Reserved | Reserved | MXINT (McBSP) | MRINT (McBSP) | Reserved | Reserved | SPITXINTA (SPI) | SPIRXINTA (SPI) |
| INT7 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| INT8 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| INT9 | Reserved | Reserved | ECAN1INT (CAN) | ECAN0INT (CAN) | SCITXINTB (SCI-B) | SCIRXINTB (SCI-B) | SCITXINTA (SCI-A) | SCIRXINTA (SCI-A) |
| INT10 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| INT11 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| INT12 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |

Fig. 47 : Tableau de synthèse des interruptions des périphériques du F281x [TEX]

Ce tableau donne les priorités (INTx) où la 1ère est la plus forte dans le traitement du CPU. Pour chaque priorité du CPU, plusieurs interruptions sont possibles et sont gérées par le routeur en exclusion mutuelle car elles remplissent des fonctions de priorités équivalentes. C'est ici que l'on identifie les interruptions périodiques, sporadiques et apériodiques.

WAKEINT, XINT1 et XINT2 sont apériodiques (watchdog notamment et interruptions externes). TINT0 est plutôt réservée au RTOS et aux séquenceurs. Les timers (TxxxINT) sont périodiques. Les captures (CAPINTx) sont sporadiques ou périodiques, comme les périphériques SPI (pVIII), SCI, eCAN et McBSP. C'est donc a priori compatible avec une analyse type RMA.
On remarque que l'interruption de l'ADC, celle qui sert au contrôle de l'électronique de puissance, est l'interruption la plus prioritaire dans le fonctionnement que l'on a choisi (sauf interruption apériodique PDP : Power Drive Protection).

## b) L'analyse temporelle

Par défaut, dans le fonctionnement en interruptions directes, le DSP n'est pas préemptif. En effet, toutes les interruptions du PIE sont en exclusions mutuelles, et la priorité prend effet dans le cas d'interruptions mémorisées (file d'attente) ou synchrones. Ainsi, une routine qui est commencée ira jusqu'à son terme. La suivante sera celle de plus grande priorité. Dans ce cas, pour la lisibilité des relevés, les interruptions (des figures suivantes) ont des déclenchements décalés pour ne pas avoir de déclenchements synchrones. On peut alors définir la marge de temps disponible pour chaque interruption dès la conception. Les temps sont alors bornés par l'interruption la plus fréquente, ce qui est pénalisant lorsqu'il y a une excursion significative entre les différentes bases de temps. Voyons cette différence Fig. 48 (le rapport cyclique positif est image du temps d'exécution de la routine) :

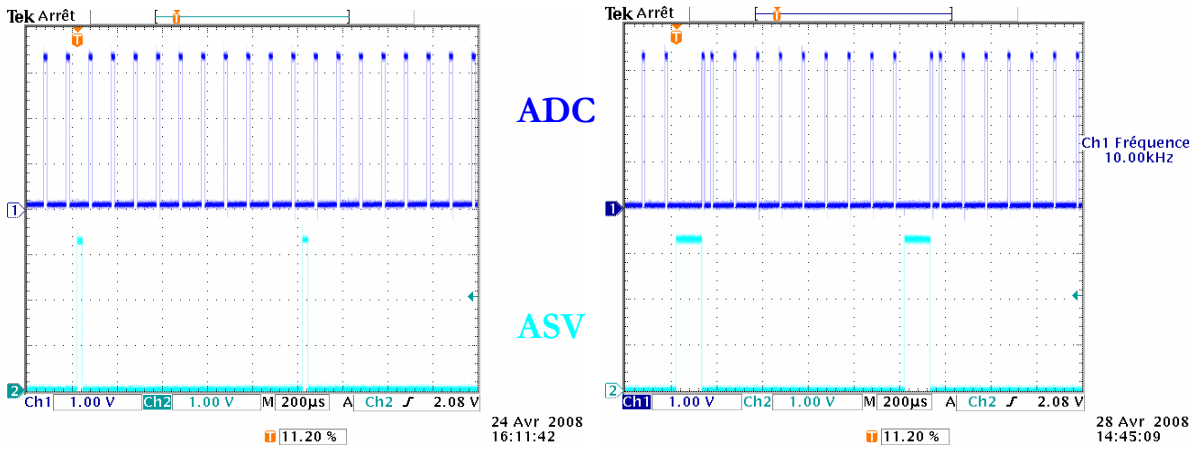



Fig. 48 : Relevés avec I.T. décalées non préemptives, cas temps réel (1) et cas surchargé (2)

Sur les deux relevés, les I.T. sont décalées pour garantir dès la conception une marge sur l'I.T. la plus rapide. Dans les deux cas, on fonctionne en interruptions directes, avec des priorités statiques sans préemption, ce qu'on rappelle être le fonctionnement intrinsèque du DSP. En

décalant volontairement les déclenchements des routines, on ne fait pas apparaître les priorités, peu importe ici. Le cas (1) représente le code actuel, avec les asservissements implémentés. Le cas (2) représente ce que pourrait être le code dans le futur avec une interruption d'asservissements plus longue dû aux changements de repères, et des correcteurs d'ordres plus élevés (plus d'équations de récurrence). Dans le second cas, les routines de contrôle moteur ne peuvent s'exécuter tant que l'interruption en cours des asservissements (ASV) n'est pas terminée (non préemption). Il faut alors fonctionner dans le mode préemptif pour pouvoir bénéficier d'un temps pour les asservissements plus important (ici décuplé) si l'on veut conserver le temps réel.

La conception du code doit maintenant prendre en compte un caractère préemptif. Cette démarche va sans doute complexifier la certification. Voyons l'évolution avec des routines incorporant le mécanisme de préemption géré par le concepteur :

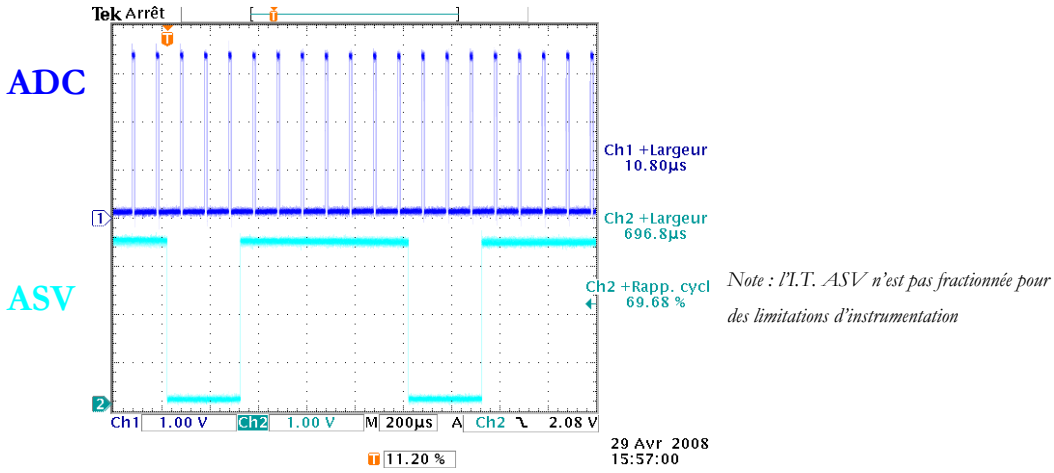



Fig. 49 : Relevé avec I.T. décalées avec préemption

Le relevé en Fig. 49 montre bien que l'on est revenu à un fonctionnement temps réel. Le DSP est donc bien adapté aux reconfigurations. Bref, l'I.T. des asservissements a été augmenté de 10000 itérations pour simuler du code (code ultérieur fictif). Pour la suite du projet, on travaille avec des interruptions à priorités statiques et déclenchements préemptifs sans RTOS (ou sans séquenceur). On confirme les priorités statiques par un relevé avec persistance qui met en avant les « zones » d'exécutions :

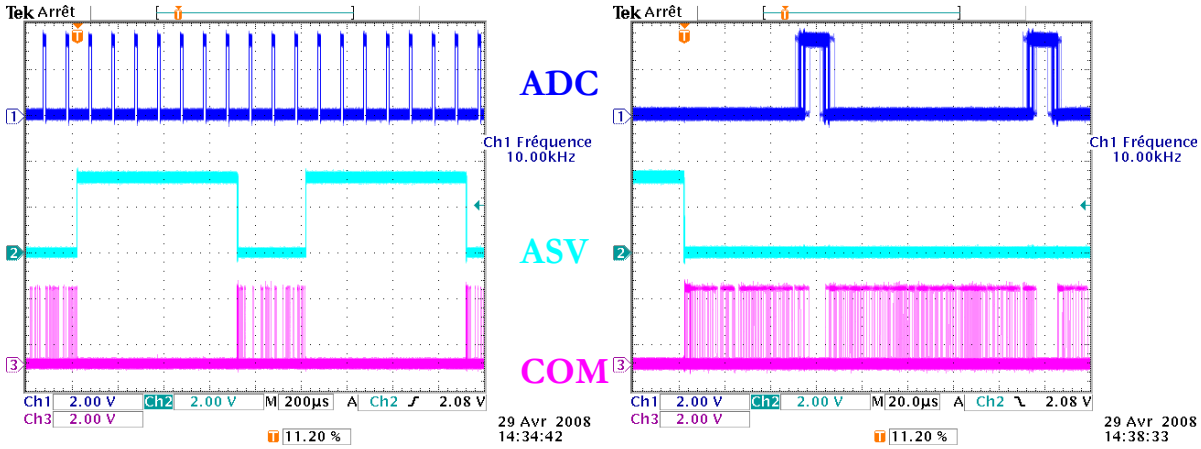



Fig. 50 : Relevés des 3 I.T. à priorités statiques avec préemption, vue période ASV (1), vue période ADC (2)

On voit nettement en Fig. 50 que l'I.T. ADC est prioritaire et préemptive sur les 2 autres, sur ASV (voir 1) et sur la COM (voir 2). L'I.T. d'asservissement est quant à elle prioritaire sur la COM. Comme la communication est sporadique, la routine s'exécute durant la tranche de temps disponible après l'interruption d'asservissement. Lorsque l'I.T. de COM apparaît pendant que les autres I.T. sont activées (notamment l'I.T. d'asservissement), celle-ci est mémorisée et exécutée juste après celle des asservissements (de plus les bases de temps sont différentes : 1ms / 20ms).

Sur le plan fonctionnel, les consignes des asservissements sont issues de la communication, donc on respecte les enchaînements. L'I.T. ADC est indépendante et la mise à jour rapide des données est transparente pour les autres fonctions.

Maintenant, revenons à l'analyse des évènements et des défaillances associées. Si la routine d'interruption est figée, il faut bien distinguer si c'est logiciel ou matériel :

- blocage logiciel : seules les données issues ou vers des périphériques sont fausses :
  - communication : perturbations des cycles de balayages, perte de contrôle de l'antenne. Elle continue comme à la dernière consigne ;
  - contrôle des amplificateurs : maintien des consignes en couple. L'antenne va forcer en butée ;
  - asservissements : maintien des consignes en couple. L'antenne va forcer en butée.
- blocage matériel : les ressources matérielles (périphériques) sont perturbées :
  - communication : plus de transmission, plus de trames sortantes, trames entrantes ignorées. L'effet est le même que pour le blocage logiciel ;
  - contrôle des amplificateurs : plus de signal PWM, surintensité (via impédance moteur)

- asservissements : les calculs sont bloqués, il n'y a plus de rafraîchissement des consignes en courant. L'effet est le même que pour le blocage logiciel.

On voit donc que le blocage logiciel n'est pas le pire cas, et c'est celui qui est le plus probable (dans l'échelle des probabilités). Le blocage matériel n'est possible que par un défaut de l'horloge principale qui est un défaut matériel, ou par une configuration ou un arrêt spécifique des timers.

L'interruption la plus importante est donc celle qui contrôle des ressources pouvant engendrer un dommage. Ainsi, l'interruption de contrôle moteur doit fonctionner quels que soient les aléas. Son temps d'exécution doit être déterminé et doit tenir avec une certaine marge dans le temps imparti. Comme les autres interruptions vont consommer des cycles processeur, il faut que l'interruption la plus critique bénéficie des ressources processeur en priorité, d'où le niveau de priorité de l'interruption. Le temps non consommé est ainsi laissé aux autres interruptions (même système de priorité), et le total de tous les temps d'exécutions doit être inférieur au temps total disponible maximal. On doit tenir compte des variations de temps, qui sont corrélés aux différents calculs ou mécanismes à exécuter.

Inversement, l'interruption la moins importante est l'interruption de communication. Si celle-ci se bloque, on perd certes le contrôle de l'antenne, mais le fonctionnement reste dans les limites prévues des balayages (mécanique et électrique). Si le blocage est dû à une surabondance de données, la charge CPU va saturer, mais avec une priorité faible de la communication, les interruptions importantes vont garder (ou reprendre) leurs temps d'exécution.

On peut synthétiser cette analyse dans un tableau en Fig. 51 (sans code fictif) :

marge : 30 % de ressources dispos désirées — ressources disponibles (%) 45,4

| I.T. | | amplificateur | asservissements | communication |
|---|---|---|---|---|
| rôle | | contrôle des semi-conducteurs de puissance (et des moteurs) | calculs périodiques des asservissements, C-BIT | envoie et réception des consignes, informations diverses vers / provenant du calculateur |
| fréquence | kHz | 10 | 1 | 0,05 |
| période | µs | 100 | 1000 | 20000 |
| temps exécution | I.T. (µs) | 16 | 24 | 8 |
| | max (µs) | 70 | 54 | 30 |
| | % temps restant | 37,8 | 21 | 15,4 |
| I/O | entrées soft | consignes en couple | consignes internes (boot), consignes externes (calculateur via com), positions codeurs | informations de fonctionnement (P-BIT, C-BIT, positions…) |
| | entrées hard | mesures analogiques, codeurs via EPLD | néant | buffer RX, consignes de balayage (positions, vitesses, pas…) |
| | sorties soft | C-BIT | consignes en couple, C-BIT | consignes de balayage (positions, vitesses, pas…) |
| | sorties hard | PWM | néant | buffer TX, informations de fonctionnement (P-BIT, C-BIT, positions) |
| risque | soft | maintien des consignes en couple, antenne en butée | maintien des consignes en couple, antenne en butée | perte de contrôle de l'antenne, continue le dernier balayage. Plus d'info sur l'état de l'antenne |
| | hard | plus de PWM (figée), surintensité | maintien des consignes en couple, antenne en butée | perte de contrôle de l'antenne, continue le dernier balayage. Plus d'info sur l'état de l'antenne |
| type de défaillance | | défaillance électrique et mécanique | défaillance mécanique | défaillance de contrôle |
| conséquences physiques | | dommages électrique couple moteur possible jusqu'au max admissible (travail en butée) | couple moteur possible jusqu'au max admissible (travail en butée) | néant |
| priorité | | 1 | 2 | 3 |

Note : les calculs sont obtenus pour le pire cas de fonctionnement (toutes les I.T. en même temps.)
La période la plus petite impose la base de temps de référence

Fig. 51 : Synthèse de l'analyse des interruptions

L'interruption de contrôle des amplificateurs (moteurs) est la seule qui peut conduire à un dommage matériel suite à un arrêt matériel du périphérique. L'analyse AMDEC permettra d'apporter une réponse à ce problème. Un bogue logiciel ne peut que bloquer l'antenne avec un couple constant sur une butée : c'est mécaniquement admis dans le dimensionnement. Avec la priorité la plus élevée, le contrôle des amplificateurs est assuré d'être exécuté en premier. Comme la marge est suffisante (marge de conception + marge restante), on n'a pas le problème de surcharge processeur. Le programme doit avoir un temps d'exécution constant pour que l'analyse soit pertinente, ou être dans le pire cas de fonctionnement (toutes les I.T. en même temps[7] avec l'exécution la plus longue[8] de chaque routine).

## c) L'analyse fonctionnelle

L'analyse fonctionnelle menée en complément des méthodes temporelles, est comme explicitée au premier chapitre, un outil générique servant à cibler les conséquences de défaillances potentielles. On peut donc passer en revue chaque niveau fonctionnel du système, tel que le

[7] Obtenu par mesures (besoin d'un bon déclenchement) ou par déduction.

[8] Le temps d'exécution le plus long est obtenu avec les statistiques de l'oscilloscope qui instrumente les points tests d'entrée et de sortie des interruptions.

composant dans son environnement électronique, le RTSU et ses interactions avec les différents composants, les routines d'interruptions avec les I/O, les routines avec les autres codes logiciels. Pour illustrer un exemple, on donne en Fig. 52 un extrait de l'analyse de l'interruption des amplificateurs.

On remarque dans cet extrait que la défaillance redoutée est le court-circuit. On compte donc sur le PFC en tête pour limiter le courant si l'on veut conserver l'intégrité du premier RTSU. En effet, si le fusible est altéré, le RTSU sera définitivement hors service jusqu'à la réparation. En cas d'un SEU par exemple, on aurait alors la possibilité de récupérer le 1$^{er}$ RTSU si la limitation amont fonctionne (pour le moment, le scénario du SEU parait improbable sur un temps long car les données sont écrasées à chaque I.T.). L'erreur (dans le sens de dérive) des asservissements paraît le scénario perturbateur le plus probable, car les marges des asservissements sont des points non définitifs et évolutifs (donc assujettis aux erreurs de conception). A priori, les correcteurs sont stables, mais on a le problème de la détection puisque les correcteurs sont le dernier maillon de la chaîne, et la surveillance des correcteurs ne peut pas être aussi avancée que les autres moyens de surveillance. Les autres erreurs possibles sont dans l'immédiat bien moins graves. Le risque majeur (en ce qui concerne cette I.T.), c'est donc bien les courants, et donc les composants. Dans tous les cas, le PFC en tête aura des protections, mais elles ne protègeront que lui-même et les dispositifs en amont, sauf si on prend en compte les RTSU. Pour les moteurs, on bénéficie de l'inertie. Si un défaut survient, la constante de temps des moteurs va jouer en notre faveur. On notera que beaucoup de défauts possibles, ayant des causes différentes, ont des conséquences communes. Bien que ces conséquences sont vues plus ou moins loin de la cause, les constantes de temps de détection sont assez faibles. On parle ici des conséquences telles que les erreurs des asservissements qui témoignent d'un défaut amont.

Finalement, cette analyse fonctionnelle comme présentée n'est qu'une synthèse de haut niveau, qui ne donne pas forcément des détails quantitatifs. Cette analyse est menée au fur et à mesure de la conception de l'ensemble.

| composant | modes de défaillance | cause | Effet | | Evaluation du risque | | | détection | | | risque général (RxD) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | local | sur le système | gravité (G) | occurrence (O) | risque (R=GxO) | détection (D) | moyens de détection possibles | solutions | |
| ISR_Adc | sur-intensités | mesures de courants erronnées | sur-intensité bus DC, destruction fusible | panne RTSU, plus de mouvements antenne | 4 | 1 | 4 | 4 | erreur des asservissement (temps de détection très long) | limitation en courant du PFC en tête | 16 |
| ISR_Adc | consigne de couple | mauvaise consigne issue du correcteur de vitesse | commande erronnée de mouvement | trajectoire antenne erronnée, risque de percuter les butées | 3 | 1 | 3 | 1 | erreur des asservissements | arrêt forcé des moteurs, changement de consigne par le calculateur | 3 |
| ISR_Adc | moteurs figés | codeurs figés | autopilotage figé | plus de mouvements de l'antenne | 4 | 1 | 4 | 1 | erreur des asservissements | redondance RTSU | 4 |
| ISR_Adc | moteurs avec couple réduit, ou mauvais sens de rotation | échec de l'indexage | mauvais contrôle moteur | mauvaises performances de balayage ou mauvais sens de rotation | 2 | 1 | 2 | 1 | erreur des asservissements, protection des butées | reset DSP | 2 |
| ISR_Adc | forme d'onde des courants | divergence des correcteurs | mauvaise poursuite des courants, destruction du fusible | mauvaises trajectoires, sur-intensités du PFC | 4 | 1 | 4 | 3 | erreur des asservissements (temps de réaction lent) | arrêt forcé des moteurs, changement de consigne par le calculateur, reset | 12 |
| ISR_Adc | mauvais contrôle moteur | erreurs sur les positions | courants chahutés | couple faux (réduit ou inversé), mauvais mouvements de l'antenne et/ou vibrations | 2 | 1 | 2 | 1 | erreur des asservissements, protection des butées, détection EPLD | redondance RTSU | 2 |

Fig. 52 : Extrait de l'application de la méthode AMDEC sur l'ISR des amplificateurs

Lorsque l'on soulève avec ce type de méthode des points délicats, on adapte à chaque type de difficulté une autre méthode qui va permettre de répondre au point dur. Ce type d'analyse a justement permis de soulever des interrogations quant à l'échange des données entre les différentes ISR (ou code non critique). Une étude des flux de données a donc été menée.

## d) L'analyse des flux de donnée

Avec un fonctionnement évènementiel, on n'a pas la succession de fonctions qui s'échangent des paramètres et des variables de manière formellement déterminées (mécanismes de passage). Les routines récupèrent des variables ou en mettent à jour indépendamment des autres routines, donc il faut une certaine vigilance lors de la conception.

Les DSP dédiés ont la particularité d'avoir des bus d'adresses typiques, et pour maximiser les performances, le bus d'adresse permet en un seul coup d'horloge d'adresser une « case » mémoire ou registre. Ainsi, on a un bus d'adresse de 22 bits. Le mode d'adressage est ainsi fait que chaque case mémoire est identifiée par une adresse de 22 bits, que la variable soit locale ou globale. La stratégie de programmation est donc optimisée pour l'utilisation de variables globales. Les interruptions vont s'échanger et se partager ce type de variables avec les mécanismes de priorités définis à la conception.

L'analyse en Fig. 53 permet d'identifier qui manipule les variables, et comment.

| | utilisé? (O/N) | Analog_Mes.X | Antenne_Cons.X | antenne_move.X | Calage_mot_C | Calage_mot_E | choix_C_E | compteur_transmit_IrDA | Control_Servo_Moteur | Control_Servo_Systeme | CorCourant_C_la.X | start_indexage | temoin_prioritaire | validate_asservissement | vect_gagant |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAL_main | O | | | | | | | | Wi | Wi | | | Wi | iW | |
| PAL_Calage_sequencement | O | | | | | | W | | | | | iW/R | | | |
| PAL_Calage_moteur | O | | | R | W/R | W/R | R | | | | | | | | R |
| PAL_Boot_PointageAuto | O | | W/R | | | | | | | | | | | | |
| PAL_Boot_PCTXParam | O | | | | | | | | | | | | | | |
| PAL_Boot_ModeSelect | O | | | W | | | | | | | | | | | |
| PAL_AutoTest_Boot | O | R | | | | | | | | | | | | | |
| PAL_AutoCalibration_Boot | O | W | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| ISR_SCI | O | | W | | | | | W | W/R | W/R | | | | | |
| ISR_ASV_1k | O | | W/R | W/R | | | | | | | | | | | |
| ISR_Adc_calage | O | R | | W | | | R | | | | | W/R | | | W/R |
| ISR_Adc | O | W/R | | W/R | | | | | | | W/R | | | | |
| | | | | | | | | | | | | | | | |
| FCT_visu_SERVO | O | | | | | | | | | | | | R | R | |
| FCT_test_CBIT | O | R | | | | | | | | | | | | | |
| FCT_sgti3c | O | | | | | | | | | | | | | | |
| FCT_SelectGeneCons | O | | W | | | | | | | | | | | R | |
| FCT_SCI_PC_TX | O | | | R | | | | | | | | | | | |
| FCT_SCI_IrDA_TX | O | | | | | | | W/R | | | | | | | |
| FCT_GenConsVit | O | | W | | | | | | | | | | | | |
| FCT_GenConsPos_xxxx | O | | | | | | | | | | | | | | |
| UpDown_32bits | N | | W/R | | | | | | | | | | | | |
| trajectoire_simple_xx | O | | | | | | | | | | | | | | |
| pointage_32bits | O | | W/R | | | | | | | W | | | | | |
| 32bits | N | | W/R | | | | | | | | | | | | |
| positionnement_32bits | O | | W/R | | | | | | | W | | | | | |
| FCT_ACQ_ADC_non_IT | O | W | | | | | | | | | | | | | |
| **REMARQUES :** | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 68 | 69 | 70 | 71 |

Légende :

W Write
iW Write et pré-initialisation
Wi Write initialisation
R Read

Note:
on ne tient pas compte des I/O physiques, constantes, static, locales

Fig. 53 : Synthèse de l'analyse des flux de données

Chaque interruption est conçue pour ne pas être dépendante d'un enchaînement entre une donnée (n) et une autre donnée (n+1), c'est-à-dire qu'il n'y a pas de lien temporel ou fonctionnel entre les données ; chaque interruption fonctionne seule avec des entrées interprétées comme venant d'un système extérieur. Chaque donnée n'est accessible en écriture que par une seule fonction, et dans les cas particuliers (ex : initialisations), la justification est détaillée. L'indépendance des interruptions et le caractère exclusif des accès permettent d'étayer la sécurité du logiciel.

Lorsque les données manipulées sont des structures, on fait la même analyse en descendant encore d'un niveau (donné en Fig. 54) pour aboutir à la donnée « mot » comme on peut le voir sur le cas suivant (donnée N°3 de la Fig. 53) :

| | utilisé ? (O/N) | Analog_Mes.X | Antenne_Cons.X | antenne_move.X | | | | | | | | Calage_mot_C | Calage_mot_E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | circulaire | | | | élévation | | | | | |
| | | | | vitesse_estime_oeil | vitesse_estime_IQ | now_position | last_position | vitesse_estime_oeil | vitesse_estime_IQ | now_position | last_position | | |
| PAL_main | O | | | | | | | | | | | | |
| PAL_Calage_sequencement | O | | | | | | | | | | | | |
| PAL_Calage_moteur | O | | | | | R | | | | R | | W/R | W/R |
| PAL_Boot_PointageAuto | O | | W/R | | | | | | | | | | |
| PAL_Boot_PCTXParam | O | | | | | | | | | | | | |
| PAL_Boot_ModeSelect | O | | | | | W | W | | | W | W | | |
| PAL_AutoTest_Boot | O | R | | | | | | | | | | | |
| PAL_AutoCalibration_Boot | O | W | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| ISR_SCI | O | | W | | | | | | | | | | |
| ISR_ASV_1k | O | | W/R | W/R | W/R | R | W/R | W/R | W/R | R | W/R | | |
| ISR_Adc_calage | O | R | | | | W | | | | W | | | |
| ISR_Adc | O | W/R | | | | W/R | | | | W/R | | | |
| | | | | | | | | | | | | | |
| FCT_visu_SERVO | O | | | | | | | | | | | | |
| FCT_test_CBIT | O | R | | | | | | | | | | | |
| FCT_sgti3c | O | | | | | | | | | | | | |
| FCT_SelectGeneCons | O | | W | | | | | | | | | | |
| FCT_SCI_PC_TX | O | | | R | | | | R | | | | | |
| FCT_SCI_IrDA_TX | O | | | | | | | | | | | | |
| FCT_GenConsVit | O | | W | | | | | | | | | | |
| FCT_GenConsPos_xxxx | O | | | | | | | | | | | | |
| UpDown_32bits | N | | W/R | | | | | | | | | | |
| trajectoire_simple_xx | O | | | | | | | | | | | | |
| pointage_32bits | O | | W/R | | | | | | | | | | |
| 32bits | N | | W/R | | | | | | | | | | |
| positionnement_32bits | O | | W/R | | | | | | | | | | |
| FCT_ACQ_ADC_non_IT | O | W | | | | | | | | | | | |
| REMARQUES : | | 1 | 2 | 3 | | | | | | | | 4 | 5 |
| | | | | 3,1 | 3,2 | 3,3 | 3,4 | 3,5 | 3,6 | 3,7 | 3,8 | | |

Fig. 54 : Détail de l'analyse des flux de données (exemple sur item n°3)

Cette analyse est très fastidieuse, mais elle permet de vérifier chaque accès de chaque donnée, donc on peut écarter des mauvais fonctionnements de conception. Ici, on peut voir des accès multiples en écriture de « now_position », mais on notera qu'il s'agit en premier lieu d'un accès au démarrage du système pour avoir la position de l'antenne, qui est remplacé par un accès en « mode indexage moteur » en interruption pour suivre le mouvement de l'antenne, puis cette interruption est remplacée par l'interruption du fonctionnement normal. A chaque étape, la

donnée n'est manipulée en écriture que par une seule fonction. L'analyse est ainsi reproduite pour les autres structures et ISR.

## IV. La protection de la mécanique

Le déplacement de la tête qui supporte le plateau hyperfréquence est borné par des butées mécaniques intégrées à la structure pour les deux axes, puisque c'est un radar à balayage dans le radôme de l'avion (voir Fig. 20). Ces butées ne sont prévues que pour la protection du plateau hyperfréquence, comme dans le cas de l'antenne hors tension. La cinématique azimut est bornée à ±90° et à ±45° pour l'élévation. Les moteurs et leurs commandes sont quant à eux capables de tourner sans limitation angulaire. Il est donc nécessaire que le logiciel prenne en compte ces données. Le véritable problème apparaît quand l'avion est en vol et que le fuselage n'est pas aligné avec le sol (roulis, tangage et cap, voir Fig. 75). En effet, dans ce cas, le repère « antenne » qui est fixe et aligné avec le fuselage n'est plus confondu avec le repère terrestre. Il faut alors prendre en compte les attitudes de l'avion pour corriger les trajectoires de l'antenne pour qu'elle balaye toujours dans le repère terrestre duquel proviennent les consignes de balayages. Dans ce cas, il existera un grand nombre de cas de figure où le balayage demandé sera en dehors des zones atteignables par l'antenne (hors des butées mécaniques). Imaginons que le radar balaye la piste comme illustré sur le cas d'école en Fig. 55 et que seul le cap change sans que les consignes terrestres changent :

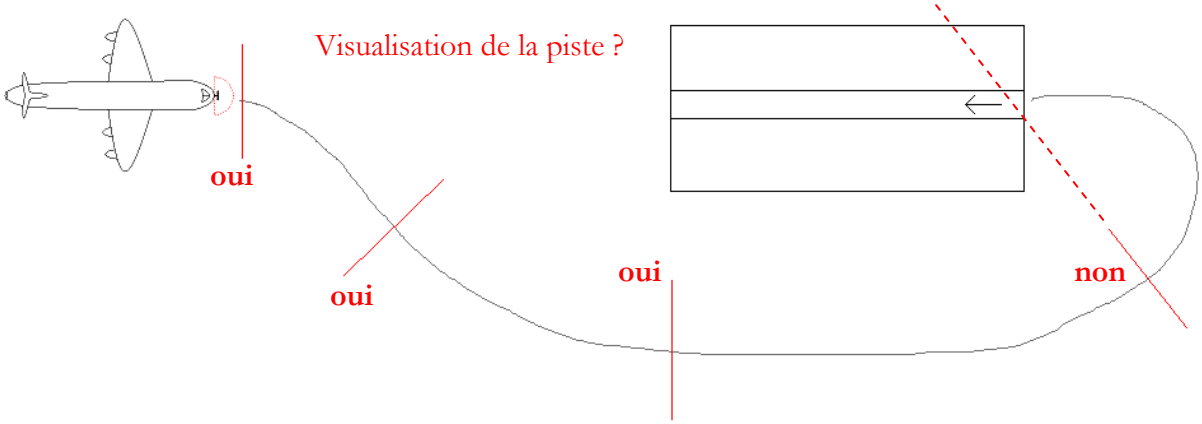



Fig. 55 : Exemple d'une trajectoire avion et de la zone de balayage radar

Il faut donc empêcher l'antenne de balayer contre les butées. On a le même genre de problème lorsque l'antenne balaye à pleine vitesse proche des extrémités, et que l'avion bouge rapidement suivant les trois axes avec en plus l'effet de l'accélération du porteur. Il faut donc calculer les

changements de repère, mais aussi anticiper les temps et angles avant les butées en fonction des attitudes et des accélérations pour pouvoir limiter les mouvements de l'antenne à une zone de marge avant les butées mécaniques (qui ne doivent jamais servir). Les mouvements du porteur modifient les trajectoires (donc la « position » des butées) tandis que les accélérations limitent les capacités des amplificateurs à contrôler les moteurs (en temps et en angle). On a donc une forte corrélation entre cette problématique et les puissances électriques des moteurs. Les simulations Matlab®/Simulink® intègrent les changements de repère, et un simulateur sous Visual Studio® permet de calculer les temps et les angles avec les limites que l'on se fixe par un calcul très rapide au format du DSP compte tenu des dynamiques et pour limiter la charge de calcul du DSP. Des environnements de simulations différents ont été préférés pour rendre l'ensemble moins complexe à développer et à utiliser.

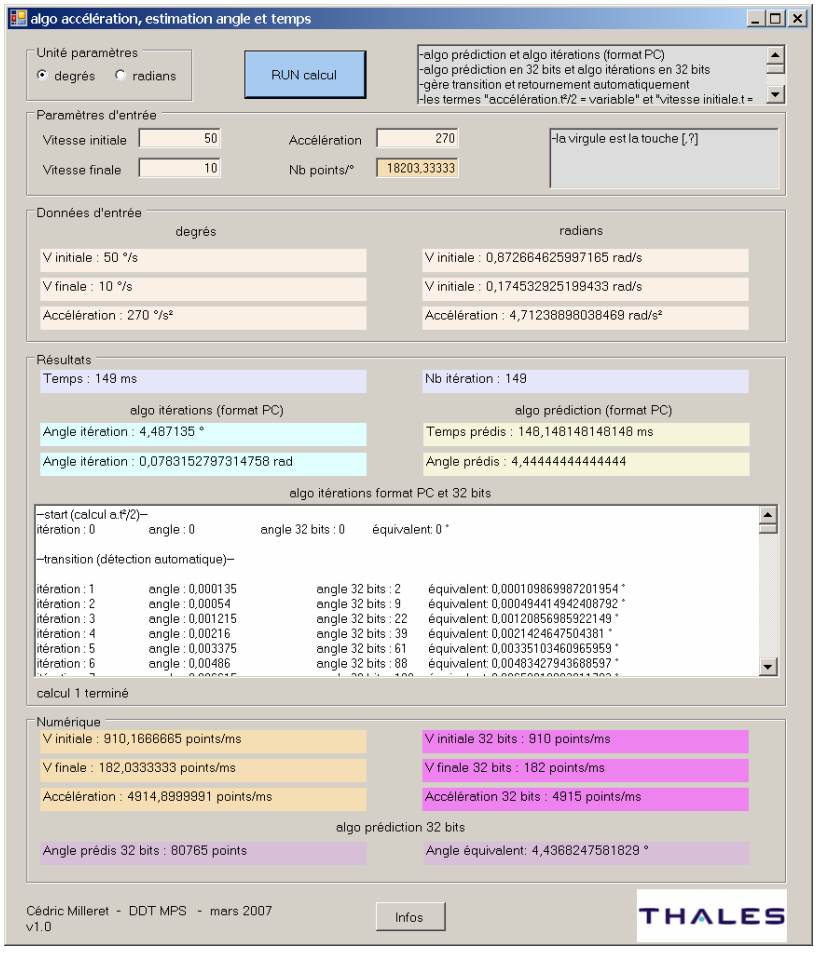



Fig. 56 : Simulateur des angles et des temps pour les trajectoires

Pour le fonctionnement en poursuite, la vitesse finale vaut « 0 », la vitesse initiale est rafraîchie à chaque calcul, la capacité d'accélération des moteurs est fixe. L'angle et le temps limites avant la marge (avant les butées mécaniques) sont retournés à la fonction de surveillance qui décide de piloter l'antenne au lieu du générateur de consigne qui définit la trajectoire initiale.

# V. Bilan de la simulation/conception

La simulation d'un système complexe sous un environnement unique construite par un ingénieur généraliste est encore une perspective assez utopiste. En effet, malgré les progrès encourageant des simulateurs pluridisciplinaires tel que Matlab®, il est encore difficile d'intégrer toutes les contraintes du système à simuler, et surtout d'en analyser les résultats. L'efficacité du code généré est là aussi un problème majeur puisque la compacité et la rapidité d'exécution du code généré sont primordiales dans l'implémentation d'un système embarqué aux ressources limitées. Il n'en reste pas moins que les résultats attendus ont été obtenus avec des moyens déployés certes plus importants. Les simulations Matlab®/Simulink® ont demandé plusieurs étapes, chacune étant itérative. Les simulations et bancs de tests sous Visual Studio® ont été des compléments à Matlab®. Les résultats définitifs ont été exploités sous CCS™ avec un codage manuel en virgule fixe nécessitant plus de 2000 lignes de code en interruption (ISR critique) parmi les 12000 lignes du projet suivant les normes et référentiels de développement.

# Chapitre 3 : indexage moteur

# I. L'indexage usuel

## A. La problématique

Les actionneurs de l'antenne sont des machines synchrones triphasées à aimants permanents montés en surface (SPMSM : Surface Permanent Magnet Synchronous Motor). A ce titre, c'est une machine qui génère un couple maximal si et seulement si le champ magnétique appliqué au stator par un onduleur de tension (voir Fig. 57) est en quadrature avec le champ fixe et constant du rotor, dans son repère, obtenu par des aimants permanents (voir Fig. 58). Tout écart avec cet angle de pilotage de la machine [BER02] se traduira par une baisse du couple avec la même commande électrique, c'est-à-dire une baisse du rendement.

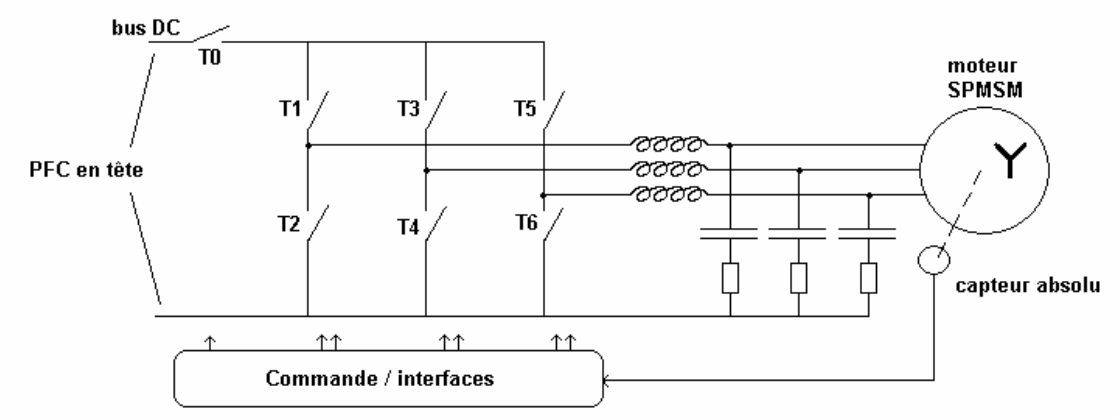



Fig. 57 : Onduleur du radar pour un axe, avec ses connexions au reste du système

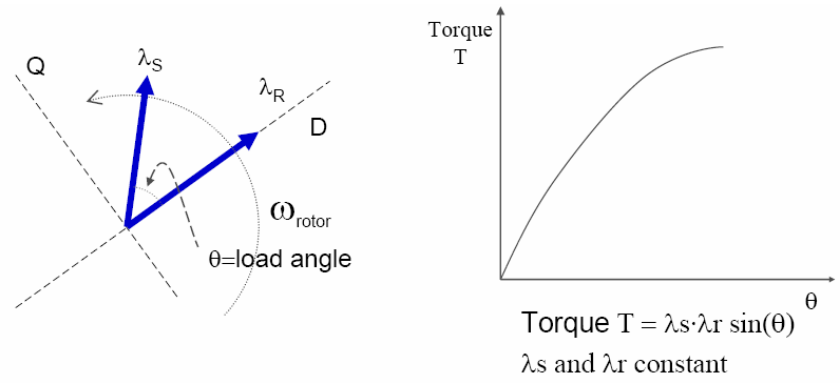



Fig. 58 : Angle de pilotage et couple

La structure onduleur (un pour chaque axe) est un pont triphasé (T1 à T6) sans neutre, dont l'alimentation de la puissance est activée par un interrupteur principal (nommé MAIN par la suite). Cette validation principale de l'alimentation est commune aux deux onduleurs. Le PFC en

tête assure un bus continu constant. Sur un bras de pont, les interrupteurs sont complémentaires, on ne peut pas laisser une phase déconnectée. L'onduleur est chargé par un filtre LC amorti (simplifié sur le schéma) sur chaque phase, suivi du moteur en étoile. La nécessité des filtres est justifiée par le contrôle moteur basse vitesse qui demande des signaux de commande sinusoïdaux, avec une fréquence de découpage qui ne doit pas se propager sur la connectique des moteurs, qui est à proximité du plateau rayonnant hyperfréquence. Cette structure permet une commande du moteur à basse vitesse avec un couple constant.

La commande en phase « exacte », permettant un couple constant, est primordiale sur une machine synchrone pour ce type de fonctionnement, c'est pour cela qu'il faut un capteur rotorique, ou alors s'en passer avec une commande sans capteurs « sensorless » [BRU96] qui fonctionne avec des algorithmes adéquats, tel que des observateurs. Dans tous les cas, la commande par un capteur reste la solution qui donne les meilleures performances, mais qui est coûteuse ; et il faut toujours une commande d'un champ par rapport à l'autre. La rotation de la machine est obtenue en maintenant un champ tournant au stator en avance de phase sur le champ du rotor. La position des deux champs doit être connue. Le champ du stator est caractérisé d'une part par la commande qui le définit, et par rapport à la position des enroulements statoriques sur le châssis. La position du rotor est définie par la position des aimants sur l'axe. Donc, pour synchroniser les deux champs, il faut connaître la phase de la commande statorique, ce qui est a priori sans problème car elle est générée par le DSP, mais il faut surtout connaître les positions mécaniques du stator et du rotor. Le repérage de ces deux positions est un indexage, peu importe la méthode pour y parvenir. La position mécanique du stator est fixe et doit être référencée, la position du rotor évolue avec la rotation, d'où le capteur ou un autre moyen (estimation) qui soit référencé ou qui le permette. Dans notre cas, on a une commande avec capteur, car on fonctionne à basse vitesse, et l'on dispose déjà d'un capteur pour les fonctions d'asservissements. On doit donc déterminer la position relative du capteur par rapport à l'axe polaire d'une paire de pôles et du champ statorique.

En fonction du nombre de paires de pôles, les angles électriques de la machine varient, ainsi, avec notre moteur de dix paires de pôles, une période électrique correspond à 36° mécaniques. Le référencement mécanique devient d'autant plus précis que l'angle électrique devient petit, car plus l'incertitude du référencement augmente (pour une même précision d'assemblage), plus le couple disponible se disperse.

## *B. Les contraintes*

Le contexte du radar aéroporté limite les choix techniques et technologiques, et notamment pour référencer les positions initiales. En effet, le radar aéroporté est un système à « bas » coût (l'aéronautique est aussi concurrentielle), et l'ensemble des restrictions de sûreté et de fiabilité s'ajoute aux contraintes de coût en ce qui concerne les stratégies de conception. Le matériel est limitant d'un point de vue électromécanique puisque le fonctionnement du radar dans son environnement, de surcroît en entraînement direct, ne fait pas appel à un fonctionnement traditionnel des actionneurs. Les principales contraintes sont données ci-dessous :

- Contraintes de conception :
  - Il n'y a pas de mémorisation dans une mémoire embarquée (ni dans le DSP, ni dans l'antenne, ni reliée sur le RTSU). Les RTSU doivent être interchangeables. Une mémoire coûte trop cher et n'est pas fiable (il faut voir la mémoire avec tout ce qu'elle implique, comme la connectique, le blindage, la gestion des articles, etc.). Une simple mémoire auxiliaire engendre un réel surcoût. Durant la vie de l'antenne (environ 25 ans), on peut changer de mécanique, donc l'assemblage des moteurs ne correspondra plus à la version précédente. On peut tout aussi bien changer l'électronique embarquée, qui est indépendante d'une quelconque configuration (positions des moteurs par rapport aux codeurs). On ne peut procéder ni à un indexage en usine (ex : intervention manuelle) qui est trop coûteux, ni à un indexage au 1$^{er}$ démarrage qui n'est pas en adéquation avec l'interchangeabilité (l'électronique est indépendante des moteurs, des codeurs, de l'antenne en général).
  - L'algorithme d'indexage est soumis aux mêmes contraintes que pour un fonctionnement opérationnel : on n'utilise que la mémoire interne, avec les règles de codages imposées.
  - La mémoire morte « programme » du DSP est figée, et n'a pas le droit d'être modifiée, par lui-même (le programme du DSP se met à jour par la modification de paramètres pré initialisés), ou un autre mécanisme (contraintes de la DO-178). Tout le programme doit être conçu à l'avance.
  - L'indexage a une durée fixe et limitée. Il est compris dans le temps total de démarrage.

- Le fonctionnement opérationnel ne doit pas être perturbé/dégradé. Le fonctionnement opérationnel est le fonctionnement de l'antenne avec toutes ses séquences de fonctionnement.

- Contraintes électromécaniques :
  - L'antenne peut être soumise à de fortes vibrations, ou accélérations pendant le vol (de 3G à 6.5G suivant l'axe). Ces contraintes sont valables pour le régime fonctionnel, indexage compris.
  - L'antenne peut subir une réinitialisation en vol (redémarrage), par perte d'alimentation, ou suite à un problème. L'antenne n'a pas de réserve d'énergie, donc il n'y a pas de mémorisation de courte durée. Le mode opérationnel et fonctionnel implique que l'antenne soit à performances nominales quel que soit le cas de figure. L'indexage fait partie du mode opérationnel.
  - Les moteurs doivent être indexés à ±5° électriques, afin que le couple maximal disponible soit constant. Les asservissements sont déterministes, et corrélés avec le faisceau hyperfréquence.
  - L'antenne ne doit pas percuter les butées. Elles ne sont prévues que pour la sécurité de la mécanique.
  - La procédure d'indexage utilise les mêmes composants matériels que pour le fonctionnement en balayage (pas de composants supplémentaires). C'est la même chaîne de mesure (capteurs compris).
  - L'indexage respecte les conditions maximales de consommations : courants de phases, et courant d'alimentation. Le PFC en tête a un dimensionnement unique ! On doit aussi fonctionner avec une tension d'alimentation étant au minimum du gabarit (normes réseau).
  - Les paramètres moteurs ne sont ni constants en utilisation (variation avec la température de -40°C à +70°C), ni constants en fabrication (L est donnée à ±30% et R à ±10%).
  - Les moteurs sont précédés de filtres (cellules LC) pour reconstituer des motifs de courants sinusoïdaux de basse fréquence, nécessaires pour préserver la qualité du faisceau hyperfréquence, mais aussi pour le bon fonctionnement du contrôle moteur et des asservissements. L'inductance série a donc sa propre tolérance qui s'ajoute pour chaque phase.

Le contrôle moteur ne peut fonctionner que si l'on peut synchroniser les champs. L'indexage est donc une étape initiale au contrôle moteur. Selon le type de commande, l'indexage n'aura pas les

mêmes contraintes. Les contraintes citées ci-dessus ne sont pas forcément valables dans chaque stratégie de commande.

## *C. Les démarrages des contrôles-moteurs*

Il existe de multiples manières de parvenir à un référencement des champs, selon le type de contrôle moteur utilisé. Notre créneau est particulier puisque l'on veut utiliser des stratégies utilisées pour les problématiques sans capteurs alors que nous avons un capteur, car nous ne pouvons pas mémoriser les paramètres d'indexage (détaillé au §I.B précédent). Dans le cas des méthodes de contrôles moteurs sans capteurs comme la détection de la F.E.M. (pVIII), le démarrage est un problème car il n'existe encore pas à ce moment là une mesure qui permette de déterminer la position du rotor. L'indexage est donc une étape distincte souvent basée sur des essais préliminaires avec des courants qui ne sont pas directement liés à la commande par retour de F.E.M. Cette commande n'est pas intéressante pour notre application car c'est une stratégie qui fonctionne pour des vitesses de rotation significatives du moteur où la F.E.M. devient mesurable (bon rapport signal/bruit). Les commandes par observateurs sont aussi des commandes moteur sans capteur, dont les performances et les architectures sont assez dispersées [ZHE07] [ZOL96] selon l'aspect qui est privilégié (couple au démarrage, ondulation de couple, stabilité en vitesse, etc.). Le démarrage est un transitoire d'initialisation, même pour une commande en couple. Les observateurs convergent à partir de la vitesse nulle afin de se stabiliser en un minimum de temps pour commencer à bien contrôler la machine. Ce temps dépend de l'architecture de l'observateur. L'indexage n'est pas ici une stratégie différente du contrôle moteur par observateurs, donc elle n'est pas utilisable pour notre application.

Pour l'application du radar météo, nous avons une commande autopilotée par courants sinusoïdaux qui permet d'avoir une commande du moteur à très basse vitesse, avec un couple constant sans ondulation. Cependant, nous avons la contrainte de connaître la position des champs avec une précision de 5° électriques pour satisfaire à l'obligation de déterminisme du couple, exigé pour les asservissements. Le référencement des axes doit être connu à l'avance, car le démarrage du contrôle moteur n'est pas un transitoire comme le traitent les méthodes par observateurs. L'indexage initial est une étape distincte.

# II. L'indexage automatique

## *A. Le phénomène recherché*

Les stratégies possibles pour indexer le moteur au démarrage dans le cas du radar météo sont celles qui sont distinctes du contrôle moteur. Comme pour la commande par back-EMF (F.E.M.), le référencement des axes par des essais en courant semble approprié.

On s'appuie sur la méthode INFORM (INdirect Flux détection by On-line Reactance Measurement) [SCH88] [SCH90] [SCH97] [SCH02] [SCH07] qui est basée sur la distribution du flux dans la machine par les aimants permanents, ce qui provoque des différences sur les inductances Xd et Xq suivant l'axe direct et l'axe en quadrature (transformation de Park) dues à des effets de saturation localisés. Xd et Xq sont fonction des inductances de fuites provoquées par les couplages entre les phases, et les couplages rotor/stator comme montré en Fig. 59.

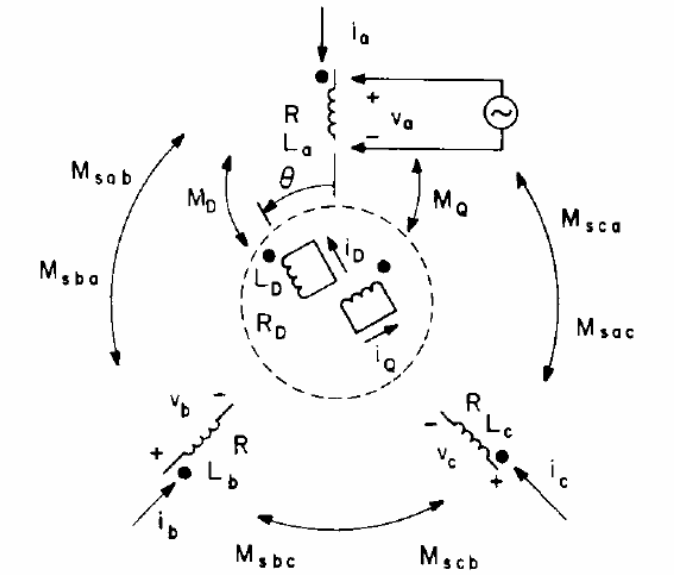



Fig. 59 : Représentation des inductances couplées [GOR88]

Les effets de saturations localisées [PER07] sont mesurables à travers des acquisitions de courants et de tensions. La machine synchrone à aimants permanents présente une relation entre la position du rotor et la valeur des inductances des enroulements variant avec le déplacement du circuit magnétique [CAR93]. La représentation des machines synchrones du radar météo, avec dix paires de pôles, est donnée Fig. 60 :

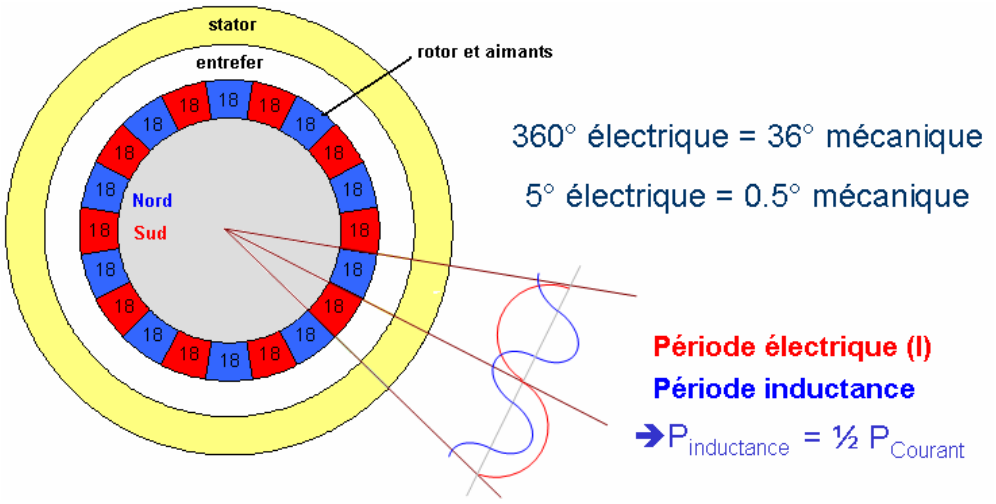



Fig. 60 : Période électrique en fonction de la période d'inductance

La période de l'inductance vue aux enroulements du stator est le double de celle de la période électrique de la machine, donc en apparence on ne peut pas différencier les pôles par des mesures (courants faibles [KIE02]) puisqu'ils sont symétriques. La variation sinusoïdale des inductances (Fig. 61), mesurées pour le relevé avec un pont d'impédance aux fréquences proches de nos essais, permet par contre de trouver plus facilement les extremums par une rotation de la machine. On voit qu'avec cette technologie de moteurs (aimants collés vraisemblablement avec une faible saillance de montage), la variation n'excède pas 15% de la valeur moyenne. Cette faible variation nous permet de faire des impulsions équivalentes quelle que soit la position initiale (inconnue) du rotor, pour la suite présentée au §II.B (donc sans exploiter l'effet de saillance par une rotation, mais seulement à partir d'une position statique qui nous indiffère).

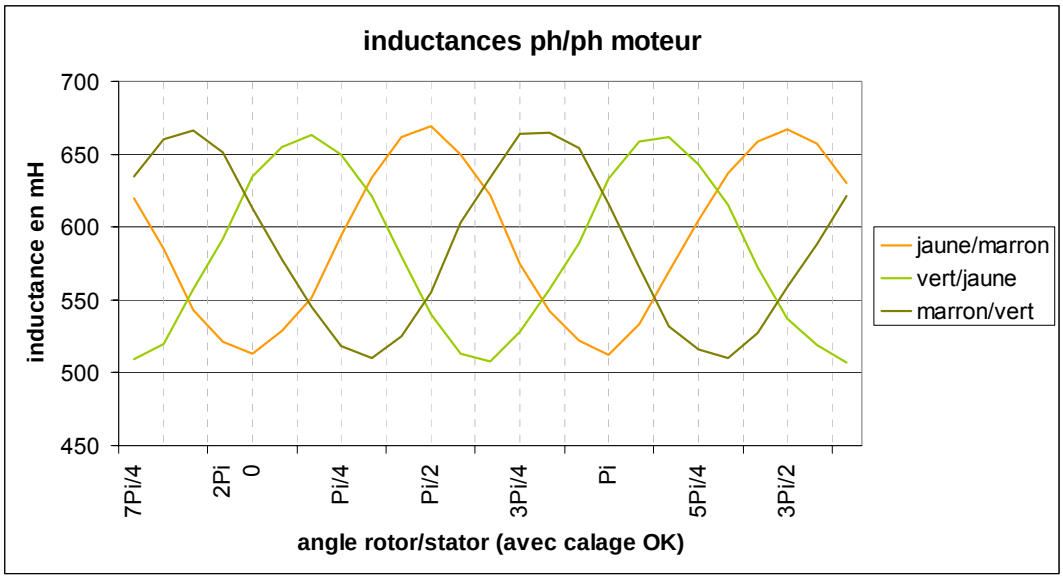



Fig. 61 : Relevé inductance moteur ph/ph en fonction des positions

Les essais de courant sur les enroulements ont pour but de mettre en évidence, via les mesures (non représentées), les variations des inductances statoriques par l'effet d'induit, où une impulsion aura un effet additif, et l'autre soustractif par la création de pôles statoriques face aux pôles rotoriques déjà présents. L'idée est de toujours réaliser les mesures avec des conditions initiales équivalentes. Le point de départ, défini par la polarisation initiale due aux aimants (Fig. 62), ne nous intéresse pas, seule l'amplitude finale avec les deux trajets possibles est la grandeur que nous exploitons (Fig. 63).

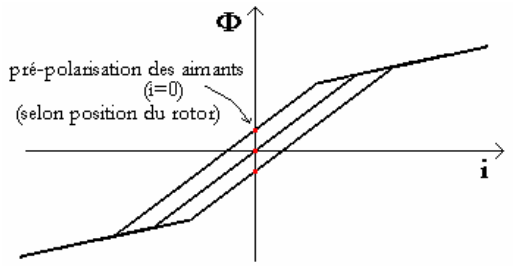



Fig. 62 : Illustration du cycle i(Φ) et de la polarisation initiale des aimants

Suite à un créneau de tension constant en amplitude et en temps appliqué entre deux phases (détaillé en Fig. 70), on crée une variation de flux dans le moteur. Comme l'inductance équivalente résulte de la position des aimants du rotor dans le circuit statorique, on mesure la réponse du courant (car Φ=L.I.).

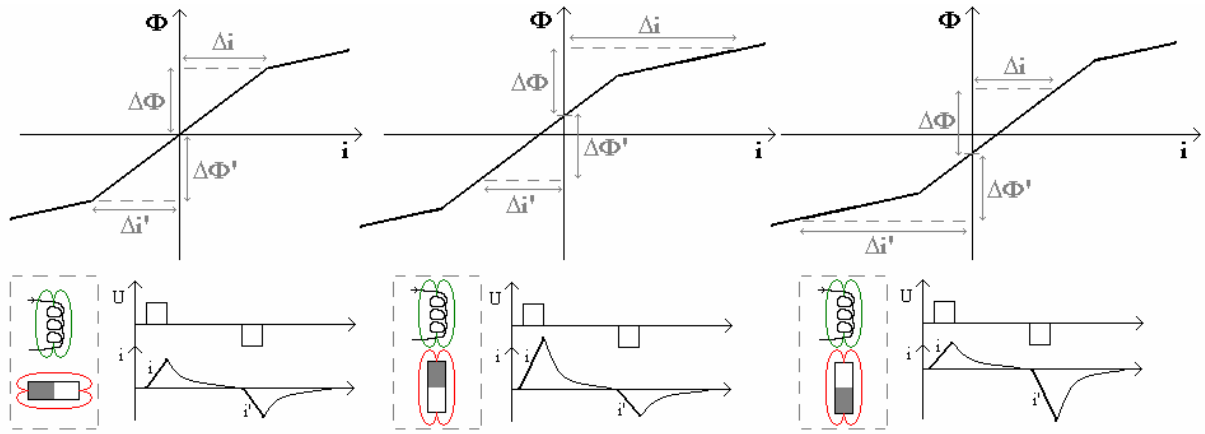



Fig. 63 : Illustration de l'influence de la position des aimants sur le cycle i(Φ) pour le courant mesuré

Dans notre problématique d'indexage, on exploite le phénomène de saturation, apparaissant sur le cycle i(Φ), mais avec des amplitudes assez peu différentes. Les méthodes de type INFORM permettent de mesurer cette petite variation.

## *B. La méthode des essais de courant par impulsions*

A l'arrêt, des essais avec des $\Delta I/\Delta t$ (mesure de la variation de l'amplitude du courant pendant un temps donné suite à une impulsion de tension) permettent de borner les positions des pôles magnétiques. C'est en exploitant la variation de l'inductance (par le phénomène de saturation) que l'on va pouvoir mesurer des amplitudes de courant différentes en fonction des impulsions générées. On retrouve les méthodes d'identification des pôles par la combinaison des vecteurs (expliqué plus loin). La méthode INFORM permet de localiser les pôles avant de démarrer, les résultats étant réutilisés pour initialiser l'observateur faisant tourner la machine sans capteurs [SCH90].

Pour le fonctionnement à vitesse non nulle, une commande dans le repère de Park, avec un estimateur de Kalman[9] en plus des informations de « bruit » de la méthode INFORM, est mise en œuvre pour estimer la position. La méthode exposée est mise en défaut lors de variations de couple. Il faut alors utiliser un procédé du 1er ordre Gauss-Markov[10]. L'ensemble de la méthode manipule beaucoup d'équations et de matrices, dont nombre de paramètres sont prédéfinis via un modèle devant être très précis. On notera des imprécisions sur la position. Cette méthode est complexe en ce qui concerne les lois de commande, mais l'indexage n'est cette fois ci pas seulement l'obtention de conditions initiales pour aboutir à la convergence de filtres, mais c'est une réelle localisation de pôles par des mesures physiques (courants). Les précisons obtenues ne sont pas spécifiées, mais sont généralement inférieures à 30° électriques avec ce type d'essais.

On trouve principalement des précisions d'indexage de 30° électriques ou moins [TAK96] [JAK05] (avec aussi des mesures de tensions [HAS07]). Les travaux qui ont permis d'obtenir de meilleures résolutions ont été faits avec des machines appropriées dans des conditions adéquates, avec des commandes numériques [NAK00] [TUR03] [POP07]. Une grosse partie des travaux que l'on trouve sur les machines concernent celles de moyennes puissances [OST96] [WIS07] où le coût du capteur est significatif et où l'on a besoin d'un rendement et/ou d'un démarrage en charge bien supérieur aux petites machines « d'usage courant » ou peu contraint.

Ces méthodes, telles qu'elles sont présentées, sont limitées à des cas de fonctionnements presque idéaux ou avec trop de capteurs pour obtenir de meilleures résolutions [NAI92] [STI99] [ICH04] [HAR05a] [HAR05b] [RAU07]. Ces méthodes ont été testées pour l'application du radar

[9] Le filtre de Kalman permet de traiter des informations dans le bruit (ex : vitesse faible) où ici, le but est d'homogénéiser la qualité des estimations sur toute la plage de fonctionnement.

[10] Le procédé de Gauss-Markov permet de mettre en œuvre des estimateurs linéaires non biaisés par la méthode des moindres carrés. Ici, on traite les variables d'état non mesurées de vitesse et de position dont l'estimation (via les filtres de Kalman) est moins bonne avec des variations de charges.

météo, mais sans succès car il n'a pas été possible de mesurer rapidement les variations de l'inductance (via l'effet de saturation, pas celui de la saillance liée à la position) avec les essais en courant, ou d'utiliser plus de capteurs, ou aussi de faire des essais à courants réduits. La dispersion des phases est aussi un point limitant. Notre machine présente une trop forte linéarité (développé plus loin) comparée à la plupart des technologies de machines [NOG98] [MAT94] [KIM03] (IPM et même SPM, pVIII), et les capteurs de courants ne sont pas assez précis pour mesurer des petites grandeurs compte tenu de leurs grandes excursions.

Pour améliorer le fonctionnement dans les faibles vitesses, on trouve une évolution [SCH07] dont le synoptique Fig. 64 montre une commande combinée EMF/observer avec un démarrage par INFORM, qui permet de faire un démarrage (à vitesse nulle) avec un fort couple de charge ; puis lorsque la vitesse augmente (apparition de back-EMF), une estimation des positions rotoriques par observateurs s'ajoute au référencement initial.

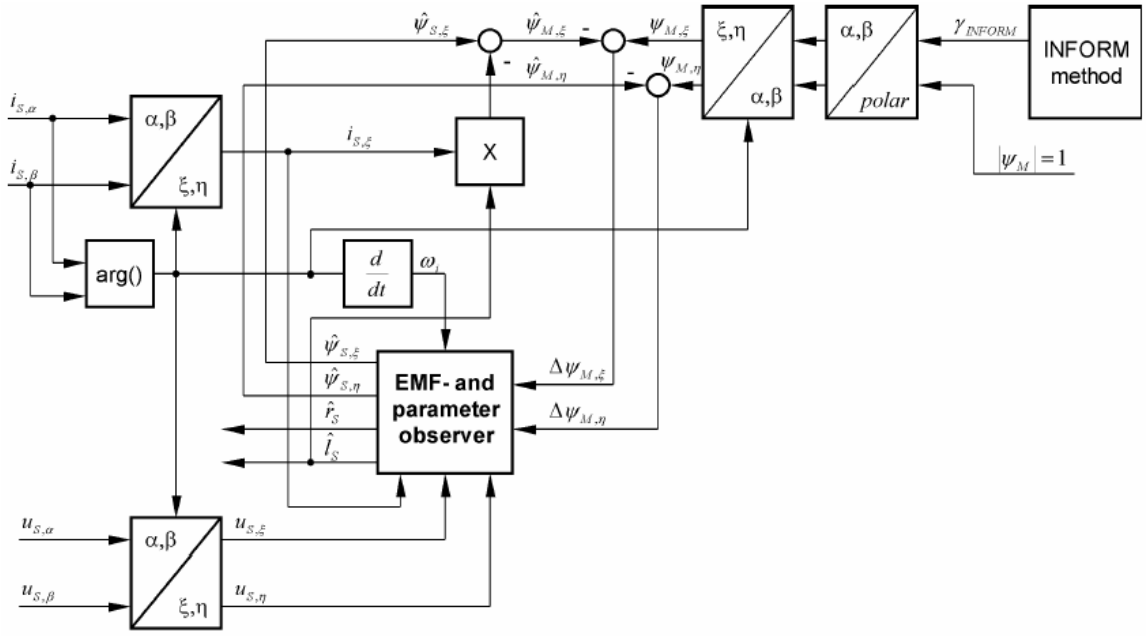



Fig. 64 : Synoptique d'une commande combinée E.M.F./observer avec un démarrage par la méthode INFORM [SCH07]

On travaille toujours dans le repère de Park, donc avec des correcteurs sur (d,q) comme dans la plupart des commandes numériques. Les commandes sans capteurs sont toujours délicates dans les faibles vitesses et les forts couples. La méthode INFORM est limitée aux démarrages compte tenu des essais en courant, qui sont réalisés de surcroît sans F.E.M. parasite. La commande devient assez complexe avec un fonctionnement combinant les machines à états finis, et des lois de commandes numériques. Le DSP se prête alors très bien à cet exercice. On voit en Fig. 65 cette commande dans un DSP TMS320F2407©, modèle inférieur au F2812 utilisé

pour le radar, où on commute entre la localisation des pôles par INFORM et une commande par observateurs selon la vitesse de rotation du moteur (on n'est plus dans l'exemple en Fig. 64 où il y a une utilisation permanente de la valeur de référencement).

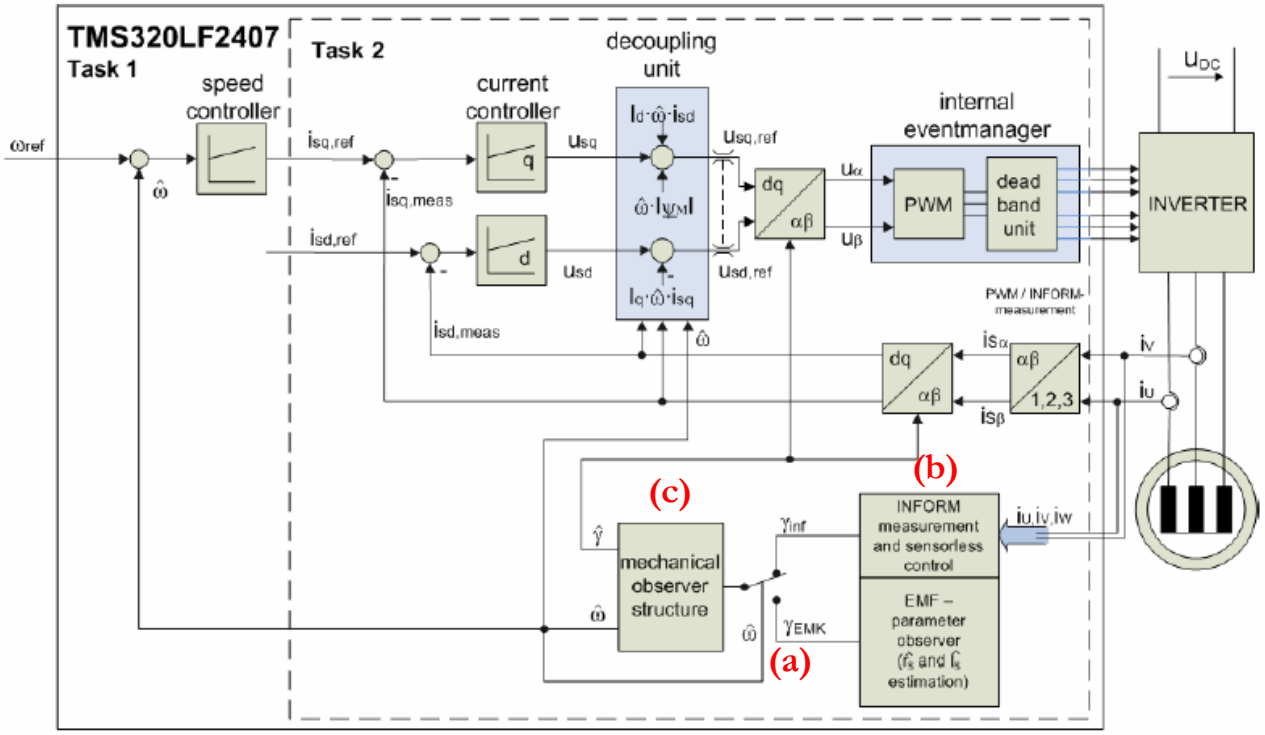



Fig. 65 : Synoptique d'une commande combinée EMF/observer avec démarrage INFORM dans un DSP TMS320 [SCH07]

La commutation (a) entre les 2 types d'information de position (b) est assez bien visible, avec un «passage» par palier de la vitesse $\hat{\omega}$ donné par l'estimateur de vitesse (c). Le reste de la commande a déjà été vu, et est découplé de cet aspect novateur. On remarque que la « task2 » est a priori l'interruption rapide (si réalisée en interruption évidement), et cette tâche est commune pour les correcteurs de courant, les estimations de positions et tous les artifices intermédiaires. On peut en déduire que le DSP est tout à fait capable de réaliser ces commandes rapides.

La méthode INFORM est adaptée à notre problématique de démarrage pour référencer le système pour toute la durée du fonctionnement. C'est bien un moyen de démarrage distinct et numérique qui peut être combiné à n'importe quelle méthode de contrôle moteur. Notre application a la particularité d'avoir un capteur mécanique (voir Fig. 57) de forte résolution sur l'axe de rotation (donc du rotor), qui n'est pas justifiée par le contrôle moteur. En effet, pour les asservissements de la tête radar, chaque axe a son propre codeur absolu d'une résolution conséquente, puisque l'on asservit à 0.25° (la résolution du codeur est donc supérieure à la précision de l'asservissement, et notamment à la résolution nécessaire au contrôle moteur), mais

le capteur est référencé au zéro des asservissements (repère antenne). L'indexage initial que nous allons utiliser va permettre de référencer le moteur au codeur, puis le contrôle moteur très basse vitesse va se servir des informations du codeur et de sa référence pour son propre fonctionnement. Il faut bien faire la distinction entre la procédure d'indexage et l'utilisation du codeur absolu (qui n'est pas exploitable directement au démarrage), qui sera ensuite exploité par l'autopilotage.

Les travaux de [NAK00] s'appuient sur la stratégie INFORM pour indexer une machine synchrone à aimants permanents montés en surface de 400W à deux paires de pôles, de tension réseau 230V/400V. Le principe consiste à exploiter les phénomènes de magnétisations non linéaires du stator de la machine en fonction des aimants du rotor, via des vecteurs déterminés en amplitude et en temps. Ces vecteurs sont obtenus par des impulsions générées par les bras d'onduleur (Fig. 66). On fait la différence entre les combinaisons complémentaires (en haut du trait de séparation) qui sont toujours possibles avec des drivers classiques, et les combinaisons nécessitant des drivers séparés des composants de puissance (en bas de la figure).

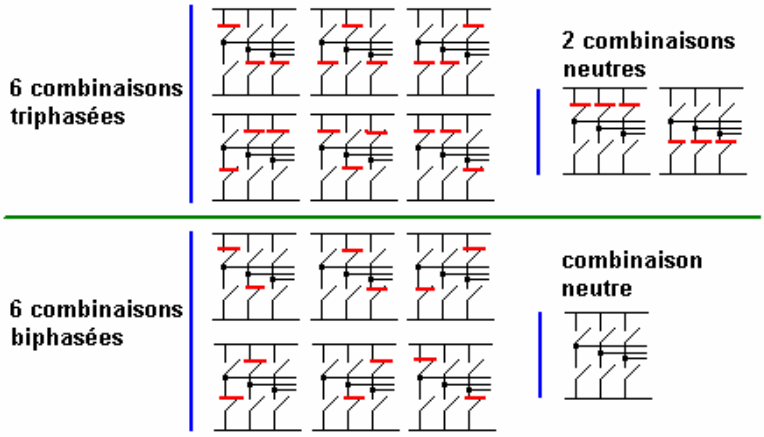



Fig. 66 : Combinaisons possibles des interrupteurs

Ces vecteurs sont soit directs avec la commutation d'un bras, soit issus d'une combinaison d'impulsions successives afin de générer un vecteur qui n'est pas directement issu d'une commutation. Les vecteurs sont tous appliqués pendant une même durée pour obtenir des amplitudes comparables, dont la durée est dépendante des caractéristiques du moteur. Cette méthode permet de réaliser un indexage sans mouvements de la machine, avec une précision inférieure à 5° électriques. La position fixe du rotor va être sollicitée par un panel de vecteurs, qui réagiront différemment selon l'état magnétique du rotor (nord, sud, intermédiaire…), ces vecteurs créant des pôles complémentaires au stator. A chaque vecteur appliqué (avec ou sans combinaison d'interrupteurs), la mesure en amplitude du courant associée pendant le temps fixe de l'échelon de tension va permettre de déterminer le vecteur suivant le plus adéquat ayant pour

but de se rapprocher du vecteur le plus compatible avec le pôle nord du rotor. En d'autres mots, chaque vecteur qui se succède (pôles au stator) sera de plus en plus proche du pôle magnétique nord du rotor, ayant pour finalité de connaître la phase exacte du dernier vecteur pertinent. Ainsi, le point de référencement pourra être déduit. Cette méthode aboutit à un résultat au bout de quelques dizaines de millisecondes.

Ce principe est régi par l'organigramme en Fig. 67. Le repère (d,q) est utilisé, mais on travaille bien à partir des mesures triphasées du repère (a,b,c).

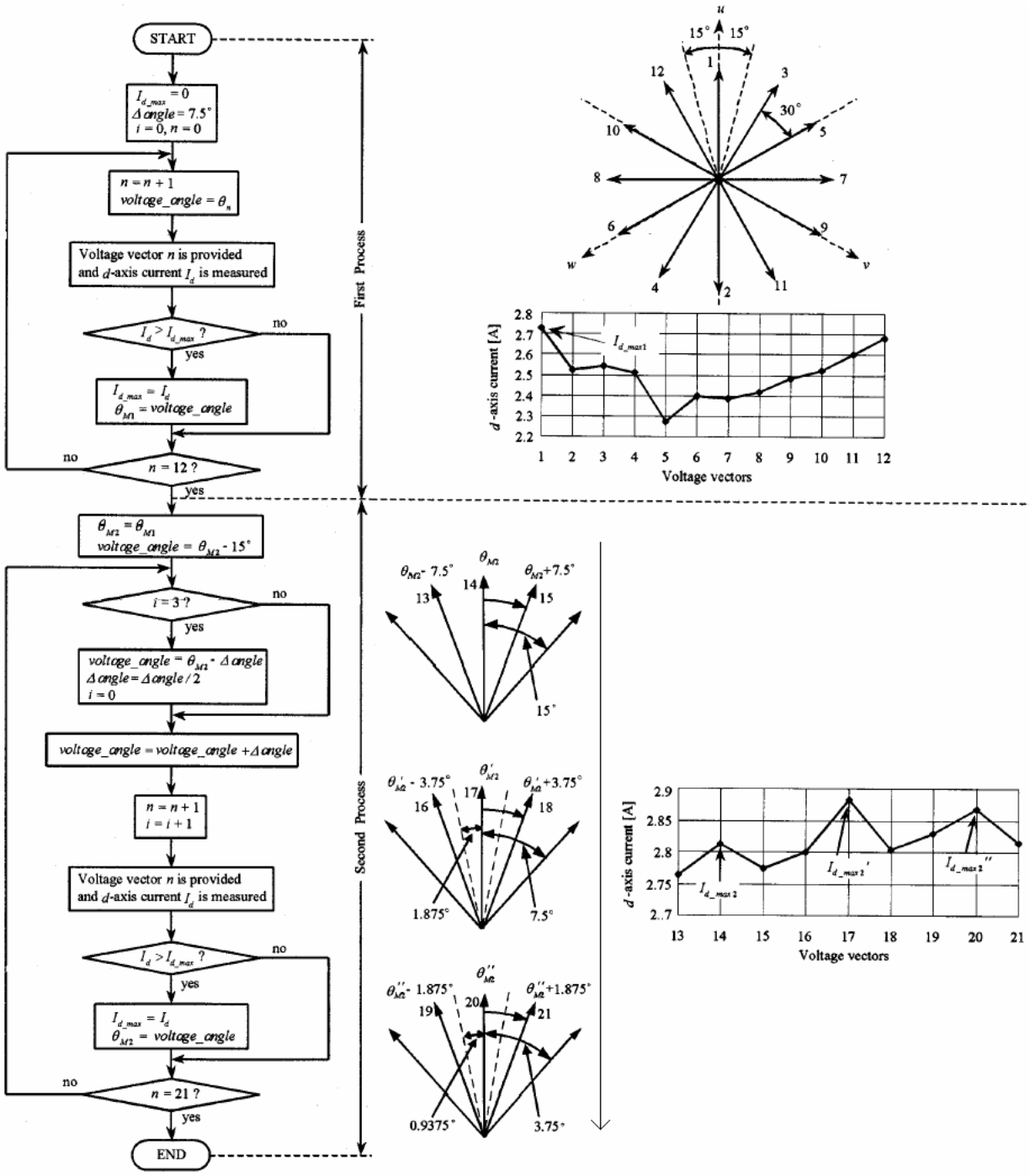



Fig. 67 : Organigramme de la méthode de localisation du pôle nord [NAK00]

L'exemple donné avec les vecteurs cités est un cas d'école. La description faite ci-dessus est donnée pour le cas où le moteur est référencé à 0° dès le départ. Ainsi, dans la 1ère phase du processus, le vecteur 1 est trouvé comme le plus proche du pôle nord ($\Delta I/\Delta t$ max) en sollicitant le moteur avec les 12 vecteurs possibles (6 avec les 3 phases, et 6 avec 2 phases et l'autre en l'air). La seconde phase du processus, itérative, va affiner cette localisation au travers de vecteurs intermédiaires, issus de combinaisons des vecteurs obtenus avec le pont triphasé. Le vecteur 14 est le vecteur 1 pour la phase suivante. La mesure de 13 et 15 réduit la fenêtre de recherche. Comme c'est un cas d'école, le 14 est à nouveau retenu, puis est remplacé par le 17 et ensuite par le 20. Ainsi, à la fin des itérations, la position 0° à ± 0.9375° est obtenue. Si l'on n'avait pas été exactement sur 0°, on aurait pu trouver comme dernier vecteur le 19 ou le 21. Toutes ces étapes ont lieu avec un rotor parfaitement immobile, sinon les mesures n'auraient aucune signification. L'amplitude et la durée d'application de ces vecteurs sont donc déterminées, et liées à la machine sous test. Ce n'est pas une méthode qui est transportable immédiatement sur une autre machine. Il faut donc préalablement déterminer les vecteurs adéquats par des essais, afin que l'excursion en courant soit suffisante pour les mesures, sans que le rotor soit soumis à un couple supérieur au couple d'arrachement.

La première phase est fiable puisque ce sont des vecteurs obtenus directement par une seule combinaison des interrupteurs du pont triphasé. En effet, dans la 1ère phase, chaque paire (ex : 1 et 2) résulte d'une mesure commune (même capteur et chaîne de mesure, même enroulement) où seule la commande des interrupteurs d'un même bras est inversée pour changer le signe du courant. Par contre, la seconde phase est délicate car elle impose une somme des courants de vecteurs, donc nécessitant une homogénéité des mesures des phases (offsets, gain, bruit). De plus, les conditions dans lesquelles cette méthode est testée sont idéales, dans le sens où la machine présente de fortes variations de l'état magnétique malgré des pôles montés en surface. Les mesures sont très peu bruitées et sont de puissance suffisante.

Le relevé en Fig. 68 montre la variation de courant dans une phase du moteur en fonction de l'angle électrique, avec la mesure des deux courants, respectivement obtenus avec l'impulsion négative, puis positive d'un échelon de tension. Le $\Delta I$ représente 15% de la pleine échelle de mesure. Le $\Delta I$ mentionné n'est pas celui du $\Delta I/\Delta t$. On parle bien ici de l'excursion de courant caractérisant la différence qui existe entre les deux pôles. On verra au §II.C la différence avec notre machine, qui présente une allure du $\Delta I$ équivalente, mais de moindre amplitude (non représentée).

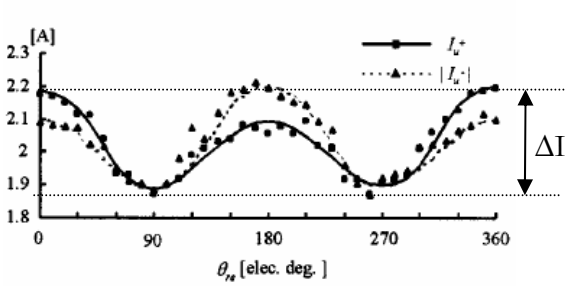



Fig. 68 : Variation de courant d'une phase en fonction de l'angle électrique de vecteurs stimuli, avec l'impulsion positive, puis négative [NAK00]

## *C. Les limitations de la méthode*

Les taux d'erreur ne sont pas abordés, tout comme la nécessité d'avoir une très bonne homogénéité des phases, donc par conséquent un offset triphasé nul (somme des courants). Théoriquement, l'ensemble des tests est réalisé en moins de quelques dizaines de millisecondes.
Cette méthode a été testée pour le radar météo : la première phase se déroule correctement, bien qu'il soit difficile d'obtenir le point de fonctionnement idéal pour les conditions de tests. Il faut un $\Delta I/\Delta t$ suffisant, sans pour autant atteindre la saturation de la machine. Par contre, la seconde phase itérative n'a pas pu être mise en œuvre avec une fiabilité et une reproductibilité suffisantes. Les niveaux des amplitudes de courant mesurés sont trop faibles. De plus, les trois phases ne sont pas équivalentes, ce qui conduit à une somme des courants des phases lors des combinaisons de vecteurs (évoqués en Fig. 66) qui ne sont pas comparables entre l'essai de l'impulsion positive, puis négative. La somme des courants qui est erronée, cumulée à la dispersion due aux erreurs de mesures, donne un résultat dont l'excursion est de même échelle que la différence des amplitudes attendues lors des essais avec les impulsions positives et négatives. On n'est donc pas capable de traiter le résultat de manière fiable. Pour avoir des niveaux de mesure plus élevés, il faut une machine présentant une non linéarité assez accentuée.

Lorsque l'on examine notre machine synchrone de plus près, on peut qualifier et quantifier les caractéristiques limitantes :

- L'ondulation de couple représente 0.2% du couple maximal. Le $\Delta I$ n'excède pas 3% de la valeur nominale du courant pour une commande à pleine échelle, contrairement à la publication qui met en œuvre une machine ayant plus de 15% de variation pour une commande à mi-échelle.
- La dispersion des paramètres intrinsèques : les inductances propres de chaque phase sont à $\pm 30\%$ (effets de saillances compris) et les résistances séries à $\pm 10\%$. L'essai de 2

vecteurs opposés ne pose aucun problème, puisque c'est la même phase qui est testée, mais la somme est donc entachée d'erreurs cumulées par la disparité entre les phases. La résistance des enroulements est faible en ce qui concerne les essais de courant, donc une variation n'a que peu d'incidence sur l'atténuation. On n'en tient pas compte pour la suite.

Quand le test fonctionne, la précision est effectivement présente, mais le taux d'erreur est très important. Aucun moyen à ce moment là n'a pu être mis en œuvre pour améliorer ou garantir un bon fonctionnement. Voyons maintenant l'effet recherché malgré les limitations techniques :

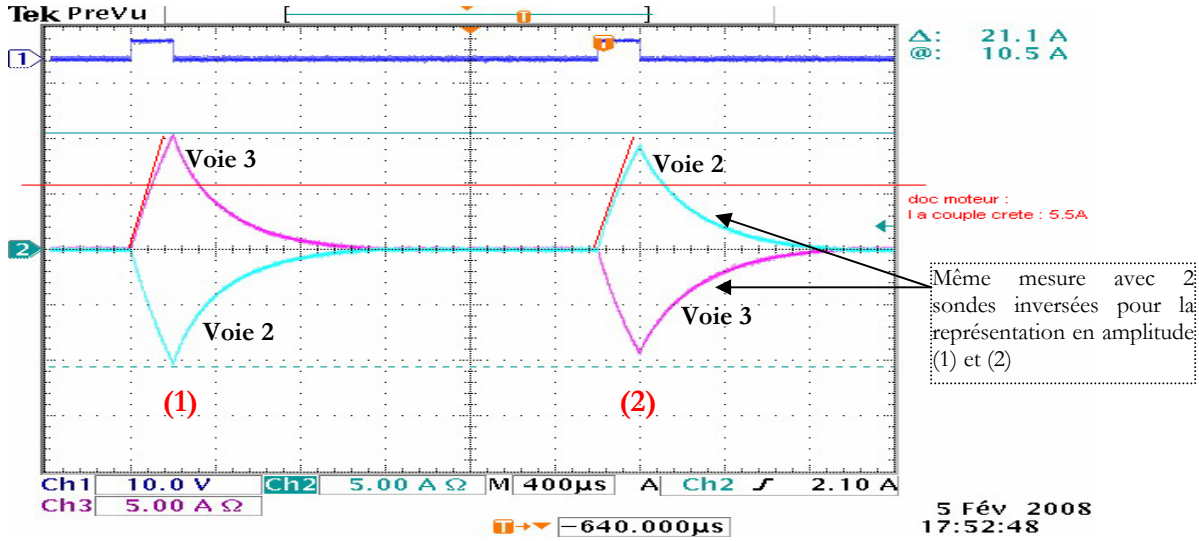



Fig. 69 : Relevé pour déterminer le point de fonctionnement pour les tests sur le moteur du radar météo

Le relevé en Fig. 69 représente la variation de courant observable sur une phase du moteur, les deux autres étant reliées via une configuration de l'onduleur. Cette phase est soumise à deux vecteurs de signes opposés (voir Fig. 70 dont la flèche indique le courant mesuré). Ces deux vecteurs sont issus de deux impulsions (par une combinaison de bras) dont le sens du courant est inversé, comme représenté en Fig. 70 avec les combinaisons (a) et (b) par exemple. L'absence de neutre et la commande complémentaire des six interrupteurs ne donne la possibilité de générer que trois paires d'impulsions (a-b, c-d et e-f) pour appliquer la tension du bus continu aux phases du moteur.

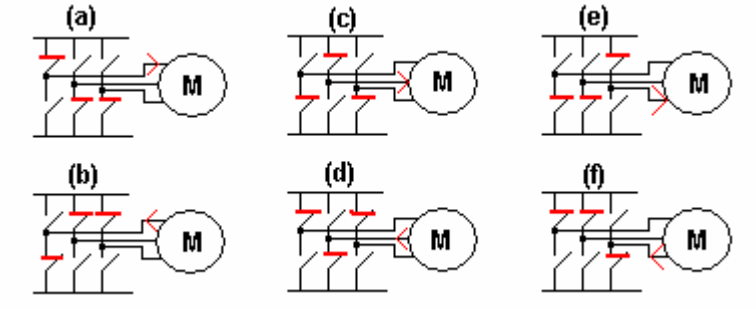



Fig. 70 : Combinaisons des interrupteurs pour générer les impulsions

La voie 1 (de la Fig. 69) sert de référence temporelle (synchronisation de l'oscilloscope) ; les fronts montants et descendants du créneau coïncident avec les instants de début et de fin de l'échantillonnage. Les voies 2 et 3 sont les mesures du même courant de phase (du moteur), où l'on a sur chaque voie une sonde de courant distincte inversée l'une de l'autre : voie 2 = - voie 3. Le but est de comparer avec un même curseur la valeur maximale du signal pour chacun des deux essais en (1) et en (2). Le test a été réalisé de sorte à montrer cette variation à son maximum. Ainsi, le 1$^{er}$ pic positif (1) correspond à un champ soustractif entre le champ d'induit et le champ des aimants (on sature moins vite) alors que l'autre pic positif (2) correspond à un champ additif (qui sature plus vite). Sachant les pôles que l'on crée au stator, on peut repérer les pôles au rotor, donc localiser le pôle nord et le sud du rotor. On voit nettement que dès que l'on dépasse le courant nominal, la saturation apparaît. Pour obtenir une différence mesurable, il faut appliquer de forts courants pour visualiser la variation entre deux vecteurs de signes opposés. Cet aspect est délicat, et il faut converger vers un compromis pour avoir un essai en courant suffisamment élevé pour la mesure, mais sans travailler dans une zone de saturation significative pour cette même mesure (atténuation des essais). On notera qu'il faudra tenir compte d'un effet compensatoire sur la durée des impulsions pour supporter des variations du bus continu si la tension n'est pas constante (gabarit sur la tension d'alimentation).

Le compromis choisi impose aussi la dynamique de la commande (mesures comprises). En régime fonctionnel, l'alimentation du moteur est découpée à 125 kHz, avec un filtrage des mesures adéquat. Pour l'étape d'indexage, il faut opérer à 2.5 kHz, avec des temps intermédiaires entre les mesures pour laisser s'effectuer la démagnétisation. On doit travailler avec deux fréquences bien différentes. La fréquence de coupure est dans l'intervalle 2.5kHz/125kHz. La présence des filtres n'est donc pas adaptée pour l'indexage (filtres qui vont transformer les fronts), mais est nécessaire pour le contrôle moteur. On opte donc pour une stratégie où les filtres sont déconnectables (Fig. 71). Les filtres sont donc déconnectés lors de l'étape d'indexage par l'ouverture de la branche des capacités de la cellule LC. Chaque inductance des cellules reste

présente et se cumule à l'inductance de la phase qui lui est commune. Ici, comme on ne cherche pas à caractériser le moteur, l'inductance série n'a pas d'impact significatif (même avec sa dispersion qui lui est propre) sur l'essai en courant qui a pour but de mettre en avant une variation, et non une valeur absolue.

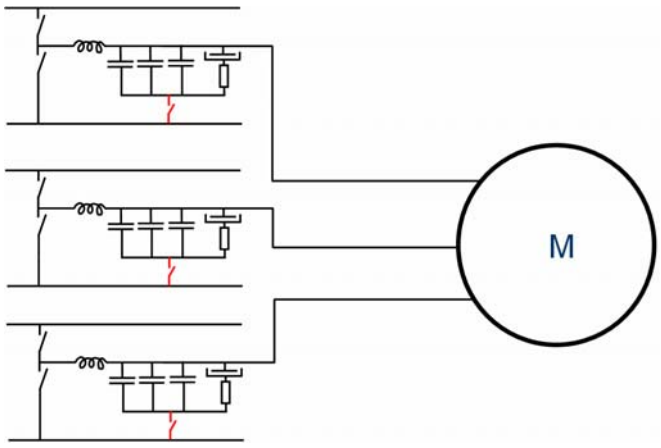



Fig. 71 : branche capacitive déconnectable de la cellule LC,

La méthode exposée dans la publication [NAK00] impose le fait qu'il est indispensable que le rotor soit immobile ; c'est à dire qu'il ne doit pas bouger sous l'effet du test, mais également sous l'effet de l'environnement.

Ce dernier point doit être appréhendé sous deux aspects :

- Le cas de l'initialisation au sol. L'antenne ne subit pas de mouvements provoqués par le porteur, mais peut subir des vibrations avec la mise en marche des systèmes (réacteurs, générateurs auxiliaires, etc.).
- Le cas de l'initialisation en vol. C'est le cas le plus critique car les mouvements porteurs sont tout à fait significatifs, et ont un impact direct sur les stimulations mécaniques de l'antenne (donc du rotor). L'environnement n'est donc a priori pas adapté à cette phase d'initialisation. De plus, il n'est pas possible d'accepter une dégradation des performances (précision de l'indexage) ou une augmentation du temps nécessaire à cette initialisation. Si l'on utilisait sans modification la première partie de la méthode présentée dans la publication [NAK00], le rotor bougerait fortement. Pour un indexage durant 660ms (c'est le temps pour une opération réussie dans notre cas, avec des moyennages, et avec un retour à l'équilibre de l'hystérésis magnétique), l'antenne pourrait se déplacer de bien plus de 30° électriques (bilan à la Fig. 77) et donc les valeurs obtenues seraient incorrectes.

Comme les variations de courant sont assez faibles, il est nécessaire d'utiliser une chaîne d'instrumentation du courant qui permette des mesures très précises. Le radar étant un système à

« bas » coût, il ne nous est pas possible d'avoir cette chaîne de mesure. La dispersion des inductances des phases est aussi un point contraignant qui remet en cause la somme des courants.

Pour parvenir à un indexage initial dans l'intervalle des ±5° électriques, on va procéder dans un premier temps à une localisation vectorielle réduite des pôles (avec les seules 6 combinaisons usuelles données en Fig. 66) que l'on appellera la phase statique, (en prenant en compte les points bloquants développés en II.C), puis dans un second temps en affinant l'angle par un positionnement électromécanique, que l'on nommera la phase électromécanique.

## D. La phase statique

Tout d'abord, le principe de la phase 1 [NAK00] est exploitable dans notre cas, hormis pour l'utilisation des 12 vecteurs que l'on peut physiquement obtenir à l'aide des combinaisons des interrupteurs. Nous sommes limités à 6 vecteurs, résultat de la combinaison des 3 phases (d'où localisation vectorielle réduite). En effet, il nous est impossible de laisser une phase en l'air du fait de l'utilisation de drivers ayant une complémentarité intrinsèque des commandes des transistors de puissance sur un même bras. En nous limitant à 6 vecteurs, on obtient un intervalle dans lequel le vecteur recherché est compris à ±30° maximum par rapport au vecteur trouvé après la 1ère phase de test. Cette précision de ±30° électriques donne un couple maximal disponible (pour la commande moteur) de 85% au minimum comme montré en Fig. 72. Ainsi, à partir de la fin de cette phase, on est capable de piloter le moteur dans le bon sens, avec au moins 85% du couple maximal disponible. On rappelle que 85% n'est pas suffisant pour l'application de contrôle du radar puisque les moteurs sont dimensionnés au plus juste, et que 15% d'incertitude sur le couple a une influence non négligeable sur le déterminisme des asservissements, mais 85% de couple disponible est suffisant pour contrôler le moteur dans une phase d'initialisation électromécanique.

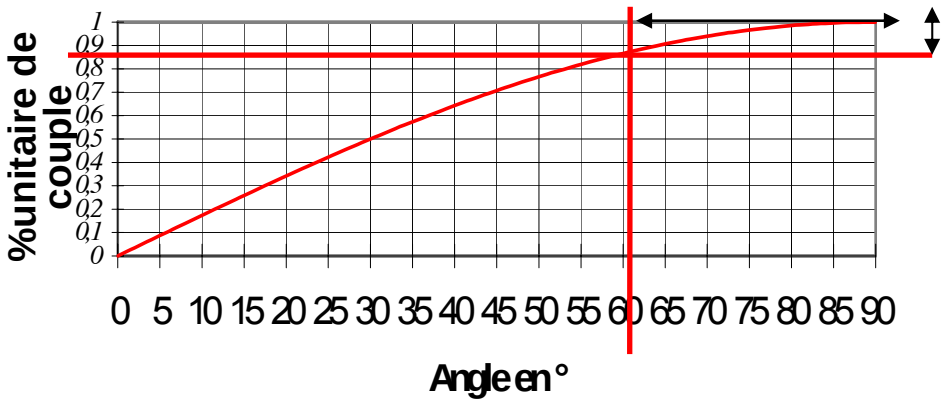



Fig. 72 : Couple en fonction de l'angle d'autopilotage

Pour exécuter la phase 1 en entier avec les conditions citées ci-dessous, il faut 660 ms. Ce temps est plus long que celui donné dans la publication [NAK00], puisque pour fiabiliser les mesures, on procède à des séries de moyennages. Un algorithme traite les résultats de deux manières différentes afin d'extraire le vecteur le plus proche du pôle nord. Ce temps est volontairement long car les mesures ne bénéficient pas d'un rapport signal/bruit favorable, ce qui est pénalisant. La fréquence de l'essai à 2.5 kHz, évoquée au §II.C, est lié à notre moteur. La durée écoulée s'explique par les termes suivants :

-$T_{mesure}$ = 200µs *(noté T1) pour l'essai à 2.5 kHz (2x200µs avec la montée/descente)*

-$T_{démagnétisation}$ = 1000µs *(noté T2), marge sans démagnétisation forcée*

-$T_{démagnétisation\ sécurité}$ = 1000µs *(noté T3), marge confortable pour ne pas tenir compte des dispersions*

-Nombre de moyennages par phase = 100, *pour fiabiliser les mesures*

-Nombre de phases à tester = 3, *c'est triphasé*

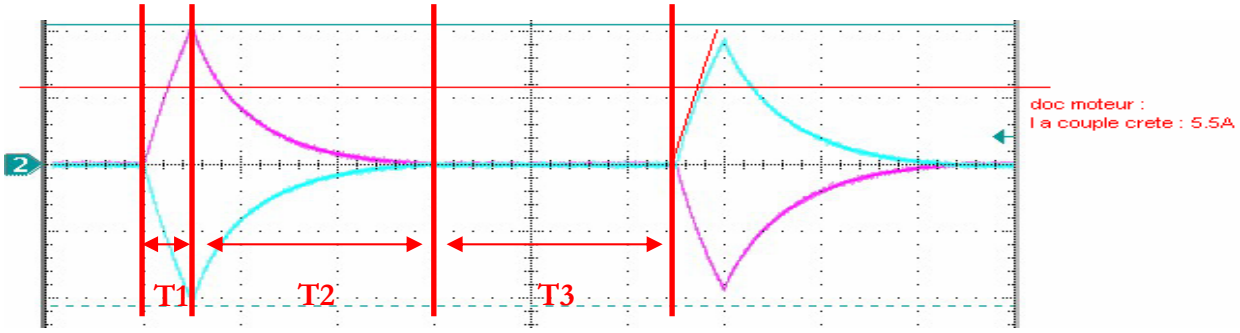



Fig. 73 : Situation des temps relatifs à l'indexage

On en déduit : $T_{indexage} = (200 + 1000 + 1000) \times 100 \times 3 = 660ms$

Le temps de mesure (T1) est imposé notamment par les caractéristiques des inductances du moteur (car il ne faut pas oublier les inductances du filtre déconnecté), par le niveau de tension du bus DC et par les courants maximaux admissibles par les amplificateurs, qui définissent la rampe de courant. Le temps de démagnétisation est obtenu pour une démagnétisation naturelle, en reliant les phases entre elles (activation de tous les interrupteurs du côté de la masse). Ce temps peut être réduit par une démagnétisation forcée, à 250µs (200µs de temps de descente forcée + 50µs de temps de démagnétisation naturelle en cas de dissymétrie des fronts). Le temps de sécurité est un temps corrélé à la démagnétisation naturelle, afin de garder une marge pour l'effet de la dérive en température des paramètres de la machine et des filtres, ainsi que l'effet de la dispersion des paramètres des moteurs (L à ±30%) ; mais il est vrai que 1000µs est une marge confortable. Le relevé en Fig. 73 montre un résultat pour une condition de test donnée à température ambiante.

Le nombre d'échantillons du moyennage pourrait être réduit avec une meilleure qualité des mesures. Ce sont donc là les degrés de liberté disponibles pour réduire le temps de l'indexage statique. Durant cette phase, on peut être confronté à une perturbation extérieure (cas du redémarrage en vol). Il faut donc vérifier que la durée de l'indexage statique ne soit pas trop grande pour que le déplacement parasite ne soit pas pénalisant. Le document de calcul (voir Fig. 76) propose différents paramètres à modifier (avec les contraintes exposées ci-dessous). Le but est de voir leurs influences respectives sur l'angle balayé durant l'indexage statique, ou alors de voir lesquels vont permettre de réduire l'angle balayé. Si l'angle balayé est trop grand, le pré-indexage (indexage statique) échoue puisque les valeurs mesurées n'ont plus de sens.

L'angle balayé n'est contraignant que dans le cas d'un redémarrage en vol, puisque l'antenne est « excitée » par les mouvements du porteur. Les accélérations « G » sont nos principales sources d'ennuis, puisqu'il faut pouvoir assurer un régime fonctionnel de 3G en latéral, de 6.5G en vertical montant, et de 4.5G en vertical descendant. Ceux-ci créent une rotation de l'antenne par le bras de levier entre le centre de gravité et l'axe de pivot, résultant de la dissymétrie de l'antenne.

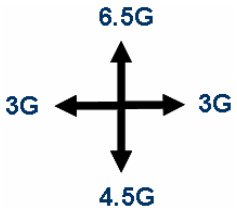



Fig. 74 : Accélérations maximales dues au porteur

Le tangage est corrélé avec l'élévation, et le cap est corrélé avec l'axe circulaire, donc on peut faire l'hypothèse que l'effet du tangage et du cap est compris dans les accélérations latérales et verticales. Le roulis est différent : sa présence crée une rotation des axes moteurs par rapport aux axes du repère terrestre, et par exemple une accélération verticale va avoir un effet sur l'axe circulaire. (Pour vérifier, on fait un test où un roulis de 90° donne une accélération latérale de 6.5G et verticale de 3G en travaillant avec les projections des « G » sur chaque axe).

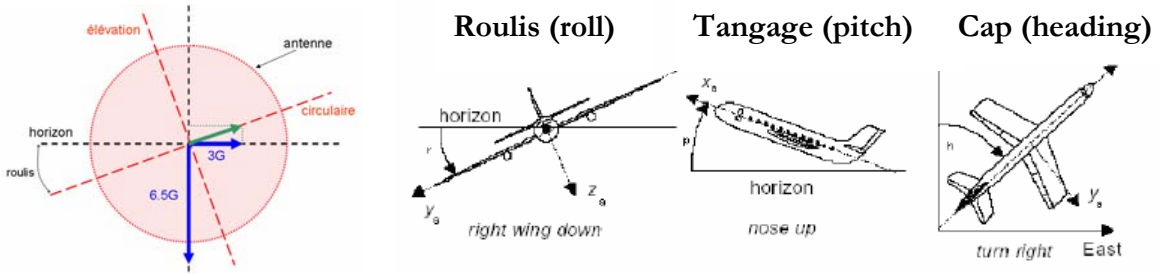
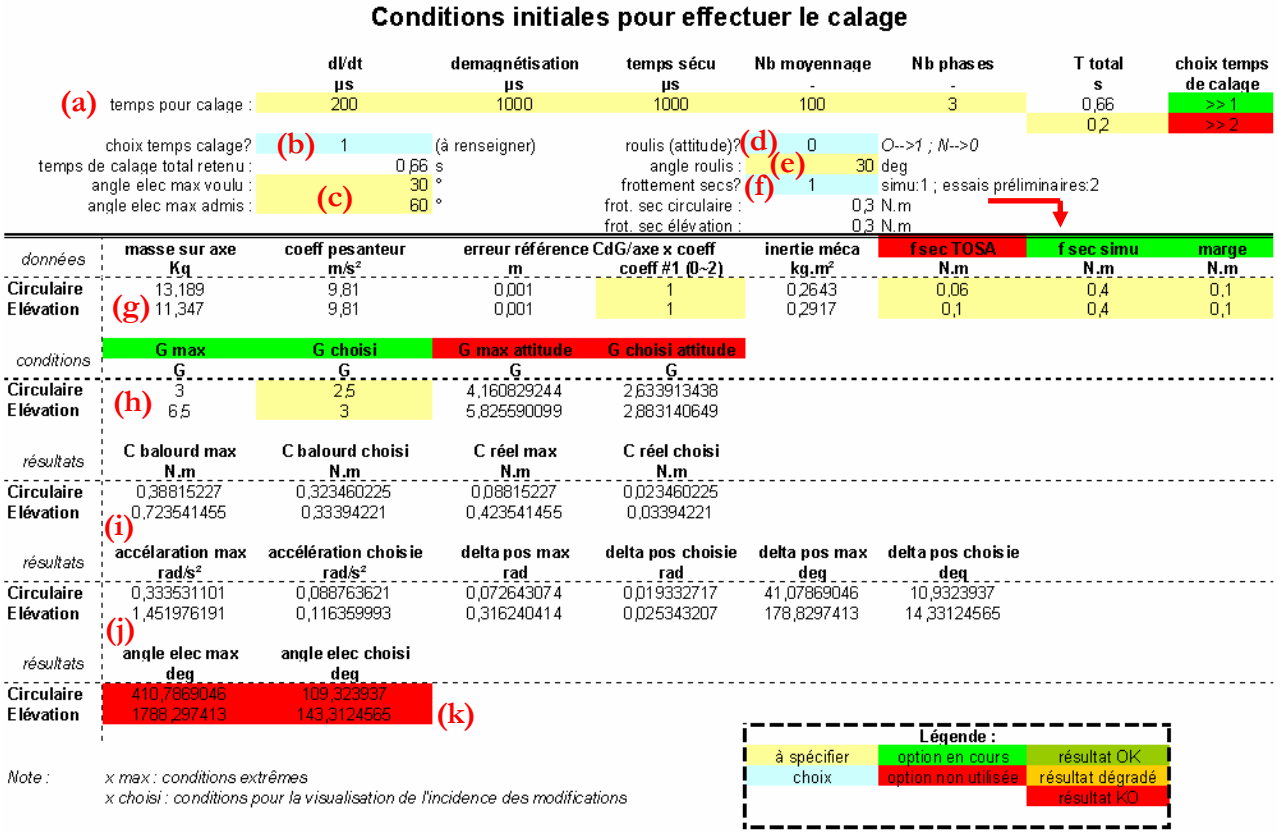



Fig. 75 : Roulis antenne et attitudes avion, roulis – tangage – cap

Le schéma en Fig. 75 montre que le roulis crée une composante verticale et latérale sur chaque axe. On peut avoir deux cas de figures, soit le roulis amplifie ou soit il diminue l'effet des accélérations sur chaque axe. La synthèse électromécanique (voir Fig. 76) permet de visualiser l'impact des différents paramètres sur l'angle électrique balayé durant le temps d'indexage. Les déplacements « parasites » créés par les mouvements du porteur sont pour nous issus de couples de balourd. Ces couples sont le résultat de la masse portée par chaque axe, dont le bras de levier n'est pas nul (marges de conception). En effet, la distance entre le centre de gravité et le centre de l'axe est faible, mais suffisant pour que le couple créé puisse avoir un impact significatif sur le déplacement angulaire de l'antenne. La distance maximale de 1mm est celle retenue pour les simulations. C'est a priori une valeur qui restera maximale, résultant aussi de l'incertitude de conception et de réalisation. On verra son influence au §II.E. Chaque axe porte une masse différente et l'axe circulaire porte l'axe élévation. Comme l'axe élévation comporte un décentrage, il devrait avoir une influence sur l'axe circulaire, du moins lorsque le piédestal est penché. Si on s'intéresse aux matrices qui décrivent la mécanique, on s'aperçoit que ce désaxage crée des termes diagonaux. Cependant, ces termes sont très faibles, et on peut les négliger. Cela veut dire que les axes sont dé-corrélés, donc du roulis ne va pas changer le poids de l'axe circulaire (donc l'inertie) en fonction des angles d'inclinaison du porteur.

## *E. Résultats de dimensionnement de la procédure d'indexage de la phase statique*

Voyons l'exemple en Fig. 76, avec les valeurs du cas exposé en phase statique ; le but est de visualiser les excursions maximales de positions compte tenu des critères (de fonctionnement) que l'on applique. Une variation de 3° mécanique est faible, mais c'est aussi une variation de 30° électriques qui devient pénalisante pour le processus d'indexage. On donne le détail des résultats pour les conditions maximales de chaque contrainte, et le détail des résultats avec des conditions différentes, afin de visualiser les incidences de ces différentes variables, pour détecter et comparer les plus significatives. Les essais préliminaires sont issus d'un ensemble de résultats de la caractérisation de l'antenne prototype sur pot vibrant en chambre d'essais. De ces différents essais, notamment avec différentes températures, on a gardé les valeurs les plus défavorables.



**Conditions initiales pour effectuer le calage**

| | dI/dt µs | demagnétisation µs | temps sécu µs | Nb moyennage - | Nb phases - | T total s | choix temps de calage |
|---|---|---|---|---|---|---|---|
| (a) temps pour calage : | 200 | 1000 | 1000 | 100 | 3 | 0,66 | >> 1 |
| | | | | | | 0,2 | >> 2 |

choix temps calage? (b) 1 (à renseigner) — roulis (attitude)? (d) 0 *O-->1 ; N-->0*
temps de calage total retenu : 0,66 s — angle roulis : (e) 30 deg
angle elec max voulu : (c) 30 ° — frottement secs? (f) 1 simu:1 ; essais préliminaires:2
angle elec max admis : 60 ° — frot. sec circulaire : 0,3 N.m
frot. sec élévation : 0,3 N.m

| *données* | masse sur axe Kg | coeff pesanteur m/s² | erreur référence m | CdG/axe x coeff coeff #1 (0~2) | inertie méca kg.m² | f sec TOSA N.m | f sec simu N.m | marge N.m |
|---|---|---|---|---|---|---|---|---|
| Circulaire | 13,189 | 9,81 | 0,001 | 1 | 0,2643 | 0,06 | 0,4 | 0,1 |
| Elévation (g) | 11,347 | 9,81 | 0,001 | 1 | 0,2917 | 0,1 | 0,4 | 0,1 |

| *conditions* | G max G | G choisi G | G max attitude G | G choisi attitude G |
|---|---|---|---|---|
| Circulaire | 3 | 2,5 | 4,160829244 | 2,633913438 |
| Elévation (h) | 6,5 | 3 | 5,825590099 | 2,883140649 |

| *résultats* | C balourd max N.m | C balourd choisi N.m | C réel max N.m | C réel choisi N.m |
|---|---|---|---|---|
| Circulaire | 0,38815227 | 0,323460225 | 0,08815227 | 0,023460225 |
| Elévation (i) | 0,723541455 | 0,33394221 | 0,423541455 | 0,03394221 |

| *résultats* | accélaration max rad/s² | accélération choisie rad/s² | delta pos max rad | delta pos choisie rad | delta pos max deg | delta pos choisie deg |
|---|---|---|---|---|---|---|
| Circulaire | 0,333531101 | 0,088763621 | 0,072643074 | 0,019332717 | 41,07869046 | 10,9323937 |
| Elévation (j) | 1,451976191 | 0,116359993 | 0,316240414 | 0,025343207 | 178,8297413 | 14,33124565 |

| *résultats* | angle elec max deg | angle elec choisi deg |
|---|---|---|
| Circulaire | 410,7869046 | 109,323937 |
| Elévation | 1788,297413 | 143,3124565 (k) |

*Note :* *x max : conditions extrêmes*
*x choisi : conditions pour la visualisation de l'incidence des modifications*

Légende :
à spécifier | choix | option en cours | option non utilisée | résultat OK | résultat dégradé | résultat KO

Fig. 76 : Feuille de calcul de l'angle électrique balayé, dans le cas de paramètres non optimisés

Utilisation du document de calcul :

- (a) : on renseigne les temps utilisés dans notre algorithme, pour les essais de courants ;
- (b) : on choisit entre la somme des temps spécifiés (>>1) ou un temps arbitraire (>>2) en (a), pour pouvoir faire des comparaisons sur les résultats intermédiaires et finaux ;
- (c) : on spécifie notre intervalle de précision souhaité ;
- (d) : on active ou non le roulis ;
- (e) : si oui, son angle (de roulis) ;
- (f) : on choisit les frottements secs, issus de la simulation, ou ceux mesurés (suivre la flèche). On peut mettre une marge sur ces frottements ;
- (g) : on indique les paramètres mécaniques ;
- (h) : on indique nos contraintes d'accélérations ;
- (i) : on obtient le couple de balourd (avec et sans attitudes);
- (j) : on obtient les accélérations engendrées, et les variations de positions pendant le temps donné, en angles mécaniques ;
- (k) : on donne le résultat de la variation de position sur chaque axe en degrés électriques.

Avec un temps d'indexage assez long (660ms), sans roulis, avec des conditions maximales d'accélérations (3G et 6.5G), on obtient un balayage de 410° électriques sur l'axe circulaire et 1800° sur l'axe élévation. Le pré-indexage n'a donc aucune chance de fonctionner. Même en dégradant les accélérations maximales à 2.5G et 3G, on n'obtient pas mieux qu'un balayage de 110° et 144°. Il est donc nécessaire, soit d'améliorer le pré-indexage, soit de définir un cahier des charges différent entre le mode fonctionnel de balayage et d'indexage du moteur. Le bilan (Fig. 77) récapitule les différentes possibilités obtenues avec les degrés de libertés dont on disposait.

Ces valeurs (variations de positions) sont obtenues pour des accélérations maximales et constantes. Une variation de 1800° électriques, c'est un demi-tour mécanique. On rappelle que durant la phase d'indexage, les moteurs ne sont pas alimentés, donc aucun asservissement ne stabilise l'antenne.

On notera que les frottements sont une donnée pénalisante. En effet, il faut les minimiser pour les asservissements, mais s'ils sont trop faibles, le déplacement dû à une perturbation sera plus important durant la phase d'indexage, c'est un compromis à fixer. C'est pour cela que les frottements les plus faibles sont ceux du pire cas (frottements mesurés sur le prototype non terminé, sans les couples de rappels de la connectique, et avec de faibles pré-charges des roulements).

| conditions / types d'essais | | temps de calage ms | G latéral m/s² | G vertical m/s² | roulis deg | erreur CdG/axe mm | F secs N.m | angle balayé C deg (élec) | angle balayé E deg (élec) | OK/KO ? |
|---|---|---|---|---|---|---|---|---|---|---|
| lent, G max | | 660 | 3 | 6,5 | 0 | 1 | 0,3 | 410 | 1800 | KO |
| lent, G min | (a) | 660 | 2,5 | 3 | 0 | 1 | 0,3 | 110 | 144 | KO |
| moins de balourd, G max | | 660 | 3 | 6,5 | 0 | 0,5 | 0,3 | 0 | 260 | KO |
| moins de balourd, G min | | 660 | 2,5 | 3 | 0 | 0,5 | 0,3 | 0 | 0 | KO |
| moins de balourd, G max | (b) | 660 | 3 | 6,5 | 30 | 0,5 | 0,3 | 0 | 103 | KO |
| moins de balourd, G min | | 660 | 2,5 | 3 | 30 | 0,5 | 0,3 | 0 | 0 | KO |
| démagnétisation plus rapide | | 150 | 3 | 6,5 | 0 | 1 | 0,3 | 22 | 93 | KO |
| démagnétisation plus rapide | (c) | 150 | 2,5 | 3 | 0 | 1 | 0,3 | 6 | 8 | KO |
| avec roulis | | 150 | 3 | 6,5 | 30 | 1 | 0,3 | 58 | 76 | KO |
| avec roulis | (d) | 150 | 2,5 | 3 | 30 | 1 | 0,3 | 10 | 5 | KO |
| balourd réduit | | 150 | 3 | 6,5 | 30 | 0,5 | 0,3 | 0 | 0 | KO |
| balourd réduit | | 150 | 2,5 | 3 | 30 | 0,5 | 0,3 | 0 | 6 | KO |
| moyennage meilleur | (e) | 75 | 3 | 6,5 | 0 | 1 | 0,3 | 6 | 24 | KO |
| moyennage meilleur | | 75 | 2,5 | 3 | 0 | 1 | 0,3 | 2 | 2 | KO |
| avec roulis | | 75 | 3 | 6,5 | 30 | 1 | 0,3 | 15 | 19 | OK |
| avec roulis | (f) | 75 | 2,5 | 3 | 30 | 1 | 0,3 | 3 | 2 | KO |
| faible balourd | | 75 | 3 | 6,5 | 30 | 0,5 | 0,3 | 0 | 0 | KO |
| faible balourd | | 75 | 2,5 | 3 | 30 | 0,5 | 0,3 | 2 | 0 | KO |
| avec faibles F secs | (g) | 75 | 3 | 6,5 | 30 | 0,5 | 0,06 / 0,1 | 13 | 13 | KO |
| avec faibles F secs | | 75 | 2,5 | 3 | 30 | 0,5 | 0,06 / 0,1 | 7 | 4 | KO |
| moyennage plus long | | 150 | 3 | 6,5 | 30 | 0,5 | 0,06 / 0,1 | 51 | 49 | KO |
| moyennage plus long | | 150 | 2,5 | 3 | 30 | 0,5 | 0,06 / 0,1 | 27 | 14 | KO |
| avec faibles F secs | (h) | 75 | 3 | 6,5 | 30 | 1 | 0,06 / 0,1 | 29 | 30 | OK |
| avec faibles F secs | | 75 | 2,5 | 3 | 30 | 1 | 0,06 / 0,1 | 17 | 13 | KO |
| moyennage plus long | | 150 | 3 | 6,5 | 30 | 1 | 0,06 / 0,1 | 116 | 120 | KO |
| moyennage plus long | | 150 | 2,5 | 3 | 30 | 1 | 0,06 / 0,1 | 66 | 49 | KO |
| balourd plus important | | 75 | 3 | 6,5 | 30 | 1,5 | 0,06 / 0,1 | 31 | 37 | KO |
| balourd tres important | | 75 | 6 | 6,5 | 30 | 2 | 0,06 / 0,1 | 47 | 55 | KO |
| | | | | | | | | | | KO |
| | | | | | | | | | | KO |

**Conditions opérationnelles limitantes actuelles:**

| | | |
|---|---|---|
| Max angle balayé admis : | 30 | deg (elec) |
| roulis opérationnel : | 30 | deg |
| accélération latérale opérationnelle : | 3 | G |
| accélération verticale opérationnelle : | 6,5 | G |
| erreur CdG / axe réalisable : | 1 | mm |

Note : c'est OK si TOUTES les conditions limitantes sont satisfaites. Ce tableau ne fait pas de calculs, il se contente de faire apparaître les résultats saisis de la feuille précédente, selon des critères limitants. Les types d'essais explicités peuvent être relatifs à des essais de la ligne précédente! attention à la lecture.

Fig. 77 : Bilan des essais des angles balayés en fonction des paramètres limitants

Cette synthèse (Fig. 77) montre différents résultats en démarrant (a) par le cas où le temps d'indexage est long. C'est un temps où l'on prend des marges sur le fonctionnement, et qui est nécessaire pour fiabiliser les mesures. Puis, pour réduire l'angle balayé, on réduit l'impact de chaque contrainte (indiqué par le type d'essais) pour évaluer son incidence afin d'identifier les contraintes les plus pénalisantes, et ceux pour éventuellement soulever des points trop restrictifs du cahier des charges fixé en interne en avance de phase sur les demandes du client.

- (a) : on commence par le cas simple, cela ne fonctionne pas ;
- (b) : on réduit le balourd, on voit qu'il est significatif ;
- (c) : on remet le balourd, mais on diminue le temps des tests. Cela commence à s'améliorer ;
- (d) : on met du roulis. Cela a un impact significatif pénalisant;
- (e) : on améliore le moyennage sans roulis. L'amélioration est visible ;
- (f) : on remet du roulis, cela fonctionne avec des frottements secs forts ;
- (g) : on recommence avec moins de frottements ;
- (h) : on trouve une autre solution avec les pires contraintes.

Ainsi, pour retrouver les contraintes initialement imposées (roulis, balourd, frottements secs, accélérations), on avance pas à pas en rétablissant les contraintes une à une et en diminuant le temps d'indexage avec les degrés de liberté que nous avons (temps de démagnétisation, temps de sécurité après la démagnétisation, nombre de moyennages). Pour garder la fiabilité sur les mesures, il est alors nécessaire d'améliorer la chaîne d'acquisition, fait démontré avec cette démarche. Il ne reste alors que peu de solutions si l'on garde tous les paramètres au maximum spécifié dans le cahier des charges. En l'occurrence, il nous reste deux solutions viables, l'une avec les frottements secs retenus pour la simulation, l'autre avec les frottements secs de l'expertise (qui semblent trop faibles sur le prototype par rapport à la version série). Ainsi, on peut fonctionner dans le pire cas si l'on parvient à effectuer un pré-indexage très rapide. Pour atteindre 75ms, il faut une démagnétisation forcée (avec un temps de sécurité amoindri), et une amélioration des mesures pour utiliser un minimum de points dans le moyennage. Le tableau de synthèse ne fait pas apparaître les essais sans roulis à accélérations maximales, qui diminuent l'effet sur l'axe circulaire, mais qui maximisent l'effet sur l'axe élévation. Dans le cas avec des frottements secs très faibles, on est à la limite que l'on s'est fixée.

D'autres solutions peuvent apparaîtrent si l'on remet en cause les conditions maximales de fonctionnement, ou si l'on parvient à réduire le couple de balourd. Cependant, cela paraît

difficile d'obtenir une erreur <1mm entre le centre de gravité d'un axe, et l'axe réel, du moins, de façon économique.

## F. La phase électromécanique

Une fois que le pré-indexage est terminé (indexage statique), le moteur est indexé avec une erreur maximale de ±30° électriques. En effet, l'angle de déplacement de la dernière rotation est de 10° électriques dans le pire cas dans le test aux limites de la solution rapide. A chaque essai (de ±180° électriques d'une paire de vecteurs), l'angle de rotation parasite est pris en compte. Si c'est une vibration, alors l'impact est imperceptible, et si c'est une rotation qui se cumule dans le même sens, alors à chaque essai, cet angle est pris en compte dans la localisation intermédiaire. On dispose alors d'un couple minimal disponible de 85% (du couple maximal atteignable) pour le contrôle moteur, pour exploiter l'idée d'un balayage de finition dans le but de créer une rotation mécanique des moteurs. Le couple est suffisamment élevé en amplitude pour donner à l'antenne un mouvement de sens défini grâce à l'indexage statique précédent. On va créer un déplacement du rotor avec le contrôle moteur qui vient d'être référencé à ±30° électriques par l'indexage statique. Les conditions fonctionnelles imposent un couple élevé pour tenir les performances des asservissements, or, pendant la phase d'indexage électromécanique, 85% du couple nous suffit pour contrer les perturbations dues aux mouvements du porteur afin de terminer l'indexage en deux étapes par un balayage de finition (post-positionnement électromécanique). En effet, l'angle mécanique restant à balayer, pour se fixer sur la position de référence, est très faible (<3° mécanique). Comme on connaît le vecteur d'équilibre déterminé par l'indexage statique, la position actuelle, le couple disponible minimal et le sens de rotation, on peut se positionner en boucle ouverte sur la position de « 0 de couple » (voir Fig. 78) pour réduire l'incertitude de l'angle d'indexage. Une fois stabilisé, il suffit de relever la position, et d'en déduire la référence de l'indexage (<5° électriques).

Le relevé (Fig. 78) montre le courant d'une phase du moteur. On démarre dès la mise sous tension. On a successivement :

- (a) : mise sous tension, la commande numérique démarre ;
- (b) : auto-tests du système électronique, défauts sur les composants ;
- (c) : connexion des interfaces (commandes, communications, amplificateurs) ;
- (d) : impulsions de courants de l'étape d'indexage statique ;

- (e) : contrôle moteur progressif, déplacement du moteur ;
- (f) : contrôle moteur en positionnement fixe ;
- (g) : séquence de c à f pour l'autre moteur, courant non instrumenté sur ce relevé.
- (h) : début du contrôle moteur en mode balayage

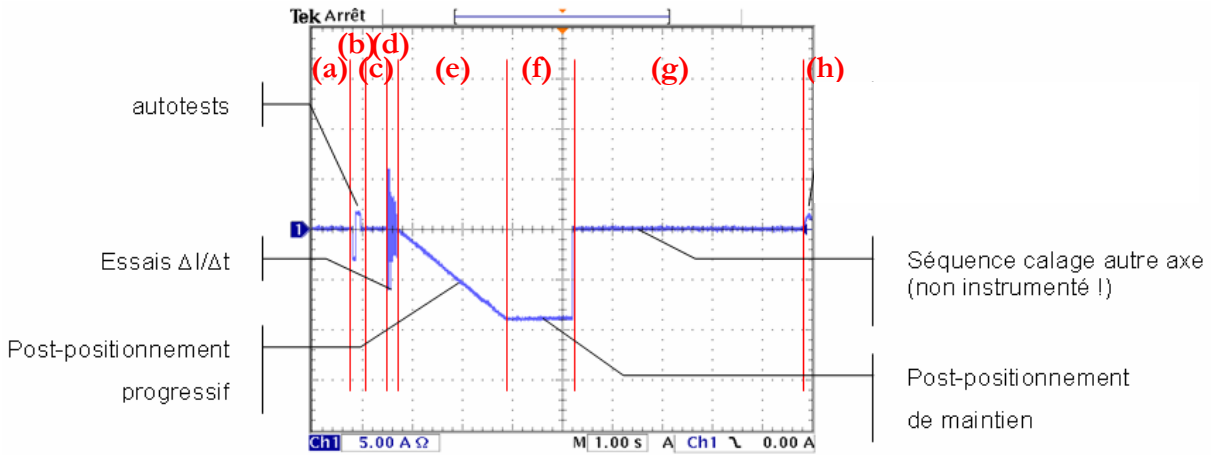



Fig. 78 : Relevé du courant d'une phase du moteur circulaire, visualisation des séquences de fonctionnement, notamment les phases statiques et électromécaniques de l'indexage automatique

Afin de garantir un temps fixe de post-positionnement, on va moyenner les éventuelles oscillations mécaniques autour de la position finale (on ignore les positions transitoires). Un « time-out » achève l'indexage dans la phase de post-positionnement de maintien. Des protections logicielles sont implémentées pour garantir le succès de l'opération et vérifier que l'offset de la position soit dans l'intervalle maximal de 30° électriques par rapport au vecteur d'équilibre. Dans la stratégie « système », on pourrait vérifier les grandeurs en les comparant avec les valeurs déterminées au sol et mémorisées dans un calculateur déporté.

## *G. L'organigramme fonctionnel*

L'ensemble des commandes est géré par un DSP et son usage en aéronautique dans le domaine des alimentations est novateur. On ne s'intéresse ici qu'à la partie liée à l'indexage. Des travaux existent sur des calculateurs [BEN03], mais pas sur un DSP qui représente à lui seul le système embarqué (SETR). La démarche d'une forte intégration est dans la continuité des travaux présentés dans le second chapitre. En effet, les contraintes du code s'ajoutent aux règles des algorithmes, mais on n'en parlera pas ici.

L'indexage automatique au démarrage est réalisé en interruption pour l'identification du pôle Nord par $\Delta I/\Delta t$, et sans interruption pour le post-positionnement, et cela pour les 2 axes. Un séquenceur pour l'indexage permet de passer d'une fonction à une autre, car il faut reconfigurer des timers et des variables à chaque étape. L'organigramme en Fig. 79 est une machine à états finis qui permet de réaliser cette fonction. Seule l'interruption d'indexage est utilisée, les autres étant inhibées. Cette partie du programme est réalisée après l'initialisation du système et des autotests.

## 1. Machine à états finis de l'indexage

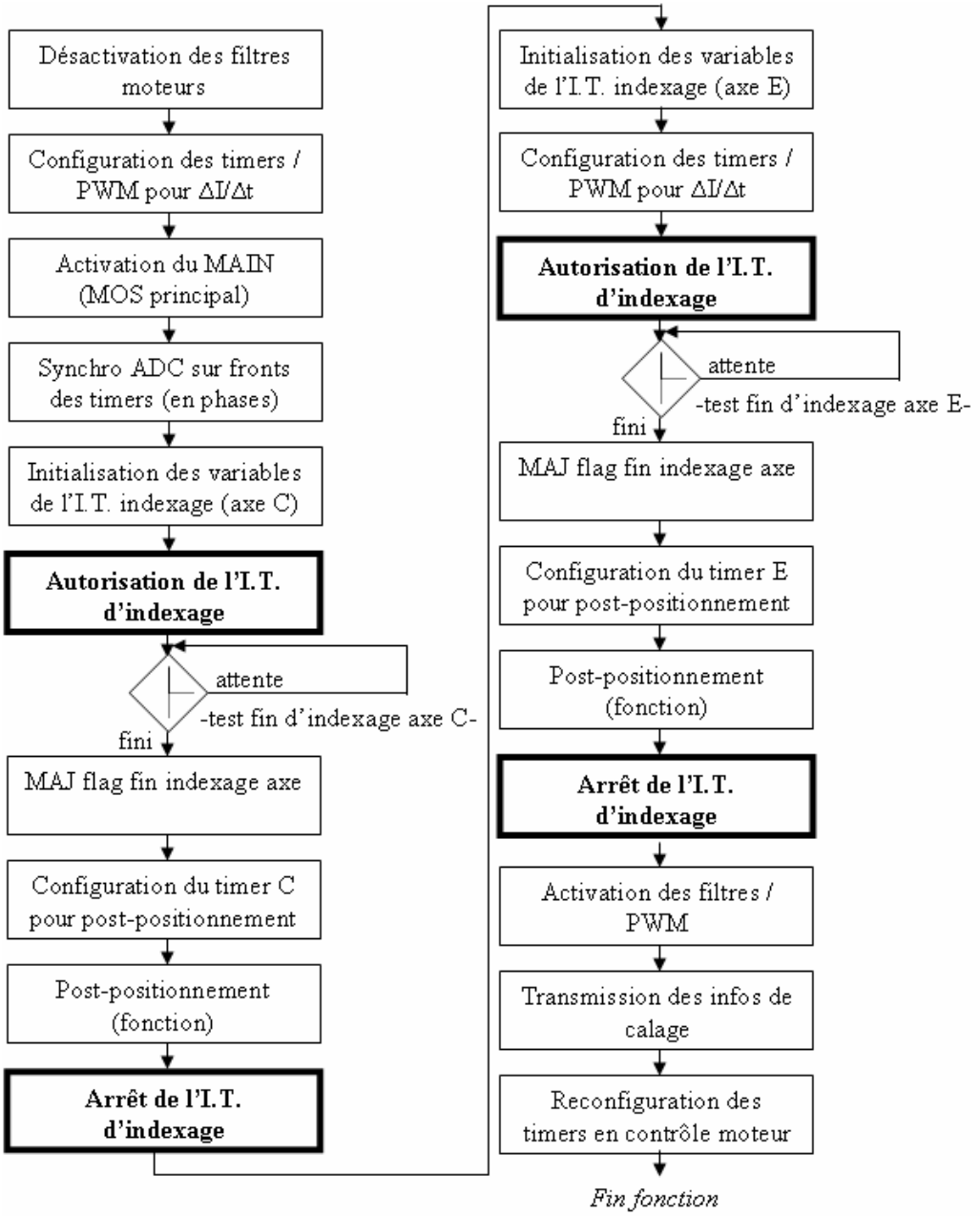



Fig. 79 : Fonction de séquencement de l'indexage automatique

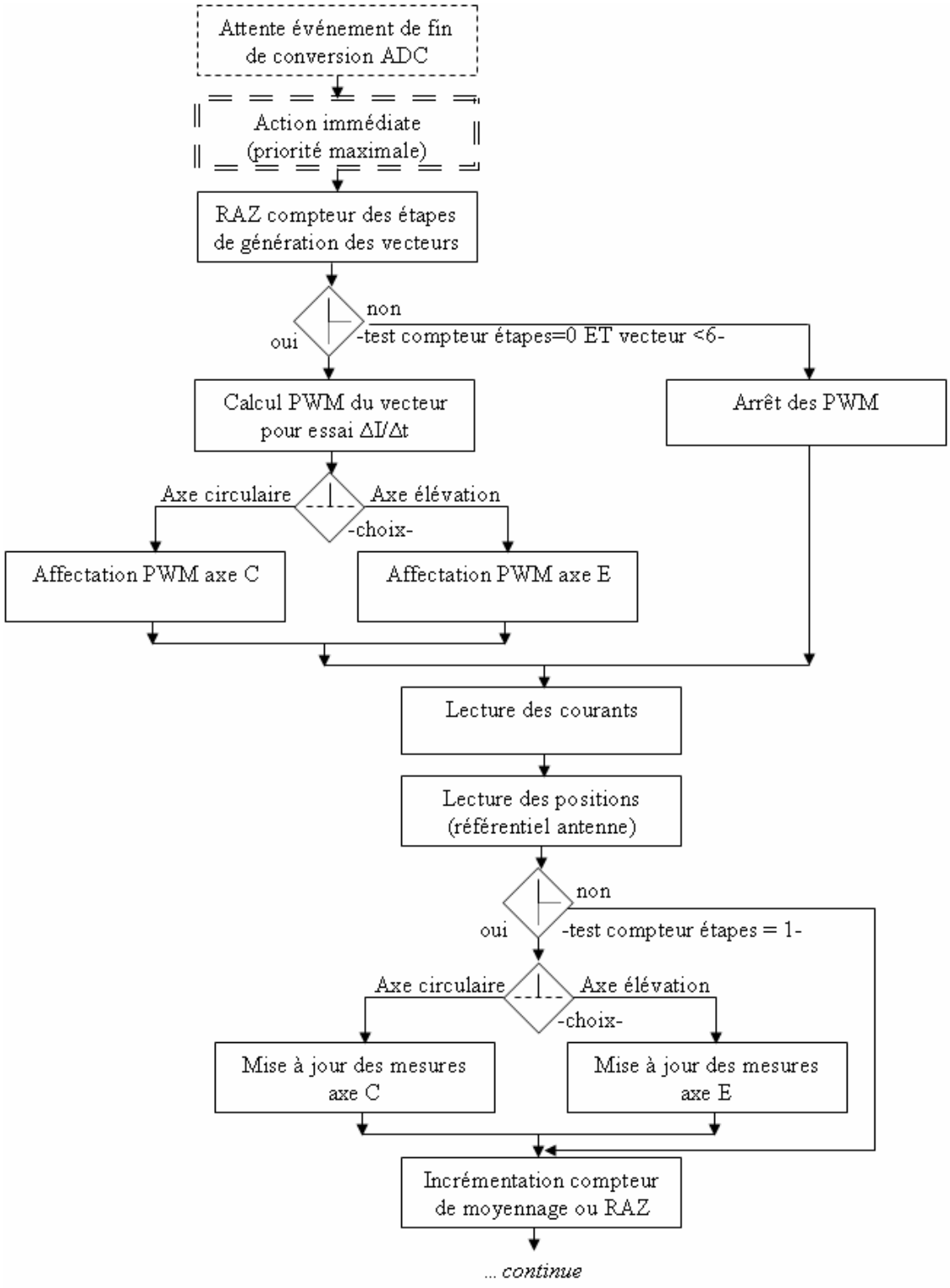
Explication des blocs :

- **Désactivation des filtres moteurs :** dès le démarrage de l'indexage, on inhibe les filtres afin de pouvoir réaliser les essais de $\Delta I/\Delta t$ ;
- **Configuration des timers/PWM pour $\Delta I/\Delta t$ :** les essais de $\Delta I/\Delta t$ sont réalisés à 2.5 kHz, (<<125 kHz qui est la fréquence de découpage des onduleurs), les timers sont synchronisés entre eux, et l'ADC est synchronisé sur le timer de la PWM maître ;
- **Activation du MAIN (MOS principal) :** on remet la puissance sur les onduleurs (connexion au bus DC coupée à la fin des autotests) pour démarrer l'indexage ;
- **Synchro ADC sur fronts des timers (en phases) :** l'ADC et les PWM fonctionnant sur des timers distincts puisque les fréquences ne sont pas identiques en régime fonctionnel (10kHz et 125kHz), il faut qu'ils fonctionnent en synchronisme pour que la mesure de courant se fasse au moment de commuter le signal PWM en mode de démagnétisation ;
- **Initialisation des variables de l'I.T. indexage (axe X) :** cela permet de démarrer l'interruption des essais en $\Delta I/\Delta t$ avec les variables initialisées pour l'axe X (flag, compteurs) ;
- **Autorisation de l'I.T. d'indexage :** le vecteur d'interruption est validé ;
- **Aiguillage de fin d'indexage axe X :** un flag commuté par l'I.T. permet à la machine à états finis de continuer l'indexage après la fin des essais en $\Delta I/\Delta t$ ;
- **MAJ flag fin indexage axe :** L'I.T. est en attente pour le post-positionnement (on ne garde que la lecture de position). Cela permettra aussi (pour les évolutions de finalisation) de faire un accès cyclique à l'EPLD ;
- **Configuration du timer X pour post-positionnement :** on remet les PWM à 125 kHz, pour ne plus entendre le 2.5 kHz et pour limiter le stress des composants de puissance ;
- **Post-positionnement :** cette fonction est détaillée plus loin ;
- **Arrêt de l'I.T. d'indexage :** l'axe est indexé. Pour passer au suivant, il faut reconfigurer l'I.T., donc on la stoppe ;
- **Activation des filtres/PWM :** quand les deux indexages sont terminés, on réactive les filtres pour le contrôle moteur avant de remettre en fonctionnement les interruptions ;
- **Transmission des infos de calage :** les valeurs sont envoyées au banc, et peuvent servir pour le calculateur en cas de stratégie de redémarrage en vol sans indexage ;

- **Reconfiguration des timers en contrôle moteur :** on remet les PWM à 125 kHz, déphasés de 180°, et un déclenchement autonome de l'ADC sur son propre timer pour permettre le fonctionnement en contrôle moteur.

Note : le séquencement traite chaque axe séparément pour bénéficier du maximum de courant possible pour chaque essai et indexage.

## 2. Programme de la phase statique



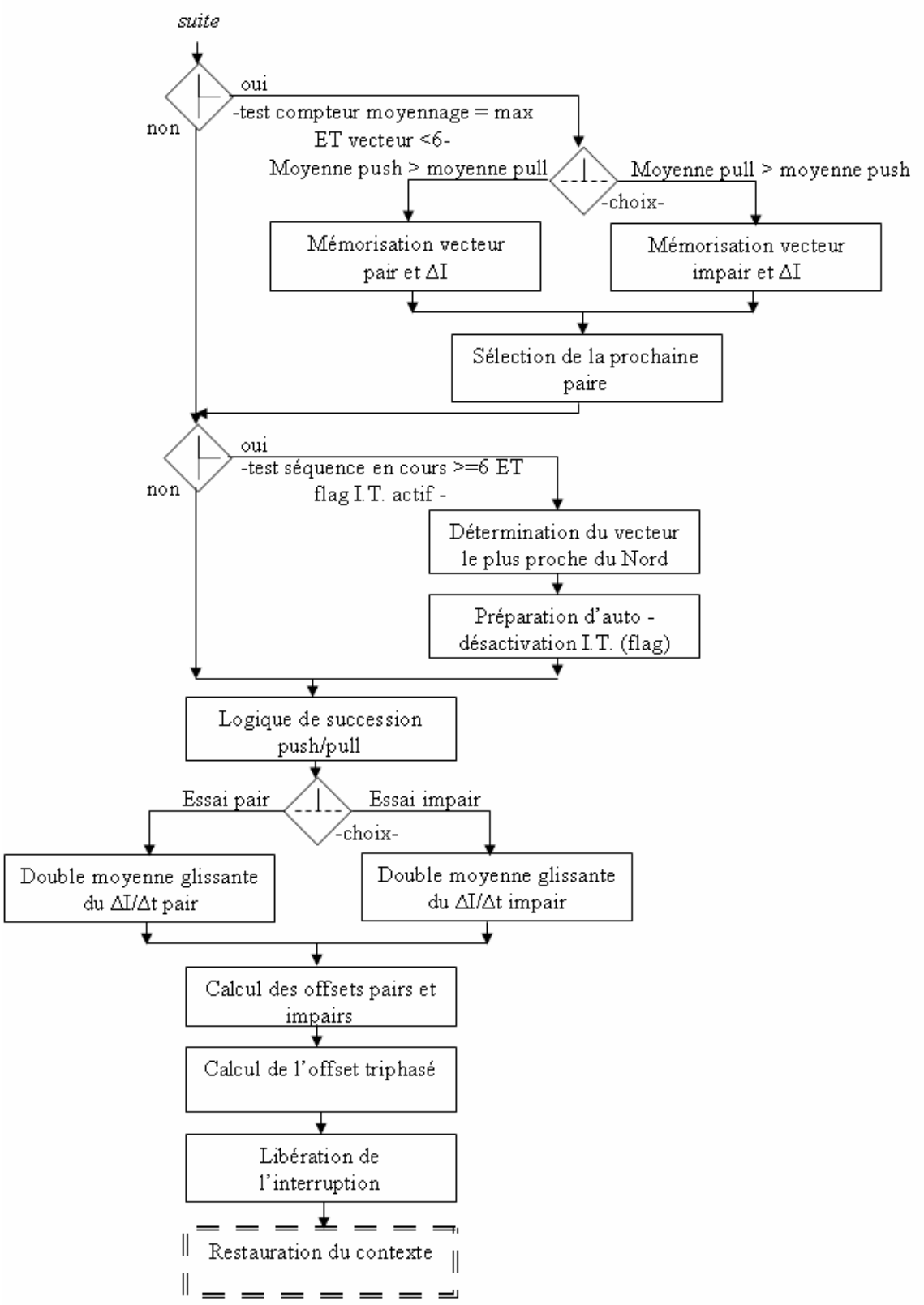



Fig. 80 : Interruption d'indexage automatique

Explication des blocs :

- **Attente évènement de fin de conversion ADC :** c'est un mécanisme matériel qui déclenche la commutation d'un flag. Le top d'un timer commute un flag qui est visible depuis le système d'interruptions (même mécanisme que l'interruption de contrôle moteur) ;
- **Action immédiate :** priorité maximale de l'I.T. comme celle du contrôle moteur. Cette I.T. n'est jamais utilisée en même temps que son homologue ;
- **RAZ compteur des étapes de génération des vecteurs :** cela permet de créer une machine à états finis via l'interruption. Chaque interruption a donc une distinction cyclique ;
- **Aiguillage du compteur d'étape :** quand on est au début de la machine à états finis, et que l'on a pas fait le test sur les trois paires, on génère le vecteur de l'essai ; sinon, il n'y a pas d'actions sur les PWM ;
- **Calcul PWM du vecteur pour essai $\Delta I/\Delta t$ :** génération du vecteur pour l'essai ;
- **Arrêt des PWM :** pendant le déroulement de la machine à états finis, cette étape permet notamment la démagnétisation ;
- **Aiguillage de choix de l'axe :** comme les codes 'C' sont communs (même I.T.), le « main » définit l'axe à prendre en compte pour l'indexage (l'interruption sera donc utilisée deux fois). Pendant toute la phase d'indexage de l'axe, ce paramètre est constant ;
- **Affectation PWM axe X :** le vecteur précédemment généré est appliqué aux PWM ;
- **Lecture des courants :** lecture du buffer dont le remplissage a été effectué en synchronisation avec un timer ;
- **Lecture des positions :** cela permet de mesurer les déplacements pour la mise au point (voir pré-indexage) et pour le post-positionnement ;
- **Aiguillage du compteur d'étapes :** au niveau de la machine à états finis, on est au moment de la fin de l'acquisition de la mesure. Si la mesure est déjà faite (ex : phase de démagnétisation), on ne rafraîchit pas les mesures qui ne veulent rien dire (l'ADC ne s'arrête pas !) ;
- **Mise à jour des mesures axe X :** on tient compte de la mesure correspondant au maximum de $\Delta I/\Delta t$ ;
- **Incrémentation compteur de moyennage ou RAZ :** cela permet de générer un compteur de moyennage pour les différentes moyennes imbriquées ;

- **Aiguillage compteur moyennage :** si la fin de la moyenne est atteinte, et que l'on est toujours dans le cadre des essais des trois paires, on peut procéder au bilan de l'essai ;
- **Aiguillage moyenne push et pull :** selon le résultat du $\Delta I/\Delta t$ le plus important, on va finaliser les résultats avec la moyenne prépondérante ;
- **Mémorisation vecteur x.pair et $\Delta I$ :** on mémorise le numéro du vecteur dont l'essai est majeur, avec son courant associé ;
- **Sélection de la prochaine paire :** l'essai terminé, on passe à la paire suivante ;
- **Aiguillage de la séquence en cours :** si tous les essais sont terminés, on passe dans un autre mode, celui où l'I.T. va laisser sa place au post-positionnement ;
- **Détermination du vecteur le plus proche du Nord :** avec les trois essais, on détermine le vecteur le plus proche du pôle Nord ;
- **Préparation de l'auto-désactivation de l'I.T.** : un Flag est commuté, celui-ci sera détecté par le « main » (séquence d'indexage) qui va stopper l'I.T. ; qui se termine alors elle-même ;
- **Logique de succession push/pull :** permet d'alterner des essais « push », puis « pull », puis « push »…durant toute la phase de moyennage ;
- **Aiguillage pair/impair :** selon l'essai en cours, on va rafraîchir la moyenne adéquate ;
- **Calcul des offsets pairs et impairs :** on distingue deux offsets triphasés distincts selon le sens du courant, car ils ne sont pas symétriques par rapport au zéro ;
- **Calcul de l'offset triphasé :** c'est la combinaison des offsets précédents pour obtenir un offset moyen, afin de pouvoir comparer les essais par rapport au zéro ;
- **Libération de l'interruption et restauration du contexte :** action software/hardware pour le fonctionnement de l'interruption.

## 3. Programme de la phase électromécanique

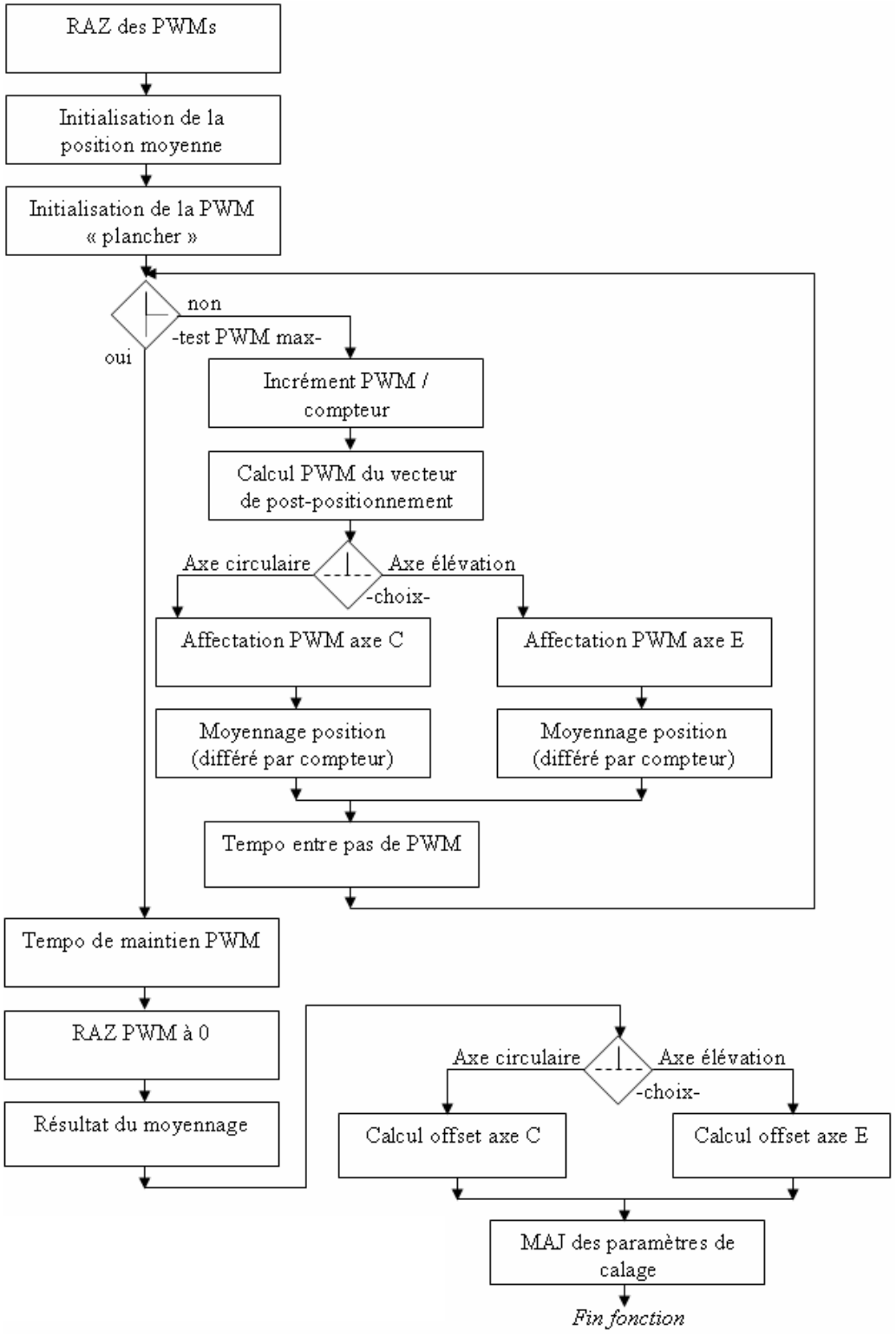



Fig. 81 : Fonction de post-positionnement

Explication des blocs :

- **RAZ des PWM :** on met les PWM en standby ;
- **Initialisation de la position moyenne :** RAZ de la position ;
- **Initialisation de la PWM « plancher » :** le post-positionnement commence avec une PWM créant un couple d'arrachement suffisamment grand pour un mouvement immédiat ;
- **Aiguillage PWM max :** le post-positionnement est réalisé avec une PWM croissante pour créer un couple qui augmente linéairement jusqu'à la position d'équilibre (pour limiter les oscillations). Quand la phase linéaire est terminée, on achève le post-positionnement ;
- **Incrément PWM / compteur :** cela permet de créer un compteur pour l'évolution linéaire de la PWM et pour le moyennage de la position jusqu'au courant max. admis ;
- **Calcul PWM du vecteur de post-positionnement :** on définit les bras d'onduleur à commuter en fonction du vecteur le plus proche du nord à générer ;
- **Aiguillage de l'axe :** comme c'est une fonction commune, le séquenceur d'indexage définit pour toute la durée d'indexage l'axe en cours d'indexage;
- **Affectation PWM axe X :** génération du vecteur sur les PWM ;
- **Moyennage position (différé par compteur) :** pour s'affranchir des éventuelles oscillations autour du point d'équilibre, on moyenne la position sans tenir compte de la position initiale et du transitoire de déplacement pour minimiser le temps de moyennage ;
- **Tempo entre pas de PWM :** comme cette fonction n'est pas en interruption (pas de cycle), la tempo permet de faire des temps d'attente entre chaque évolution de PWM pour coller avec les inerties mécaniques ;
- **Tempo de maintien PWM :** lorsque le post-positionnement est terminé, on maintient la position si on veut vérifier la stabilité de l'antenne ;
- **RAZ PWM à 0 :** on relâche les contraintes sur l'antenne ;
- **Résultat du moyennage :** on en déduit la position d'équilibre. Le calcul n'est pas fait avant pour ne pas arrondir les nombres ;
- **Calcul offset axe X :** la position d'équilibre permet de déterminer l'offset avec la position d'assemblage (qui est modulo et dont le zéro absolu est la référence) ;
- **MAJ des paramètres de calage :** les offsets identifiés seront utilisés pour le contrôle moteur.

# III.Bilan de l'indexage automatique :

L'indexage mis en œuvre est dédié au fonctionnement du radar aéroporté ou à un dispositif présentant un cycle robotique équivalent. En effet, on doit pouvoir faire des mouvements au démarrage, même s'ils sont faibles. L'antenne se comporte comme une charge de masse et de dimensions constantes, dont le couple délivré par l'ensemble moteurs - amplificateurs pour créer le mouvement d'indexage nécessite d'être inférieur au couple maximal disponible. Il faut aussi se réserver une marge (de couple) si l'environnement est chahuté. Cette méthode ne s'applique donc pas au démarrage à pleine charge.

Les essais en laboratoire sur l'antenne ont donné de bons résultats, reproductibles et constants. On indexe les deux axes avec une précision inférieure à 5° électriques. Tous les tests n'ont pas pu être réalisés concernant notamment les fortes vibrations car cela nécessite une mise en place d'un banc, impossible dans les délais de la thèse. Par contre, toute l'approche théorique a été faite avec la même rigueur que les simulations fonctionnelles et systèmes. De la même façon, certains algorithmes mineurs de protection restent à implémenter.

On peut considérer que les travaux présentés ici sont destinés à une « niche » d'applications. Il existe néanmoins plusieurs aspects novateurs, au-delà du caractère très ciblé pour l'application. En effet, les outils mis en œuvre sont réutilisables. L'usage du DSP, vu sous l'aspect fonctionnel de l'indexage seul ou dans l'ensemble du projet, est une réelle progression dans le domaine des alimentations (ici extension aux servomécanismes). Pour des applications plus conventionnelles des moteurs synchrones (hors des cycles radars), l'indexage statique peut être amélioré si on utilise des machines plus adaptées.

Les travaux réalisés pour le radar font actuellement l'objet de dépose de Brevet d'invention. Plusieurs aspects novateurs ont été démontrés. Les démarches entreprises pour le dépôt de Brevet sont plutôt longues à l'échelle de la thèse, elle-même ne présentant pas un avancement assez rapide, donc les publications ne peuvent pas être citées en l'état dans le contenu du manuscrit.

# Conclusion

De nos jours, l'électronique de puissance est omniprésente dans la gestion et le traitement de l'énergie. Les convertisseurs statiques existent dans quasiment tous les endroits où se trouve de l'énergie électrique. Les domaines d'applications couvrent toutes les gammes de puissance, et chaque application définit les critères prépondérants : le MTBF, le coût, les performances, la taille.

Cette thèse traite des domaines d'application de l'aéronautique civile, mais pour élaborer un système « sur étagère » visant également les applications militaires, qui sont les créneaux de THALES Systèmes Aéroportés, équipementier des avions pour la navigation et les servomécanismes. Les critères prépondérants sont nombreux, et sont les plus contraignants parmi l'ensemble des domaines possibles. En effet, l'aéronautique pousse l'exigence dans les extrêmes, compte tenu des risques encourus et des budgets mis en jeu.

Le fond de la thèse repose sur l'implémentation, le design de convertisseurs statiques lorsque l'on remplace les systèmes analogiques existants pour exploiter les performances et les atouts de compacité des systèmes numériques. Cette évolution est réglementée par les normes aéronautiques, où les composants numériques sont vus comme de véritables électrons libres. Les normes sont donc là pour donner des règles de développements, de tests et de vérifications. Cependant, la norme n'a pas pour rôle de donner des directives aux développeurs, et c'est là qu'interviennent les méthodes proposées dans cette thèse.

Cette thèse a couvert un large éventail du domaine des sciences de l'ingénieur. Le but initial n'était pas celui-ci, puisque la thèse initiale se focalisait plus sur la commande d'un simple onduleur, et sur des points plus précis. Le contexte industriel a fait évoluer le cours de la thèse vers l'application « Radar de pointe », qui a bouleversé le contenu et les objectifs de la thèse. C'était à la fois très intéressant pour ma formation et mes objectifs après la thèse, mais cela a été une contrainte sous-estimée quant aux difficultés vis-à-vis de mes compétences.

Certains points n'ont pas pu être suffisamment approfondis d'un point de vue académique, pour être réutilisés, notamment pour l'impact des normes sur les méthodes de conception. De plus, l'indexage moteur aurait pu être bien plus rapide à développer avec des experts du domaine.

Néanmoins, le projet est un succès puisque le prototype est opérationnel et conforme aux attentes. Un gros travail documentaire a été de réaliser en parallèle de la thèse, un document (DCM) concernant tout le développement du système, et la thèse ne montre pas les outils qui ont

été développés uniquement pour le projet. Le travail avait dès le début été sous-estimé, et le départ prématuré de mon encadrant industriel a été pénalisant.

Sur un plan plus technique, les espoirs placés dans la génération automatique de code ne sont pas totalement récompensés, mais au moment de la thèse, les outils qui étaient encore tous récents (version béta) n'étaient pas assez matures. En revenant dessus vers la fin de ma thèse, bon nombre de difficultés n'étaient pas encore résolues. Pour que les outils évoluent plus vite, il faudrait à mon sens un meilleur retour d'expérience sur les outils vers Matlab® : ces retours sont plutôt sommaires vu le caractère très particulier des applications.

Le projet ainsi réalisé est mené au terme des objectifs initiaux, et la majorité du travail devrait pouvoir être reprise pour l'industrialisation. Par contre, pour le « système sur étagère », il reste encore du travail puisque toute l'implémentation repose sur un dimensionnement spécifique, et dédié au fonctionnement du radar.

Finalement, on a soulevé beaucoup de questions ; des difficultés restent encore à résoudre. Contrairement au début du projet, on a maintenant une vision assez précise du travail à fournir, dans quels axes, que ce soit à moyen ou à long terme sur les projets intégrant un DSP ; ou que ce soit sur les technologies à venir et comment en tirer le meilleur parti : coût/performances/mise en œuvre. Même si les technologies évoluent encore beaucoup en terme d'intégration, la problématique du code en électronique de puissance sera toujours la même.

# Table des figures.

# Bibliographie.

**Chapitre 1 :**


[ABD100] :Norme ABD0100.1.2 Part1 Chap2 AIRBUS Industrie, « Environment »

[AMDEC] :Outil AMDEC, Pascal Guesdon, Supélec, 1997

[AZZ04] : Abedenour Azzedine, UFR SSI, UBS (Bretagne Sud) ; « Outil d'analyse et de partitionnement/ordonnancement pour les systèmes temps réels embarqués », Thèse, 2 juin 2004.

[BAB05] : Jean-Philippe Babau, L3i/CITI - INSA Lyon ; « Formalisation et structuration des architectures opérationnelles pour les systèmes embarqués temps réel », HDR, 12 décembre 2005.

[BON03] : Frédéric Boniol, Gérard Bel, Jérôme Ermont ; « Trois approches pour la modélisation et la vérification de systèmes embarqués » ; techniques et sciences informatiques, VOL.X, N°X/2003, pages 1 à X. (Chercheurs ONERA / ENSEIHT)

[RLAB] : http://www.boeing.com/assocproducts/radiationlab/index.htm

[DO178] : Norme DO-178B : Software Consideration in Airbone Systems and Equipment Certification, RTCA, December 1, 1992

[DO254] : Norme DO-254 : Design Assurance Guidance for Airbone Electronic Hardware, RTCA, April 19, 2000

[DOD03] : Paul E. Dodd, Lloyd W. Massengill ; « Basic Mechanisms and Modeling of Single Event Upset in Digital Microelectronics » ; IEEE Trans on Nuclear Science, volume 50, n°3, juin 2003

[FOR90] : Ray Ford, « Monitoring Distributed Embedded Systems », DCS University, Kansas, IEEE TH0307-9/90/0237, 1990

[GUI03] : Jérémie Guiochet ; « Maîtrise de la sécurité des systèmes de la robotique de service, approche UML basée sur une analyse du risque système » ; Thèse INSA Toulouse, LESIA, N°695, 9 juillet 2003

[HIE05] : David M. Hiemstra, Miladinovic B., Chayab F. ; « Single Event Upset Characterization of the SMJ320C6701 DSP using Proton Irradiation » ; IEEE Radiation effects Data Workshop, p42-45, 11-15 juillet 2005

[ISO01] : Damir Isovic, Gerhard Fohler, DCE Suède, « Efficient Scheduling of Sporadic, Aperiodic and Periodic Tasks with Complex Constraints », The 21st IEEE, Real-Time Systems Symposium, p207-216, 27-30 novembre 2000, Orlando, FL, USA

[KAL04] : Hamoudi Kalla, « Génération automatique de distributions / ordonnancements temps réel, fiables et tolérants aux fautes », thèse INPG spécialité Informatique « Systèmes et Logiciels », INRIA, 17 décembre 2004

[LAP89] : Jean-Claude Laprie, LAAS-CNRS ; « Sûreté de fonctionnement des systèmes informatiques et tolérance aux fautes », Techniques de l'ingénieur R7-595, paru en octobre 1989

[LIU73] : C. L. Liu, James W. Layland, M.I.T. ; « Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment », projet MAC, JACM, volume 20, issue1, p46-61, janvier 1973, New York, USA.

[MAJ95] : Peter P. Majewski, Eugene Normand, Dennis L. Oberg ; « A new Solar Flare Heavy Ion Model and its Implementation througth MACREE, an improved Modeling Tool to Calculate Single Event Effects rates in Space » ; Boeing Defense and space group, Seattle, WA 98124-2499, IEEE Trans. On Nuclear Science, volume 42, issue 6, part1, p2043-2050, décembre 1995, Madison, WI, USA, meeting : 17-21 juillet 1995.

[NOR93] : Eugene Normand, Taber A. ; « Single Event Upset in Avionics » ; Boeing Defense and space group, IEEE Trans. On Nuclear Science, volume 40, issue 2, p120-126, avril 1993

[NOR95] : Eugene Normand, J. L. Wert ; « Single Event Upset and latchup Measurements in avionics Devices using the WNR neutron Beam and a new neutron-induced latchup Model » ; Boeing Defense and space group, IEEE Radiation Effects Data Workshop, NSREC '95 Workshop Record, p33-38, 19 juillet 1995, meeting : 17-21 juillet 1995, Madison, WI, USA

[NOR96] : Eugene Normand ; « Single Event Upset a ground level » ; Boeing Defense and space group, IEEE Trans. On Nuclear Science, volume 43, issue 6, Part 1, p2742-2750, décembre 1996, meeting 15-19 juillet 1996, Indian Wells, CA, USA

[NOR97] : Eugene Normand, Wert J. L., Oberg D. L., Majewski P. R., Voss P., Wender S.A. ; « Neutron-Induced Single Event Burnout in High Voltage Electronics » ; Boeing Defense and space group, IEEE TRANS. on Nuclear Science, volume 44, issue 6, Part 1, p2358-2366, décembre 1997, meeting : 21-25 juillet 1997, Snowmass Village, CO, USA.

[NOR98] : Eugene Normand ; « Extensions of the burst Generation rate Method for wider Application to proton/neutron induced Single Event Effects » ; Boeing Defense and space group, IEEE Trans. On Nuclear Science, volume 45, issue 6, Part 1, p2904-2914, décembre 1998, meeting : 20-24 juillet 1998, Newport Beach, CA, USA.

[OBE93] : Dennis L. Oberg, Jerry L. Wert, Eugene Normand ; « Measurement of Single Events Effects in the 87C51 Microcontroller » ; Boeing Defense and space group, Seattle, WA 98124-2499, IEEE Trans. On Radiation Effects Data Workshop, p43-50, 21 juillet 1993

[PET02] : Dennis K. Peters, David Lorge Parnas ; « Requirements-based Monitors for Real-Time Systems » ; IEEE Trans. On Software Engineering, volume 28, issue 2, p146-158, Feb.2002

[QIU00] : Xiaobing qiu, Wolfgang Wimmer ; « Applying object-orientation and Component Technology to Architecture Design of Power System Monitoring », ABB Power Automation, Suisse, IEEE 0-7803-6338-8/00, Proceedings PowerCon 2000 International Conference, Power System Technology, volume 2, p589-594, 4-7 décembre 2000, Perth, WA

[RIC03a] : P. Richard, M. Richard, F. Cottet ; « Analyse holistique des systèmes temps réel distribuées : principes et algorithmes » ; LISI/ENSMA, Hermès 2003

[RIC03b] : Pascal Richard, LISI-ENSMA ; « Analyse du temps de réponse des systèmes temps réels », AETR., Actes de l'Ecole d'été Temps Réel, 2003

[SPU96] : Marco Spuri, INRIA ; « Analysis of deadline Scheduled Real-Time Systems », rapport de recherche n°2772, janvier 1996.

[TCHIN] : Techniques de l'ingénieur

[THI04] : Lothar Thiele, DITEE ETH Zürich ; « Design for Timing Predictability », Revue Real-time systems, volume 28, N°2-3, p157-177, Kluwer Academics Publisher, Springer Netherlands, collection Computer Science, novembre 2004, ISSN 1573-1383.

[TIN03] : Ken Tindell, Hermann Kopetz, Fabian Wolf ; « Safe Automotive Software Development », LivesDevices et Volkswagen, IEEE Procedings of the design, automation and test in Europe Conference and Exhibition, p616-621, 2003.

f

**Chapitre 2 :**


[ABD01] : R. Ben Abdenour, P. Borne, M. Ksouri, F. M'Sahli ; « Identification et commande numérique de procédés industriels, méthodes et pratiques de l'ingénieur », Technip, Paris, 2001, INIST L27809

[DO160] : Norme DO-160E : Environmental Conditions and Test Procedures for Airbones Equipment, December 4, 2004

[LEE98] : Chang-Gun Lee, DCE, Corée ; « Analysis of Cache Related pre-emption Delay in Fixed Priority pre-empting Scheduling », IEEE TRANS. on Computer, volume 47, issue 6, juin 1998.

[MAH07] : Arnaud Mahe ; « Dimensionnement électromécanique d'une antenne aéroportée », Thales A.S., Brest, 2007

[MAU03] : Résonances mécaniques et servomécanismes, D. Maurel, TASFR00201972-DOP.UI/BEP, Thales A.S. Brest, 11 avril 2003

[MAR06] : Corrections des attitudes avions par les servomécanismes de l'antenne radar météo, M. Marcant, TASFR00471698, Thales A.S., Brest, 11 mai 2006

[OZE06] : Damien Ozenne ; « Etude des asservissements d'une antenne aéroportée », Thales A.S., Brest, 2006

[RCM] : DR.RCM n°88/505 : Référentiels utilisés pour le radar aéroporté, Thales

[REFSA] : IRS, CIDS, PIDS, ICD THALES Systèmes Aéroportés™, 2006-2008

[SCS04] : Software Coding Standard, THALES Avionics™ J40894 issue AA-03, April 05, 2004

[TEX] : Sprs174n, spru060, spru065, spru078, spru095, Texas Instruments™, www.ti.com

**Chapitre 3 :**


[BEN03] : Benchaib Abdelkrim, Alacoque Jean-Claude, Poullain Serge, Thomas Jean-Luc ; « Initial Rotor Position Detection of Permanent Magnet Synchronous Motor » ; EPE topic 9a, Permanent Magnet Machines and Drives, 2-4 septembre 2003, Toulouse

Brevet d'invention Européen EP1398869A1: Benchaib Abdelkrim, Alacoque Jean-Claude, Poullain Serge, Thomas Jean-Luc ; ALSTOM ; « Procédé et calculateur de détermination de la position angulaire à l'arrêt d'un rotor, unité de commande et système incorporant le calculateur », INPI 2004

[BER02] : Nicolas Bernard ; « Machine Synchrone : de la boucle ouverte à l'autopilotage » ; Revue 3EI pp24-39, n°30, septembre 2002

[BRU96] : Cyrille Bruguier ; « Commande d'une machine synchrone à aimants sans capteur mécanique », L.E.G., Thèse INPG, 28 octobre 1996

[CAR93] : Laurent Cardoletti ; « Commande et réglage de moteurs synchrones auto-commutés par capteurs indirects de position », EPFL, Thèse n°1118, 1993

[GOR88] : S.F. Gorman ; « Determination of Permanent Magnet Synchronous Motor Parameters for use in Brushless DC Motor Drive Analysis », IEEE TRANS. On Energy Conversion, volume 3, issue 3, p674-681, septembre 1988

[HAR05a] : Michael C. Harke, Dejan Raca, Robert D. Lorenz ; « Fast and Smooth Initial Position and Magnet Polarity Estimation of Salient and near zero saliency PMSM » ; IEEE International Conference on Electric Machines and Drives, p1037-1044, 15 mai 2005

[HAR05b] : Michael C. Harke, Dejan Raca, Robert D. Lorenz ; « Implementation issues for fast Initial Position and Magnet Polarity Identification of PMSM with near Zero Saliency »; Power Electronics and Applications, p10, EPE 2005 – Dresde, Allemagne

[HAS07] : Chihiro Hasegawa, Shoji Nishikata ; « A Simple starting Method for self controlled Synchronous Motors in Electric Propulsion Systems for Ships » ; Power Electronics and Applications, EPE, 2-5 septembre 2007, Aalborg Danemark

[ICH04] : Shinji Ichikawa, Mutuwo Tomita, Shinji Doji, Shigeru Okuma ; « Initial Position Estimation and low speed Sensorless Control of Synchronous Motors in Consideration of Magnetic Saturation based on System Identification Theory » ; IEEE 39th annual meeting conference Record, Industry Applications Conference, volume 2, p971-976, IAS, 3-7 octobre 2004

[JAK05] : Piotr Jakubowski, Wlodzimierz Koczara, Nazar Al-Khayat ; « Method of the Poles Position Identification for Brushless Axial Flux Permanent Magnet Motor Drive

System » ; European Conference on Power Electronics and Applications, ISBN 90-75815-09-3, EPE, 2005, Dresde Allemagne

[KIE02] : Juergen Kiel, Stephan Beineke, Andreas Buente ; « Sensorless Torque Control of Permanent Magnet Synchronous Machines over the whole Operation Range » ; EPE-PEMC, T9-053, Dubrovnik & Cavtat, 9 septembre 2002

[KIM03] : Hyunbae Kim, Kum-Kang Huh, Michael Harke, Jackson Wai, Robert D. Lorenz, Thomas M. Jahns ; « Initial Rotor Position Estimation for an integrated starter Alternator IPM Synchronous Machine » ; EPE, 2003, Toulouse

[MAT94] : Nobuyuki Matsui, Takaharu Takeshita ; « A novel starting Method of Sensorless Salient Pole Brushless Motor » ; IEEE Conference Record of the Industry Applications Society annual Meeting, volume 1, p386-392, 2-6 octobre 1994, Denver, CO, USA

[NAI92] : Malakondaiah Naidu, bimal K. Bose ; « Rotor Position Estimation Scheme of Permanent Magnet Synchronous Machine for high Performance Variable Speed Drive » ; IEEE Conference Record of the Inductry Applications Society Annual Meeting, volume 1, p48-53, 4-9 octobre 1992, Houston, TX, USA

[NAK00] : Shin Nakashima, Yuya Inagaki, Ichiro Miki ; « Sensorless Initial Rotor Position Estimation of Surface Permanent Magnet Synchronous Motor » ; IEEE Trans. Industry Applications, volume 36, issue 6, p1598-1603, nov/dec 2000

[NOG98] : Toshihiko Noguchi, Kazunori Yamada, Seiji Kondo, Isao Takahashi ; « Initial Rotor Position Estimation Method of Sensorless PMSM with no Sensitivity to Armature Resistance » ; IEEE Trans. Industry Applications, volume.45, issue 1, p118-125, février 1998

[OST96] : Stefan Östlund ; « Sensorless Rotor Position Detection from Zero to rated speed for an integrated PMSM Drive » ; IEEE Trans. Industry Applications, volume 32, issue 5, p1158-1165, Septembre/octobre 1996

[PER07] : Jan Persson, Miroslav Markovic, Yves Perriard ; « A new standstill Position Detection Technique for non salient PMSM using the Magnetic Anisotropy Method », Annual Meeting Conference Record of the Industry Applications, volume 1, p238-244, 2-6 octobre 2005

[POP07] : Dumitru Daniel Popa, Liviu Mario Kreindler, Raducu Giuclea, Aurelian Sarca ; « A Novel Method for PMSM Rotor Position Detection » ; EPE 12th European Conference on Power Electronics and Applications, 2-5 septembre 2007, p1-10, Aalborg Danemark

[RAU07] : Reiko Raute, Cedric Caruana, Joseph Cilia, Cyril Spiteri staines ; « A Zero Speed Operation Sensorless PMSM Drive without additional Test Signal Injection » ; EPE, 2-5 septembre 2007, Aalborg Danemark

[SCH88] : M. Schroedl ; « Detection of the Rotor Position of a Permanent Magnet Synchronous Machine at Standstill » ; ICEM, p195-197, 1988, Pise, Italie

[SCH90] : Manfred Schroedl, « Operation of the Permanent Magnet Synchronous Machine without a Mechanical Sensor », International Conference on Power Electronics and Variable-speed Drives, p51-56, 17-19 juillet 1990, Londres, Angleterre

[SCH97] : Peter Schmidt, Michael Gasperi, Glen Ray, Ajith H. Wijenayake ; « Initial Rotor Angle Detection of a non Salient Pole Permanent Magnet Synchronous Machine », IEEE Trans. Industry Applications Society Annual Meeting, IAS, volume 1, p459-463, New Orleans, Louisiana, 5-9 octobre 1997

[SCH02] : Manfred Schroedl, « Industrial Sensorless Permanent Magnet Synchronous Motor Drives based on the INFORM Method for high Performance including Standstill », EPE-PEMC, Dubrovnik & Cavtat, Croatia, September 9-11, 2002.

[SCH07] : Manfred Schrödl, Matthias Hofer, Wolfgang Staffler ; « Extended EMF and Parameter Observer for Sensorless controlled PMSM Machines at low Speed » ; EPE, 2-5 septembre 2007, Aalborg Danemark

[STI99] : M. Stiebler, Y. Li ; « Detection of the Rotor Position of a Permanent Magnet Synchronous Motor at Standstill » ; European Transactions on Electrical Power, ETEP, volume 9, issue1, p43-47, janvier 1999

[TAK96] : Takaharu Takeshita, Nobuyuki Matsui ; « Sensorless Control and Initial Position Estimation of Salient Pole brushless DC Motor » ; 4th International Workshop on Advanced Motion control, volume 1, p18-23, AMC, 18-21 mars 1996

[TUR03] : Marco Tursini, Roberto Petrella, Francesco Parasiliti ; « Initial Rotor Position Estimation Method for PM Motors » ; IEEE Trans. Industry Applications, volume 39, issue 6, p1630-1640, nov.dec 2003

[WIS07] : Janusz Wisniewski, Piotr Jakubowski, Wlodzimierz Koczara ; « Poles Position Identification of Permanent Magnet Axial Flux Motor using PIPCRM Sensorless Method » ; EPE, 2-5 septembre 2007, Aalborg Danemark