

UNIVERSITÉ DE MONTRÉAL

CONCEPTION D'UNE ARCHITECTURE LOGICIELLE POUR L'INTÉGRATION
DES SYSTÈMES DE POSITIONNEMENT, CONTRÔLE ET PROPULSION D'UN
DISPOSITIF FERROMAGNÉTIQUE ENDOVASCULAIRE AUTONOME GUIDÉ
PAR IRM

ARNAUD CHANU
INSTITUT DE GÉNIE BIOMÉDICAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE BIOMÉDICAL)

AOÛT 2006

© Arnaud Chanu, 2006.



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-19288-7
Our file *Notre référence*
ISBN: 978-0-494-19288-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CONCEPTION D'UNE ARCHITECTURE LOGICIELLE POUR L'INTÉGRATION
DES SYSTÈMES DE POSITIONNEMENT, CONTRÔLE ET PROPULSION D'UN
DISPOSITIF FERROMAGNÉTIQUE ENDOVASCULAIRE AUTONOME GUIDÉ
PAR IRM

présenté par : CHANU Arnaud

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. BEAUDOIN, Gilles, Ph.D., président

M. MARTEL Sylvain, Ph.D., membre et directeur de recherche

M. BOUDREAU, Yves, Ph.D., membre

REMERCIEMENTS

Mes premiers remerciements vont à mon directeur de recherche, le Professeur Sylvain Martel qui a rendu possible cette Maîtrise et m'a permis à maintes reprises de me démarquer et de saisir pleinement le contexte du domaine de la recherche.

Je tiens également à remercier chaleureusement mes parents qui m'ont soutenu dans tout ce que j'ai pu entreprendre au cours de toutes ces années. Ils ont su me léguer l'essentiel et ont toujours cru en moi. Sans eux, ce mémoire comme beaucoup d'autres choses n'aurait été qu'un vague songe.

Merci à mon plus fidèle ami qui a été plus qu'un frère pour moi, Fadi Asmar. Tu as su tout comprendre avant même que j'y pense parfois et jamais je n'ai connu une telle amitié qui, je l'espère, durera encore longtemps. Après plus de 10 ans de partage et de conseils, on peut difficilement ne pas être rassuré et fier d'avoir un ami comme toi.

Un grand merci à ma copine, Valérie Langlois qui, jours après jours, me soutient et me réconforte dans les moments difficiles et m'accompagne dans les moments plus heureux pour ne faire qu'un.

A tous mes collègues du projet MR-Sub, un énorme merci pour leur soutien et leurs conseils. Merci à Jean-Baptiste Mathieu pour les rares 'party' que tu as organisés au cours de ces dernières années mais qui furent si agréables. Tu es mon gestionnaire de projets informatiques préféré et le roi du MR-Sub. Ce fut un honneur de travailler avec toi JB (mais ce n'est pas fini). Un énorme merci à Ouajdi Felfoul, pour toutes les conversations stimulantes et frustrantes que l'on a pu avoir. Les CDs cassés sous la rivalité et les nuits enivrantes de Palo-Alto. Alan Hume serait fier de toi ! Un grand

merci au paisible et calme Samer Tamaz qui sait maîtriser toutes les situations en fermant les yeux et en inspirant profondément... quel contrôle ! Merci à Eric Aboussouan pour son caractère si antipathique mais si drôle durant les nuits à l'IRM. Il est la preuve vivante qu'on a toujours besoin d'un plus petit que soi. Bon courage dans ta future carrière et sache que Ouajdi n'a pas de famille à Dubai, c'était une diversion !

Un énorme merci à tous mes 'NanoCollègues' qui ont su me rendre la vie plus agréable au cours des deux dernières années. Merci au nouveau Charles Charbonneau Tremblay pour son emprise tyrannique sur le laboratoire, sans toi, nous serions libres mais innocents. Un grand merci à Martin Mankiewicz pour sa folie à mon début de Maîtrise et à sa version 2.0 plus calme et plus sereine dans les derniers mois. Peu importe ta version, elle est toujours agréable à côtoyer. Merci à Pascal pour ses commentaires parfois innovateurs et sa musique trance qui a su garder le lab à la fine pointe des plus chauds 'after hours'. Un énorme merci à Pierre Pouponneau pour son support lors de nos défaites à l'IRM et pour sa potion magique dans laquelle il est probablement tombé étant petit (non Pierre, je ne dirai rien de plus). Un grand merci à mon ancien étudiant de l'UQAM préféré, à un ex-coloc de courte durée et un imitateur de secrétaires hors pairs, j'ai nommé Olivier Don Truong. Tu as su nous déranger tant de fois pour si peu mais ta joie et ton entrain ont toujours su faire renverser la balance. Merci à Ramzi Bey Oueslati pour toutes ces grandes discussions si profondes, si amicales, si réciproques. Tu as exprimé tant d'amitié envers moi qu'il est dur d'en retenir le meilleur moment. Merci à Dominic St-Jacques qui m'a légué la merveilleuse responsabilité de 'Sysadmin' du Laboratoire de Nanorobotique. Tu avais raison Dom, ce

n'est pas un travail à faire en Maîtrise. Merci à Wael Sabra pour son moral exemplaire et ses conseils judicieux dans mon travail. Un grand merci à Daniel Parent Vézina qui nous a permis de démystifier un peu plus les possibilités de l'IRM. Tu voulais que je te tienne au courant des avancements, ce mémoire devrait te satisfaire. Merci à tous les autres que j'ai moins côtoyés mais bien appréciés, merci donc à Walder André, à Anh Tuan Nguyen, à Zhao Lu pour ses conseils avant mon voyage en Chine et à Mahmood Mohammadi.

Un énorme merci à Neila Kaou pour son aide et sa disponibilité durant la rédaction de ce mémoire mais également en tout temps. On est si heureux de te revoir parmi nous Neila ! Merci également à Kwang Soo Kim pour ses questions parfois pertinentes et ses danses du Nanolab si envoûtantes. Je n'ai qu'une seule chose à dire : 'Mozilla!'.

Un très grand merci à M. Gilles Beaudoin à la fois pour bien avoir accepté de faire partie de mon jury mais également pour avoir été souvent disponible pour le projet MR-Sub et finalement pour m'avoir accompagné au 'Workshop' à Chicago où je fus confronté pour la première fois à une angoissante présentation. Merci également à M. Yves Boudreault d'avoir sacrifié des jours de vacances pour faire partie de mon jury.

Finalement, un dernier merci à M. Gino Potvin, technologue au CHUM, pour toutes ces nuits en IRM passées avec l'équipe du MR-Sub et pour nous avoir soutenu dans nos défaites et félicités dans nos victoires. Merci également aux autres intervenants du CHUM pour leur bonne humeur et leurs conseils très précieux.

RÉSUMÉ

Le présent mémoire de maîtrise porte sur la mise sur pieds d'une architecture informatique visant à intégrer les méthodes de propulsion, positionnement et contrôle d'un dispositif ferromagnétique en vue d'une navigation à l'aide d'un appareil d'imagerie par résonance magnétique (IRM). Un tel dispositif est développé dans l'optique d'obtenir un moyen de déplacement autonome pour d'éventuelles libérations ciblées de médicaments ou autres interventions chirurgicales dans les vaisseaux sanguins. L'utilisation des bobines de gradients magnétiques, déjà présentes sur les systèmes IRM, pour propulser, guider et contrôler un dispositif ferromagnétique assurerait à la fois une opportunité d'utilisation dans la plupart des hôpitaux ou centres médicaux équipés de ce genre de modalité d'imagerie tout en fournissant les ressources nécessaires à la navigation d'un tel dispositif.

Une description détaillée du projet MR-Sub du Laboratoire de Nanorobotique de l'École Polytechnique de Montréal est présentée afin de spécifier le logiciel et les requis matériels pour la navigation de ce dispositif. Une revue des méthodes de propulsion, de positionnement et de contrôle permet de mieux comprendre les enjeux et les nécessités d'un tel projet. La plateforme logicielle conçue et présentée ainsi que l'intégration des différents modules de propulsion, de positionnement, de contrôle et de 'Roadmap' assure la présence d'un système de navigation autonome utilisant les caractéristiques temps réel des appareils d'imagerie par résonance magnétique cliniques modernes. Cette architecture propose un modèle de classes qui assurent une cohérence fonctionnelle menant à une séquence IRM temps réel élaborée pour la navigation du dispositif dans

des environnements avec et sans écoulement de fluide. La validation de l'architecture a été effectuée avec succès sur un IRM clinique Avanto 1.5 T de la compagnie Siemens. La possibilité de réaction temps réel fourni par cet appareil assure l'atteinte des contraintes d'asservissement pour la navigation *ex vivo* d'une bille ferromagnétique de 1.5 mm de diamètre dans divers environnements. Les résultats de contrôle en une dimension et en deux dimensions permettant le suivi d'un chemin constitué de plusieurs balises sont présentés. Une fréquence opératoire de 30 Hz est atteignable et valide l'architecture présentée ainsi que l'intégration des modules de propulsion magnétiques, de positionnement et de contrôle de la bille avec des précisions de l'ordre du millimètre.

ABSTRACT

The present report relates the elaboration of a software architecture for the integration of the propulsion, tracking and control techniques needed for the navigation of a ferromagnetic device using a magnetic resonance imaging system (MRI). Such device is being developed in order to obtain a new autonomous transportation method for eventual targeted drug deliveries or surgery procedures in the human blood vessels. The use of magnetic gradient coils, already present on MRI systems, provides the external power to propel, guide and control a ferromagnetic device that could be used in the vast majority of hospitals or medical centers equipped with this kind of imaging techniques.

A detailed description of the MR-Sub project engineered at the Nanorobotic Laboratory located at Ecole Polytechnique of Montreal is presented in order to specify the necessary software and hardware requirements for the navigation of this device. A review of the propulsion, tracking and control methods makes it possible to better understand the stakes and the needs for such a project. The elaborated and presented software platform in conjunction with the integration of the propulsion, tracking, control and 'Roadmap' modules ensures the presence of an autonomous navigation system using the real time capabilities of the modern clinical magnetic resonance imaging scanners. The software architecture introduced proposes an object oriented class model leading out to a real time MRI sequence used for the device's navigation in environments with and without flow. The validation of the software architecture was carried out successfully on a Siemens Avanto 1.5 T clinical MRI. The real time feedback capabilities of the Avanto

provides the necessary requirements for the servoing constraints needed for the *ex vivo* navigation experiments on a 1.5 mm diameter ferromagnetic bead. One dimension and two dimensions control experiments using pre selected waypoints were carried out successfully and are presented. An operational frequency of 30 Hz is reachable and validates the presented software architecture as well as the integration step of the elaborated magnetic propulsion, magnetic tracking and control modules with a millimeter precision in most cases.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	vii
ABSTRACT	ix
TABLE DES MATIÈRES	xi
TABLE DES FIGURES.....	xv
INDEX DES TABLEAUX	xviii
INTRODUCTION	1
CHAPITRE 1. LE PROJET MR-SUB.....	5
Introduction.....	5
1.1 Problématique actuelle.....	5
1.2 Objectifs du projet.....	8
1.3 Les piliers du projet.....	10
1.3.1 Roadmap	10
1.3.1.2 Segmentation en vue de la reconnaissance des vaisseaux	13
1.3.1.3 Création des balises.....	14
1.3.1.3.1 Création manuelle des balises	14
1.3.1.3.2 Création automatique des balises	14
1.3.2 Propulsion magnétique.....	15
1.3.3 Positionnement magnétique	16
1.3.3.1 MS-SET	18
1.3.4 Contrôle.....	18
1.4 Objectifs et requis de la présente étude.....	19
1.4.1 Vue éclatée de l'intégration	19
1.4.2 Requis.....	20
Conclusion	23
CHAPITRE 2. ARCHITECTURE LOGICIELLE ET INTÉGRATION	24
Introduction.....	24

2.1 Contraintes du projet.....	24
2.2 Description de l'environnement informatique du système d'imagerie par résonance magnétique retenu	28
2.3 Séquences IRM	31
2.4 Environnement de développement : IDEA / ICE.....	35
2.4.1 IDEA	36
2.4.2 ICE	37
2.4.3 Protocole de la séquence et communication entre IDEA et ICE	41
2.5 Capacités temps réel de l'Avanto.....	44
2.5.1 Utilisation des possibilités temps réel.....	44
2.5.2 Fonctionnement de l'aspect temps réel.....	46
2.6 Interaction, intégration des modules et vue d'ensemble de l'architecture	54
2.6.1 Interaction des modules.....	56
2.6.1.1 Prise en compte du mouvement respiratoire et de l'aspect pulsatif de l'écoulement sanguin	58
2.6.2 Intégration des modules	62
2.7 Constitution des classes.....	68
2.7.1 Classes du programme de reconstruction d'images.....	68
2.7.1.1 Classe MPosition.....	68
2.7.1.2 Classe MChemin	69
2.7.1.3 Classe MVolumeImage.....	69
2.7.1.4 Classe MControleur	69
2.7.1.5 Classe MPlanificateur	71
2.7.1.6 Classe MGestionnaireImages.....	71
2.7.1.7 Classe MGestionnairePositionnement	71
2.7.1.8 Classe MGestionnairePropulsion	71
2.7.2 Classes de la séquence	72
2.7.2.1 Classe mCommunication	72
2.7.2.1 Classe mMrsubKernel.....	72

2.8 Diagramme de classes	73
2.9 Diagrammes de séquence	76
2.10 Validation et tests de l'architecture	80
2.10.1 Approche 'boîte blanche'	80
2.10.1.1 Complexité cyclique.....	81
2.10.1.2 Test des conditions.....	81
2.10.2 Approche 'boîte noire'	82
2.10.2.1 Tests par graphes.....	82
2.11 Allure de la séquence temps réel du MR-Sub.....	83
Conclusion	86
CHAPITRE 3. ANALYSE ET RÉSULTATS	87
Introduction.....	87
3.1 Méthodologie	88
3.2 Résultats	95
3.2.1 Résultats de contrôle 1D	97
3.2.1.1 Montage et paramètres	97
3.2.1.2 Contrôle sans écoulement	99
3.2.1.3 Contrôle avec écoulement constant.....	101
3.2.1.4 Contrôle avec écoulement pulsé.....	102
3.2.2 Résultats de contrôle 2D	104
3.2.2.1 Montage et paramètres	104
3.2.2.2 Contrôle sans écoulement en présence d'un chemin.....	105
3.2.2.5 Contrôle sans écoulement en présence d'un chemin dans un fantôme vasculaire.....	107
3.3 Discussion	109
3.4 Conclusion	112
CHAPITRE 4. CONTRIBUTION, TRAVAUX FUTURS ET CONCLUSION.....	113
4.1 Contribution de la présente étude au projet MR-Sub.....	113
4.2 Travaux futurs	115

4.3 Conclusion	116
RÉFÉRENCES.....	118
ANNEXE I: ILLUSTRATION DES ENVIRONNEMENTS DE COMPILATION DES SÉQUENCES ET DES PROGRAMMES DE RECONSTRUCTION.....	124

TABLE DES FIGURES

Figure 1.1 : Cathéter ballon utilisé dans les procédures gastriques [35].....	6
Figure 1.2: Schéma bloc du fonctionnement du MR-Sub.....	10
Figure 1.3 : Schéma bloc du module roadmap.....	11
Figure 1.4: Exemple d'une image d'angiographie haute résolution [7].....	12
Figure 1.5 : Illustration de la mise en commun des piliers pour l'obtention d'une application autonome	20
Figure 1.6: Illustration des différents paramètres physiques en fonction des vaisseaux sanguins humain [10]	22
Figure 2.1: Vue d'ensemble des composants matériels du système IRM Avanto de Siemens ainsi que du système d'axes considéré	30
Figure 2.2: Illustration du déroulement d'une séquence de type écho de spin classique.	33
Figure 2.3: Déroulement typique d'une séquence sur l'Avanto	35
Figure 2.4: Illustration des différents processus et des fonctions se trouvant dans l'infrastructure ICE ainsi que des possibilités d'appel à ses fonctions en provenance de la séquence	41
Figure 2.5: Illustration de l'interface usager Syngo MR de Siemens	43
Figure 2.6: Illustration de la succession des blocs d'événements à partir de leur programmation jusqu'à leur exécution sur le scanner	48
Figure 2.7 : Principe d'interruption du MPCU et de son réveil par l'envoi d'un signal de 'wakeup' en provenance du bloc d'événements	49
Figure 2.8 : Ordre et exécution des événements lors d'une communication temps réel entre une séquence et un programme de reconstruction d'images.....	52
Figure 2.9: Vue d'ensemble du fonctionnement d'une séquence d'un point de vue structurel utilisant ou non les capacités temps réel de l'Avanto	55
Figure 2.10 : Illustration de la succession des modules dans une opération typique du MR-Sub.....	57
Figure 2.11: Illustration du complexe P-Q-R-S-T de l'onde cardiaque.....	61

Figure 2.12: Vue globale des interactions et de l'intégration des modules dans l'environnement de développement de l'Avanto	65
Figure 2.13 : Diagramme de classes de l'architecture logicielle du MR-Sub.....	74
Figure 2.14 : Diagramme de séquence de la préparation du système	78
Figure 2.15 : Diagramme de séquence de l'exécution du système	79
Figure 2.16 : Allure globale de la séquence temps réel pour la navigation du MR-Sub .	85
Figure 3.1 : Allure du fantôme vasculaire sous VolView et d'un ensemble de balises constituant un chemin à suivre pour le dispositif.....	94
Figure 3.2 : Exemple d'un fichier contenant un chemin de 10 balises (positions en mètres).....	95
Figure 3.3 : Montage utilisé pour les tests de contrôle en une dimension	97
Figure 3.4 Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 6cm sur l'axe x dans un environnement sans écoulement [51].....	100
Figure 3.5 Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 8cm sur l'axe z dans un environnement sans écoulement [51].....	101
Figure 3.6 Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 8cm sur l'axe x dans un environnement avec un écoulement constant [51]	102
Figure 3.7 : Allure de l'onde fémorale utilisée pour les tests de contrôle en écoulement pulsé [51].....	103
Figure 3.8 : Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 6cm sur l'axe x dans un environnement avec un écoulement pulsé représentant l'onde fémorale humain [51].	104
Figure 3.9 : Illustration du montage pour les tests de contrôle en deux dimensions	105
Figure 3.10 : Simulation et résultats expérimentaux du déplacement de la bille suivant un chemin pré déterminé en l'absence d'écoulement [51].....	107
Figure 3.11 : Illustration du fantôme vasculaire utilisé pour les tests de contrôle avec chemins	108

Figure 3.12: Allure du premier chemin testé pour la navigation dans le fantôme vasculaire.....	108
Figure 3.13 : Résultats expérimentaux du déplacement de la bille dans un fantôme vasculaire suivant un parcours illustré à la figure 3.12.....	109
Figure I.1: Illustration de l'interface de compilation et de simulation des séquences ...	124
Figure I.2 : Illustration de l'invite de commandes du ICE.....	125

INDEX DES TABLEAUX

Tableau 3.1 : Paramètres de la séquence de positionnement	99
Tableau 3.2 : Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction x, sans écoulement.....	99
Tableau 3.3 : Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction x, sans écoulement.....	100
Tableau 3.4 : Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction z avec un écoulement constant	101
Tableau 3.5 Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction x avec un écoulement pulsé	103
Tableau 3.6 : Balises constituant le chemin pour les résultats de la figure 3.10.....	106
Tableau 3.7 : Paramètres pour le test de contrôle en deux dimensions suivant un chemin pré déterminé.....	106
Tableau 3.8 : Paramètres pour le test de contrôle en deux dimensions suivant un chemin pré déterminé.....	108

LISTE DES SYMBOLES ET ABRÉVIATIONS

CAN	Convertisseur Analogique Numérique
CHUM	Centre Hospitalier de l'Université de Montréal
DICOM	Digital Imaging and Communications in Medicine
DLL	Dynamic Linked Library
DSP	Digital Signal Processor
FIFO	First In First Out
ICE	Image Construction Environment
IDEA	Integrated Development Environment for Applications
IGU	Interface Graphique Usager
IRM	Imagerie par Résonance Magnétique
MPCU	Measurement and Physiologic Control Unit
MRIR	Magnetic Resonance Image Reconstruction
MS-SET	Magnetic Signature by Selective Excitation Tracking
PID	Proportionnel Intégrateur Dérivatif
PMMA	Polyméthacrylate de méthyle
RF	Radio Fréquence
RSB	Rapport Signal sur Bruit
SDE	Sequence Development Environment
STL	Standard Template Library

INTRODUCTION

Les avancées technologiques dans le domaine du Génie Biomédical ont permis depuis plusieurs années d'améliorer grandement les moyens d'observation non invasifs ou minimalement invasifs chez des patients. Parmi l'ensemble de ces avancées, on note la présence de multiples moyens d'imagerie du corps humain permettant la pose de diagnostics plus rapide et plus sûre. Actuellement, dans le milieu hospitalier on retrouve plusieurs techniques d'imagerie médicale. Parmi les plus répandues on dénote les appareils à rayons X [42,29] les systèmes à ultrasons [29,28,23,56], les appareils de médecine nucléaire [29] et l'imagerie par résonance magnétique (IRM) [20, 57, 4, 27].

Sous un autre angle, les interventions chirurgicales se sont vues offrir, par l'entremise de ces modalités d'imagerie, de nouvelles approches exploitant la faculté de visionner en temps réel les différentes procédures opératoires tout en assurant aux patients une intervention sécuritaire et rapide. De plus en plus d'opérations minimalement invasives [13] sont désormais effectuées en conjonction avec une modalité d'imagerie avancée permettant au chirurgien et aux aides soignants de continuellement obtenir les signes vitaux ainsi que de connaître en tout temps la position des instruments utilisés pour l'intervention : c'est l'imagerie interventionnelle [6, 5] .

Bien que de plus en plus perfectionnées, ces interventions soutenues par l'imagerie médicale sont souvent limitées par les moyens chirurgicaux utilisés. En effet, les opérations minimalement invasives exploitant une modalité d'imagerie sont généralement des interventions guidées par cathéters [24]. Ces outils permettent, en passant par les vaisseaux sanguins, d'atteindre les organes vitaux ou des sites particuliers

nécessitant un traitement. Malheureusement, la taille ainsi que la flexibilité limitée des cathéters empêche parfois leur utilisation pour atteindre des vaisseaux plus petits ou trop sinueux. De telles régions du corps sont alors soit accessibles par des opérations beaucoup plus invasives, soit simplement inaccessibles avec les moyens actuels.

Par ailleurs, le couplage des instruments médicaux et des modalités d'imagerie n'est pas sans problème non plus. Par exemple, l'imagerie par résonance magnétique, offre de nombreux avantages pour le patient et pour le chirurgien en terme de résolution spatiale et en terme de contraste. De plus, cette méthode étant non ionisante, il n'y a aucun risque quant à l'exposition élevée aux matières radioactives, ce qui n'est pas le cas dans l'imagerie par rayons X où le patient et le chirurgien reçoivent, tout au long de l'intervention, des doses relativement élevées de rayonnement. Finalement, l'aspect tomographique de l'imagerie par résonance magnétique offre au chirurgien un moyen très précis pour la visualisation des tissus et des organes suivant n'importe quelle direction de l'espace. Cela dit, l'imagerie par résonance magnétique, en raison de son mode de fonctionnement, ne permet pas l'utilisation d'outils chirurgicaux métalliques, qu'ils soient paramagnétiques, diamagnétiques ou ferromagnétiques [19, 17, 39, 18, 15]. En effet, de tels matériaux créent des distorsions de susceptibilité magnétique rendant très difficile, voir impossible l'imagerie interventionnelle sur ce type d'appareils. Cela dit, certaines procédures chirurgicales de type interventionnelle par cathéter utilisent des techniques de positionnement actif pour observer le mouvement de celui-ci en temps réel lors de l'opération [21]. Par ailleurs, certains cathéters compatibles IRM existent sur le marché [24], mais il n'en demeure pas moins qu'ils se buttent aux mêmes problèmes

que les cathéters standards décrits plus haut en terme de rigidité et de dimension. Le projet Magnetic Resonance Submarine (MR-Sub) a vu le jour au Laboratoire de Nanorobotique de l'Ecole Polytechnique de Montréal en janvier 2002. Ce projet vise à utiliser les gradients magnétiques d'un appareil d'imagerie par résonance magnétique pour propulser et guider un dispositif endovasculaire autonome sans fil dans les vaisseaux sanguins humain. Comme nous le décrirons plus tard, ce dispositif pourrait être une solution potentielle aux problèmes de taille et de flexibilité des cathéters actuels en plus d'utiliser une modalité d'imagerie non ionisante pour le patient ou le personnel soignant. Le robot endovasculaire (dorénavant nommé MR-Sub) pourrait en effet permettre à certaines applications, encore aujourd'hui impossibles ou en phase de recherches intensives, de voir le jour. La libération ciblée de médicaments ainsi que la chirurgie minimalement invasive sans fil sont deux applications potentielles envisagées par un dispositif de ce genre. Bien qu'il n'existe, à notre connaissance, aucun projet similaire de guidage d'un dispositif endovasculaire par IRM, certains instruments comme l'appareil Niobe de la compagnie Stereotaxis [49] se rapportant au domaine valent d'être mentionnés.

Un projet comme le MR-Sub regroupe une multitude de défis qui se doivent d'être tous relevés indépendamment avant de pouvoir éventuellement obtenir une autonomie robotisée de la sorte. Un tel dispositif nécessite la mise en commun de plusieurs champs d'action qui passent par la propulsion magnétique, le contrôle, le positionnement et un système de gestion de navigation pour un guidage précis du MR-Sub. Ces quatre aspects forment les piliers fondateurs du MR-Sub. Bien entendu, pris

indépendamment bien que fonctionnels, ces piliers ne peuvent assurer un couplage adéquat au dispositif et se doivent donc d'être intégrés dans un environnement logiciel commun où chaque aspect a son rôle à jouer et assure ainsi l'accomplissement de la tâche auquel est dédié le dispositif. C'est de cette intime et complexe intégration qu'il sera question dans ce mémoire

Le premier chapitre qui est abordé décrit plus en détails le projet MR-Sub en plus de cibler précisément le contexte de ce mémoire. Les différents constituants du projet que sont la propulsion, le contrôle et le positionnement et le système de navigation (nommé dorénavant 'Roadmap') seront également présentés pour tout d'abord permettre une meilleure compréhension des constituants du projet en plus d'illustrer les défis à relever en terme d'intégration fonctionnelle. Un second chapitre présentera les requis du projet ainsi que l'architecture retenue pour l'intégration des différents piliers du MR-Sub. Point central de ce mémoire, ce chapitre décrira le modèle logiciel développé ainsi que son couplage avec l'architecture déjà en place de l'IRM devant permettre une parfaite symbiose à travers la création d'une séquence d'acquisition spécialement conçue pour le guidage temps réel du dispositif dans un environnement vasculaire réel. Un dernier chapitre traitera de la méthodologie et des résultats obtenus afin de vérifier et quantifier dans un premier temps, le fonctionnement des différents constituants pris indépendamment et dans un second temps, les mêmes piliers intimement intégrés dans l'architecture logicielle retenue.

CHAPITRE 1. LE PROJET MR-SUB

Introduction

Le projet MR-Sub a connu, au cours de ces quatre dernières années, une croissance exemplaire, relevant plusieurs défis comme celui de la validation de la méthode de propulsion magnétique [33] par IRM ainsi que celui du positionnement d'objets ferromagnétiques en IRM [15]. Il s'est également vu attribuer deux importantes subventions. La première assure le support financier au projet et lui a été remis par le conseil de recherches en sciences naturelles et en génie du Canada (CRSNG). La seconde provient de la fondation Canadienne pour l'innovation (FCI) et devra permettre l'achat et l'installation d'un appareil d'imagerie par résonance magnétique à l'École Polytechnique de Montréal. Par ailleurs, les travaux antérieurs sur la propulsion magnétique par IRM et sur les techniques de positionnement de ferromagnétiques en IRM ont tous deux fait l'objet d'une demande de brevet.

1.1 Problématique actuelle

Un grand nombre d'interventions chirurgicales requièrent une modalité d'imagerie médicale pour être en mesure d'assister une procédure par cathéters. Ceux-ci étant utilisés pour pouvoir atteindre l'endroit cible de l'opération. Les traitements chirurgicaux coronaires les plus répandus tels que la procédure d'angioplastie, de placement d'endoprothèses vasculaires ('stent'), d'embolisation ou de libérations ciblées de médicaments sont toutes des méthodes par cathétérisation.

La cathérisation consiste à introduire un tube, plus ou moins rigide selon l'application, dans une cavité du corps après avoir effectué une mince ouverture pour atteindre celle-ci. Il existe une multitude de cathéters sur le marché selon l'utilisation que la chirurgie nécessite. Certains comportent des ballons gonflables à leur extrémité qui sont par exemple utilisés pour altérer des vaisseaux obstrués dans le cas d'artériosclérose alors que d'autres permettent le passage de guides métalliques pour la manipulation d'outils à l'embout du cathéter. La figure 1.1 suivante illustre un cathéter ballon utilisé dans les chirurgies gastriques pour augmenter la taille de la poche gastrique.

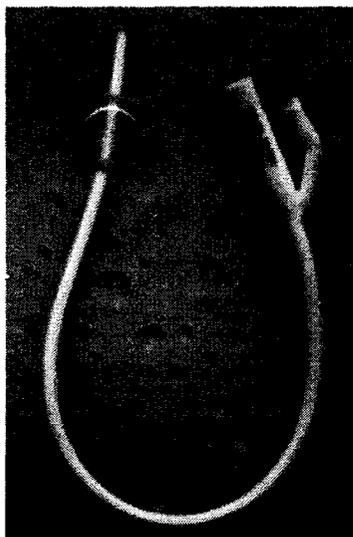


Figure 1.1 : Cathéter ballon utilisé dans les procédures gastriques [35]

Bien que la plupart de ces procédures se déroulent sans aucun problème grâce à l'utilisation de cathéters très fiables, il n'en demeure pas moins que certaines chirurgies sont beaucoup plus risquées, voir dangereuses pour le patient en raison des inconvénients qui accompagnent l'utilisation de cathéters. La flexibilité ainsi que la

taille limitée de ces instruments rendent parfois impossible l'accès à des vaisseaux de petites tailles ou des régions sinueuses du système vasculaire. Les plus petits cathéters ont des diamètres de l'ordre du millimètre alors que les capillaires ont une épaisseur moyenne de quelques micromètres. Il existe donc une certaine gamme de vaisseaux et donc de sites physiologiques qui sont simplement inatteignables par les cathéters actuels en raison de leurs tailles trop importantes. Par ailleurs, certaines sections vasculaires sont trop sinueuses pour pouvoir y insérer et y faire circuler des instruments comme le cathéter. En effet, lors du déplacement du cathéter dans le système vasculaire, celui-ci doit être assez flexible pour pouvoir accommoder l'aspect sinueux des vaisseaux tout en évitant de dégrader ceux-ci. Une trop grande distance à parcourir couplée avec une importante sinuosité peut entraîner un important frottement entre le cathéter et les cellules endothéliales.

Dans le cas de libération ciblée de médicaments, ces limites peuvent réduire de manière significative l'efficacité du traitement en raison de la difficulté d'atteinte du site ciblé. Dans le cas des cancers par exemple, il est préférable de pouvoir se rapprocher le plus près possible du site tumoral en raison du danger que représente le médicament libéré pour les cellules saines aux alentours des cellules tumorales. Ces médicaments étant généralement nocifs pour l'ensemble des cellules, il y a tout intérêt à larguer ce principe actif dans une région la plus rapprochée possible du site cancéreux pour éviter tous dommages collatéraux aux autres cellules saines du corps.

Ces divers problèmes liés essentiellement à l'utilisation d'instruments médicaux pouvant provoquer des complications lors d'interventions ou simplement limiter les

types de traitement accessibles aux patients exposent un besoin pour une technologie mieux adaptée et complémentaire visant à réduire les risques associés à l'utilisation d'instruments comme le cathéter dans certaines procédures médicales ou chez certains patients plus à risque.

La nanomédecine tente de répondre à une partie du présent problème. En effet, cette discipline est en pleine explosion dans le domaine de la recherche médicale et a déjà vu quelques applications se réaliser [58, 9, 12]. La nanomédecine vise essentiellement l'intégration des nanotechnologies dans les applications médicales. Le domaine le plus étudié est sans contredit les nouvelles méthodes d'administration de médicaments sous forme de nanoparticules visant à augmenter l'efficacité et la précision des traitements disponibles aujourd'hui tout en réduisant les doses administrées au patient.

1.2 Objectifs du projet

Le principe du projet MR-Sub est de proposer un moyen alternatif de guidage et de navigation endovasculaire pouvant être utilisé dans plusieurs cas de figure. Constitué d'un corps ferromagnétique, ce dispositif est guidé par les gradients magnétiques d'un système IRM typique que l'on retrouve dans la plupart des environnements hospitaliers ou cliniques. Il pourrait donc aussi bien permettre l'atteinte de régions éloignées dans les vaisseaux sanguins de trop petites tailles pour être atteignables par cathéter que permettre le transport et la libération de médicaments dans des endroits difficiles ou impossible d'accès par les méthodes traditionnelles. Il pourrait également être utilisé comme concentrateur thermique dans des traitements par hyperthermie ou encore agir

comme agent embolisant. Une autre application potentielle de ce dispositif consisterait à s'en servir comme attracteur magnétique pour de plus petites particules elles-mêmes magnétiques.

En somme, le MR-Sub ne se veut pas un remplacement d'une technologie déjà implantée et bien rodée que sont les cathéters mais plutôt offrir un complément technologique pour faciliter les applications déjà existantes ou même permettre l'essor de nouvelles applications encore aujourd'hui impossibles ou faisant face à de multiples contraintes d'un point de vue technique. La figure 1.2 suivante expose le schéma bloc anticipé du fonctionnement typique du MR-Sub.

Pour réaliser un tel projet, il convient de posséder une plateforme intimement liée à l'appareil de résonance magnétique permettant la navigation d'un tel dispositif. Une architecture informatique doit permettre le bon fonctionnement lors de son déplacement et doit assurer l'atteinte des objectifs du MR-Sub. C'est de cette plateforme informatique qu'il est question dans ce mémoire. La mise en commun d'un moyen de propulsion, de positionnement et de contrôle ainsi que l'élaboration des classes constituantes de l'architecture informatique sont décrits dans les lignes qui suivent. Des contraintes d'ordre temporel très strictes se doivent d'être respectées pour être en mesure d'obtenir un système de navigation permettant le contrôle d'un tel dispositif dans un environnement complexe qu'est le système vasculaire humain.

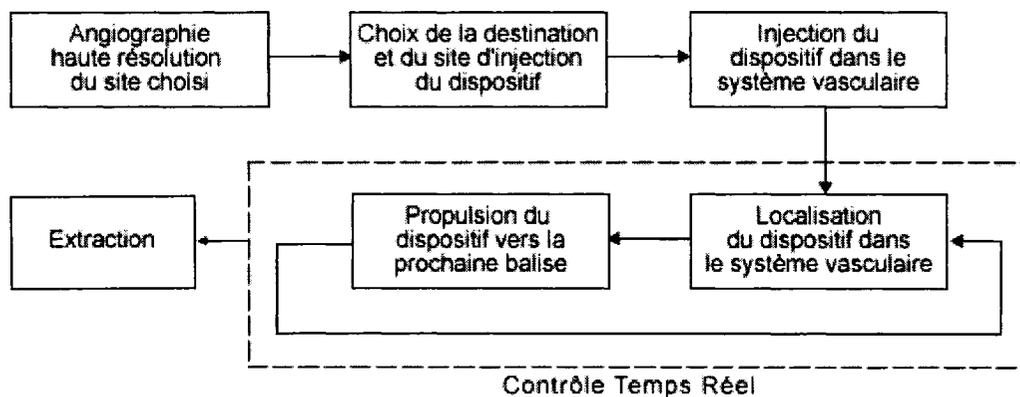


Figure 1.2: Schéma bloc du fonctionnement du MR-Sub

1.3 Les piliers du projet

Dans cette sous-section nous abordons plus en détails les requis fonctionnels du projet en plus de décrire leur mode de fonctionnement en vue de l'analyse et de la description de leur intégration au chapitre 2. Sur la figure 1.2, on peut déduire quatre requis fonctionnels principaux: le système de routage que l'on nommera 'Roadmap', le système de propulsion magnétique, le système de positionnement magnétique et le système de contrôle pour la navigation du dispositif.

L'ensemble des piliers présentés ici, constituera les futurs modules qui seront intégrés dans une architecture logicielle en vue d'automatiser l'ensemble des tâches liées à la navigation et à la rencontre des requis du projet.

1.3.1 Roadmap

Premier pilier fonctionnel du projet MR-Sub, la feuille de route ou 'roadmap', doit assurer l'obtention, d'une façon manuelle ou automatisée, d'un ou plusieurs chemins valides que devra suivre le dispositif dans l'environnement en vue d'atteindre sa destination finale. Le concept final de ce module est donc de permettre, à partir d'une

image angiographique illustrant en détails l'ensemble des vaisseaux dans une région donnée de l'hôte, à un usager de choisir un point d'insertion du dispositif ainsi qu'un point terminal dans le système vasculaire qui constituera la destination du MR-Sub. La génération des points intermédiaires nommés balises (ou 'waypoints') pourra se faire soit de façon automatique par l'entremise d'un algorithme spécialisé de recherche d'un meilleur chemin, soit par un choix manuel de l'utilisateur. La figure 1.3 illustre le schéma bloc de l'utilisation du module roadmap.

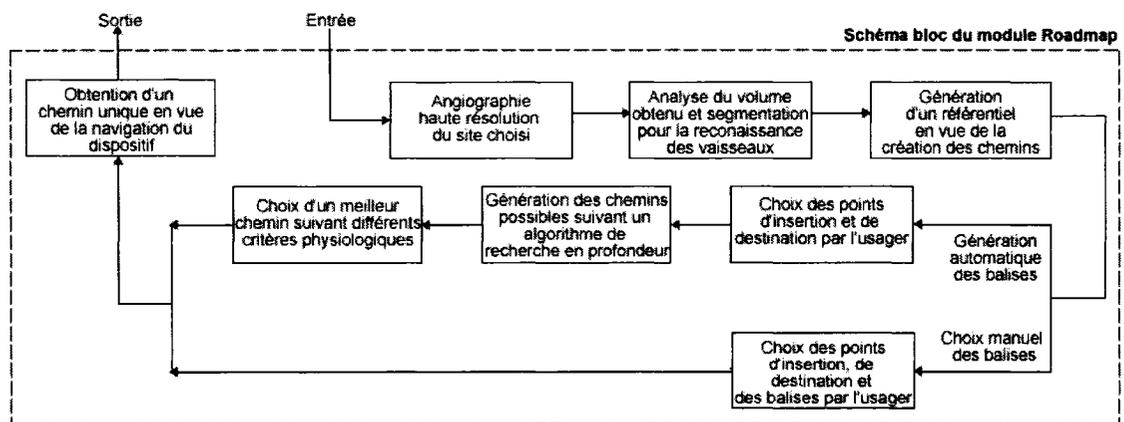


Figure 1.3 : Schéma bloc du module roadmap

Dans le cas d'une procédure automatique de génération des points intermédiaires d'un chemin potentiel pour le dispositif, il est envisageable, à l'issue de l'utilisation des algorithmes de meilleurs chemins retenus, d'obtenir plusieurs chemins applicables. Bien entendu, il conviendra alors d'effectuer une sélection d'un chemin optimal que l'on conservera en vue de la navigation du dispositif. Ainsi, bien que plusieurs chemins puissent exister pour mener le MR-Sub d'un point initial A à un point final B, nous ne retiendrons et n'utiliserons qu'un seul de ceux-ci en fonction de plusieurs critères. Ceux-ci étant d'ordre principalement physiologiques comme la taille des vaisseaux parcourus,

le débit sanguin maximal ou la sinuosité des vaisseaux qui influent tous sur les paramètres de contrôle du dispositif en terme de contraintes physiques ou perturbations externes à considérer lors de la navigation. Il est également important de rappeler qu'en raison des perturbations magnétiques que cause le dispositif une fois placé dans le tunnel de l'appareil d'imagerie par résonance magnétique rendant la prise d'image standard excessivement bruitée, les étapes du module roadmap décrites à la figure 1.3 ci-haut se doivent d'être faites avant l'injection du MR-Sub dans le milieu hôte.

L'angiographie effectuée en tout début de procédure permet d'obtenir une image très détaillée du réseau vasculaire sanguin. Un exemple d'image obtenue suite à une telle procédure d'angiographie sur un appareil MedSpec de 4 Tesla de la compagnie Bruker Biospin est illustrée à la figure 1.4 ci-dessous.

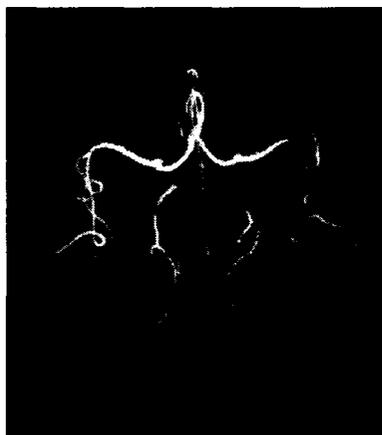


Figure 1.4: Exemple d'une image d'angiographie haute résolution [7]

A l'issue de cette angiographie, un volume ou une succession de tranches sont obtenus et sont exportables en format DICOM [37, 41]. Ce format, spécialement conçu pour le domaine de l'imagerie médical, facilite la distribution, le visionnement et le traitement des images obtenues à partir de différentes modalités d'imagerie ayant des

modes de fonctionnement très différents. Le format DICOM est généralement le standard pour la circulation de données en provenance d'une modalité d'imagerie médicale. Une multitude de logiciels gratuits ou à licences payantes existent sur le marché pour traiter et visualiser les images de ce format.

1.3.1.2 Segmentation en vue de la reconnaissance des vaisseaux

Avant de pouvoir générer une quelconque forme de chemin dans les vaisseaux sanguins en vue de la navigation du dispositif, il est préférable, en premier lieu, d'être en mesure de reconnaître, à partir de l'image angiographique obtenue, les zones de l'image représentant le système vasculaire et ses parois des autres zones qui constituent le milieu avoisinant. Cette étape d'extraction des données permettra d'obtenir une image binaire, constituée de pixels d'intensité maximale ou nulle distinguant les zones acceptées des zones rejetées. Dans le cadre du MR-Sub, ce clivage permettra de connaître les coordonnées des points recouvrant uniquement les vaisseaux et ainsi connaître les régions pertinentes pouvant être utilisées pour la création des chemins dans les étapes à venir. La famille des méthodes permettant d'effectuer une telle distinction sur une image (noir et blanc ou même couleur) se nomme segmentation. Il existe une multitude de méthodes de segmentation basées sur plusieurs critères de l'image ou du contexte [45, 36, 55, 3, 11, 25, 2, 16, 44]. Certaines de ces méthodes peuvent aussi bien s'appliquer sur une image standard 2D que dans un volume 3D de points. L'étape de segmentation est une des plus critiques de tout le module roadmap car sa qualité va directement affecter celle de la création et la gestion des chemins. Une segmentation trop grossière risque de diminuer ou d'augmenter le nombre de points considérés comme faisant partie

de la région vasculaire que l'on désire extraire pouvant mener à des chemins inexistantes réellement ou simplement plus restreint qu'ils ne le sont en réalité.

1.3.1.3 Création des balises

Comme l'illustre la figure 1.3, cette étape peut s'effectuer de deux façons : soit automatiquement une fois les points de départ et d'arrivée du dispositif sélectionnés dans le volume, soit de façon manuelle par l'entremise d'un usager qui impose l'ensemble des balises pour le chemin.

1.3.1.3.1 Création manuelle des balises

Dans le cas manuel, l'utilisateur va donc choisir, à partir du volume qui lui est présenté, l'ensemble des points de passage qui, mis ensemble, constitueront le chemin à suivre pour le MR-Sub. L'intérêt de la présence d'une telle méthode est avant tout relié au développement du projet. En effet, en vue des tests effectués pour caractériser l'architecture et l'intégration des modules de propulsion, de positionnement et de contrôle du projet, il était essentiel de posséder une méthode simple et rapide de génération de chemins. La possibilité de spécifier un chemin à partir d'un volume 3D pour ensuite en exporter les données assure une facilité accrue lors des premiers tests sans avoir à implémenter une méthode automatique de création de chemins qui est relativement fastidieuse et qui n'est pas nécessaire pour les premiers essais d'intégration.

1.3.1.3.2 Création automatique des balises

La création automatique des balises est beaucoup plus raffinée technologiquement que la méthode manuelle et offre une assurance en terme de précision et d'optimisation du chemin choisi en plus de limiter les éventuelles erreurs

humaines lors de la sélection des balises. L'utilisateur n'a qu'à choisir le point d'injection du dispositif et la destination désirée dans le volume du système vasculaire qu'il observe. Une fois ces points validés, une méthode de recherche de chemin débute en vue de déterminer les différents chemins possibles pouvant relier les deux conditions initiales proposées par l'utilisateur. Une fois ces chemins choisis ou trouvés suivant le cas, un choix du meilleur chemin est alors effectué devant mener à l'obtention d'un parcours paramétré suivant plusieurs critères d'ordre physiologique. On retrouve entre autre parmi ces critères : les débits sanguins dans les différentes régions, la taille et l'épaisseur des vaisseaux traversés, la sinuosité de ceux-ci et la complexité global du parcours. Des paramètres d'ordre structurels du dispositif comme son volume, sa masse et sa forme sont également à considérer lors du choix du meilleur chemin. D'autres paramètres d'ordre techniques comme l'amplitude et la durée maximale des gradients magnétiques de l'IRM considéré sont également à prendre en compte en raison de la limitation de la force de propulsion que ces deux paramètres imposent.

1.3.2 Propulsion magnétique

Seconde pierre angulaire du projet MR-Sub, le module de propulsion magnétique est fondé sur les travaux antérieurs au Laboratoire de Nanorobotique [33, 32, 34, 31, 30]. Il se penche sur la faisabilité de la propulsion magnétique à l'aide des gradients magnétiques d'un système d'imagerie par résonance magnétique. Ce module prend en charge le déplacement du dispositif dans son environnement. Comme exposé plus tôt, le MR-Sub est utilisé en conjonction avec un appareil d'imagerie par résonance magnétique. Ce système lui fournit la force extérieure sous la forme de gradients

magnétiques en vue de sa propulsion. Le module de propulsion du MR-Sub consiste donc en l'application des gradients magnétiques dans une direction donnée de l'espace et avec une amplitude donnée qui produit une force sur le dispositif. Il faut toutefois savoir que les gradients magnétiques que l'on retrouve en IRM sont souvent limités en amplitude et en temps d'application. Il n'est donc pas toujours possible d'appliquer un gradient d'une amplitude donnée pendant une durée indéfinie car cette durée dépend des amplitudes de tous les gradients appliqués et répétés dans la séquence IRM. Ces limitations proviennent essentiellement des contraintes techniques au niveau des amplificateurs des bobines de gradient que l'on retrouve sur tout IRM. Il n'est donc pas possible de propulser le dispositif dans une direction donnée avec une force excédant les capacités des bobines de gradient tout comme il n'est pas possible de le propulser au delà d'un certain temps t sans qu'il n'y ait interruption des bobines de gradient en raison d'une dissipation thermique trop importante. Ces limites en terme d'amplitude de gradients et en terme de temps d'application se doivent d'être prises en compte lors du contrôle du dispositif afin de permettre, dans les contraintes du système, d'atteindre la destination fixée.

1.3.3 Positionnement magnétique

Une fois le chemin à suivre pour le dispositif obtenu et la propulsion magnétique fonctionnelle, il convient ensuite de pouvoir, à tout moment, positionner le dispositif ferromagnétique dans son environnement. Une analyse rapide du problème pourrait proposer l'utilisation des images de densité protonique classique obtenues par les séquences d'imagerie standard de l'IRM. Cependant, pour deux raisons essentielles,

cette solution n'est pas viable. Tout d'abord, un tel dispositif se doit d'être contrôlé et navigué avec des contraintes de temps excessivement restreintes en vue d'obtenir un temps de réaction 'temps réel' dans l'ordre de quelques dizaines de millisecondes. Un délai trop important lors de la navigation du dispositif pourrait rendre celle-ci inutilisable dans un contexte *in vivo* ou lorsque plusieurs perturbations viennent influencer le contrôle du dispositif. La création d'une image, même avec les techniques d'acquisition rapide comme les séquences d'IRM 'Echo Planar Imaging' (EPI) [20] ne peuvent produire des images ayant des durées de l'ordre de 50 à 100 millisecondes. S'il fallait utiliser une telle image pour localiser le dispositif, il faudrait également considérer le temps supplémentaire au traitement et à l'analyse de ces images à chaque déplacement du MR-Sub en vue de le positionner. Cette tâche alourdirait grandement le temps de réponse du dispositif et ne garantirait pas un contrôle acceptable dans des délais très courts. De plus, la présence d'une particule magnétique dans une région imagée avec un système d'imagerie par résonance magnétique provoque des distorsions en raison de la différence de susceptibilité magnétique entre la particule métallique et son environnement [19, 17, 39, 18, 15, 48, 50]. Ce phénomène est grandement amplifié avec la catégorie des ferromagnétiques.

Pour ces raisons, il convient donc de trouver une méthode plus rapide et plus précise pour localiser un dispositif ferromagnétique comme le MR-Sub. La solution décrite ici est basée sur les travaux antérieurs réalisés au Laboratoire de Nanorobotique qui consistaient à étudier les différentes avenues possibles pour localiser un corps ferromagnétique en IRM [15, 14].

1.3.3.1 MS-SET

La méthode de signature magnétique par excitation sélective [1] (Magnetic Signature by Selective Excitation Tracking ou 'MS-SET') consiste à exciter uniquement les 'spins' se trouvant à une certaine distance du centre de la bille et qui ont une fréquence de précession qui est fonction de leur distance du centre de la bille en raison des lignes du champ magnétique produit par la bille ferromagnétique. Le lecteur désirent en apprendre davantage sur cette technique peut se rapporter à [14].

Cette méthode de localisation est incluse dans une séquence IRM spécialement développée. Les parties d'analyse et de traitement des données sont effectuées sur la même machine que celle habituellement utilisée pour la reconstruction d'images lors des séquences d'imagerie standard. Il est possible de positionner le dispositif avec un taux de rafraîchissement d'environ 83 Hz par axe. Une localisation 3D peut donc se faire à une fréquence de 27 Hz.

1.3.4 Contrôle

Dernier pilier du projet MR-Sub d'un point de vue de l'intégration, le module contrôle est en réalité le chef d'orchestre de la navigation du dispositif. Il occupe une place centrale dans l'intégration et dans l'architecture présentées au prochain chapitre. Son rôle est d'asservir le déplacement du MR-Sub tout en respectant le plus possible le chemin et les balises choisis au départ. En se basant sur la position du dispositif, le contrôleur détermine la commande idéale en terme d'amplitude et de direction des gradients magnétiques pour la propulsion du MR-Sub. La modélisation et les

développements présentés ont été réalisés par l'équipe du projet au Laboratoire de Nanorobotique [51, 53, 54, 52].

Les travaux de modélisation ont permis la mise en place d'un premier contrôleur de type PID (Proportionnel Intégrateur Dérivatif) pour le MR-Sub [53]. La raison pour laquelle un contrôleur simple comme le PID a été choisi fut la simplicité de mise en oeuvre dans le but de rapidement évaluer l'impact des forces et des contraintes considérées dans la réponse du système.

Le module 'contrôle' sera placé au coeur de l'architecture logicielle et constituera le chef d'orchestre de la navigation du MR-Sub.

1.4 Objectifs et requis de la présente étude

L'ensemble des piliers étant présenté, une courte introduction des objectifs d'intégration est présentée afin de résumer le travail à effectuer en vue de sa description détaillée au prochain chapitre. Par ailleurs, une liste des requis est également présentée pour cadrer les attentes que devra remplir l'architecture développée.

1.4.1 Vue éclatée de l'intégration

Pour permettre au dispositif de naviguer dans un environnement donné, il est nécessaire de rassembler les méthodes de propulsion, de positionnement et de contrôle autour d'une application comportant un gestionnaire qui se chargera des communications entre les différents piliers en plus de proposer une interface simplifiée pour l'utilisation du système de navigation. La figure 1.5 ci-dessous résume

graphiquement la mise en commun des éléments et le travail qui a été effectué et qui est plus longuement décrit au prochain chapitre.

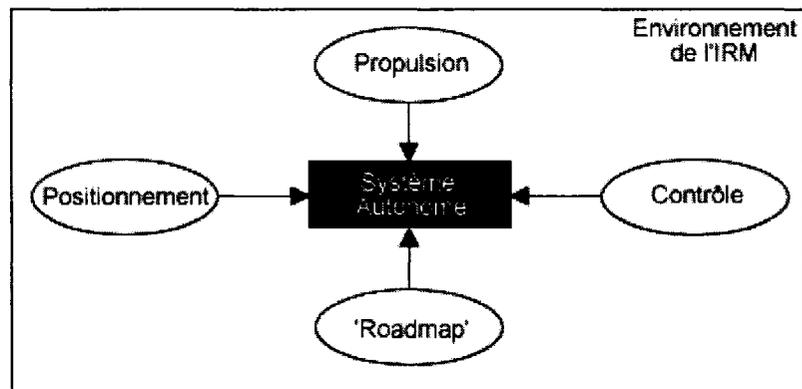


Figure 1.5 : Illustration de la mise en commun des piliers pour l'obtention d'une application autonome

Cette mise en commun doit se faire dans le respect des contraintes temporelles qui entrent en jeu lors d'une communication temps réel. L'intégration et la mise en place de l'architecture doit également permettre une compatibilité entre les appareils d'imagerie par résonance magnétique cliniques de la compagnie Siemens qui permettent une interaction temps réel des événements.

1.4.2 Requis

Le MR-Sub doit avant tout permettre la navigation et le transport éventuel de médicaments entre un point d'insertion préalablement fixé et un point de destination également fixé dans le système vasculaire humain. Une fois atteint, le dispositif devra également soit être en mesure d'effectuer le chemin inverse jusqu'à la région d'insertion pour son extraction ou jusqu'à une région où il pourra être aisément récupéré.

Sa taille finale dépendra de plusieurs facteurs comme la taille du site de destination ainsi que du chemin emprunté. La dimension des différents vaisseaux du système vasculaire varie énormément au fur et à mesure que les ramifications vasculaires s'étendent. Plus on s'éloigne du coeur, plus la vitesse de l'écoulement sanguin diminue également. Celle-ci passe de plusieurs dizaines de cm/s au niveau de l'aorte à quelques cm/s au niveau des artérioles [43]. La figure 1.6 tirée de [10] illustre les différences entre les types de vaisseaux sanguins en terme de vitesse d'écoulement, d'aire de la section des vaisseaux, de pression moyenne et de pourcentage du volume sanguin total. Ainsi, la taille du MR-Sub dépendra du site de destination. Elle dépendra également du chemin à parcourir et des différentes vitesses sanguines pulsées rencontrées afin d'assurer un contrôle continu du dispositif tout au long de son trajet.

Sa forme est aussi un paramètre décisif ayant des conséquences non négligeables sur sa navigation. Bien que pour l'instant et dans le cadre de ce mémoire, celle-ci sera considérée sphérique, il conviendra d'étudier l'impact des différentes formes possibles sur la navigation et le contrôle du MR-Sub. La résistance à l'écoulement est un des paramètres importants du choix de la forme du dispositif. Celle-ci peut, avec les mêmes paramètres de puissance fournie par l'appareil de résonance magnétique, produire une différence notable en terme de vitesse de déplacement et de maniabilité.

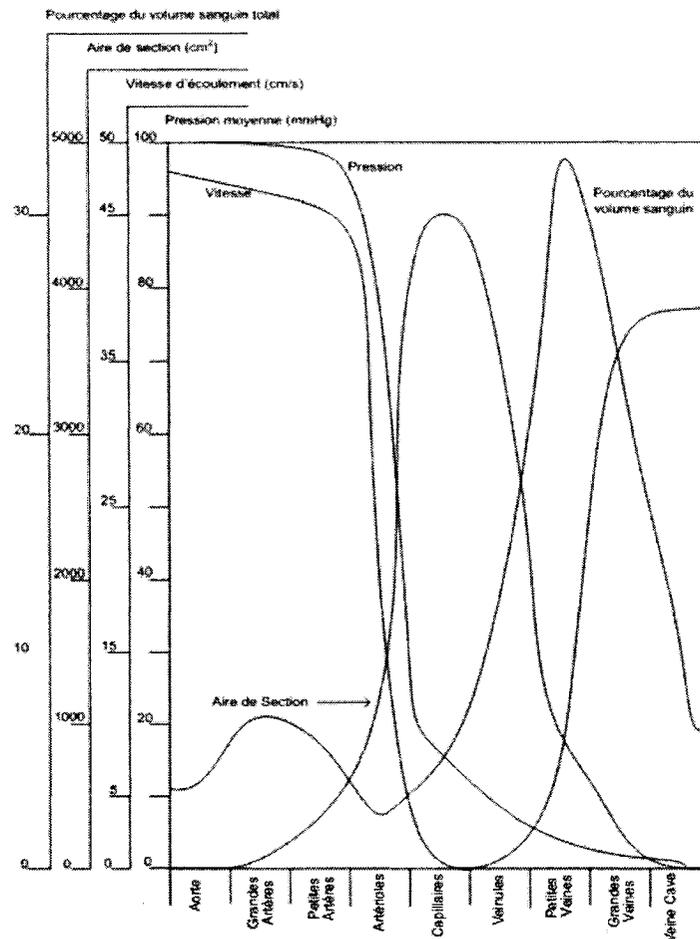


Figure 1.6: Illustration des différents paramètres physiques en fonction des vaisseaux sanguins humain [10]

Dernier requis d'ordre physique du dispositif, la biocompatibilité de celui-ci devra également permettre d'assurer une navigation sans danger pour le patient et sans complications pour le dispositif; complications que pourrait provoquer une réaction immunitaire majeure suite à l'insertion du dispositif.

En plus de ces requis physiques, il faut s'assurer des possibilités d'interactions temps réel de l'IRM considéré. En effet, la propulsion s'effectuant par application de gradients magnétiques dans une certaine direction avec une certaine amplitude, ceux-ci

doivent pouvoir être modifiés aux bons grés de l'architecture dans des délais très courts pour être en mesure de réagir rapidement sur la navigation du dispositif. Ce requis d'ordre structurel est sans contredit le plus important en termes de faisabilité d'un projet comme celui du MR-Sub. L'absence d'une telle possibilité de réaction rapide sur les gradients appliqués empêcherait toute forme de contrôle en temps réel du dispositif, rendant son usage et sa navigation inutile. Comme il sera décrit sous peu, ces contraintes de temps de l'ordre de quelques millisecondes sont essentielles et primordiales au bon fonctionnement du MR-Sub. Pour ces raisons et pour des raisons d'ordre d'accessibilité aux ressources, l'architecture présentée ici ainsi que les tests effectués et présentés au chapitre 3 ont été réalisés sur un appareil d'imagerie par résonance magnétique Avanto 1.5 T de la compagnie Siemens permettant une application temps réel des gradients ainsi que d'autres éléments importants à la navigation qui seront également présentés.

Conclusion

Dans ce chapitre, nous avons présenté les différents aspects du projet MR-Sub aussi bien en terme d'objectifs qu'en termes de requis fonctionnels. Les différents piliers nécessaires au fonctionnement du dispositif ont été présentés en vue de leur intégration dans une architecture logicielle visant à assurer l'interaction de ces modules pour obtenir un système de navigation autonome. Le chapitre suivant traite précisément de cette intégration et de la description de l'environnement logiciel de l'IRM dans lequel sera introduit cette architecture.

CHAPITRE 2. ARCHITECTURE LOGICIELLE ET INTÉGRATION

Introduction

Dans ce chapitre, nous présentons la conception et l'intégration de la plateforme logicielle devant permettre une navigation autonome du MR-Sub. Dans un premier temps, les contraintes spécifiques du projet sont énoncées de manière à fixer les limites fonctionnelles d'un tel dispositif. Dans un second temps, l'environnement informatique de l'IRM choisi est détaillé afin de mieux saisir le fonctionnement global de celui-ci en plus d'avoir une meilleure idée du matériel utilisé pour réaliser un tel projet. Nous présentons ensuite l'environnement de développement dans lequel est posée l'architecture logicielle. En effet, il convient de greffer adéquatement l'architecture développée avec celle déjà présente sur l'appareil d'imagerie par résonance magnétique. Dans cette même section, nous introduisons également l'emplacement et l'interaction des différents modules introduits au chapitre 1. Par la suite, nous examinerons le fonctionnement d'une séquence IRM typique et présenterons la séquence temps réel développée pour la navigation du dispositif. L'allure et la description des différentes classes établies seront ensuite présentées pour finalement donner un aperçu du fonctionnement du système à travers des diagrammes de séquence.

2.1 Contraintes du projet

Avant de développer une architecture logicielle pour le fonctionnement d'un dispositif endovasculaire comme le MR-Sub, il convient de préalablement rassembler la

liste des contraintes que l'on sait pouvoir causer des difficultés lors de l'avancement du projet et pour lesquelles des dispositions sont prises lors de l'édification de l'architecture. Certaines d'entre elles sont cependant sans impact direct sur l'intégration des composantes mais nécessitent tout de même une première réflexion pour le développement futur du projet. Une fois cette étape effectuée, il est alors possible de concevoir l'interaction et l'intégration des modules nécessaires au bon fonctionnement du MR-Sub.

Les contraintes sont essentiellement les aspects du projet qui se doivent d'être présents et qui, en quelque sorte, limitent les avenues ou les solutions face aux requis émis. Dans le cadre du projet MR-Sub, ces contraintes sont d'ordre physique en ce qui a trait à la composition du dispositif, d'ordre physiologique en raison de l'environnement complexe dans lequel le dispositif se doit d'évoluer et d'ordre structurel en ce qui a trait aux limitations externes de l'environnement et des moyens utilisés.

D'un point de vue structurel, la contrainte majeure du dispositif réside dans sa composition. En effet, c'est en raison de sa nature ferromagnétique à haute magnétisation que l'on peut utiliser un appareil comme l'IRM et ses gradients magnétiques pour propulser le MR-Sub. Cependant, c'est également en raison de sa nature ferromagnétique que l'on ne peut utiliser une séquence d'imagerie standard pour sa localisation, ce qui a amené une étude approfondie des méthodes de positionnement d'un ferromagnétique en IRM [14]. Sa nature ferromagnétique hautement toxique pour le corps est également un frein majeur en terme de biocompatibilité naturelle. Pour cette

raison, il a fallu également étudier diverses avenues en terme de recouvrements biocompatibles en vue d'applications *in vivo*.

Les contraintes physiologiques sont nombreuses et constituent la raison pour laquelle un contrôle temps réel est requis. Comme présenté à la section 2.1.1, le système vasculaire qui constitue l'environnement de navigation du MR-Sub est très complexe. L'aspect pulsé de l'écoulement sanguin produit une variation continue de la force de traînée que subit le dispositif au cours de son déplacement. Il est cependant concevable avec les moyens actuels sur les appareils d'imagerie par résonance magnétique de prévoir et de connaître la fréquence cardiaque à l'aide d'un électrocardiogramme (ECG) relié au système pour ensuite compenser cette perturbation. Cela dit, le débit sanguin ne peut quant à lui être mesuré précisément dans l'ensemble des vaisseaux et ne peut qu'être estimé. De plus, l'aspect souvent sinueux et restreint que prennent les vaisseaux sanguins dans le système vasculaire implique une navigation complexe en terme d'orientation et de propulsion. Une dernière contrainte physiologique d'envergure est la dilatation et le déplacement des vaisseaux en raison de la respiration du patient. Ces perturbations se doivent d'être continuellement compensées par le contrôleur. La déformation continue des vaisseaux produit des incohérences dans le suivi du chemin établi. De telles variations peuvent entraîner des déplacements inconsistants alors qu'initialement, ceux-ci étaient prévus. On peut alors concevoir des scénarios où le déplacement entre deux balises initialement anticipé se répercute en un déplacement erroné voire à risque lors de l'opération. Bien que certaines méthodes de recalage permettent de compenser en temps réel [59, 38, 8, 22, 46]le déplacement de tissus et

d'organes, aucune n'est réellement sans faille et aucune ne permet de compenser précisément l'ensemble des déformations lors d'opérations interventionnelles. Ce problème de recalage est un domaine très étudié en recherche et n'offre pas encore de solutions pour tous les cas de figure de déformation.

D'un point de vue structurel, la contrainte majeure réside dans la limitation de la force maximale produite par les bobines de gradients de l'appareil de résonance magnétique. En effet, sur l'IRM Avanto 1.5 T de la compagnie Siemens considéré, l'amplitude maximale des gradients se situe à 40 mT/m. Selon [34], la force créée par de telles bobines de gradient sur une bille composée d'acier et de carbone mesurant 3.14 mm de diamètre et ayant une magnétisation de $M_{sat} = 1.375680 \times 10^6$ A/m est insuffisante pour faire léviter celle-ci. Cependant, une solution actuellement en cours consiste à modifier les bobines de gradients d'imagerie par des bobines plus puissantes qui permettraient une telle lévitation en plus d'une propulsion à contre courant sanguin ce qui est également très difficile voire impossible avec les bobines de gradient actuelles sur l'Avanto.

En terme d'architecture logicielle, la contrainte majeure est le respect du délai d'opération devant permettre le fonctionnement du contrôleur sans que celui-ci ne devienne instable ainsi que le respect des délais de communication pour la réaction en temps réel des opérations. La contrainte secondaire réside quant à elle dans la difficulté de développement d'un tel système avec un accès restreint à la ressource principale qu'est l'IRM. L'architecture présentée s'est donc édifiée en grande partie en l'absence

de l'IRM qui n'était utilisé que pour valider les travaux effectués préalablement impliquant un développement rapide mais efficace.

2.2 Description de l'environnement informatique du système d'imagerie par résonance magnétique retenu

Avant de décrire l'environnement de développement spécifique à l'Avanto qui devra accueillir l'architecture d'intégration des modules, il convient de présenter une vue d'ensemble du fonctionnement et de l'interaction des différents composants structurels de l'IRM.

L'Avanto est composé de quatre cibles matérielles indépendantes. Il y a tout d'abord l'ordinateur permettant à l'utilisateur de contrôler les opérations et nommé 'host', il permet la compilation des séquences IRM ainsi que des programmes de reconstruction d'images qui sont détaillées à la section 2.3 ci-dessous. Il permet également de lancer l'exécution d'une ou plusieurs séquences successives en plus de fournir une interface pour le visionnement des images produites par l'appareil. Le système d'exploitation utilisé pour le 'host' est Windows XP. Une seconde machine, le 'Measurement and Physiologic Control Unit' ou MPCU reçoit les événements de la séquence comme les gradients, les ondes radio fréquence (RF) ou les convertisseurs analogique à numérique (CAN) en provenance du 'host'. Un ordonnancement temps réel des événements est alors effectué avant de fournir au scanner les instructions via un DSP. Le système d'exploitation du MPCU est VxWorks, un système temps réel dédié au traitement de tâches ayant des contraintes temporelles très élevées. Le scanner, troisième appareil du

lot de quatre, reçoit les instructions de déroulement de la séquence en provenance du MPCU et effectue les opérations demandées dans les temps imposés. Ces événements sont généralement les constituants de la séquence comme les gradients à appliquer, les ondes RF et les CAN. Le scanner gère également l'ensemble des amplificateurs de puissance pour les gradients et pour les ondes RF. Une fois l'acquisition terminée, les données brutes récupérées par le CAN du scanner sont envoyées à une quatrième machine nommée MRIR ('MR Image Reconstruction') qui se charge de reconstruire les images à partir des données brutes récupérées sur le scanner. Il faut se rappeler qu'en imagerie par résonance magnétique, le signal récupéré lors de la prise d'images est directement obtenu dans le domaine spectral. Il convient donc d'effectuer une succession de transformées de Fourier inverses pour reconstruire l'image, ce que fait en partie le MRIR. Une fois les images reconstruites, celles-ci sont envoyées au 'host' pour le visionnement. La figure 2.1 suivante schématise l'environnement matériel de l'Avanto, ainsi que le système d'axes considéré dans l'environnement.

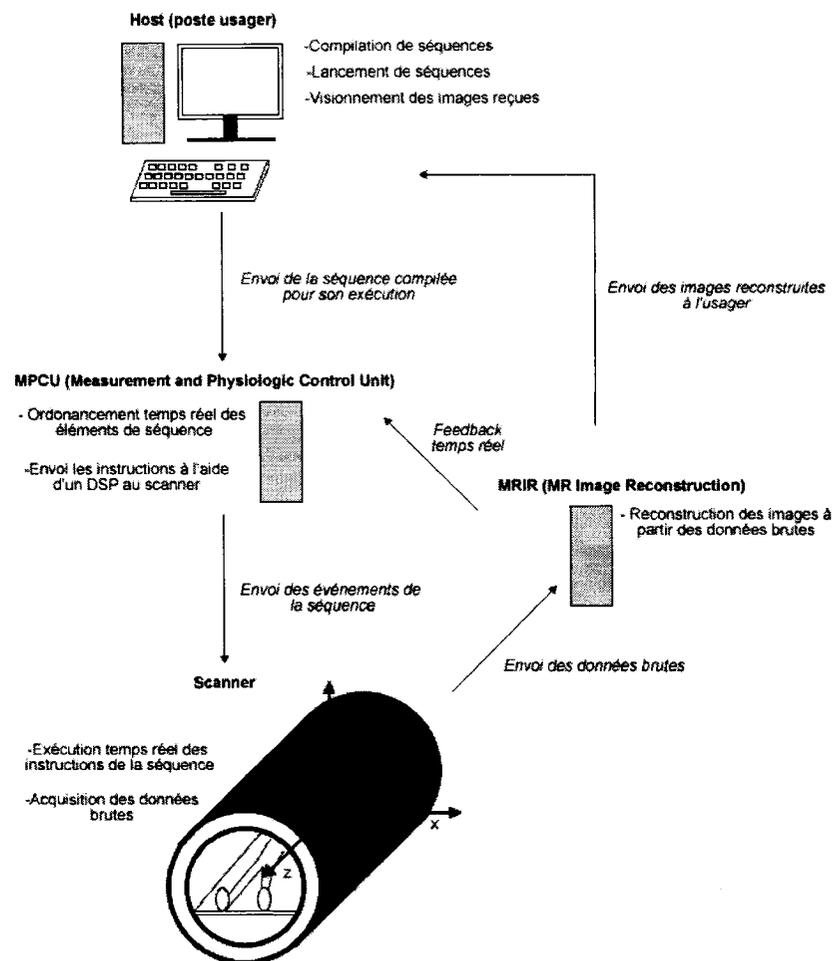


Figure 2.1: Vue d'ensemble des composants matériels du système IRM Avanto de Siemens ainsi que du système d'axes considéré

Bien que détaillé dans les sections 2.4 et 2.5 suivantes, il est important de mentionner la présence d'un lien physique unidirectionnel existant entre le MRIR et le MPCU. Ce lien est ce qui rend un projet comme le MR-Sub ou toute autre procédure d'imagerie interventionnelle réalisable. En effet, une fois les données brutes de l'acquisition envoyée, il est possible, à partir du MRIR, de modifier la séquence en cours d'exécution sur le scanner en communiquant avec le MPCU. Ce lien permet en fait de terminer une

boucle de communication entre l'ordonnanceur d'instructions (le MPCU), l'exécuteur de ces instructions (le scanner) et l'observateur des données (le MRIR). C'est en effet grâce à cette boucle dont le fonctionnement est plus longuement discuté à la section 2.5 qu'il est possible de réagir à tout moment sur le déroulement de la séquence pendant son exécution sur le scanner.

2.3 Séquences IRM

Avant de continuer la présentation de l'environnement que l'on retrouve sur l'IRM, il convient de présenter plus en détails ce qu'est une séquence IRM. Bien que le fonctionnement détaillé de cette modalité d'imagerie ne soit pas l'objectif de ce mémoire, le lecteur peut se rapporter à plusieurs ouvrages [20, 57, 4, 27] pour approfondir ses connaissances sur son fonctionnement.

Une séquence IRM est un diagramme chronologique représentant la succession des événements nécessaires à la formation d'un signal et à sa récupération en vue d'une reconstruction de celui-ci en une image. Les événements typiques que l'on retrouve dans une séquence IRM sont les gradients magnétiques, les ondes RF et les CANs. L'ordonnancement temporel strict de ces événements mène à la formation d'un signal émanant du volume observé. Ce signal est une superposition d'ondes ayant des amplitudes, des fréquences et des phases bien précises que l'on récupère par un CAN avant de les placer dans un tableau en vue de la reconstruction finale. Ce tableau, nommé espace k contient donc un échantillonnage des signaux suivant leurs fréquences et leurs phases. L'exécution d'une séquence remplit généralement une ligne unique de

l'espace K. Pour remplir l'ensemble du tableau, il convient donc de répéter la séquence autant de fois que l'on désire de lignes dans l'espace K. Entre deux répétitions de la séquence, il existe généralement un temps mort T_r ou temps de récupération pendant lequel aucun événement n'est présent afin de laisser le temps aux protons de revenir à leur état d'équilibre.

En imagerie par résonance magnétique, les gradients magnétiques servent à encoder dans les trois directions de l'espace le signal provenant du volume observé alors que l'onde RF sert à exciter les protons d'hydrogène pour les mettre dans un état où il est possible d'en récupérer un signal. Suivant le type de séquence, une excitation permet de faire basculer les protons avec un angle déterminé par rapport à leur orientation d'équilibre. Une fois l'onde RF relâchée, les protons, tout en se déphasant, vont tranquillement revenir à leur position d'équilibre dans le plan longitudinal, c'est la relaxation. C'est durant cette relaxation que l'on mesure le signal provenant du volume. En raison du déphasage qui augmente au fur et à mesure que les protons se relaxent, il devient de plus en plus difficile de capter un signal cohérent en provenance du volume. Pour cette raison, on effectue généralement une refocalisation servant à remettre en phase les protons. Cette refocalisation peut s'effectuer de deux manières, soit par une onde RF à 180 degrés, soit par une succession de gradients ayant deux polarités différentes, qui eux aussi, agissent sur la phase des protons. L'emploi de l'onde RF dans la refocalisation mène à des séquences que l'on nomme écho de spin ou 'spin echo' alors que l'emploi de gradients pour la refocalisation mène à des séquences d'écho de gradient ou 'gradient echo'. Une fois la refocalisation effectuée, il faut attendre un temps T_E ou

temps d'écho, avant d'observer le retour en phase complet des protons. C'est à cet instant que l'on mesure le signal et que l'on remplit l'espace K. Cette séquence est ensuite répétée autant de fois qu'il faut pour remplir toutes les lignes de l'espace k, toujours séparée d'un temps T_R entre deux acquisitions. Une fois complété, ce tableau spectral est reconstruit par le MRIR pour former une image d'intensité du volume observé. La figure 2.2 suivante illustre une séquence écho de spin classique.

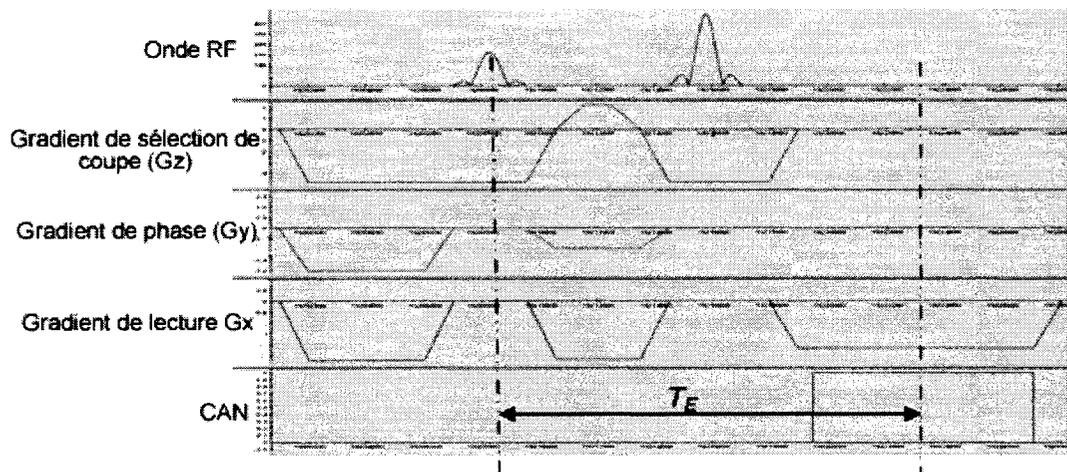


Figure 2.2: Illustration du déroulement d'une séquence de type écho de spin classique

Sur l'Avanto, d'un point de vue programmation, une séquence s'illustre sous la forme d'une série de fonctions C++ regroupées dans un fichier qui est par la suite compilé et inclus dans une librairie dynamique ou DLL. Ce DLL est ensuite appelé lors de son exécution par l'infrastructure de l'environnement logiciel de Siemens qui se charge de l'envoyer au MPCU pour son exécution. Lors de la compilation, le compilateur utilise un fichier de compilation 'makefile' pour connaître les dépendances entre les différents fichiers et librairies.

Sur les appareils d'imagerie par résonance magnétique cliniques de Siemens, une séquence comporte 4 fonctions obligatoires qui sont appelées lors de l'exécution. La première se nomme 'fSEQinit' et sert généralement à spécifier certaines limites lors de l'exécution, le type du système de résonance magnétique utilisé, le protocole par défaut de la séquence et l'initialisation de variables statiques ou qui ne dépendent pas des choix de l'utilisateur avant l'exécution de la séquence. La seconde se nomme 'fSEQPrep' et permet de vérifier la validité du protocole en cours, ou de le rendre valide le cas échéant. Elle s'occupe également de la sélection du programme de reconstruction d'images qui sera utilisé une fois les acquisitions effectuées ainsi que de l'envoi des informations nécessaires à l'initialisation de celui-ci. La troisième fonction à être appelée est 'fSEQCheck' qui se charge de vérifier que l'amplitude des gradients utilisés lors de la séquence ne dépasse pas un seuil critique au delà duquel, ceux-ci pourraient occasionner des complications comme des stimulations nerveuses au patient. Finalement, la fonction 'fSEQRun' contient la chronologie des événements de la séquence devant s'exécuter sur le scanner. Comme décrit plus haut, l'utilisation de CANs dans une séquence permet de récupérer le signal en provenance du volume imagé. C'est donc à travers ces CANs que sont envoyées au programme de reconstruction d'images les données récupérées lors de l'acquisition. Cette chronologie d'événements est exécutée autant de fois qu'il est nécessaire pour remplir l'espace k. La portion répétée de la fonction 'fSEQRun' est donc souvent nommée noyau ou 'kernel'. Seul ce noyau faisant partie de la fonction 'fSEQRun' et qui rassemble la suite temporelle des événements est répété; les autres fonctions ne sont exécutées qu'une seule fois lors du déroulement de la séquence. La

figure 2.3 ci-dessous illustre le déroulement d'une séquence typique au niveau de la succession des fonctions.

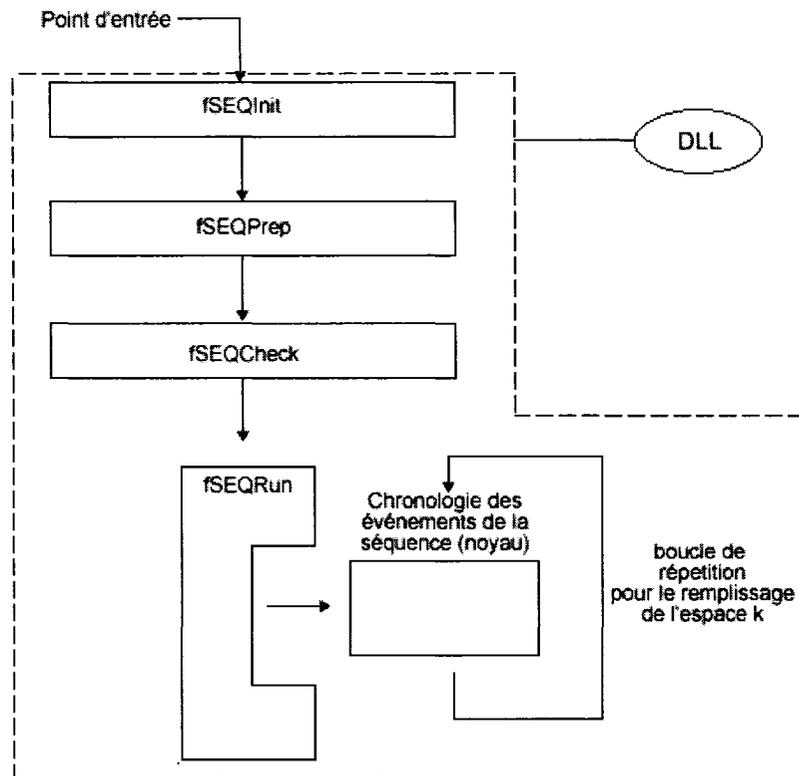


Figure 2.3: Déroulement typique d'une séquence sur l'Avanto

2.4 Environnement de développement : IDEA / ICE

Une fois les constituants matériels présentés, il convient désormais d'introduire l'environnement de développement présent sur l'Avanto et sur tous les autres systèmes d'imagerie par résonance magnétique de Siemens. En effet, l'environnement décrit dans ces lignes se retrouve sur la plupart des IRM cliniques fabriqués par la compagnie Siemens ce qui assure une compatibilité de l'intégration et de l'architecture présentée

dans ce mémoire sur l'ensemble des systèmes d'imagerie par résonance magnétique cliniques temps réel de cette compagnie. Cet environnement permet à la fois le développement et la compilation des séquences et des programmes de reconstruction d'images aussi bien sur l'IRM lui-même que sur des postes externes n'étant pas reliés au système d'imagerie. Il est ainsi possible de développer des applications et des séquences sans toutefois pouvoir les tester au sein même du Laboratoire de Nanorobotique. Comme il sera présenté sous peu, il est également possible de simuler l'exécution de telles séquences IRM 'offline'. Cela dit, une telle simulation fonctionnelle sur un poste externe n'assure en aucun cas le bon fonctionnement de cette séquence sur l'IRM lui-même. En fait, la simulation des séquences à l'extérieur de l'IRM ne sert qu'à assurer uniquement la bonne synchronisation des événements constituant la séquence.

2.4.1 IDEA

L'environnement de développement de l'Avanto comporte deux sous environnements ayant chacun un rôle bien précis dans l'élaboration, le test et l'exécution des séquences. Ces deux sous environnements se retrouvent aussi bien sur l'IRM que sur les postes externes pour la compilation 'offline' des séquences. Le premier sous environnement est celui responsable de l'interface permettant la compilation et la simulation des séquences IRM développées et se nomme 'Integrated Development Environment for Applications' ou IDEA. Il consiste en une invite de commande nommée 'Sequence Development Environment' ou SDE permettant la sélection et la compilation de séquences déjà présentes ainsi que la création de nouvelles séquences. Le

lecteur peut se rapporter à la figure I.1 se trouvant en annexe pour une illustration de l'interface SDE.

IDEA comporte également un outil de simulation nommé MRISeqSimVis permettant de visualiser la chronologie des événements la constituant. Obtenu en lançant la commande 'sim' dans le SDE, MRISeqSimVis simule un DSP virtuel exécutant un à un les événements de la séquence avec leur chronologie respective. La séquence devant s'exécuter dans un environnement temps réel, il existe trois cibles de compilation. La première est la cible permettant la compilation en mode 'debug'. Elle est utilisée pour générer un DLL pouvant être analysé avec l'outil MRISeqVis. La seconde est la cible permettant la compilation en mode 'release'. C'est la cible utilisée pour la version devant être exécutée sur l'IRM. Finalement, il existe une cible 'i086' pour l'exécution de la séquence sur le MPCU dans l'environnement VxWorks. Les deux dernières cibles doivent obligatoirement être compilées pour pouvoir exécuter une séquence sur le scanner.

2.4.2 ICE

Le second sous environnement est celui de reconstruction d'images nommé 'Image Construction Environment' ou ICE. Ce sous environnement permet, sur la machine host ou sur un poste externe au système IRM de pouvoir compiler les programmes de reconstruction d'images à l'aide d'une invite de commande. Dans le cas où cette compilation s'effectue sur la machine host, le programme de reconstruction est alors envoyé au MRIR pour son exécution en parallèle avec la séquence. Chaque

séquence se doit d'avoir un programme de reconstruction d'images pour recevoir et traiter les données brutes récupérées durant le déroulement de la séquence. A nouveau, le lecteur peut se rapporter à la figure I.2 en annexe qui illustre l'allure de l'invite de commande pour la compilation des programmes de reconstruction d'images. Bien que ce sous environnement possède lui aussi un outil de simulation, il n'est que rarement utilisé. En effet, la simulation d'un programme de reconstruction est relativement complexe et ne permet que de s'assurer du bon fonctionnement d'un tel programme tant qu'il n'implique pas de communication temps réel, ce qui n'est pas le cas de l'architecture présentée ici. De plus, étant donné qu'un programme de reconstruction d'images s'attend à recevoir des données d'acquisition en provenance de la séquence et qu'il n'est pas possible de simuler l'interaction entre ceux-ci, la pertinence d'un tel outil de simulation pour les programmes de reconstruction d'images paraît atténuée.

En ce qui concerne les programmes de reconstruction d'images, ils sont constitués, tout comme les séquences, de points d'entrées sous la forme de fonctions écrites en C++. Ces fonctions sont en fait des sous fonctions d'un programme beaucoup plus global que constitue l'infrastructure logicielle ICE. Cette infrastructure se charge d'appeler les différentes fonctions au fur et à mesure du déroulement de la séquence. Toutes ces fonctions font partie de trois processus qui tournent simultanément sur le MRIR. Ainsi, toute fonction, qu'elle constitue un point d'entrée obligatoire ou non, doit se retrouver dans l'un des trois processus concourants qui sont actifs sur le MRIR. Le premier processus se nomme 'OnIce' et regroupe les fonctions effectuant des tâches durant l'exécution de la séquence. Le second processus se nomme 'OffIce' et regroupe

quant à lui les fonctions effectuant des tâches une fois les acquisitions de la séquence terminées. En fait, si la séquence nécessite du post-traitement, il sera effectué sous la forme d'une fonction et placé dans le processus 'OffIce'. Finalement, un processus nommé 'DAFE' est chargé de la fonction d'opération temps réel présentée sous peu.

Bien qu'il existe un grand nombre de fonctions pour le traitement des données en plus de celles pouvant être incluses par les concepteurs, l'infrastructure ICE ne comporte que trois fonctions obligatoires et deux fonctions pour l'opération temps réel. La première fonction constituant le premier point d'entrée obligatoire se nomme 'prepare' et est appelée durant la préparation de la séquence avant l'exécution des événements qu'elle contient. La fonction 'prepare' permet essentiellement l'initialisation des variables ainsi que l'exécution de routines avant le lancement de la séquence. Cette fonction se situe dans le processus 'OnIce'. La seconde fonction obligatoire se nomme 'online' et est appelée à chaque acquisition de la séquence. En fait, chaque CAN de la séquence peut être configuré pour envoyer ses données à une fonction déterminée dans le programme ICE. Durant une séquence standard, ce CAN est configuré pour envoyer ces données à la fonction 'online' du programme de reconstruction. Ainsi, chaque fois que le CAN récupère des données, la fonction 'online' reçoit ces données et effectue sa routine en vue de la reconstruction. Toute opération devant s'effectuer lors du déroulement de la séquence à la suite des acquisitions, se doit de se trouver soit dans la fonction 'online' elle-même, ou se doit d'être appelée par celle-ci. La fonction 'online' se retrouve naturellement dans le processus 'OnIce'. La dernière fonction obligatoire devant se trouver dans tout programme de reconstruction se nomme 'offline' et se

charge des routines devant s'effectuer en fin de séquence. Lorsque la dernière ligne d'acquisition de celle-ci est terminée, la fonction 'offline' est alors invoquée en vue des opérations post-acquisition éventuelles. Cette fonction se retrouve dans le processus 'OffIce' de l'infrastructure. Dans le cas d'applications temps réel, deux autres fonctions se rajoutent à celles déjà présentées. Elles constituent des points d'entrée facultatifs qui deviennent obligatoires dans le cas d'application temps réel au niveau de la séquence. La première fonction se nomme 'rtfeedback' et est appelée par un CAN de la séquence spécialement configuré pour cela. Cette fonction, tout comme son homologue 'hpfeedback' qui constitue la seconde fonction temps réel, permet l'évaluation des données acquises au moment de son appel ainsi que le remplissage d'un tampon de données temps réel (qui est expliqué en profondeur à la section suivante) en vue de son envoi à la séquence dans des délais très courts. Ce tampon sera alors reçu par un point d'entrée dynamique spécial du côté de la séquence et permettra la modification de celle-ci. La différence majeure entre les deux fonctions 'rtfeedback' et 'hpfeedback' réside dans le processus dans lequel elles se trouvent. La fonction 'rtfeedback' se situe dans le processus DAFE alors que 'hpfeedback' se situe dans le processus OnIce. De plus, selon les données de Siemens, la fonction 'rtfeedback' est dédiée aux tâches devant s'exécuter dans des contraintes de temps excessivement restreintes alors que 'hpfeedback' permet l'exécution de code légèrement plus lourd et plus long à s'exécuter. Malheureusement, aucun délai n'est fourni par le constructeur pour définir ce qu'il entend par contraintes temporelles sévères.

La figure 2.4 suivante regroupe les fonctions et les processus décrits plus haut en plus d'illustrer leur utilisation au cours de l'exécution de la séquence.

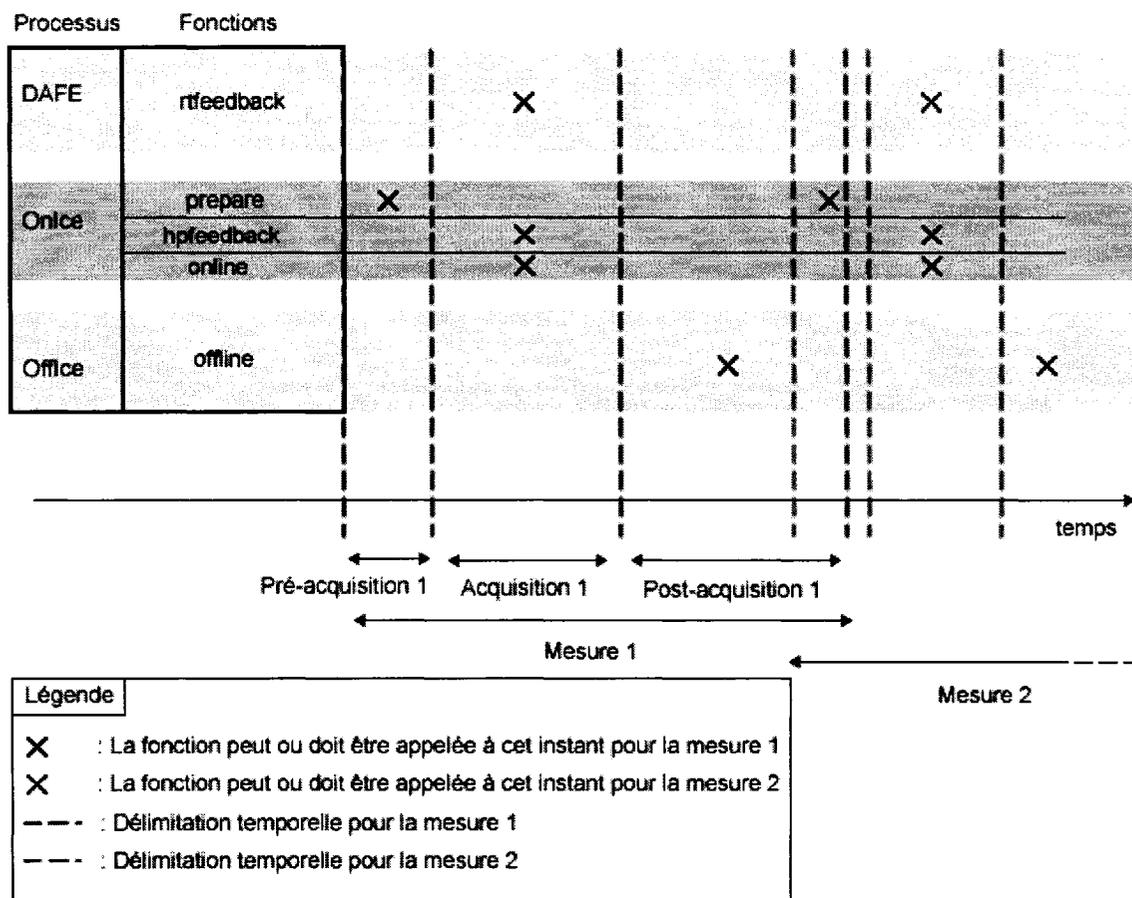


Figure 2.4: Illustration des différents processus et des fonctions se trouvant dans l'infrastructure ICE ainsi que des possibilités d'appel à ses fonctions en provenance de la séquence

2.4.3 Protocole de la séquence et communication entre IDEA et ICE

L'utilisateur désirant exécuter une séquence qu'elle soit temps réel ou non, doit avant tout, par l'entremise de l'interface graphique usager (IGU) Syngo MR de la compagnie Siemens, fixer certains paramètres avant le démarrage de celle-ci. L'ensemble des paramètres nécessaires au fonctionnement de la séquence se nomme le

‘protocole’ de la séquence. Ce protocole contient à la fois des informations nécessaires au déroulement de la séquence comme l’angle de flip, le temps de répétition T_r ou la résolution ainsi que d’autres données envoyées au MRIR lors de l’initialisation. Certains de ces paramètres sont fixés et envoyés directement par l’infrastructure de l’environnement logiciel de Siemens alors que d’autres sont modifiables par l’utilisateur à travers Syngo MR. La figure 2.5 illustre l’allure de l’interface Syngo permettant de remplir le protocole, lancer la séquence désirée et observer les images reçues.

Lors du démarrage de la séquence, certaines informations comme le nombre d’acquisitions à effectuer ou la résolution sont envoyées au programme de reconstruction situé sur le MRIR. Pour ce faire, il existe deux moyens. Si les paramètres ne changent pas durant le déroulement de la séquence, il est possible d’envoyer des données du protocole au programme de reconstruction via un tampon de données nommé ‘Yet Another Parameter Set’ (YAPS). Lors de la phase de préparation de la séquence (qui n’est donc pas encore démarrée), les données sont envoyées et sont récupérées lors de l’appel de la fonction ‘prepare’ du programme de reconstruction d’images.

remplis par le système ainsi que des champs réservés au programmeur de séquences qui peut les remplir avec les données qu'il désire. Par le biais de fonctions dédiées, il est ainsi possible, dans la séquence, de remplir le MDH par des données qui sont récupérables dans la fonction 'Online' du programme de reconstruction d'images.

2.5 Capacités temps réel de l'Avanto

Cette section décrit plus en détails le fonctionnement et l'utilisation de l'aspect temps réel de l'Avanto 1.5 T de Siemens. C'est sur ces capacités temps réel qu'est basé le fonctionnement du MR-Sub. En effet, la navigation du dispositif dans un environnement vasculaire nécessite à la fois de modifier la valeur des gradients appliqués en fonction de la commande reçue pendant l'exécution de la séquence mais également de déterminer et d'appliquer ces commandes dans les délais les plus courts possibles pour augmenter la fréquence d'opération du système et ainsi permettre un contrôle plus efficace. L'utilisation des capacités temps réel est tout d'abord présentée sous la forme d'interaction des fonctions utilisées pour ensuite introduire le fonctionnement de cet aspect innovateur de l'imagerie par résonance magnétique.

2.5.1 Utilisation des possibilités temps réel

L'utilisation des fonctionnalités temps réel de l'Avanto est relativement complexe à prime abord et ne sont décrits ici que les appels aux fonctions nécessaires à son utilisation durant le déroulement de la séquence. L'inclusion des bibliothèques spécifiques nécessaires à son fonctionnement n'est donc pas couverte ici.

Le fonctionnement du mode temps réel de l'Avanto débute par l'appel d'une fonction de préparation du tampon de données temps réel qui est elle-même appelée lors de l'exécution de la fonction 'prepare' du programme de reconstruction d'images. Cette fonction de préparation initialise un tampon de données de type 'void *' en lui assignant une étiquette d'identification en vue de sa reconnaissance dans la séquence. Au niveau de la séquence, il faut s'assurer de configurer un CAN pour l'envoi des données à une fonction temps réel (soit 'hpfeedback', soit 'rtfeedback'). Une fois la séquence lancée et lorsque l'événement du CAN est exécuté sur le scanner, les données récupérées sont envoyées à la fonction temps réel choisie. A cet instant, la séquence s'interrompt durant un délai donné afin de permettre à la fonction temps réel du programme de reconstruction d'effectuer sa tâche. La fonction choisie 'hpfeedback' ou 'rtfeedback' est alors exécutée et, suivant les données reçues, remplit le tampon de données temps réel. Ces données sont généralement des valeurs de gradients, des orientations de tranche ou des paramètres d'onde RF devant être modifiés dans la séquence pour les prochaines acquisitions de celle-ci. Une fois le tampon de données rempli, celui-ci est envoyé à la séquence à l'aide d'un tunnel dédié existant entre les deux machines et illustré à la figure 2.1. Un point d'entrée optionnel de la séquence est alors appelé prenant la forme d'une fonction nommée 'fSEQReceive'. Cette fonction peut être exécutée bien que le processus de la séquence soit interrompu car celle-ci réside dans un processus à part destiné précisément à recevoir des données en provenance de l'extérieur. Une fois exécutée, cette fonction analyse le tampon de données reçu en vérifiant son étiquette d'identification et effectue les opérations qui s'y rapportent. Avant la fin de la fonction

'fSEQReceive', celle-ci, à travers l'utilisation d'une variable spécialisée, informe la séquence que la boucle temps réel s'est déroulée correctement. Au réveil de la séquence, celle-ci vérifie si l'opération s'est correctement comportée et émet un message au système le cas échéant. La séquence se poursuit alors à partir du point où elle a été mise en veille. Chaque fois qu'un CAN configuré pour la communication temps réel est exécuté sur le scanner, une boucle temps réel de communication s'initie donc entre le programme de reconstruction et la séquence.

2.5.2 Fonctionnement de l'aspect temps réel

Le fonctionnement interne du mode de communication temps réel de l'Avanto dont l'utilisation vient d'être présentée est décrit plus en détails dans ces lignes. Cependant, avant de présenter comment la communication temps réel prend place entre la séquence et le programme de reconstruction, il convient tout d'abord d'introduire le concept d'événements et de blocs d'événements au niveau de la séquence.

Les événements que l'on retrouve dans une séquence comme les gradients magnétiques, les ondes RF et les CAN sont eux-mêmes considérés comme des événements temps réel car leur exécution temporelle au niveau du scanner s'effectue dans des délais de microsecondes. En effet, en IRM la formation d'images à travers le principe de résonance magnétique nucléaire et l'encodage spatial avec les gradients magnétiques implique la succession d'événements bien précis avec des temps d'application bien précis également. Ces événements se doivent d'être regroupés en blocs d'événements qui sont envoyés au MPCU pour leur exécution sur le scanner. Ces blocs constituent donc l'unité temporelle de base d'une séquence. Une fois envoyés vers

le MPCU, ils sont exécutés en temps réel à l'aide d'un DSP pour ensuite être dirigés sur le scanner pour être à nouveau exécutés en vue de la récupération de données. Lors du déroulement de la séquence, le MPCU va donc traiter les blocs d'événements et les insérer dans une file de type FIFO d'où le scanner puise les blocs événements à exécuter. Bien entendu, afin d'éviter que le FIFO ne se vide, les blocs d'événements sont traités plus rapidement par le DSP qu'ils ne prennent de temps à s'exécuter sur le scanner. Les blocs d'événements sont délimités par des instructions bien précises sous la forme de fonctions C++. Ainsi, un bloc d'événements débute toujours par l'instruction 'fRTEBInit' et se termine toujours par l'instruction 'fRTEBFinish'. Entre ces instructions se trouvent les événements temps réel comme les gradients, les ondes RF et les CAN. Ces événements sont insérés dans un bloc d'événements par l'entremise de l'instruction 'fRTEI'. La figure 2.6 suivante illustre la succession des traitements que subit un bloc d'événements temps réel entre le niveau de code et le niveau d'exécution sur le scanner.

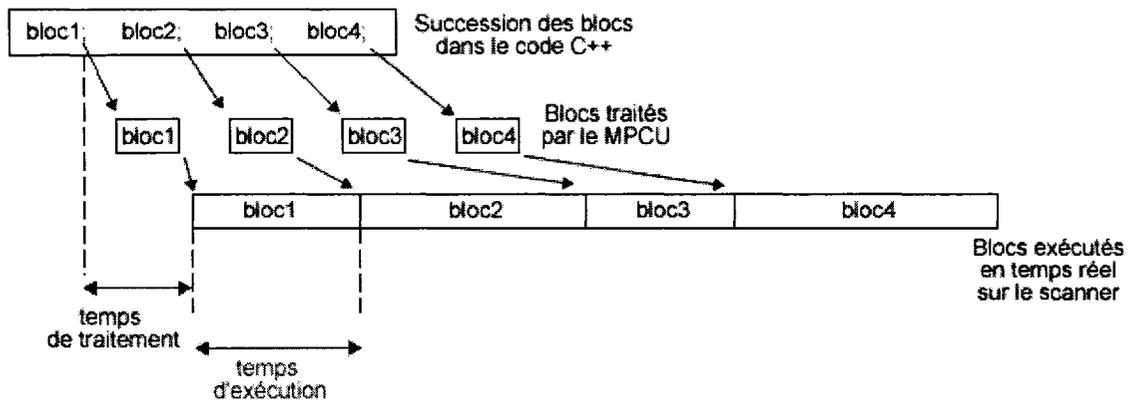


Figure 2.6: Illustration de la succession des blocs d'événements à partir de leur programmation jusqu'à leur exécution sur le scanner

Dans le cas d'une séquence temps réel, celle-ci suspend le traitement des blocs à venir sur le MPCU qui est alors interrompu (ou mis en veille) à l'aide de la fonction 'fRTWaitForWakeup' placée après le premier bloc d'événements. Pour remettre en fonction le MPCU, il faut envoyer un signal nommé 'wakeup' placé dans le bloc d'événements de telle sorte que le temps séparant l'instant d'application du signal 'wakeup' et la fin du bloc d'événements soit suffisamment grand pour effectuer la tâche temps réel qui, dans le cas présenté ici, consiste en l'exécution d'une fonction temps réel du programme de reconstruction d'images. La figure 2.7 illustre le principe des blocs d'événements temps réel et le principe de la mise en veille du MPCU et de son réveil à l'aide du signal 'wakeup'.

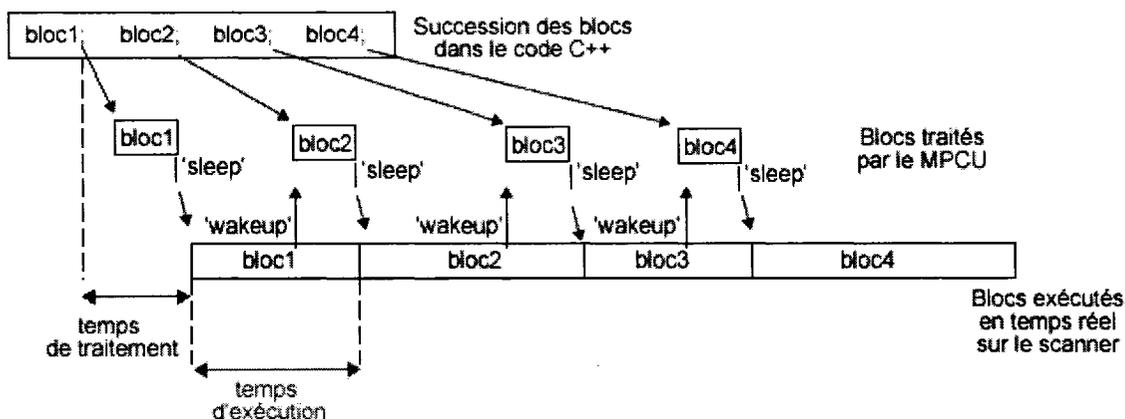


Figure 2.7 : Principe d'interruption du MPCU et de son réveil par l'envoi d'un signal de 'wakeup' en provenance du bloc d'événements

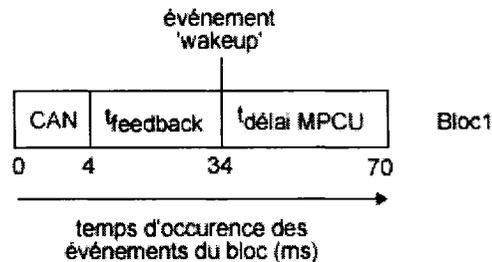
La boucle de communication temps réel permet donc d'être en mesure de modifier le déroulement d'une séquence pendant son exécution. Le déroulement d'une telle communication relativement complexe est résumé dans les prochaines lignes. Tout d'abord, il convient de déclarer une variable permettant de vérifier si la boucle de communication temps réel s'est bien effectuée. Pour se faire, on crée une variable de type booléen que l'on nomme 'bfeedbackreceived' et que l'on initialise à 'false'. Il faut ensuite être en présence de données nécessitant d'être évaluées par le programme de reconstruction à partir de la fonction 'fSEQRun' qui contient les blocs d'événements. Un CAN, spécialement configuré pour envoyer ses données à l'une des deux fonctions temps réel ('hpfeedback' ou 'rtfeedback') est ensuite incorporé dans un tel bloc d'événements que l'on nomme Bloc1. Ce Bloc1 contient également un événement pour la remise en fonction du MPCU après sa future interruption. Dans le Bloc1, l'événement 'wakeup' est inséré après un temps mort suite à l'application du CAN. Ce temps mort

doit assurer l'exécution de la fonction dédiée du programme de reconstruction ainsi que la réception des données et le déroulement de la fonction 'fSEQReceive' qui se chargera de modifier la séquence. Ce temps mort est nommé $t_{feedback}$. Suite à l'insertion de l'événement 'wakeup', un autre délai est inséré devant permettre, une fois le MPCU réveillé, de lui laisser suffisamment de temps pour à nouveau insérer d'autres blocs à exécuter par le scanner dans le FIFO. Il faut se rappeler que le traitement des blocs d'événements sur le MPCU est beaucoup plus rapide que leur déroulement sur le scanner. Le MPCU est ensuite mis en veille alors que le bloc d'événements Bloc1 débute son exécution sur le scanner. L'événement du CAN est alors lancé et envoie les données récupérées à la fonction choisie du programme de reconstruction d'images. Au même moment, le bloc d'événements exécute le temps mort $t_{feedback}$. Pendant ce temps, la fonction temps réel du programme de reconstruction est exécutée et envoie les instructions à travers le tampon de données temps réel. Par l'entremise d'un tunnel dédié, ce tampon, accompagné de son étiquette d'identification atteint le processus dans lequel se trouve la fonction 'fSEQReceive' de la séquence. Celle-ci analyse le tampon de données reçu, modifie les différentes instances des objets des classes représentant les gradients, les ondes RF etc... pour refléter le changement désiré et modifie la variable 'bfeedbackreceived' à 'true' pour informer la séquence (lorsque le MPCU se réveillera) que la boucle de communication temps réel s'est correctement complétée. Après l'exécution du temps mort (qui doit être suffisant pour que la fonction 'fSEQReveice' ait eu le temps de modifier la variable 'bfeedbackreceived' comme il vient d'être décrit), le scanner continue le déroulement du bloc d'événements et atteint l'événement du signal

‘wakeup’ qu’il exécute et qui provoque le réveil du MPCU. Celui-ci est chargé de vérifier l’état de la variable ‘bfeedbackreceived’ pour s’assurer que la boucle temps réel s’est complétée correctement. Le scanner termine alors l’exécution du Bloc1 en exécutant un second délai pour laisser le temps au MPCU qui vient d’être réveillé d’insérer d’autres blocs d’événements dans le FIFO en vue de leur exécution. Les blocs d’événements successifs sont alors traités par le MPCU et exécutés sur le scanner sans laisser le FIFO vide, ce qui aurait provoqué un arrêt brusque de la séquence, le scanner n’ayant plus de blocs d’événements à rouler. La figure 2.8 tente de résumer graphiquement le déroulement d’une communication temps réel typique, comme celle décrite à l’instant. Les temps donnés sont fictifs.

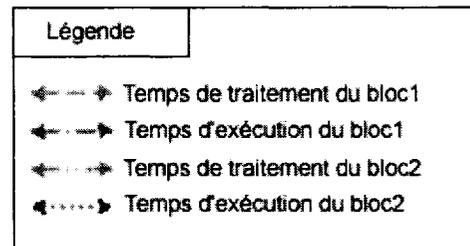
Ordre des événements

- 1) Configuration du CAN pour l'envoi des données à la fonction 'hpfeedback' du programme de reconstruction d'images
- 2) Mise à 'false' de la variable de vérification de la boucle temps réel 'bfeedbackreceived'
- 3) Création d'un bloc d'événements pour la communication temps réel, incluant le CAN de 1)



- 4) Mise en veille du MPCU

- 5) Vérification de la variable 'bfeedbackreceived'



Lors de l'exécution

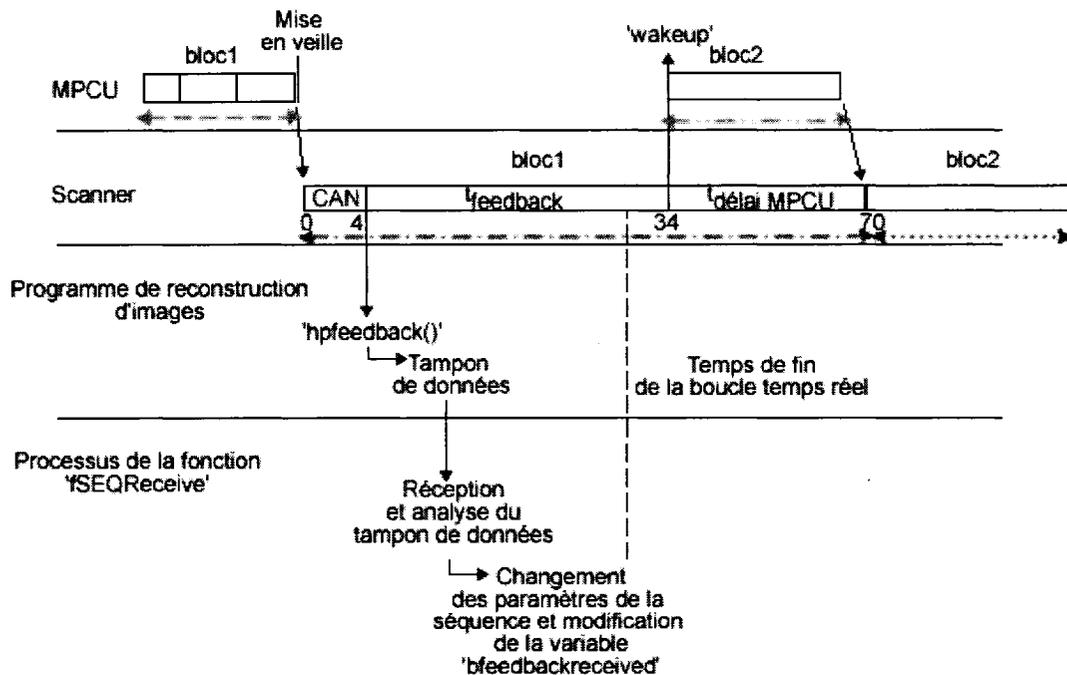


Figure 2.8 : Ordre et exécution des événements lors d'une communication temps réel entre une séquence et un programme de reconstruction d'images

Avant de terminer cette section, il convient de commenter les temps nécessaires à respecter lors d'une communication temps réel entre une séquence et un programme de reconstruction d'images en vue d'assurer au scanner et à la séquence en général un déroulement sans complication. Il faut tout d'abord s'assurer que le temps mort réservé à l'exécution de la fonction du programme de reconstruction d'images et à l'envoi et au traitement des données en vue de la modification de la séquence soit suffisamment grand pour permettre à toutes ces opérations de se terminer avant ce délai nommé $t_{feedback}$ que l'on peut observer à la figure 2.8. Le cas échéant, le signal 'wakeup' arrive avant la fin de ces opérations et le MPCU traite de nouveaux événements. Le test de validation de la boucle temps réel à travers la variable 'bfeedbackreceived' échoue et le système produit une erreur. Bien que ce comportement n'empêche pas la séquence de s'exécuter car celle-ci poursuit son cours sans modification de la part de la fonction 'fSEQReceive', il n'en demeure pas moins qu'il devient alors impossible avec le délai temps réel alloué, d'effectuer une communication temps réel valide entre la séquence et le programme de reconstruction.

Le second délai devant être assuré est celui qui suit l'événement 'wakeup' dans le bloc d'événements. En effet, une fois le signal 'wakeup' exécuté sur le scanner, le MPCU se réveille et continue à traiter et placer des événements dans le FIFO. Or, si après le signal 'wakeup', il n'existe pas d'autres événements ou un temps mort d'une certaine durée notée $t_{délai MPCU}$ sur la figure 2.8, le MPCU n'a pas le temps de traiter un autre bloc d'événements avant que le scanner finisse l'exécution du précédent bloc. En fait, il faut s'assurer, bien que théoriquement il ne soit pas possible de connaître ce délai,

que le temps de traitement du prochain bloc noté par la flèche ‘Temps de traitement du bloc 2’ sur la figure 2.8 soit inférieur au délai entre l’occurrence du signal ‘wakeup’ et la fin du bloc d’événements exécuté sur le scanner noté par la portion de la flèche ‘Temps d’exécution du bloc 1’ entre les temps 34 ms et 70 ms du bloc 1 sur la figure 2.8.

2.6 Interaction, intégration des modules et vue d’ensemble de l’architecture

L’environnement de développement de l’Avanto étant désormais présenté, les différents modules décrits au chapitre 1 et constituant les piliers du projets se doivent maintenant d’être mis en commun et placés dans l’architecture présentée dans les sections 2.4 et 2.5. Une fois effectuée, cette intégration permettra de procéder aux premiers tests de contrôle, de propulsion et de positionnement d’une façon organisée et autonome devant permettre le dévoilement d’une première preuve de concept du fonctionnement du MR-Sub.

Avant de présenter l’intégration et l’interaction des différents modules, une vue d’ensemble du fonctionnement d’une séquence typique est illustrée à la figure 2.9. On peut y voir les différents chemins entrepris par les données suivant que la séquence nécessite une communication temps réel avec le programme de reconstruction d’images ou qu’elle se déroule sans cette communication temps réel.

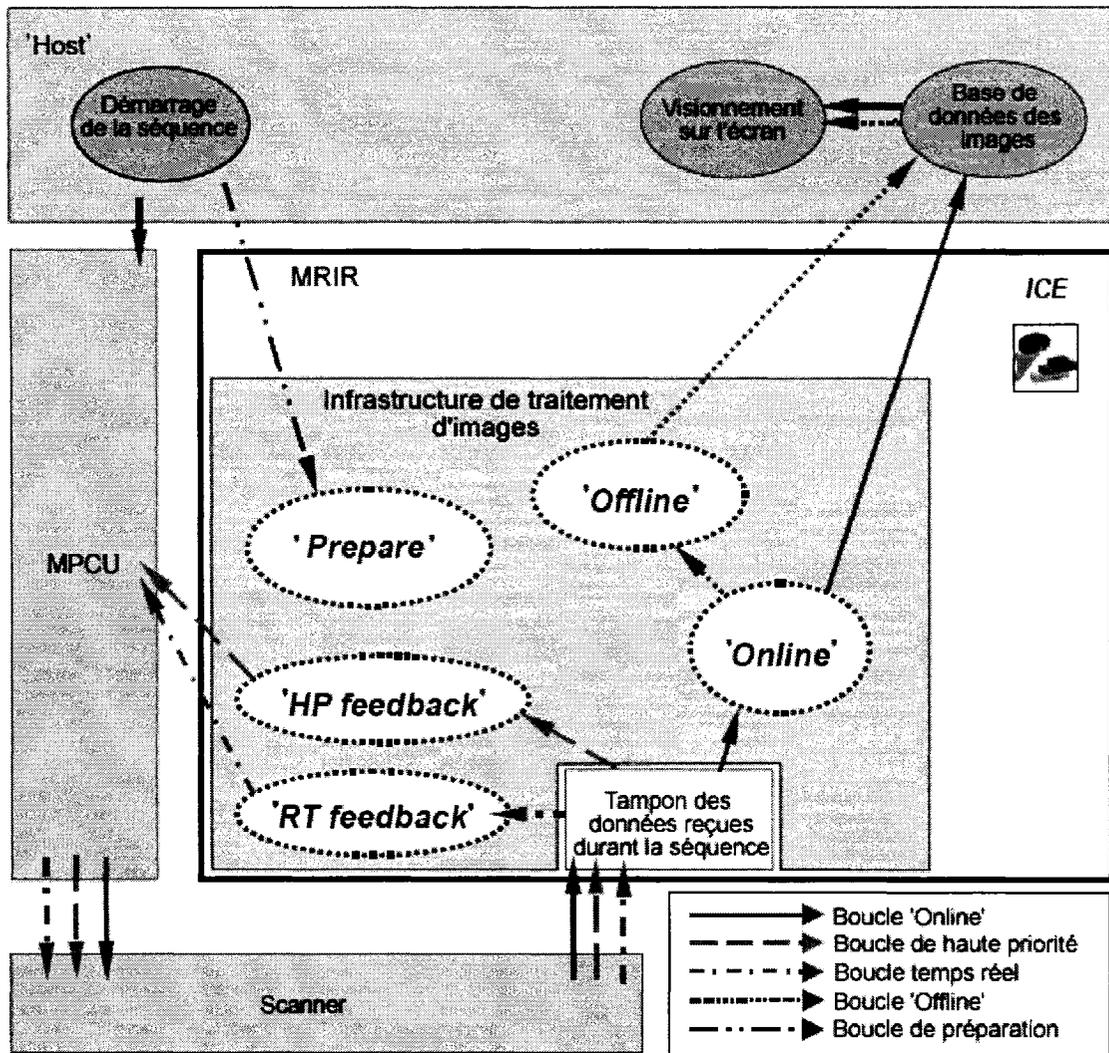


Figure 2.9: Vue d'ensemble du fonctionnement d'une séquence d'un point de vue structurel utilisant ou non les capacités temps réel de l'Avanto

On voit clairement sur la figure 2.9 que l'infrastructure de traitement d'images, étant donné ses capacités de réaction temps réel sur la séquence, constitue le centre nerveux de celle-ci où devra être insérée l'architecture du MR-Sub. En effet, bien que la séquence soit responsable de l'acquisition des données, c'est le programme de reconstruction qui les récupère, les analyse éventuellement et décide de modifier la

séquence durant son exécution. Ce programme de reconstruction donne ainsi un aspect dynamique à la séquence. Bien que les fonctions permettant de réagir ainsi sur une séquence en déroulement soient appelées par l'entremise d'un CAN se trouvant dans un bloc d'événements de la séquence et que le MPCU soit mis en veille à la suite d'une opération de la séquence, toutes les décisions en vue de la modification de celle-ci proviennent du contenu du tampon de données temps réel. C'est pour cela que dans le cadre du projet MR-Sub, c'est dans le programme de reconstruction d'images que se trouveront les différents modules de l'architecture du MR-Sub. La séquence ne sera en quelque sorte que le serviteur des actions commandées par les différents modules.

2.6.1 Interaction des modules

Au chapitre 1, quatre modules centraux avaient été présentés. Un premier module nommé module 'Roadmap' doit permettre la sélection d'un chemin à partir d'une image ou d'un volume d'angiographie. Ce module doit être utilisé avant l'insertion du dispositif en raison des perturbations de susceptibilité magnétique que cause le noyau ferromagnétique du MR-Sub. Le second module, celui de propulsion doit se charger du déplacement du dispositif suite à l'application de gradients magnétiques en provenance de l'appareil de résonance magnétique. Le troisième module de positionnement doit quant à lui prendre en charge la localisation du dispositif à l'aide d'une séquence spécialement élaborée nommée MS-SET. Finalement, le module de contrôle se charge d'interpréter la position du dispositif pour ensuite déterminer la nouvelle commande à appliquer pour faire converger le MR-Sub vers sa cible, que ce soit une balise ou son point de destination.

La succession des différents rôles qu'occupent les modules présentés guide clairement leur utilisation dans le cadre d'une opération typique du MR-Sub. Tout d'abord, on exploite le module 'Roadmap' en vue d'obtenir un chemin à suivre sous la forme d'une succession de balises dans le système vasculaire. Par la suite, on insère le dispositif au point d'insertion choisi par l'utilisateur. A cet instant, les modules de positionnement, de contrôle et de propulsion se succèdent sans relâche en vue de déplacer le dispositif de balise en balise jusqu'à sa destination. La figure 2.10 suivante illustre le déroulement typique de l'opération du MR-Sub d'un point de vue modulaire.

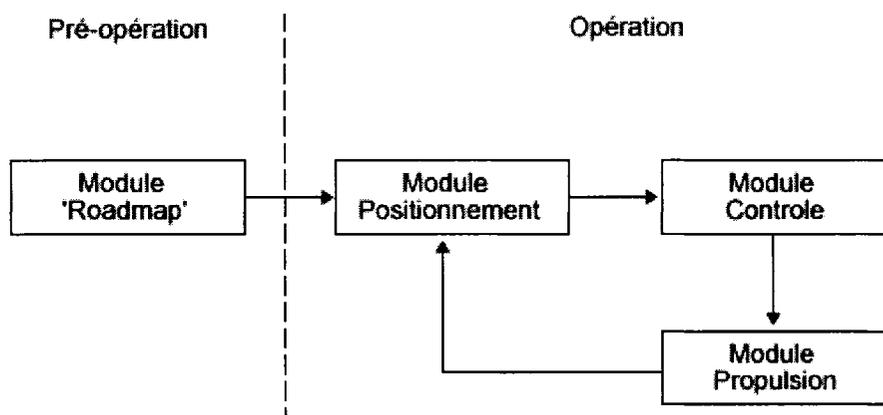


Figure 2.10 : Illustration de la succession des modules dans une opération typique du MR-Sub

Une fois l'angiographie effectuée, les images ou le volume sont analysés par un logiciel externe, présenté plus en détail au chapitre 3, pour effectuer par la suite la segmentation et la sélection des chemins toujours à partir de l'interface usager fournie par la machine 'host'. Une fois la sélection des points d'insertion et de destination effectuée, le chemin est constitué soit manuellement ou automatiquement et conservé pour la suite des événements. Le dispositif est ensuite inséré au point d'entrée

préalablement déterminé. A ce moment, l'opération débute et le dispositif est localisé avec la séquence de positionnement MS-SET qui est exécutée sur le scanner. La position est envoyée au module de contrôle qui se trouve dans le programme de reconstruction d'images pour être analysée. En se basant sur la position de la prochaine balise à atteindre, le contrôleur détermine l'amplitude et la direction du gradient magnétique à appliquer pour déplacer le MR-Sub vers cette balise. Cette commande est ensuite envoyée au module de propulsion qui se charge de demander à la séquence de modifier le prochain gradient à appliquer pour celui calculé par le contrôleur. Par le biais d'une communication temps réel, l'architecture applique alors le gradient demandé, provoquant le déplacement du dispositif. A nouveau, la séquence de positionnement est exécutée sur le scanner et la boucle recommence. Une fois la balise atteinte, l'architecture passe à la prochaine balise qui devient la nouvelle destination intermédiaire du MR-Sub. En répétant ces opérations, le système tente de faire converger le dispositif vers sa destination finale.

2.6.1.1 Prise en compte du mouvement respiratoire et de l'aspect pulsatif de l'écoulement sanguin

Comme présenté à la section 2.1.2, plusieurs contraintes comme le déplacement et la dilatation des vaisseaux lors de la respiration ou l'aspect pulsatif de l'écoulement sanguin en raison du battement cardiaque impliquent la prise en charge de mesures de la part du contrôleur pour limiter autant que possible ces perturbations. Dans le cas de la dilatation et du déplacement des vaisseaux, il faut être en mesure, sans trop diminuer la fréquence d'opération du MR-Sub, de recalibrer les données obtenues lors de la pré-

acquisition avec celles acquises lors de l'opération. Dans ce cas, il devient nécessaire de considérer la prise d'images lors du déroulement de l'opération. En plus de ralentir considérablement le temps de l'opération du MR-Sub, la présence d'artéfacts de susceptibilité magnétique rendrait l'opération de recalage beaucoup plus complexe. Bien que plusieurs méthodes de traitement et d'analyse d'images pour le recalage existent [59, 38, 8, 22, 46], aucune ne prend en considération la complexité supplémentaire que pose la présence d'un corps ferromagnétique dans le volume considéré. Bien que la solution à ce problème dépasse le cadre de ce mémoire, il convient néanmoins d'en faire part et de prévoir la présence d'une éventuelle solution lors de l'édification de l'architecture. L'aspect pulsatif de l'écoulement sanguin est une perturbation qui complexifie le déroulement de l'opération du MR-Sub. En effet, le battement cardiaque modifie constamment la force de traînée que subit le dispositif et implique que celle-ci doit être considérée comme alternative par le contrôleur. Certains contrôleurs sont en mesure de considérer une telle perturbation pour peu qu'ils connaissent la fréquence et l'allure de l'onde cardiaque. Bien que la résolution de cette contrainte sort également du champ d'action de ce mémoire, qui lui porte sur l'édification des fondations en terme d'architecture informatique, il semble important de présenter ici deux solutions envisagées par l'équipe du projet MR-Sub pour compenser une telle perturbation. Une première solution envisagée consiste à évaluer, lors de la pré-opération, l'allure de l'onde cardiaque à l'aide d'un ECG. Une fois l'onde récupérée, celle-ci est considérée similaire tout au long de l'opération. Une estimation de la compensation en fonction des points de l'onde est alors déterminée. La fréquence

d'opération de l'architecture variant très peu, celle-ci est considérée constante et il devient possible de connaître la durée de l'opération séparant deux appels au contrôleur. En synchronisant cette durée avec l'onde enregistrée, on est alors en mesure de connaître, lors de l'appel du contrôleur, la position temporelle du MR-Sub dans l'onde cardiaque et également de prévoir cette position lors de l'application des gradients issues de la commande. Le contrôleur est alors en mesure de compenser entièrement ou en partie du moins et de manière statique, la perturbation occasionnée. Une seconde solution plus dynamique consiste à utiliser un oxymètre pour synchroniser le pouls cardiaque avec une onde de débit de la zone étudiée et préalablement relevé. L'oxymètre compatible IRM utilisé lors des essais préalables procure une sortie de signaux 'transistor transistor logic' ou TTL pouvant être aisément récupérés. L'idée consiste à constamment surveiller le pouls du patient et effectuer une détection de pics sur le signal reçu en vue de la synchronisation lors du passage des sites R à S de l'onde cardiaque. La figure 2.11 suivante illustre une onde cardiaque simplifiée ainsi que ses points communément observés.

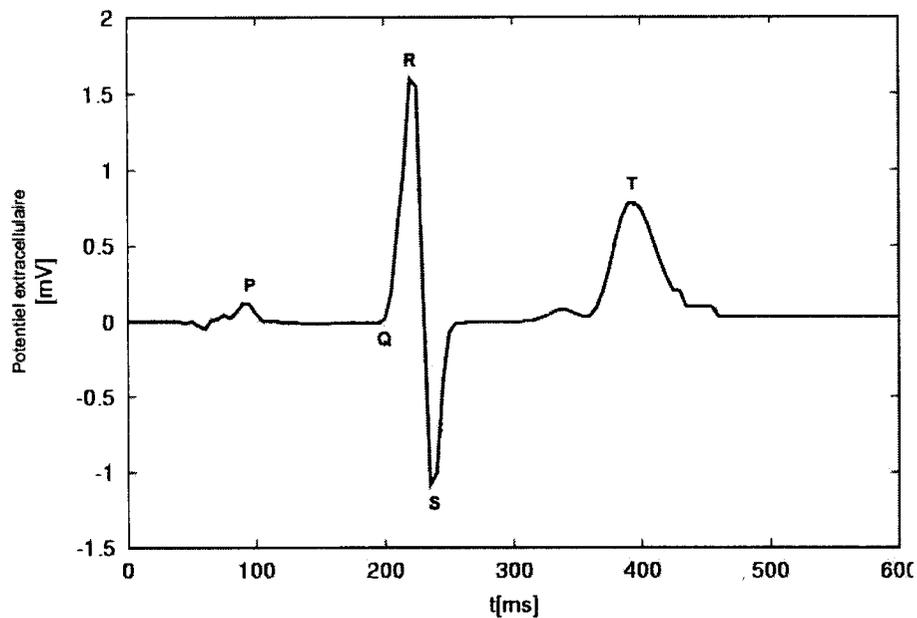


Figure 2.11: Illustration du complexe P-Q-R-S-T de l'onde cardiaque

Lors du début du pic R, un détecteur de crête envoie un signal à la séquence qui envoie à son tour, par le biais d'une communication temps réel, un message au programme de reconstruction où se trouvent les différentes instances des modules dont notamment le module contrôle. Celui-ci est averti de la dépolarisation imminente des cellules cardiaques qui précède de quelques dizaines de millisecondes le changement brusque du débit sanguin. Si l'on suppose que les données du débit sanguin dans la zone considérée ont été récupérés lors de la pré-opération et stockées dans un tableau en vue de leur utilisation par le contrôleur, celui-ci peut alors connaître la valeur du débit sanguin tout en restant synchronisé lors de l'opération.

L'architecture présentée ici est évolutive car bien qu'elle n'incorpore pas encore ces traitements supplémentaires, il ne sera pas nécessaire d'y effectuer d'importants

changements lors de l'ajout de cette phase de synchronisation. La présence d'un gestionnaire central assurant l'interaction entre les différents piliers peut sans problème se voir ajouter d'autres modules qui s'intégreront naturellement dans l'environnement.

2.6.2 Intégration des modules

L'interaction des modules venant d'être présentée, il convient maintenant de placer ceux-ci dans l'architecture IDEA/ICE pour être en mesure d'obtenir une application de navigation du dispositif complètement intégrée dans l'environnement de séquences et du programme de reconstruction d'images. Comme décrit plus haut, ces modules sont placés dans l'environnement de reconstruction en raison de la possibilité de réaction sur la séquence qu'offre cet environnement. La prise de décisions quant à la navigation du dispositif se fera par l'entremise du module contrôle. Les modules de positionnement et de propulsion s'occuperont d'effectuer les tâches demandées par le module de contrôle. Plus précisément, le module de positionnement se chargera de récupérer les données relevées par la séquence de positionnement, d'analyser celles-ci et d'en extraire la position du dispositif. Cette position est ensuite envoyée au module de contrôle qui détermine, suivant l'erreur qui sépare la position actuelle du MR-Sub de la prochaine balise, l'amplitude et la direction du prochain gradient magnétique à appliquer dans la séquence. Une fois déterminée, le module de propulsion se charge d'appliquer cette nouvelle commande à la séquence. La séquence ne fera qu'exécuter la séquence de positionnement ou appliquer le gradient de propulsion magnétique suivant les directions reçues par le biais de communication temps réel existant entre celle-ci et le programme de reconstruction. Une fois cette routine effectuée, il convient de la répéter autant de fois

que nécessaire pour atteindre le point de destination du chemin. L'ensemble de cette procédure formera donc une séquence temps réel qui sera décrite plus en détails à la section 2.10.

De manière à simplifier l'interaction entre les modules qui seront décomposés en classes C++ décrites à la section 2.7, un module central nommé 'gestionnaire de requêtes' se chargera de diriger l'ensemble des interactions entre les différents modules périphériques déjà présentés. La figure 2.12 résume le fonctionnement global des interactions entre les modules ainsi que leur intégration dans l'architecture IDEA/ICE déjà existante.

Les programmes de reconstruction d'images sur la série d'appareils d'imagerie par résonance magnétique clinique de la compagnie Siemens comme l'Avanto utilisé sont écrits en C++, facilitant l'implantation de l'architecture conçue selon un paradigme orienté objet. Comme le montre la figure 2.12 ci-dessous, le gestionnaire central de requêtes est assisté par trois gestionnaires secondaires que sont les gestionnaires d'images, de propulsion et de positionnement. Le gestionnaire central de requêtes est chargé d'interroger les différents gestionnaires périphériques pour l'exécution respectives des tâches de gestion des chemins, de calcul des commandes devant être appliquées et d'extraction de la position du dispositif. Le gestionnaire central est créé lors de l'initialisation de l'objet du programme de reconstruction d'images. Une fois créé, il se charge lui-même de créer les gestionnaires secondaires et d'effectuer les tâches d'initialisation requises à leur fonctionnement. Dans le cas du MR-Sub, ce sont essentiellement les fonctions 'prepare', 'online' et 'hpfeedback' du programme de

reconstruction qui sont modifiées pour assurer la navigation du dispositif. Plus précisément, le gestionnaire de requêtes initialise les différents gestionnaires secondaires dans la fonction 'prepare' et lance la procédure de calcul de la position à chaque appel de la fonction 'Online' par les CANs de la séquence. La fonction 'hpfeedback' est quant à elle appelée par le CAN temps réel où le gestionnaire central de requêtes se charge d'interroger le gestionnaire de propulsion pour déterminer la commande à appliquer en fonction de la distance séparant le dispositif de son objectif. Une fois calculée, la commande est envoyée à la séquence par le gestionnaire central de requêtes pour compléter la boucle de communication temps réel entre la séquence et le programme de reconstruction d'images.

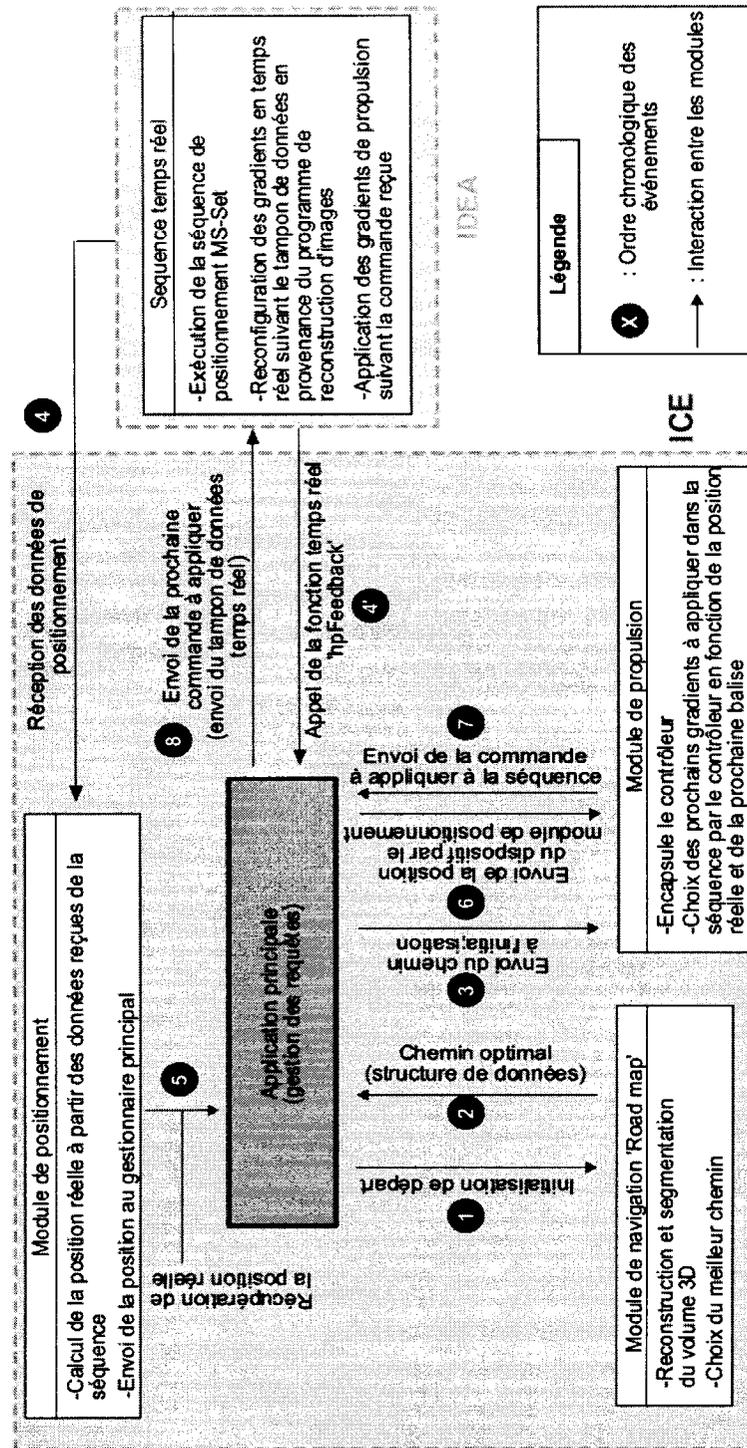


Figure 2.12: Vue globale des interactions et de l'intégration des modules dans l'environnement de développement de l'Avanto

Si l'on résume le fonctionnement global d'une opération typique du MR-Sub en se basant sur la figure 2.12, celle-ci débute par l'initialisation de départ de l'opération qui prépare à la fois la séquence et le programme de reconstruction d'images (noté 1 sur la figure 2.12). Dans ce dernier, la fonction 'prepare' décrite à la section 2.4 va débiter la création du chemin devant être suivi par le module de navigation. Celui-ci envoie le chemin au module de gestion des requêtes (noté 2 sur la figure 2.12) qui va ensuite l'envoyer à son tour au module de propulsion. Le chemin est ensuite copié et envoyé au contrôleur en vue de la navigation prochaine (noté 3 sur la figure 2.12). La séquence débute ensuite par l'exécution d'un CAN pour la communication temps réel suivi d'une section de propulsion qui au début, étant donné que le positionnement n'a pas encore été effectué, possède des gradients d'amplitudes nulles. La séquence se termine par une portion de positionnement qui n'est ni plus ni moins que la séquence MS-SET décrite à la section 1.3.3. A ce moment, deux opérations parallèles s'effectuent. En premier lieu, le CAN utilisé en début de séquence appelle la fonction 'hpfeedback' du programme de reconstruction pour démarrer la boucle temps réel et ainsi évoquer le contrôleur se trouvant dans le module de propulsion (noté 4' sur la figure 2.12). Avant de mettre le MPCU en pause, en attente de la réponse de la communication temps réel avec le programme de reconstruction d'images comme décrit à la section 2.5.2, le CAN du positionnement qui s'est alors exécuté appelle quant à lui la fonction 'Online' du programme de reconstruction (noté 4 sur la figure 2.12). Dans cette routine, se trouve, sous la forme d'un objet C++, le module de positionnement qui se met alors à calculer, à l'aide des projections et d'une opération de corrélation, la position du dispositif. Une

fois la position calculée, celle-ci est envoyée au module de gestion des requêtes qui l'envoie à son tour au module de propulsion en vue de son utilisation par le contrôleur (notés respectivement 5 et 6 sur la figure 2.12). Celui-ci détermine alors l'erreur définie comme la distance séparant, dans chacun des axes orthogonaux x, y et z du référentiel, la position actuelle du dispositif de la prochaine balise. L'algorithme de contrôle détermine, à partir de cette erreur, la commande à appliquer. Cette commande prend la forme de trois valeurs correspondant aux amplitudes des trois gradients dans le repère orthogonal x, y et z de l'IRM. Une fois la commande déterminée, celle-ci est encapsulée dans le tampon de données temps réel par le module de gestion des requêtes (noté 7 sur la figure 2.12). Celui-ci se charge d'envoyer le tampon à la séquence alors que le MPCU est encore suspendu (noté 8 à la figure 2.12). La fonction 'fSEQReceive' analyse le tampon de données et modifie la valeur des gradients suivant la commande reçue. Une fois que la séquence se réveille, elle exécute à nouveau le CAN de la communication temps réel qui appelle à nouveau la fonction 'hpfeedback'. Avant de mettre la séquence en pause à nouveau, la portion de propulsion et de positionnement de la séquence effectuent respectivement leur tâche. La première se charge de propulser le dispositif à l'aide des trois gradients orthogonaux d'amplitudes calculés lors de la dernière communication temps réel et la seconde effectue la séquence de positionnement dont le CAN appelle la fonction 'Online' en vue de déterminer la nouvelle position du dispositif. La procédure continue ainsi jusqu'à l'atteinte de la destination par le MR-Sub.

2.7 Constitution des classes

Les modules présentés aux sections précédentes se doivent désormais d'être décomposés en classes pour être en mesure de formuler l'environnement fonctionnel du MR-Sub. Ces classes seront ensuite mises sous la forme d'objets qui seront inclus dans l'environnement pour refléter le fonctionnement des opérations du MR-Sub décrites à la section 2.6. La transposition des modules en classes sera décrite méthodiquement en utilisant une approche 'bottom up', c'est-à-dire en débutant la description à partir des classes de base visant à modéliser les positions et les chemins d'un volume pour terminer par les classes de haut niveau agissant en tant que gestionnaires des opérations et utilisant les classes de base. Le nombre relativement restreint des classes créées permet la présentation succincte de l'ensemble de celles-ci ainsi que du (ou des) modules auquel (ou auxquels) chaque classe se rapporte. Le diagramme d'interaction des classes entre elles est quant à lui présenté à la section 2.8 sous la forme d'un diagramme de classes.

2.7.1 Classes du programme de reconstruction d'images

2.7.1.1 Classe *MPosition*

La classe *MPosition* permet de modéliser la position du dispositif et des balises composant un chemin. Elle représente une position donnée sous la forme d'une coordonnée dans l'espace formée de trois variables de type 'double'. Toute opération nécessitant l'écriture ou la lecture de la position d'un élément de l'architecture passe directement ou indirectement par l'interaction avec la classe *MPosition*.

2.7.1.2 Classe MChemin

La classe MChemin sert à la modélisation d'un chemin typique pour la navigation du MR-Sub. Cette classe représente un chemin donné par un ensemble d'objets de la classe MPosition. Par le biais d'une liste de la librairie 'Standard Template Library' (STL), une succession de positions sont ainsi regroupées et gérées pour former un chemin issu de l'analyse du volume d'images récupéré lors de l'angiographie initiale ou par la spécification d'un fichier texte contenant les positions du chemin. Chaque chemin permet l'ajout et le retrait d'objets de la classe MPosition pour sa construction et sa modification.

2.7.1.3 Classe MVolumeImage

La classe MVolumeImage se charge de la représentation du volume d'images en vue de son traitement pour la segmentation et la recherche du meilleur chemin éventuel. Sachant qu'il peut exister plusieurs chemins valides dans un même volume d'images, la classe MVolumeImage contient également une liste STL des différents chemins possibles pour un volume donné. Elle permet donc l'ajout et le retrait d'un chemin dans la liste ainsi que la présence d'une fonction pour la recherche du meilleur chemin dans la liste en vue de la navigation.

2.7.1.4 Classe MControleur

Cette classe est l'implantation logicielle du contrôleur élaboré suivant les travaux de l'équipe du projet MR-Sub. Elle contient à la fois l'algorithme de contrôle ainsi qu'une copie du chemin de navigation choisi. La consigne est définie comme étant le prochain point devant être atteint par le MR-Sub lors de la navigation. Bien que similaire

en définition à la balise, la consigne est unique alors qu'il existe généralement plusieurs balises. Au fur et à mesure que le dispositif évolue dans l'environnement et se rapproche à une certaine distance de la prochaine balise, un algorithme d'analyse de proximité estime si la distance séparant la position actuelle du dispositif de la prochaine balise est inférieure à une précision fixée au départ par l'utilisateur. Si tel est le cas, la position de la prochaine balise est copiée dans la consigne du contrôleur qui tente alors de diriger le MR-Sub vers celle-ci. Ainsi, l'objet MControle issu de la classe du même nom ne fait qu'essayer de diminuer l'erreur entre la consigne qu'il possède et la position actuelle du MR-Sub. L'analyse de proximité est effectuée en supposant une sphère de rayon r_p centrée sur la position de la consigne (donc centrée sur la position de la prochaine balise devant être atteinte). Lorsque la position du dispositif relevée par la séquence MS-SET se situe à l'intérieur de la sphère, la consigne est considérée atteinte et la position de la balise suivante est copiée dans la consigne du contrôleur. Le rayon de précision est fixé par l'utilisateur dans l'interface graphique Syngo MR et envoyé via le YAPS au programme de reconstruction d'images. Bien que l'asservissement du dispositif par le contrôleur doit assurer un déplacement précis lors de l'approche des balises, un dépassement de celles-ci est possible. Un placement préalablement étudié des balises assurent déjà que si un dépassement doit s'effectuer, celui-ci ne provoquera pas l'emprunt d'une mauvaise voie dans le système vasculaire. Dans ce cas, un dépassement serait rapidement annulé en ramenant le dispositif à la balise précédente.

2.7.1.5 Classe MPlanificateur

Cette classe est la matérialisation logique du module de gestion des requêtes, elle assure le bon fonctionnement de la navigation du MR-Sub. L'instance de cette classe est unique dans l'architecture et doit permettre la prise en charge de l'ensemble des événements influençant l'opération. Cette classe est en quelque sorte le chef d'orchestre de la procédure devant mener le dispositif à sa destination. Pour se faire, il est subordonné par trois gestionnaires qui sont la traduction des quatre modules présentés jusqu'ici. Ces classes de gestionnaire offrent une interface unique à l'objet MPlanificateur pour accéder aux diverses fonctions subalternes.

2.7.1.6 Classe MGestionnaireImages

Cette classe se situe à un niveau supérieur des classes de base que sont MPosition, MChemin et MVolumeImage. Elle permet de prendre en charge toutes les tâches reliées à la création et la gestion du volume d'images et des chemins le composant tout en offrant une interface unique pour l'agent central qui s'occupe de gérer le déroulement de l'application du MR-Sub comme décrit en 2.7.5.

2.7.1.7 Classe MGestionnairePositionnement

Cette classe gestionnaire offre au planificateur des opérations une interface unique pour accéder aux méthodes de positionnement. Elle permet entre autres d'effectuer l'initialisation de la fonction 'prepare' du programme de reconstruction en plus d'implanter les méthodes nécessaires au positionnement du dispositif une fois les données reçues par le CAN de la séquence MS-SET.

2.7.1.8 Classe MGestionnairePropulsion

La classe *MGestionnairePropulsion* intègre la classe *MControleur* qui se charge du contrôle du dispositif. Elle permet au planificateur central d'interagir avec le module de propulsion et le contrôleur pour les requêtes de contrôle une fois que les CAN de la séquence de positionnement ont récupéré les données.

2.7.2 Classes de la séquence

2.7.2.1 Classe mCommunication

Cette classe se charge, comme son nom l'indique, de la communication temps réel avec le programme de reconstruction. Une fois la séquence démarrée et les fonctions 'fSQEInit', 'fSEQPrep' et 'fSEQCheck' effectuées, la fonction 'Run' de la classe *MCommunication* est appelée. Celle-ci exécute un premier bloc d'événements contenant le CAN pour le démarrage de la boucle de communication temps réel. Une fois la séquence de positionnement et de propulsion effectuée, la fonction 'Run' met le MPCU en pause à l'aide de la fonction 'fRTWaitForWakeup'. Finalement, la fonction test si le retour de la communication temps réel s'est exécuté correctement.

2.7.2.1 Classe mMRsubKernel

La classe *mMR-SubKernel* implante la succession des blocs d'événements pour la portion propulsion et la portion positionnement de la séquence. Elle se charge de la préparation des objets temps réel que sont les gradients, les CANs et les ondes RF à travers les fonctions 'Init', 'Prep' et 'Check'. La fonction 'Run' de cette classe est appelée par la fonction 'Run' de la classe *mCommunication* juste après l'exécution du bloc d'événements comprenant le CAN en vue d'appel de la boucle temps réel. La fonction 'Run' de l'objet *mMR-SubKernel* exécute les deux portions de la séquence du

MR-Sub successivement. Une première portion de propulsion impliquant les gradients en vue du déplacement du dispositif est exécutée, suivie de la portion de positionnement qui comporte trois CANs (un pour chaque axe de positionnement) qui appellent chacun la fonction 'Online' du programme de reconstruction d'images.

2.8 Diagramme de classes

Le diagramme de classes permet, dans le processus logiciel normal, de donner une vue d'ensemble des interactions entre celles-ci afin d'illustrer les modélisations logicielles d'une façon organisée et hiérarchique. Le diagramme de la figure 2.13 illustre à la fois les classes du programme de reconstruction d'images ainsi que celles de la séquence. Les deux environnements dans lesquels sont inscrites ces classes sont clairement illustrés également. On retrouve sur la figure 2.13 les liens de dépendances entre les classes décrites en 2.7.

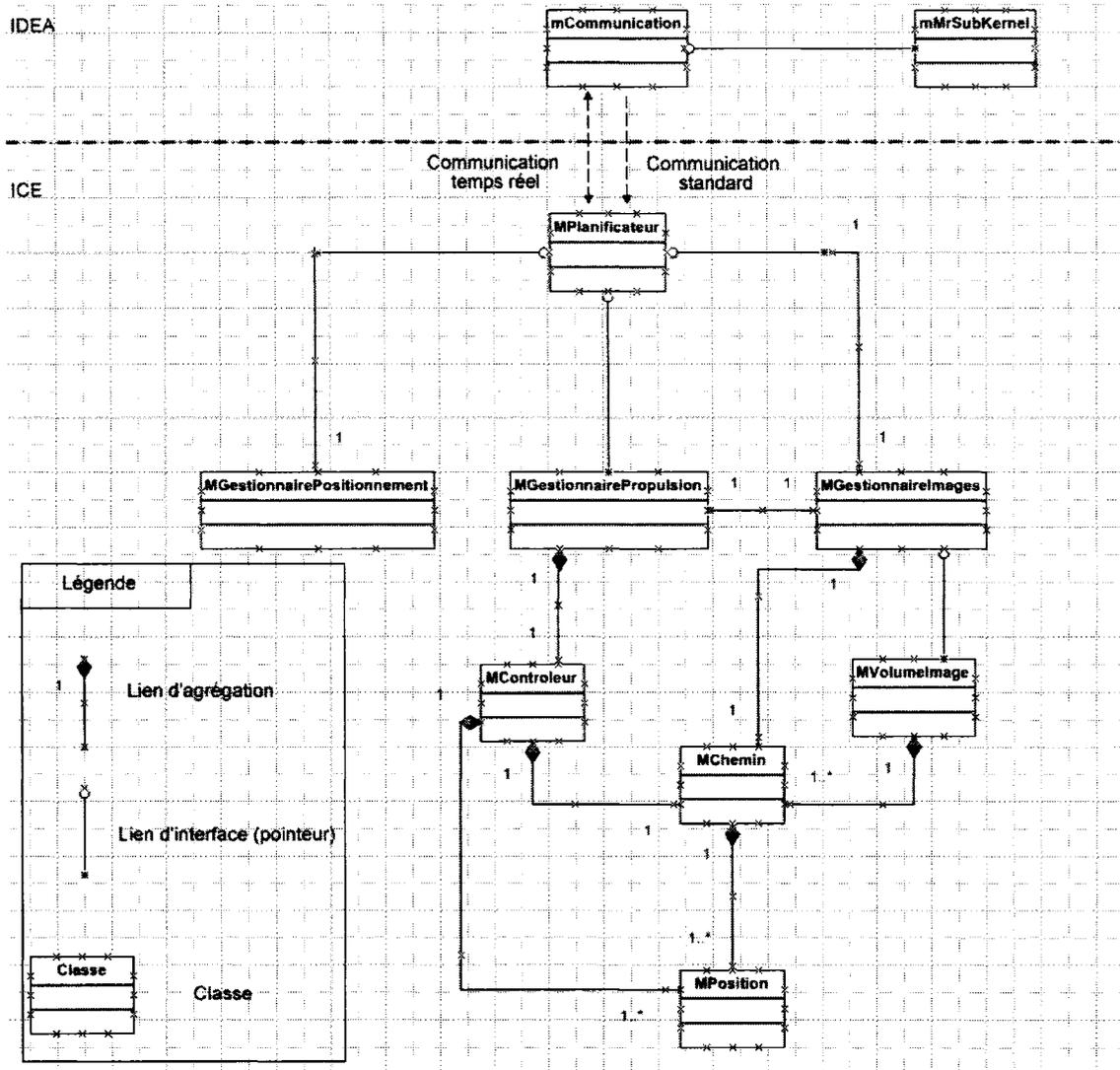


Figure 2.13 : Diagramme de classes de l'architecture logicielle du MR-Sub

La classe MPlanificateur se trouve au sommet du diagramme du côté du programme de reconstruction (ICE) et communique uniquement avec les trois gestionnaires pour orchestrer ses différentes opérations. Elle possède donc trois interfaces que sont les trois gestionnaires implantés à travers la présence de trois pointeurs représentés par le lien d'interface à la figure 2.13.

Le gestionnaire de propulsion possède un pointeur vers le gestionnaire d'images pour être en mesure de récupérer le chemin à suivre avant le démarrage de l'opération. Il possède également un objet de la classe MControleur qui se charge du contrôle lors de la navigation. Lors d'une requête de contrôle, la fonction 'hpfeedback' du programme de reconstruction appelle la fonction 'CommunicationTempsReel' de la classe MPlanificateur qui se charge d'effectuer les opérations de la boucle temps réel. Entre autre, cette fonction lie les positions écrites par la fonction 'Online' précédemment et calcule l'erreur sur chacun des axes x, y et z suivant le référentiel présenté à la figure 2.1. Ces erreurs sont ensuite envoyées au contrôleur par le biais du gestionnaire de propulsion. Le contrôleur renvoie alors les commandes à appliquer à la fonction 'CommunicationTempsReel' qui se charge de les inclure dans un tampon de données et d'envoyer celui-ci à la séquence.

Le gestionnaire d'images quant à lui possède un pointeur sur une instance de la classe MVolumeImage pour être en mesure d'effectuer les différentes tâches reliées à la création des chemins et au choix du meilleur chemin pour la navigation. Il possède également une copie du chemin courant qui sera envoyé à l'instance de la classe MControleur juste avant le départ du MR-Sub.

La classe MControleur possède une copie du chemin courant comme il vient d'être décrit ainsi qu'une instance de la classe MPosition représentant la consigne à atteindre lors du contrôle.

La classe `MVolumeImage` possède une liste STL d'objets de la classe `MChemin` qui regroupe l'ensemble des chemins valides pour la navigation du dispositif dans le volume considéré.

Comme décrit à la section 2.7.2, la classe `MChemin` comporte une liste STL d'objets de la classe `MPosition` afin de représenter les chemins du volume comme une succession de balises représentées par trois coordonnées dans l'espace, modélisée par la classe `MPosition`.

Du côté de la séquence, la classe `mCommunication` possède un pointeur vers la classe `mMR-SubKernel` en vue d'appeler la méthode 'Run' de celle-ci qui provoquera l'exécution de la séquence d'événements de propulsion et de positionnement. De plus, cette classe se charge de la communication temps réel avec la classe `MPlanificateur` du programme de reconstruction d'images par l'entremise d'un bloc d'événements contenant un CAN configuré pour appeler la fonction temps réel 'hpfeedback'.

2.9 Diagrammes de séquence

Le diagramme de classes ne permet pas de représenter l'aspect temporel de la succession des événements et des messages. Le diagramme de séquence a été conçu précisément pour mettre en valeur l'aspect chronologique de l'exécution des tâches incluses dans une opération bien précise. Une vue de la séquence illustrant la préparation des différents objets préalablement au déroulement d'une séquence typique de positionnement, contrôle et propulsion est illustré à la figure 2.14 ci-dessous. Un second diagramme illustrant l'exécution d'une telle séquence est montré à la figure 2.15. Ces

deux diagrammes permettent de mieux saisir la succession des interactions entre les objets de l'architecture.

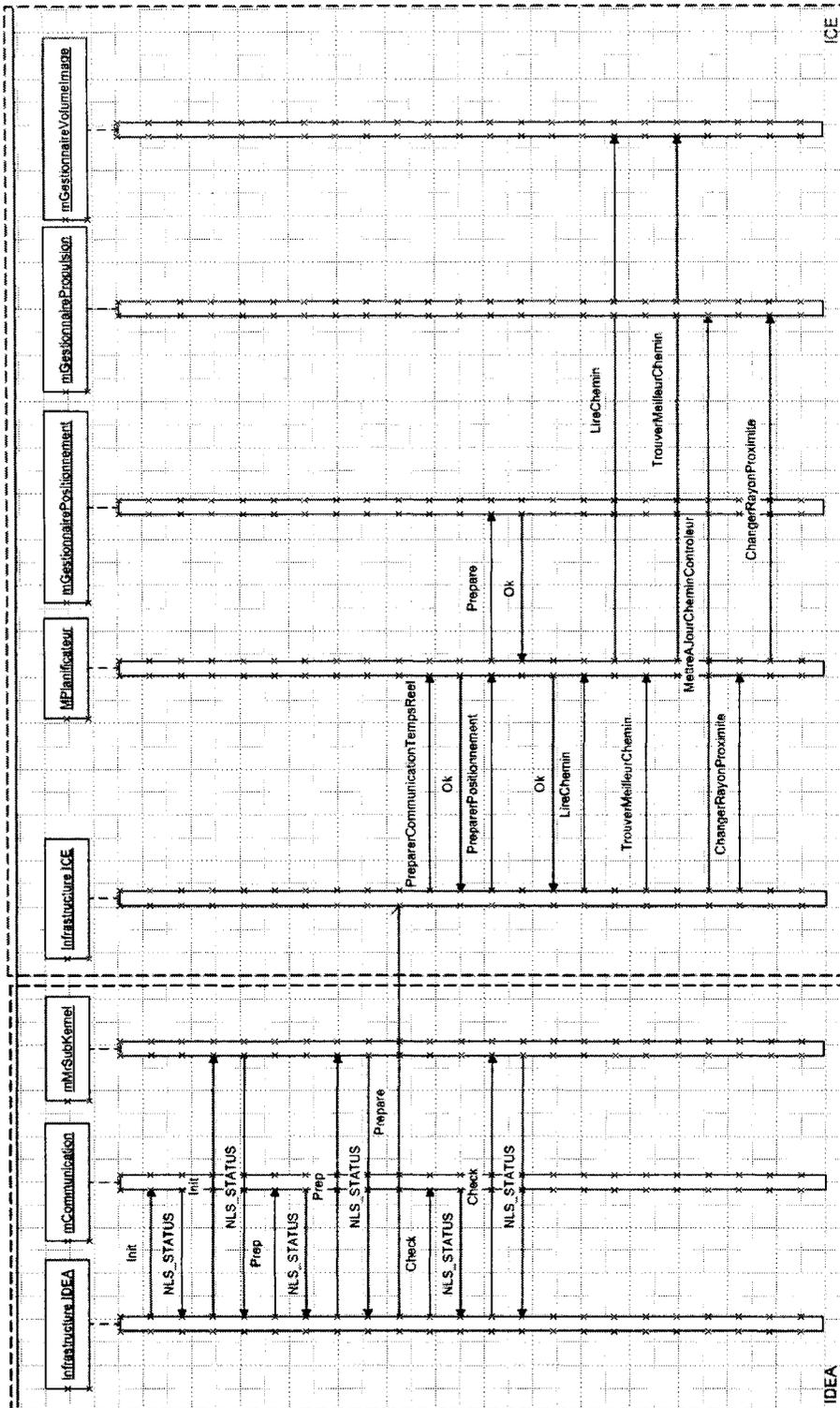


Figure 2.14 : Diagramme de séquence de la préparation du système

2.10 Validation et tests de l'architecture

Dans cette section, nous présentons un bref survol des différentes méthodes de validation et test de logiciel. Selon [40], il existe essentiellement deux approches pour la phase de test d'un logiciel, quelque soit son envergure et ses requis, une approche dite 'boîte noire' ('black box') et une approche 'boîte blanche' ('white box'). L'approche 'boîte blanche' suppose que l'on veut assurer le bon fonctionnement interne de l'ensemble des opérations et des méthodes d'un logiciel. Ceci s'effectue généralement en utilisant des tests qui tenteront d'évaluer l'ensemble des chemins possibles ainsi que l'ensemble des boucles dans leurs limites respectives en plus de valider les différents états des données internes se trouvant dans le module étudié. Bien que cette approche semble mener vers une couverture complète des cas possibles dans un contexte de test logiciel, ce n'est généralement pas le cas. L'approche 'boîte noire' quant à elle, met l'emphase sur le requis de la fonctionnalité du logiciel. Les tests issus de cette approche tentent généralement, à partir d'un ensemble de données initiales, d'assurer la fonctionnalité du ou des modules étudiés. Ces deux approches ne proposent pas deux techniques alternatives mais plutôt deux techniques complémentaires qui permettent de déceler différentes classes d'erreurs.

2.10.1 Approche 'boîte blanche'

Dans l'approche 'boîte blanche' plusieurs tests sont largement répandus. Seuls quelque uns d'entre eux sont décrits dans les prochaines lignes. Le lecteur désirant en connaître davantage est encouragé à consulter [40].

2.10.1.1 Complexité cyclique

Ce test permet de quantifier la complexité logique d'un programme. A la suite de ce test, une valeur représentative du nombre de chemins indépendants pouvant être empruntés en parcourant une exécution donnée du programme est donnée. On entend par 'chemin indépendant' une exécution impliquant la présence d'une nouvelle instruction ou d'une nouvelle condition. On représente alors souvent les programmes sous la forme de graphes de flux où chaque nœud est une instruction ou une décision indépendante. L'exécution du programme mène donc au parcours de plusieurs nœuds dans un ordre bien précis. Une fois la complexité cyclique obtenue, on connaît alors la limite supérieur en terme de chemins indépendants et par la même occasion, la limite supérieur en terme de nombre de tests à exécuter pour garantir une couverture complète de l'ensemble des instructions du programme. Ce nombre est généralement important, ce qui rend souvent la couverture complète de l'ensemble des chemins indépendants irréalisables. Pour en connaître davantage, le lecteur peut consulter [40].

2.10.1.2 Test des conditions

Sachant que beaucoup d'erreurs de programmation se retrouvent généralement dans les instructions de conditions, certains tests permettent d'assurer la qualité de ceux-ci. Les instructions conditionnelles sont généralement représentées par des expressions booléennes comme '<, >, =, !=, <=, >=' . L'objectif du test de conditions d'un programme n'est pas de seulement détecter les erreurs directement liées au résultat de la condition mais également les autres erreurs d'un programme. En effet, si un ensemble de tests pour un programme est efficace pour la détection d'erreurs dans les instructions

conditionnelles, cet ensemble est probablement également apte à détecter d'autres erreurs dans le programme en question.

2.10.2 Approche 'boîte noire'

Le rôle des tests de type 'boîte noire' est de détecter des erreurs de type bien précis. Ainsi, les fonctions incorrectes ou manquantes, les erreurs d'interface, les erreurs de structure de données ou les données externes sont des exemples d'erreurs que les tests de type 'boîte noire' sont en mesure de détecter. Contrairement aux tests 'boîte blanche' qui sont souvent effectués en début de phase de test, les tests 'boîte noire' sont généralement effectués à la toute fin de la phase de validation étant donné qu'ils ne préoccupent

2.10.2.1 Tests par graphes

Dans un modèle orienté objet, il convient souvent d'assurer la bonne communication entre ceux-ci et vérifier que chacun de ces objets possède bien les requis pour agir comme il se doit. Pour se faire, on établit généralement un graphe dont les nœuds sont les différents objets, les liens, l'interaction entre les objets et le poids des nœuds, décrit les propriétés de l'objet. Chaque interaction entre les nœuds est étudiée pour en tirer une série de tests. Par transitivité entre les objets, une batterie de tests désirant cibler un aspect de la relation entre deux nœuds devra également tester tous les prérequis à l'obtention de cet aspect. De cette manière, un test assurera à la fois la validation d'un lien en particulier entre deux objets mais également d'autres liens implicites nécessaires à l'obtention du critère voulant être testé. A nouveau, le lecteur peut se rapporter à [40] pour en apprendre davantage.

2.11 Allure de la séquence temps réel du MR-Sub

Si l'on résume le déroulement de la séquence temps réel permettant le déplacement, le positionnement et le contrôle du dispositif après lui avoir fourni un chemin de navigation donné, nous obtenons le schéma séquentiel de la figure 2.16 ci-dessous. Elle présente la succession des événements tout au long du déroulement de la séquence aussi bien du point de vue de IDEA que du côté du ICE. La séquence MR-Sub illustrée est celle qui est utilisée pour l'obtention des résultats présentés au chapitre 3. Le temps de la portion de propulsion est donné selon :

$$t_{propulsion} = t_{feedback} - t_{positionnement} \quad (2.5)$$

Où $t_{feedback}$ est le délai temps réel pendant lequel le MPCU est arrêté et $t_{positionnement}$ est le temps d'exécution de la séquence de positionnement MS-SET. Ce temps est constant et vaut $t_{positionnement} = 15$ ms. Ainsi, fixer la valeur $t_{feedback}$ revient à spécifier indirectement le temps pendant lequel le dispositif est propulsé. Le délai temps réel $t_{feedback}$ minimum pouvant être utilisé est très dépendant de la complexité du code de la fonction 'hpfeedback' du programme de reconstruction d'images. Cette fonction hébergeant le contrôleur ainsi que la fonction d'analyse de proximité pour le passage des balises, il a été expérimentalement impossible de fixer le délai temps réel $t_{feedback}$ à moins de 30 ms. En fait, selon la compagnie Siemens, il est difficilement envisageable de descendre au-delà de 12 à 15 ms et ce, même pour une fonction 'hpfeedback' ne contenant aucune instruction en raison des délais de communication et de changement de contexte entre le

MRIR et le MPCU. Par ailleurs, l'augmentation du délai temps réel provoque également une augmentation du temps de propulsion selon l'équation 2.5 et ne peut être trop élevé en raison des limitations des amplificateurs de courant alimentant les bobines de gradient. Si tel est le cas, la séquence est alors arrêtée automatiquement durant son exécution par le système pour empêcher une surchauffe des amplificateurs. En pratique, il n'est donc pas possible d'augmenter ce délai temps réel à plus de 30 ms si l'on suppose une amplitude maximale des gradients de 40 mT/m.

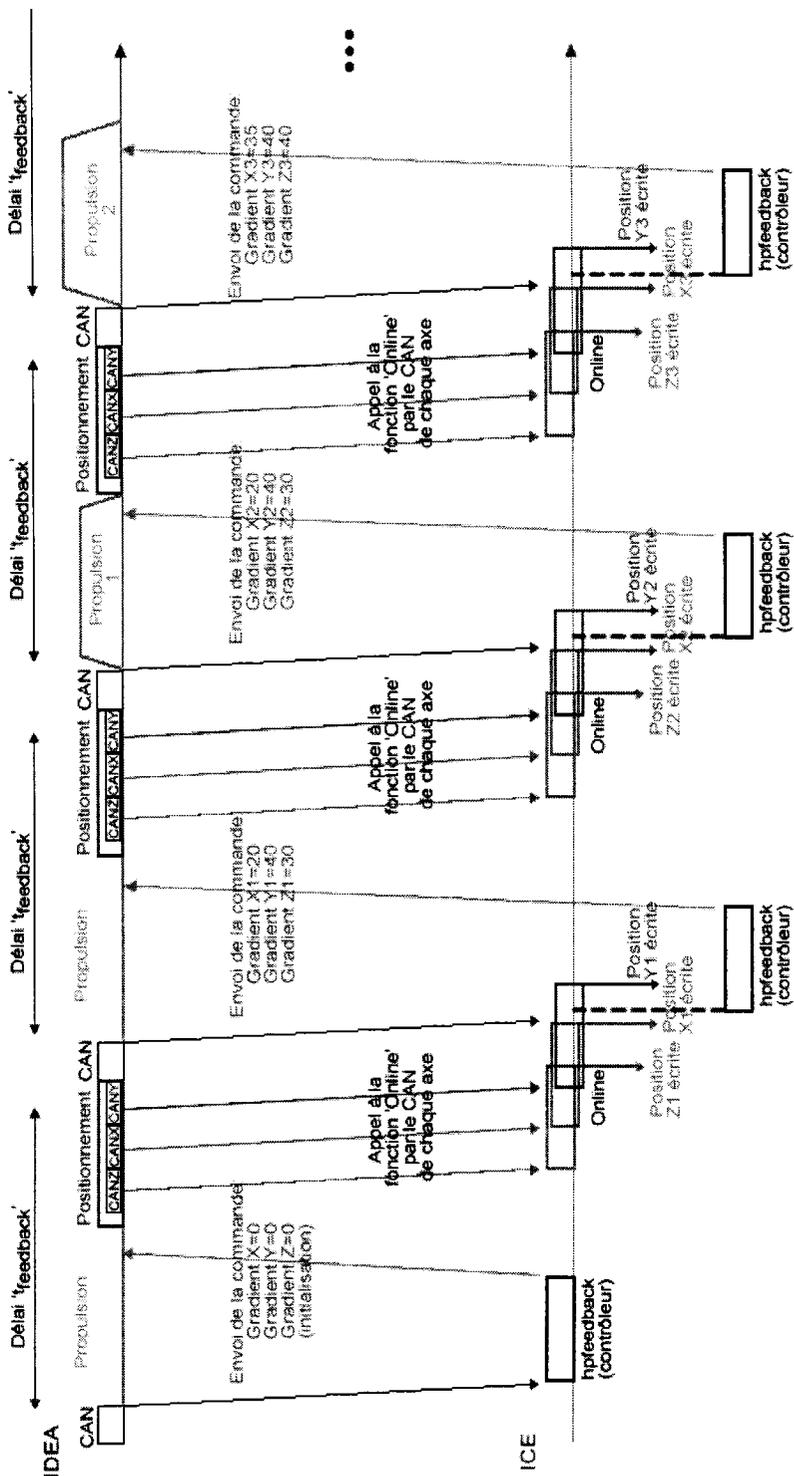


Figure 2.16 : Allure globale de la séquence temps réel pour la navigation du MR-Sub

Conclusion

Dans ce chapitre, nous avons introduit les contraintes imposées par l'environnement complexe qu'est le système vasculaire humain. Nous avons également présenté la plateforme logiciel de l'IRM avec les environnements IDEA et ICE ainsi que l'aspect temps réel nécessaire à la navigation d'un dispositif comme le MR-Sub. Le principe des séquences a été brièvement introduit en présentant les différentes fonctions nécessaires à leur fonctionnement. La description des différents modules issus des piliers ainsi que leur mise en commun autour d'un gestionnaire central a également été présentée. L'intégration de ces modules dans les environnements ICE et IDEA permettant l'édification d'un système de navigation autonome a également été décrite. Finalement, la constitution et la description des classes nécessaires au développement de la séquence temps réel ont été présentées. Maintenant l'architecture présentée, il convient de décrire au prochain chapitre la méthodologie et les résultats obtenus en IRM lors des tests expérimentaux de navigation.

CHAPITRE 3. ANALYSE ET RÉSULTATS

Introduction

Les classes ainsi que l'environnement de l'architecture étant maintenant décrits, il convient désormais d'introduire dans un premier temps la méthodologie ainsi que les tests effectués durant ces deux dernières années. Dans un second temps, les résultats découlant des tests effectués en IRM seront présentés, accompagnés de la description des montages utilisés. Bien que les résultats présentés ci-dessous soit avant tout des résultats de contrôle du dispositif dans différents contextes, ceux-ci n'auraient pu être obtenus sans la présence d'un système de navigation reposant sur l'interaction des modules présentés au précédent chapitre. Les courbes de contrôle données ci-dessus sont parfois discutables en terme de choix et de précision du contrôleur dans l'atteinte des contraintes physiques mais demeurent néanmoins d'excellents témoins du bon fonctionnement et de la bonne interaction des modules développés et intégrés. En effet, les résultats ci-dessous sont ceux d'une bille dont les déplacements contrôlés sous certaines conditions ne sont assurés que par une interaction adéquate des modules de propulsion, de positionnement et de 'Roadmap'. Une discussion des ces résultats ainsi qu'une section portant sur les travaux futurs termineront ce mémoire.

3.1 Méthodologie

Dans cette section, nous décrivons les différentes étapes au cours de ces deux dernières années qui ont permis la mise au point de l'infrastructure et de l'architecture logiciel du MR-Sub. Cette méthodologie est construite de manière chronologique et permet d'observer l'avancement des requis et les étapes singulières de l'intégration et de la création des modules. Étant donné que la validation de l'architecture mise au point devant permettre une première navigation simple du dispositif devait se faire le plus rapidement possible dans les échéanciers du projet, c'est précisément cet aspect qui est avant tout traité. La mise au point du module 'Roadmap' bien que nécessaire à l'obtention d'une application complètement autonome ne faisait pas partie de la preuve de concept initiale et a donc débutée une fois les premiers piliers de propulsion, contrôle et positionnement intégrés.

En premier lieu, l'analyse des besoins de l'intégration a permis la mise sur pied d'une première architecture logicielle incluant les classes MPosition, MChemin, MVolumeImage, les trois gestionnaires et le planificateur central. Cette première ébauche fut testée individuellement pour valider la création et la gestion de chemins. Bien entendu, il n'était alors pas encore question de tests en IRM mais seulement de la création d'un squelette de classes qui était testé dans l'environnement de développement Microsoft Visual Studio au Laboratoire de Nanorobotique. Cette première phase de l'architecture devait permettre de modéliser une première version du module 'Roadmap' à travers les classes MPosition, MChemin, MVolumeImage, MGestionnairePropulsion et MPlanificateur. A travers les différents tests effectués, la construction d'un chemin

virtuel ainsi que le passage des balises dans ce chemin ont permis de valider cette première ébauche. Chacune des méthodes des classes a été testée individuellement à l'aide d'un programme pilote qui fournissait l'interface entre les objets des classes. Par la suite, une fois l'ensemble des classes implantées, des tests de validité d'ensemble ont été effectués. Pour se faire, un programme pilote permettant de construire un chemin à partir d'une liste d'objets MPosition a été utilisé. Les méthodes d'insertion, de destruction et de recherche de balises dans le chemin ont ensuite été testées à travers les méthodes d'un objet de la classe MPlanificateur qui sera sensé s'occuper de l'ensemble des requêtes et du bon déroulement des opérations dans l'architecture finale greffée à l'environnement IDEA/ICE.

Par la suite, il a été nécessaire de prendre connaissance du contexte matériel qu'est l'IRM en plus de se familiariser avec l'utilisation, la structure et le développement de séquences IRM. Cette phase de renseignement s'est essentiellement faite à l'aide de la documentation de l'environnement de développement de Siemens que comprend IDEA et ICE. Une fois les outils de simulation, de compilation et de visualisation des séquences connus, il a été nécessaire d'analyser le contenu d'une séquence IRM typique et d'y repérer où se placeraient les classes MCommunication et MRSubKernel en vue d'une première intégration dans une séquence officielle. En parallèle, un programme de reconstruction standard (IceProgram2D) a été analysé en vue d'en déduire où et comment intégrer l'architecture testée préalablement. Les premières séquences d'imagerie furent donc compilées et simulées dans IDEA grâce à l'environnement de développement externe fourni par la compagnie. Cela dit, comme

mentionné au chapitre 2, cet environnement n'assure en rien le bon déroulement sur l'IRM d'une séquence correctement simulée au Laboratoire. Il a donc fallu effectuer les premières modifications de séquence pour ensuite les tester avec l'appareil de résonance magnétique présent au pavillon Notre-Dame du centre hospitalier de l'Université de Montréal (CHUM).

L'accès limité aux ressources fut sans contredit la contrainte majeure de la phase d'intégration des modules. En effet, il a été possible généralement d'obtenir 2 à 4 heures d'accès à l'IRM par semaine suivant les disponibilités des intervenants. A ce rythme, l'élaboration de la communication temps réel ainsi que l'intégration des classes du MR-Sub dans un programme de reconstruction d'images se sont allongés considérablement en terme de temps de développement. Il a fallu plusieurs visites à l'IRM pour obtenir une séquence qui se compile correctement sur la machine 'host' de l'environnement. Une fois ces premiers tests effectués, il a fallu ensuite se pencher sur le fonctionnement et l'utilisation de l'aspect temps réel de la séquence. La documentation officielle de Siemens étant très vague à ce sujet, il était impossible d'obtenir une communication entre le programme de reconstruction développé et la séquence après plusieurs essais. Il a donc été décidé de faire venir un des membres de l'équipe de développement de l'imagerie interventionnelle de Siemens qui nous a éclairé sur le sujet. Peu de temps après, il était alors possible de communiquer entre ces deux entités. A partir de ce moment, il n'était plus possible de valider le fonctionnement de l'architecture au laboratoire car l'environnement de simulation ne permet pas de simuler la communication temps réel entre un programme de reconstruction et une séquence. Cela

dit, la fréquence des visites à l'IRM restant toujours limitée, il a fallu trouver une solution pour assurer un fonctionnement rapide de l'architecture du côté du programme de reconstruction sans toutefois avoir accès à l'IRM. Il a donc été choisi de simuler la communication temps réel dans l'architecture du laboratoire à l'aide de deux fonctions qui ont été insérées dans la séquence. Ces fonctions ne sont exécutées que lorsque que l'on désire tester l'architecture au laboratoire de Nanorobotique. Une de ces fonctions appelait la routine du contrôleur, au même endroit où était lancée la procédure de communication temps réel dans la séquence de l'IRM. A partir de ce moment, bien qu'il était toujours impossible d'assurer que ce qui était développé en terme de routine de contrôle respectait les délais de communication temps réel sur l'IRM, il devenait néanmoins possible de valider les routines de contrôle. Bien que les objets des classes du programme de reconstruction n'étaient pas présent, cela permettait de valider l'aspect contrôle du processus. Ainsi, à partir des positions de simulation du contrôleur, l'asservissement était testé au laboratoire de Nanorobotique. Les visites à l'IRM ont ensuite permis de propulser, pour la première fois, le dispositif dans un tube de plexiglas avec une succession de gradients magnétiques appliqués dans une direction x sans toutefois être en présence d'un contrôleur pour guider celui-ci.

Par la suite, l'introduction d'un premier contrôleur 1D a permis l'obtention de premiers résultats de contrôle assurant du coup le fonctionnement de l'architecture et de la communication temps réel entre les différents modules impliqués dans la navigation du dispositif. Ces résultats sont présentés à la section 3.2 accompagnés des montages utilisés. L'équipe fut alors en mesure pour la première fois de contrôler une bille

ferromagnétique dans un tube de plexiglas sur une distance de 10 cm grâce à l'architecture développée et validée en partie au laboratoire et en partie à l'IRM. A ce moment, les méthodes des classes `MPlanificateur`, `MVolumeImage`, `MGestionnairePropulsion`, `MGestionnairePositionnement` et `MGestionnaireImages` développées au laboratoire ont pu être validées par la pratique. Par la suite, le développement et l'implantation d'un contrôleur PID 2D a permis d'effectuer les premiers tests en boucle ouverte et en boucle fermée en deux dimensions, assurant le déplacement de la bille sur une plaque de plexiglas. Le système n'étant pas encore au point, il était alors uniquement possible de spécifier au contrôleur une consigne unique sous la forme d'une position 2D (x, y) que l'on désirait atteindre. A nouveau, les résultats de ces tests sont détaillés à la section 3.2 ci-dessous.

Par la suite, l'ébauche du module 'Roadmap' développé a été validé. Pour se faire, une routine de lecture de positions dans un fichier a été ajoutée à la classe `MVolumeImage`. Le fichier contient essentiellement deux types d'information. La première ligne indique le nombre de balises contenues dans le chemin et les lignes suivantes sont des triplets représentant la position (en mètres) dans l'espace des balises à atteindre. Ce fichier est ensuite placé dans un répertoire spécifique sur le MRIR où il est lu et chargé en mémoire lors de l'appel à la fonction 'prepare' du programme de reconstruction d'images. Ces balises sont alors inscrites dans une liste STL en vue de la navigation du dispositif. Un exemple d'un tel fichier est illustré à la figure 3.2. La génération de ces points devant refléter un déplacement réel, il fallait ensuite être en mesure de posséder un outil d'analyse d'images dans lequel il serait possible de

sélectionner un chemin à parcourir à partir d'une image DICOM obtenue sur l'IRM et d'en exporter le contenu dans le fichier lu au démarrage. Pour se faire, il a été décidé d'opter pour une solution offrant un gain de temps substantiel considérant le temps restant pour le développement de ce module. Ainsi, un logiciel offrant déjà une interface graphique ainsi que des fonctionnalités de visionnement et de manipulation d'images DICOM a été recommandé. Après quelques recherches le logiciel VolView de la compagnie Kitware [26] a été retenu. Ce logiciel offre une version d'essais de 30 jours à partir de laquelle nous avons été en mesure de visualiser et prévoir les déplacements dans un fantôme vasculaire préalablement imagé en IRM en vue des tests de contrôle avec balises. Une fois le volume en format DICOM chargé, il est possible d'obtenir une vue isométrique ainsi que trois vues orthogonales de celui-ci. On peut ensuite, à l'aide de l'interface usager fournie, insérer une à une des balises dans le volume en cours pour finalement exporter celle-ci dans un fichier qui est lu au démarrage de la navigation du MR-Sub en IRM. La figure 3.1 montre l'allure de l'application VolView illustrant un fantôme vasculaire de carotide humaine utilisé lors des test de contrôle avec chemin. On peut également voir sur la figure 3.1 une succession de balises (sous la forme de sphères) placées par l'utilisateur en vue de la construction d'un chemin devant être suivi par le dispositif en IRM. Les premiers tests de contrôle utilisant activement le module 'Roadmap' développé ont alors débutés. Le passage des balises chargées au démarrage à partir du fichier est assuré par un algorithme d'analyse de proximité ajouté dans la classe MGestionnairePropulsion. Cet algorithme analyse la position du dispositif pour vérifier si elle se trouve dans une sphère virtuelle dont le centre et la coordonnée de la prochaine

balise à atteindre et dont le rayon est fixé par l'utilisateur à travers le protocole de départ. Si la position du MR-Sub se trouve dans cette sphère, le gestionnaire de propulsion charge alors la prochaine balise en tant que nouvelle consigne pour le contrôleur. Cette routine est bien entendue effectuée à chaque appelle de la routine du contrôleur. Dans les premiers essais, un déplacement unidirectionnel sur l'axe x ont permis, après quelques ajustement, d'obtenir un premier déplacement contrôlé suivant un chemin pré-établi. A partir de ce moment, les tests de contrôle en une dimension et deux dimensions se sont constamment appuyés sur ce système de 'Roadmap' simple mais efficace.

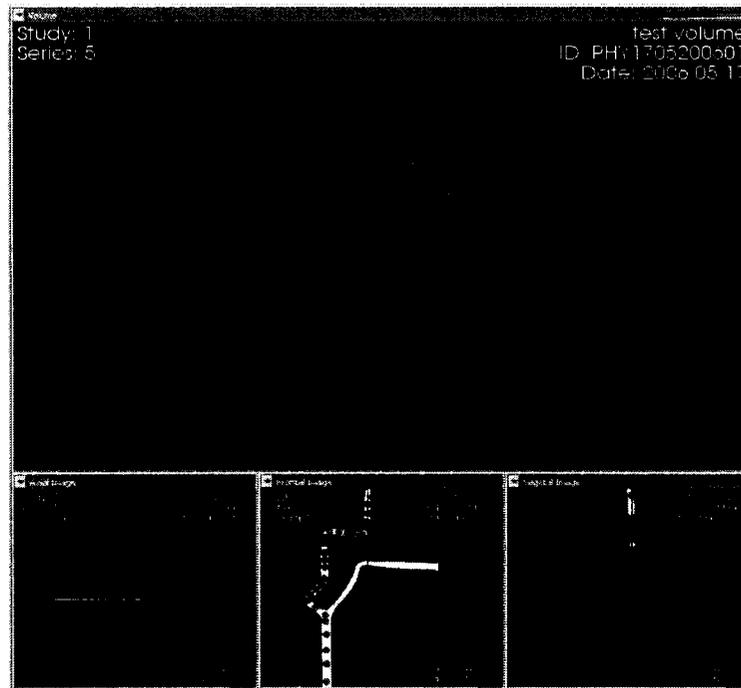


Figure 3.1 : Allure du fantôme vasculaire sous VolView et d'un ensemble de balises constituant un chemin à suivre pour le dispositif

10		
0.01	0.0	0.001
0.01	0.0	0.002
0.01	0.0	0.003
0.02	0.0	0.004
0.02	0.0	0.005
0.02	0.0	0.008
0.02	0.0	0.01
0.02	0.0	0.012
0.02	0.0	0.013
0.03	0.0	0.015

Figure 3.2 : Exemple d'un fichier contenant un chemin de 10 balises (positions en mètres)

Une fois combinés, le contrôleur 2D ainsi que le système de navigation par balises permirent alors au projet MR-Sub de passer à une étape cruciale de son développement qu'est l'habileté du dispositif à être contrôlé à travers un chemin prédéterminé par l'utilisateur. Plus détaillée à la section 3.2, cette étape a permis la réalisation de tests de contrôle 2D en boucle fermée à la fois sur une plaque de plexiglas et dans un fantôme vasculaire suivant un chemin prédéterminé par l'utilisateur.

3.2 Résultats

Dans cette section, nous présentons les résultats obtenus en terme de contrôle du dispositif et en terme de fréquence d'opération de la séquence pour la navigation du MR-Sub. Bien que dans le cas d'une architecture logicielle, il soit difficile de présenter des mesures physiques témoignant du bon fonctionnement de celle-ci, les résultats concluants en terme de contrôle du dispositif assurent à la fois le bon fonctionnement de l'architecture implantée ainsi que la preuve de concept du fonctionnement d'un tel dispositif dans un environnement ex-vivo. Comme mentionné en introduction, ces résultats de contrôle ne sont rendus possibles qu'en raison d'une communication adéquate entre les différents modules de l'architecture permettant la propulsion et le

positionnement du dispositif ainsi que le suivi des balises grâce au module 'Roadmap'. Bien entendu, la présence d'un contexte *in vivo* pour la navigation du dispositif ajoute plusieurs contraintes supplémentaires au contrôle de celui-ci. Cela dit, aucune de ces contraintes ne devrait cependant influencer l'architecture présentée car celles-ci se devront d'être prises en compte par le contrôleur qui est un module très flexible dans le système. Seule la synchronisation avec une ECG ou un oxymètre qui n'a, à ce jour, pas encore été incluse dans l'architecture décrite pourrait influencer et modifier celle-ci.

Dans les lignes qui suivent, les résultats de contrôle 1D et 2D sont présentés. Les courbes présentées sont tirées de [51]. Dans le cas des résultats en 1D, il convient de distinguer trois types que sont l'environnement sans écoulement, l'environnement avec écoulement constant et l'environnement avec écoulement pulsé. Des tests impliquant le suivi d'une succession de balises ont été réalisés dans le cadre d'un environnement sans écoulement. Pour ce qui est des résultats en 2D, nous présentons des données dans un environnement sans écoulement avec et sans balises.

En raison de la limitation en terme d'intensité des bobines de gradient utilisées comme décrit à la section 2.1.2, il est encore impossible d'effectuer un contrôle en trois dimensions du dispositif. La lévitation et la propulsion verticale de celui-ci n'étant pas possible, aucun test de contrôle en trois dimensions n'ont été effectués. L'architecture a cependant été développée dans un esprit de continuité et a assuré dès le début, un fonctionnement supportant les trois dimensions.

3.2.1 Résultats de contrôle 1D

3.2.1.1 Montage et paramètres

Le montage utilisé pour effectuer les tests de contrôle en une dimension consiste en un tube de polyméthacrylate de méthyle (PMMA) communément appelé plexiglas de 15 cm de long. Celui-ci est retenu par un support également fait de PMMA. Le tout est placé dans un bac de plastique et rempli d'eau. Les deux extrémités du tube sont munies d'un filtre empêchant la bille de s'échapper. La figure 3.3 suivante illustre l'allure du tube dans lequel sera placée la bille ferromagnétique.

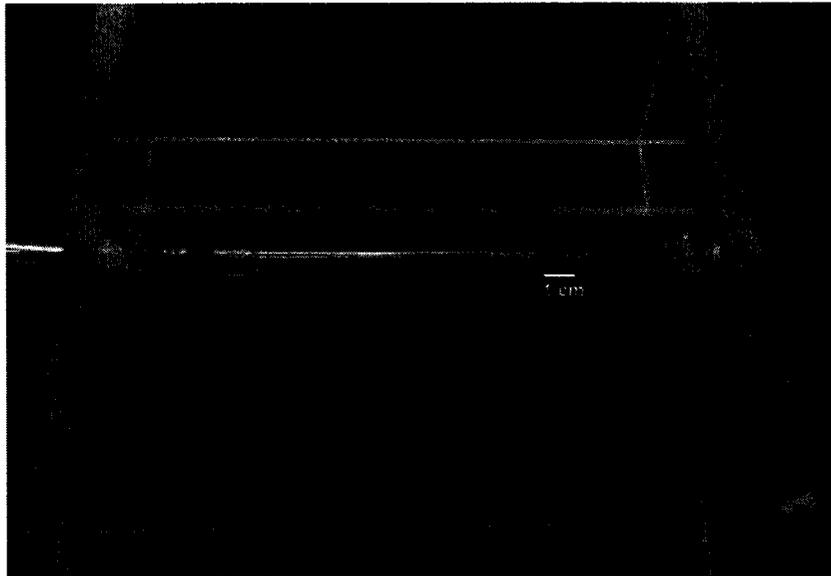


Figure 3.3 : Montage utilisé pour les tests de contrôle en une dimension

La bille faisant office de dispositif est constituée d'un alliage de chrome et d'acier ayant une magnétisation à saturation $M_{bille} = 1.35 \times 10^6 \text{ A/m}$ et un diamètre

$d_{bille} = 1.5$ mm. Celle-ci est placée dans le tube rempli d'eau qui est lui-même placé avec le bac qui le contient, au centre du scanner.

Dans le cas des tests avec écoulement constant, une pompe placée à l'extérieur de la salle du scanner est reliée par des tubes flexibles de PMMA au montage. Elle alimente celui-ci avec de l'eau se déplaçant à vitesse constante à travers le tube. Le débit circulant dans le système est réglé par une valve reliée à la pompe.

Pour ce qui est des tests avec écoulement pulsé, une pompe cardiaque programmable et compatible IRM de la compagnie Shelly [47] est utilisée. Celle-ci est en mesure de reproduire l'allure pulsée de l'écoulement sanguin que l'on retrouve dans plusieurs zones du corps humain telles la carotide ou l'artère fémorale. Dans les résultats présentés à la section 3.2.1.4 ci-dessous, une onde simulant l'écoulement dans l'artère fémorale est utilisée.

Le contrôleur utilisé pour les tests présentés ci-dessous est un PID tel que décrit selon [54, 52]. La valeur des gains est spécifiée dans un tableau qui accompagne les résultats. Le délai temps réel de la séquence noté $t_{feedback}$ sur la figure 2.16 est fixé à 30 ms. Comme détaillé à la section 2.10, ce temps est le minimum fonctionnel pour la complexité de la fonction 'hpfeedback'. Par ailleurs, les paramètres du protocole de la séquence MR-Sub influençant la qualité du positionnement sont donnés au tableau 3.1 suivant. La signification est l'impact de ces paramètres sortant du cadre de ce mémoire, le lecteur peut se rapporter à pour une explication approfondie de ces termes.

Par ailleurs, bien que la lévitation complète de la bille dans la direction verticale soit impossible avec les bobines de gradients présentes sur l'Avanto utilisé, certains

résultats illustrés ci-dessous font mention de la présence d'un gradient constant dans la direction y du repère de l'IRM afin de faciliter la mise en mouvement de la bille dans la direction de déplacement. En effet, la présence d'une telle force diminue l'impact du poids de celle-ci et facilite sa mise en mouvement.

Tableau 3.1 : Paramètres de la séquence de positionnement

Angle de flip (deg)	22
Fréquence d'offset (Hz)	1000
Amplitude des gradients 'spoilers' (mT/m)	2
Délai temps réel (ms)	30

3.2.1.2 Contrôle sans écoulement

Le premier résultat présenté est un déplacement le long de l'axe x. Une consigne de 6 cm est donnée au système au démarrage. Les gains du contrôleur ainsi que les paramètres spécifiques à ce test sont résumés dans le tableau 3.2 suivant alors que l'allure du déplacement de la bille en fonction du temps est donné à la figure 3.4.

Tableau 3.2 : Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction x, sans écoulement

Consigne X (cm)	6
Consigne Z (cm)	0
Gain proportionnel Kp	1.2
Gain intégrateur Ki	0
Gain dérivatif Kd	0
Gradient Y (mT/m)	26

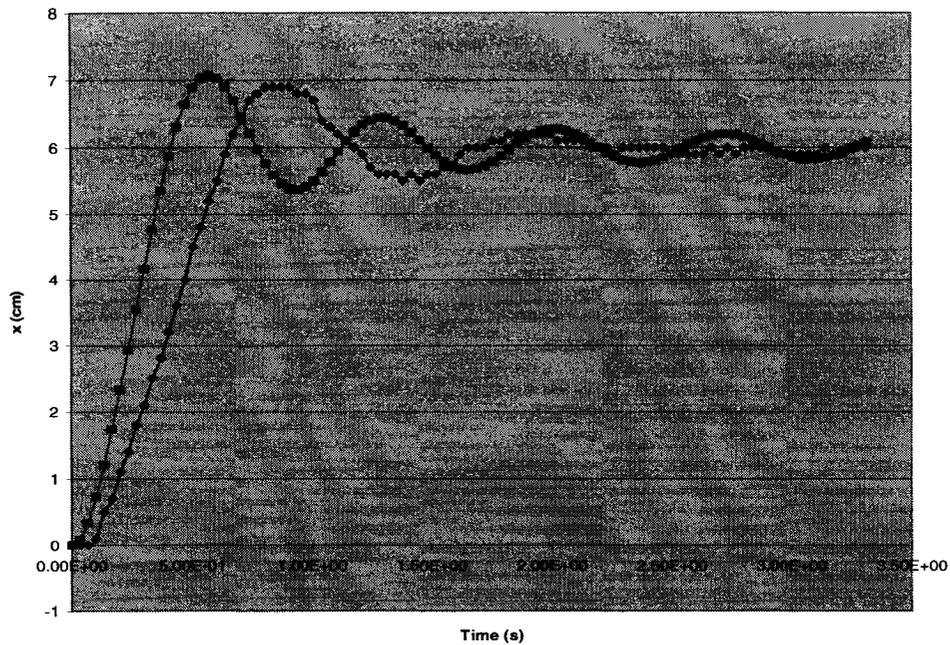


Figure 3.4 Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 6cm sur l'axe x dans un environnement sans écoulement [51]

Le prochain résultat est un déplacement dans la direction z avec une consigne de 8 cm. Les gains du contrôleur ainsi que les paramètres spécifiques à ce test sont résumés dans le tableau 3.3 suivant. Le déplacement de la bille en fonction du temps est donné à la figure 3.5.

Tableau 3.3 : Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction x, sans écoulement

Consigne X (cm)	0
Consigne Z (cm)	8
Gain proportionnel K_p	10
Gain intégrateur K_i	0.1
Gain dérivatif K_d	0
Gradient Y (mT/m)	0

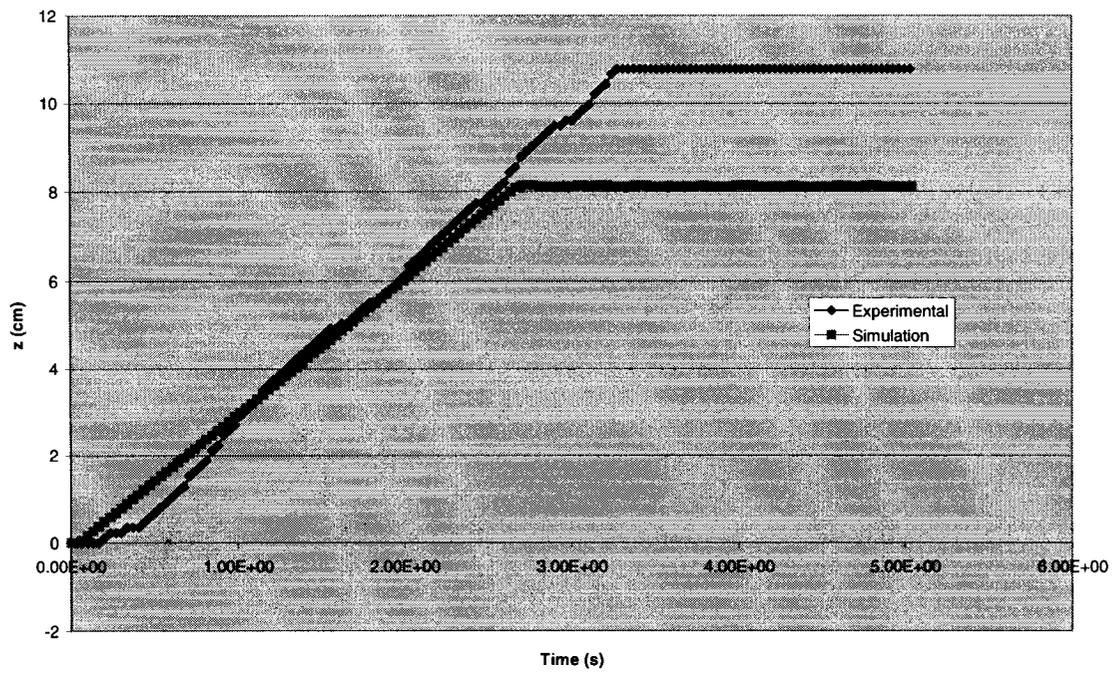


Figure 3.5 Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 8cm sur l'axe z dans un environnement sans écoulement [51]

3.2.1.3 Contrôle avec écoulement constant

Nous présentons ici le test de déplacement unidirectionnel dans la direction x dans un environnement à écoulement constant dans de l'eau. Les données relatives à ce test sont données au tableau 3.4 et la courbe représentant le déplacement de la bille en fonction du temps est présenté à la figure 3.6.

Tableau 3.4 : Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction z avec un écoulement constant

Consigne X (cm)	-8
Consigne Z (cm)	0
Gain proportionnel Kp	6
Gain intégrateur Ki	0.02
Gain dérivatif Kd	0

Gradient Y (mT/m)	0
Débit (l/min)	0.428

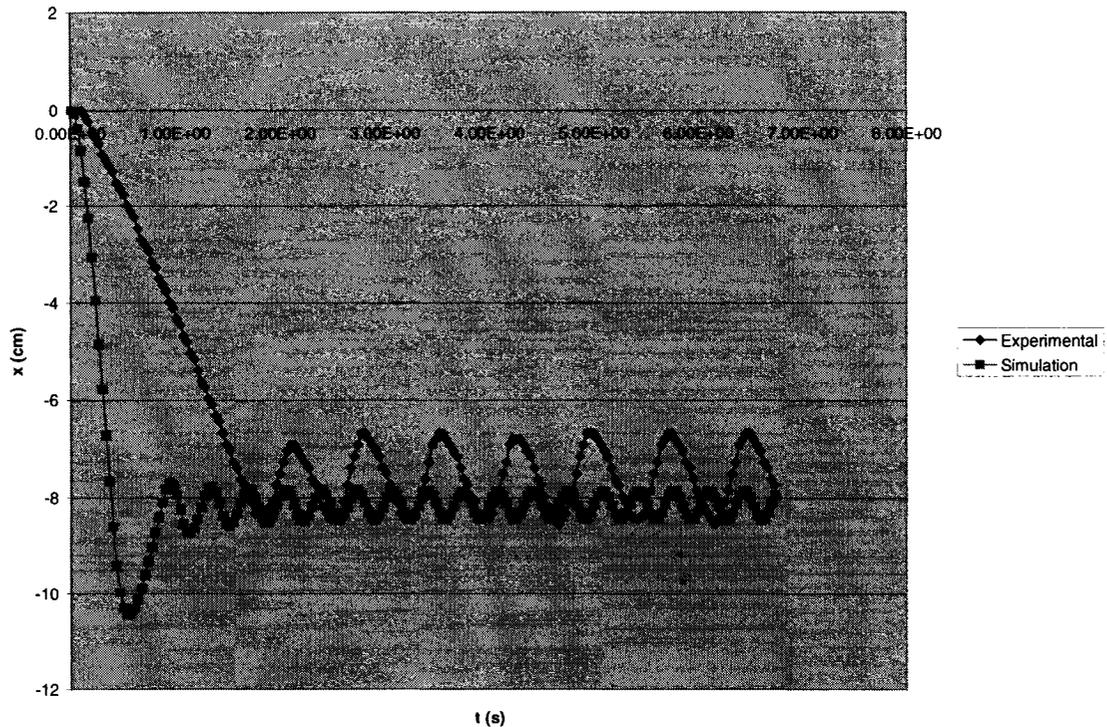


Figure 3.6 Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 8cm sur l'axe x dans un environnement avec un écoulement constant [51]

3.2.1.4 Contrôle avec écoulement pulsé

Les résultats effectués en écoulement pulsé à l'aide de la pompe CardioFlow sont présentés dans ces lignes. Un contrôle unidirectionnel en z est considéré. Une onde simulant l'écoulement dans l'artère fémorale humaine est utilisée. L'expérience de contrôle consiste à déplacer la bille sur une distance de 6 cm alors que celle-ci fait face à un écoulement pulsatif opposé. Pour se faire, la bille est en premier lieu propulsée dans la direction x négatif à l'aide d'une onde se déplaçant de droite à gauche. A l'instant où l'onde s'inverse pour traverser le tube de gauche à droite, la séquence est démarrée pour tenter de ramener la bille à 6 cm dans la direction x négatif à partir de l'endroit où elle se

trouve à cet instant. L'onde circulant de gauche à droite, la bille est alors propulsée contre celle-ci en vue d'atteindre un déplacement de 6cm. L'onde fémorale utilisée est illustrée à la figure 3.7 suivante. Les paramètres du contrôleur sont présentés au tableau 3.5. Les résultats de contrôle sont quant à eux, représentés à la figure 3.8.

Tableau 3.5 Consignes et paramètres du contrôleur pour un déplacement unidirectionnel dans la direction x avec un écoulement pulsé

Consigne X (cm)	6
Consigne Z (cm)	0
Gain proportionnel K_p	4
Gain intégrateur K_i	0.02
Gain dérivatif K_d	0.05
Gradient Y (mT/m)	0

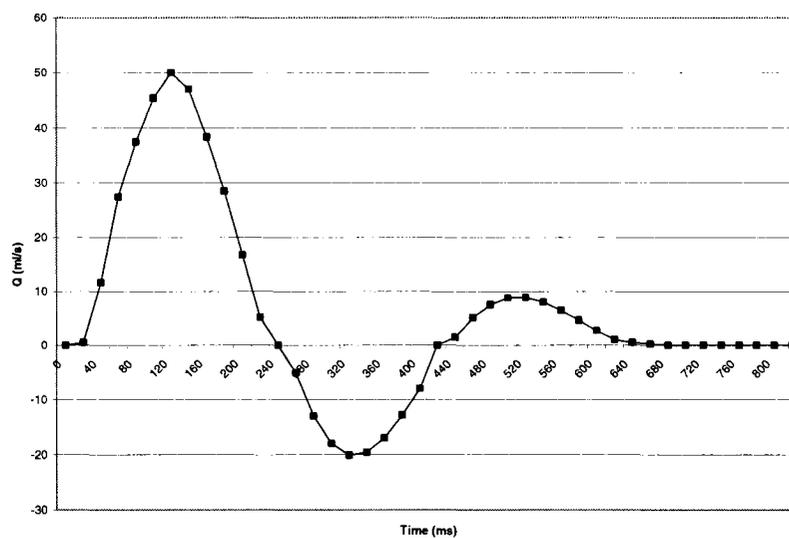


Figure 3.7 : Allure de l'onde fémorale utilisée pour les tests de contrôle en écoulement pulsé [51].

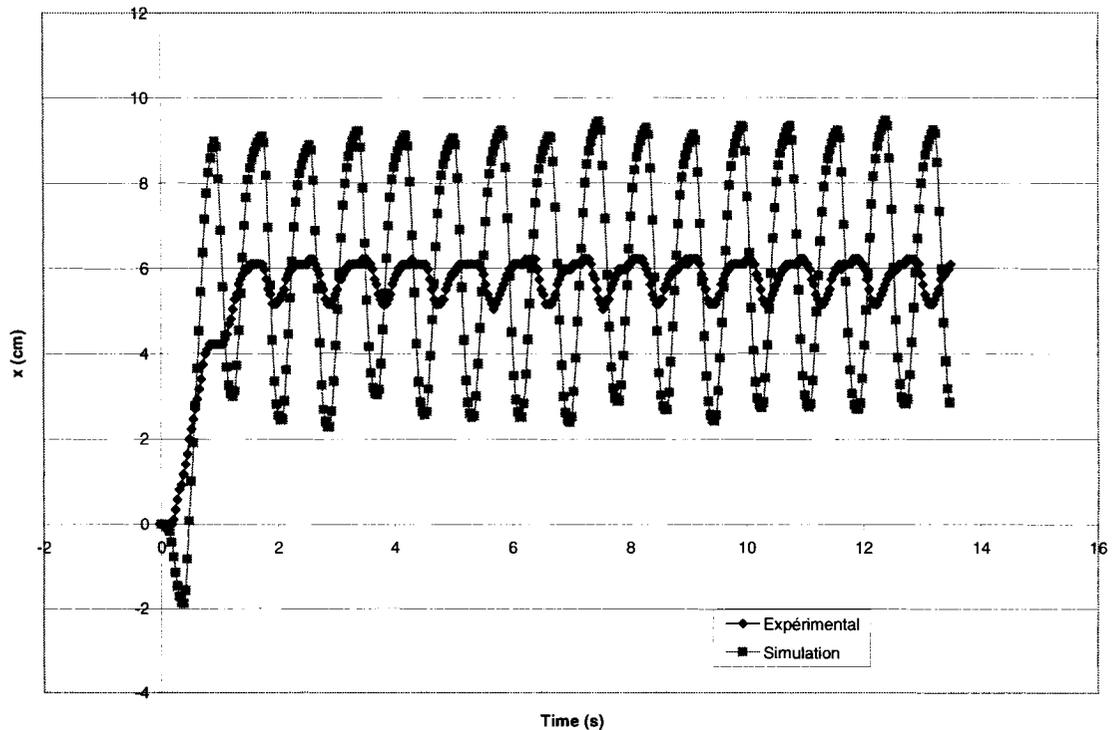


Figure 3.8 : Simulation et résultats expérimentaux du déplacement de la bille sur une distance de 6cm sur l'axe x dans un environnement avec un écoulement pulsé représentant l'onde fémorale humain [51].

3.2.2 Résultats de contrôle 2D

3.2.2.1 Montage et paramètres

Les résultats obtenus en deux dimension ont été effectués sur une plaque de PMMA mesurant 17 cm de long et 15 cm de large supportée par trois vis de plastique fixées à un bac pour contenir l'eau. Le niveau de la plaque est réglable par les trois vis. Celle-ci est entourée de petits rebords en plastique et recouvert d'une autre plaque de PMMA. Le tout permet de placer la bille entre les deux plaques tout en empêchant celle-ci d'être coincée et de quitter le montage. Le bac est ensuite rempli d'eau et on s'assure que la

cavité où se trouve la bille soit également immergé. La figure 3.9 illustre l'équipement utilisé. La bille utilisée pour les essais est la même que celle décrite en 3.2.1.1.

Le contrôleur utilisé est un PID en deux dimensions plus explicitement décrit dans [51, 53]. De plus, les paramètres de positionnement sont similaires à ceux du tableau 3.1.

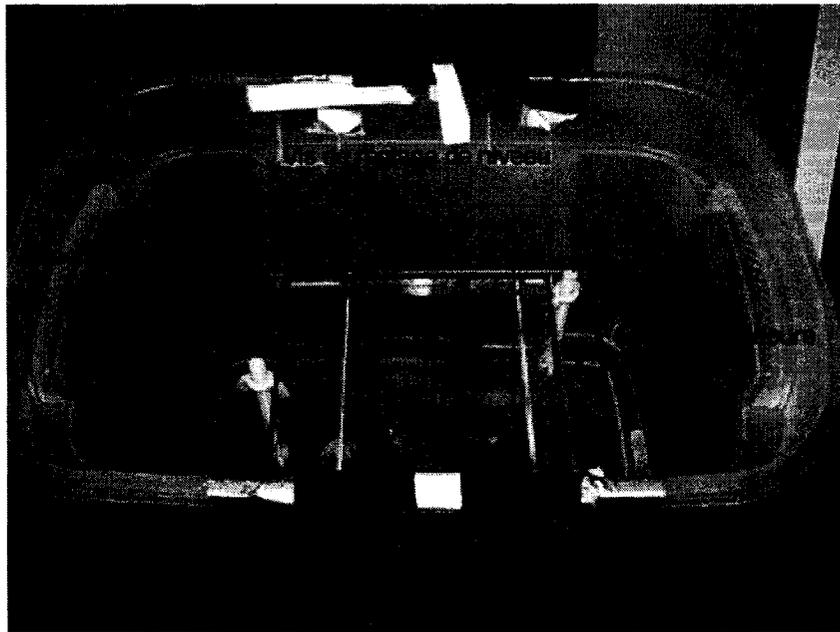


Figure 3.9 : Illustration du montage pour les tests de contrôle en deux dimensions

3.2.2.2 Contrôle sans écoulement en présence d'un chemin

Le chemin considéré est un ensemble de 5 balises dont les positions sont illustrées au tableau 3.6 ci-dessous. Pour le passage des balises, une sphère virtuelle de rayon r_p est centrée sur la prochaine balise à atteindre. Lorsque la distance séparant le dispositif du centre de la sphère de précision est inférieur au rayon r_p , le système considère que la balise est atteinte et charge la prochaine position à atteindre. Les

paramètres de la séquence et du contrôleur sont donnés au tableau 3.7 alors que les résultats de contrôle sont donnés à la figure 3.10.

Tableau 3.6 : Balises constituant le chemin pour les résultats de la figure 3.10

Numéro de balise	Position x (cm)	Position y (cm)	Position z (cm)
1	2	0	2
2	4	0	4
3	6	0	6
4	5	0	8.8
5	4	0	12

Tableau 3.7 : Paramètres pour le test de contrôle en deux dimensions suivant un chemin pré déterminé

r_p (mm)	2
Gain proportionnel Kp	2
Gain intégrateur Ki	0.01
Gain dérivatif Kd	0.5
Gain proportionnel normal	5
Gain dérivatif normal	0.1
Gradient Y (mT/m)	0

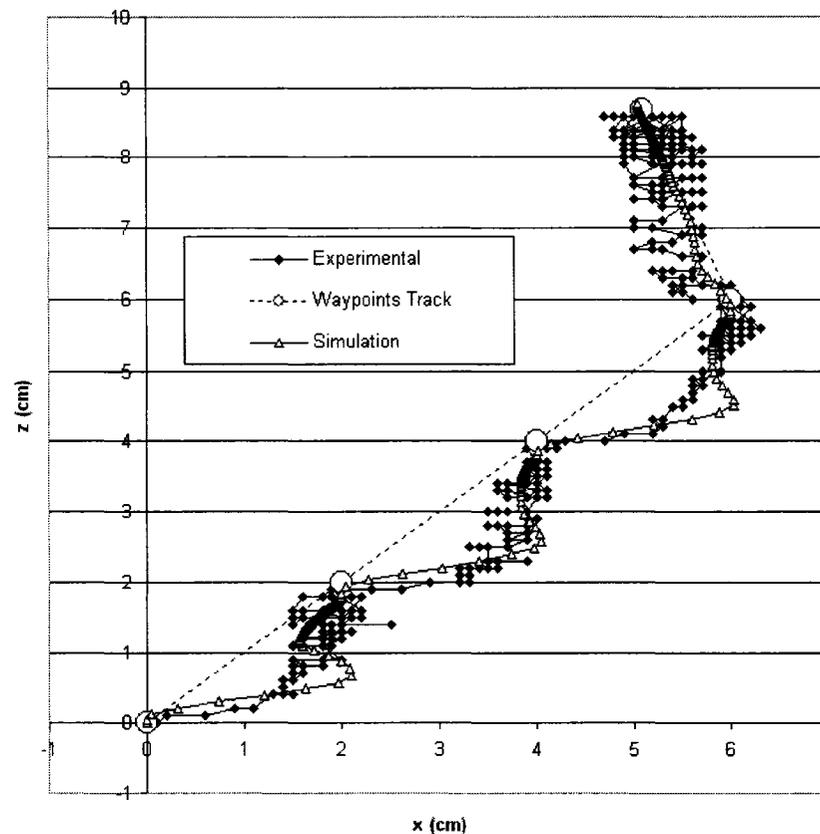


Figure 3.10 : Simulation et résultats expérimentaux du déplacement de la bille suivant un chemin pré déterminé en l'absence d'écoulement [51].

3.2.2.5 Contrôle sans écoulement en présence d'un chemin dans un fantôme vasculaire

Le fantôme vasculaire décrit à la section 3.1 est utilisé pour la navigation du dispositif à travers un chemin préalablement établi. Celui-ci est en premier lieu rempli d'eau et la bille est ensuite insérée à un endroit pré-déterminé. Par la suite, le fantôme est orienté et placé dans le tunnel de l'IRM. Un seul chemin a été testé et présenté ici. Celui-ci tente de déplacer la bille suivant le parcours illustré à la figure 3.12 contenant 26 balises. La position d'insertion de la bille correspond avec la position de la première balise dans le logiciel Volview. La figure 3.11 montre le fantôme vasculaire utilisé ainsi

que le repère considéré pour les tests de contrôle. Les paramètres de la séquence sont illustrés au tableau 3.8 suivant. Les résultats de la navigation du MR-Sub pour le chemin de la figure 3.12 se trouvent à la figure 3.13.

Tableau 3.8 : Paramètres pour le test de contrôle en deux dimensions suivant un chemin pré déterminé dans un fantôme vasculaire

r_p (mm)	5
Gain proportionnel K_p	2
Gain intégrateur K_i	0.01
Gain dérivatif K_d	0.5
Gain proportionnel normal	5
Gain dérivatif normal	0.1
Gradient Y (mT/m)	0



Figure 3.11 : Illustration du fantôme vasculaire utilisé pour les tests de contrôle avec chemins

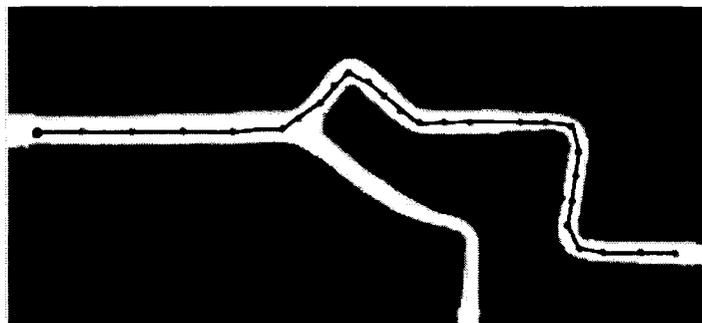


Figure 3.12: Allure du premier chemin testé pour la navigation dans le fantôme vasculaire

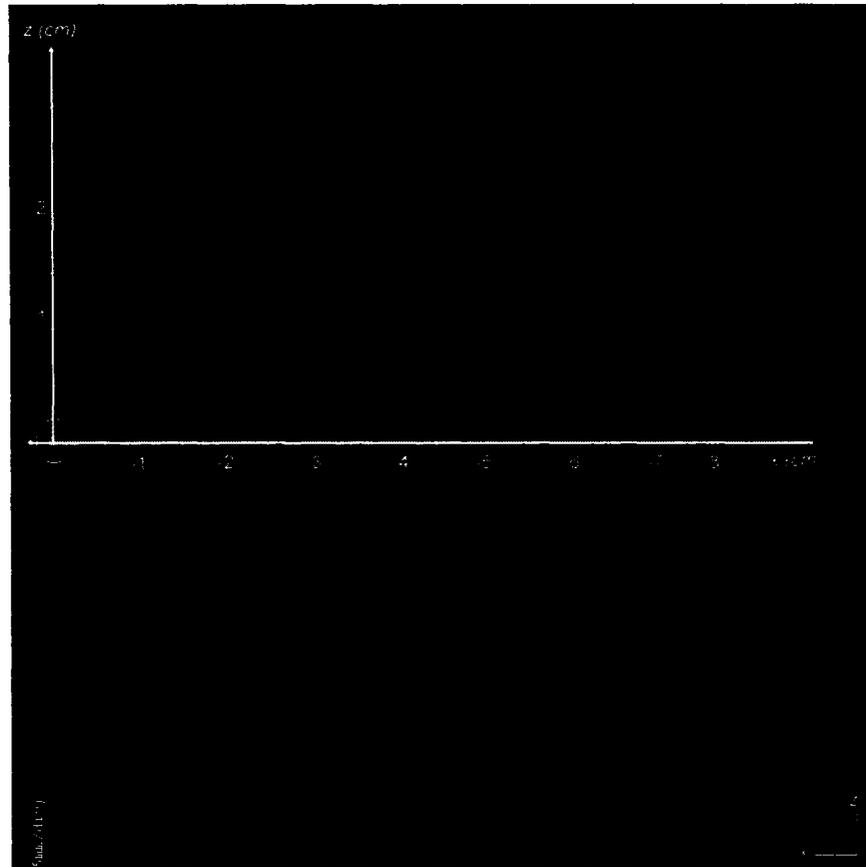


Figure 3.13 : Résultats expérimentaux du déplacement de la bille dans un fantôme vasculaire suivant un parcours illustré à la figure 3.12

3.3 Discussion

Les résultats présentés à la section précédente témoignent dans l'ensemble du bon fonctionnement de l'architecture et de la communication adéquate entre les modules développés. Bien que certaines figures semblent discutables en terme de précision du contrôleur, elles témoignent néanmoins du respect des contraintes temporelles en ce qui concerne le délai temps réel entre la séquence et le programme de reconstruction

d'images développés. Le délai temps réel considéré de 30 ms permet ainsi d'obtenir une fréquence de fonctionnement $f_{operation}$ telle que :

$$f_{operation} = \frac{1}{t_{feedback} + t_{CAN}} \quad (3.1)$$

$$f_{operation} = \frac{1}{30 \times 10^{-3} + 4 \times 10^{-3}} = 29.41 Hz$$

où $t_{feedback}$ est le délai temps réel de la séquence illustrée à la figure 2.16 et t_{CAN} est le temps d'application du bloc d'événement contenant le CAN pour la communication temps réel comme décrit en 2.5.2. La somme de ces deux délais constitue le temps d'exécution d'une itération de la séquence MR-Sub plus visuellement compréhensible à la vue de la figure 2.16. Avec l'architecture proposée et la séquence développée, il est donc possible d'effectuer la navigation (propulsion, localisation et contrôle) du dispositif à raison de 30 fois par seconde environ. Bien que cette fréquence opératoire soit suffisante pour les résultats présentés, l'ajout de traitement supplémentaire comme la présence d'un contrôleur plus complexe ou la nécessité d'analyse de données pour le recalage ou la synchronisation avec un ECG engendrerait une fréquence d'opération inférieure. Par ailleurs, la commande récupérée par la séquence en provenance du programme de reconstruction d'image n'est pas appliquée lors de la prochaine section de propulsion mais plutôt à la suivante. Ceci s'explique par le fait que la réception de la commande s'effectue pendant le temps d'arrêt du MPCU (le délai temps réel) alors que le bloc de propulsion courant se trouve déjà dans le FIFO d'exécution. Il serait envisageable de modifier la séquence actuelle pour assurer l'application de la commande à l'instant de sa réception. La fréquence d'opération serait cependant plus

élevée mais assurerait une réponse plus rapide du système lors de changements brusques de position ou lors d'une opération de recalage.

Les expériences menées permettent au dispositif d'atteindre son ou ses objectif(s) avec une précision de l'ordre du millimètre en général ou du centimètre dans certains cas. Ces erreurs de précision en régime permanent sont essentiellement dues au choix et à la mise au point du contrôleur implanté qui ne relève en rien d'une erreur en terme d'intégration et de développement de l'architecture. Seul le module 'Roadmap' qui n'a pu être totalement complété, bien qu'il soit possible au dispositif de suivre un chemin préalablement fixé par l'utilisateur, rend impossible la sélection et la construction automatique d'un chemin considéré idéal pour la navigation du MR-Sub à travers celui-ci. La portion du module devant analyser une image angiographique préalablement obtenue à la navigation du MR-Sub pour en extraire les chemins éventuels n'a pas été complétée en raison des délais excessifs dans l'atteinte des objectifs d'intégration prioritaires pour la mise au point d'un système de navigation autonome. Ces délais sont eux-mêmes essentiellement dus à l'accès restreint à l'IRM durant ces deux dernières années. Bien que l'objectif principal visant à posséder une architecture fonctionnelle pour la navigation du MR-Sub ait été atteint, l'objectif secondaire consistant en la mise au point d'un système d'analyse et de traitement d'images angiographiques pour la formation automatique de chemins reste à compléter.

3.4 Conclusion

Le présent chapitre a présenté la méthodologie choisie pour la mise sur place d'une première architecture fonctionnelle permettant la navigation du MR-Sub dans des environnements *ex vivo*. Les résultats de contrôle en une dimension et en deux dimensions d'une bille ferromagnétique de 1.5 mm de diamètre ont permis de valider le bon fonctionnement de l'architecture. Finalement, une discussion des résultats en terme de fréquence d'opération du système et d'éventuels ajouts pour la synchronisation avec un ECG ou un oxymètre sont introduits. Cette première architecture du MR-Sub a permis à l'équipe du projet d'obtenir pour la première fois des résultats de contrôle impliquant le bon fonctionnement de celle-ci. Des extensions à l'architecture restent cependant à effectuer pour compléter celle-ci et dont la description fait l'objet du prochain chapitre.

CHAPITRE 4. CONTRIBUTION, TRAVAUX FUTURS ET CONCLUSION

4.1 Contribution de la présente étude au projet MR-Sub

Comme dans la plupart des projets assurant une phase d'intégration entre divers éléments préalablement existants, il est parfois difficile de faire la part des choses entre les tâches effectuées lors de l'étude en soit et celles déjà réalisées et faisant partie du livrable final. Les lignes qui suivent ont pour objectif de tracer la frontière entre les apports de la présente étude et les données préalables au projet MR-Sub.

Le requis premier de la présente étude était de fournir au projet MR-Sub un environnement logiciel permettant au dispositif de pouvoir suivre un chemin préalablement fixé dans un environnement *in-vivo* en trois dimension. Un tel environnement doit permettre d'assurer l'asservissement du dispositif de manière autonome tout en respectant les contraintes temporelles de l'IRM en ce qui à trait aux délais de communication temps réel ainsi que la contrainte temporelle du contrôleur pour assurer la stabilité de celui-ci. Pour ce faire, la mise en commun des piliers déjà présents que sont le positionnement du ferromagnétique en IRM [14], l'étude de faisabilité d'une méthode de propulsion magnétique [34] et la mise au point d'un contrôleur PID pour l'asservissement du dispositif [51, 53, 54, 52] a été effectué. De plus, le pilier 'Roadmap' devant permettre le suivi d'un chemin préalablement choisi par l'utilisateur, a été quant à lui, développé lors de la présente étude en vue de son intégration. Seul le logiciel 'Volview' décrit plus en détails à la section 3.1, utilisé pour le

visionnement et la génération des balises sur une image en format DICOM ne fait pas parti du travail effectué dans cette étude.

Cette intégration a débuté par la modification et l'adaptation d'une séquence d'imagerie déjà fournie par la compagnie Siemens. La séquence 'MiniFLASH' a donc été analysée et s'est vue ajouter la portion de communication temps réel ainsi que le bloc d'événements comportant les gradients de propulsion et la séquence de positionnement. Ce bloc se trouve dans les fichiers `mMRSubKernel.cpp` et `mMRSubKernel2.cpp` suivant le mode de positionnement choisi (respectivement spin écho et écho de gradient). La séquence de positionnement spin écho a été préalablement développée par les membres du projet alors que la séquence utilisant l'écho de gradient a été développée durant la présente étude à la fois par les membres du projet et par l'auteur de cet ouvrage. L'intégration de ces deux séquences de positionnement a été cependant effectuée par l'auteur uniquement.

Par la suite, la mise sur pied de l'ensemble des classes illustrées dans cette étude a été élaborée par l'auteur et intégrée à la séquence initiale et au programme de reconstruction. Le contrôleur utilisé lors des tests de navigation a été développé au laboratoire par les membres du projet et son intégration dans la classe `mContrôleur` en vue des tests de navigation a été effectuée par l'auteur. La validation de l'architecture logicielle développée s'est effectuée aussi bien au laboratoire de Nanorobotique qu'en IRM par l'auteur.

En somme, seuls la séquence de positionnement et le contrôleur PID faisant partie intégrante du code de l'architecture n'ont pas été développés par l'auteur.

4.2 Travaux futurs

Bien que la grande majorité des requis du projet face à l'obtention d'une infrastructure autonome pour le guidage du dispositif aient été rencontrés, il n'en demeure pas moins que certains aspects qui avaient été mis temporairement de côté pour assurer un premier fonctionnement de l'application se doivent désormais d'être implantés.

Le premier de ces requis consiste à compléter le module 'Roadmap' pour que celui-ci soit en mesure, à même l'IRM, de prendre une angiographie du volume de travail, l'analyser et en extraire l'ensemble des chemins possibles à partir d'un point de départ et d'un point d'arrivée fixés par l'utilisateur. Il convient par la suite de déterminer le chemin le plus approprié à la navigation du dispositif suivant les contraintes physiologiques observées pour finalement exporter ce chemin final, de la même manière dont sont exportés actuellement les chemins pour les tests présentés dans la section des résultats.

La seconde étape dans les besoins du projet MR-Sub est la mise au point d'une méthode de recalage temps réel afin d'être en mesure de corriger les inévitables déformations et mouvements de l'environnement lors de la navigation du dispositif. Cette phase de correction devrait se faire dans un laps de temps très court pour être intégrable dans la séquence temps réel de navigation présentée à la figure 2.16. Une telle tâche supplémentaire impliquerait probablement la revue des délais actuels de traitement à la hausse mais assurerait que la dilatation et la contraction des vaisseaux ne compromettraient pas les déplacements du MR-Sub.

Finalement, une synchronisation avec un oxymètre ou un ECG constitue un dernier élément majeur au bon fonctionnement du dispositif dans un environnement pulsatif comme celui des vaisseaux sanguins afin d'assurer que le contrôleur soit en mesure de prédire les variations d'amplitude et de fréquence de l'écoulement auquel le dispositif fait face. Idéalement, une telle synchronisation, couplée à un algorithme d'approximation permettrait de connaître le débit en tout point du volume parcouru si la navigation du MR-Sub ne se fait pas dans une région à physiologie similaire en tout point.

4.3 Conclusion

Le projet MR-Sub est sans nul doute une idée ambitieuse qui fait douter encore bien des gens du domaine du Génie Biomédical et du monde médical. Il paraît à prime abord surréaliste de vouloir guider un dispositif dans les vaisseaux sanguins humain en vue d'y libérer des médicaments à des endroits cibles ou de permettre de simplifier des opérations comme l'embolisation. De nombreux obstacles d'ordre physiologique et physique limitent encore les avancés de ce projet. Cela dit, les progrès effectués suite à la mise en commun des méthodes de propulsion, de positionnement et de contrôle ont permis l'atteinte d'un premier objectif fondamental qui est celui de permettre à une première forme de dispositif d'être librement contrôlé dans divers environnements tout en assurant une précision non négligeable. L'architecture établie et qui a été présentée dans ce mémoire a permis d'atteindre une seconde preuve de concept du fonctionnement du MR-Sub et qui cette fois, n'assure pas uniquement la faisabilité indépendante des

piliers mais bien une interaction complète de ceux-ci menant à un premier système autonome de navigation d'un dispositif en IRM. Cela dit, beaucoup de travaux restent encore à effectuer avant d'obtenir un dispositif endovasculaire apte à la navigation dans le corps humain. Bien que très stimulants et très complexes, ceux-ci ne sont pas irréalisables.

RÉFÉRENCES

- [1] ABOUSSOUAN, A., FELFOUL, O., MATHIEU, J.B., BEAUDOIN, G., MARTEL, S. 2006 . "Real-Time Projection Based Technique for Tracking Ferromagnetic Devices". *International Society for Magnetic Resonance in Medicine*.
- [2] ATTALI, D., CHANUSSOT, J., ARESTE, R., GUYONIC, S. 2005. "3D snakes for the segmentation of buried mines in 3D acoustic images". *Oceans 2005 - Europe, 20-23 June 2005*. Brest, France : IEEE. Vol. 1, P. 442-6.
- [3] BAKALEXIS, S.A., BOUTALIS, Y.S., MERTZIOS, B.G. 2002. "Edge detection and image segmentation based on nonlinear anisotropic diffusion". *Proceedings of 14th International Conference on Digital Signal Processing (DSP2002), 1-3 July 2002*. Santorini, Greece : IEEE. vol.2, P. 1203-6.
- [4] BERNSTEIN, M.A., KING, K.F., ZHOU, X.J. 2004. *Handbook of MRI Pulse Sequences*. Academic Press.
- [5] BLANCO, R.T., OJALA, R., KARINIEMI, J., PERALA, J., NIINIMAKI, J., TERVONEN, O. 2005. "Interventional and intraoperative MRI at low field scanner - A review". *European Journal of Radiology*. 56:2. 130-142.
- [6] BLANCO SEQUEIROS, R., OJALA, R., KARINIEMI, J., PERALA, J., NIINIMAKI, J., REINIKAINEN, H. et al. 2005. "MR-guided interventional procedures: A review". *Acta Radiologica*. 46:6. 576-586.
- [7] BRUKER BIOSPIN. High-Resolution Time-of-Flight Angiography. <http://www.bruker-biospin.de/MRI/applications/angiography.html> (Page consultée le 28 2006)
- [8] BUDGE, S.E., MAYAMPURATH, A.M., SOLINSKY, J.C. 2004. "Real-time registration and display of confocal microscope imagery for multiple-band analysis". *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on.* 2, P. 1535-1539 Vol.2.
- [9] BULTE, J.W.M. *Magnetic nanoparticles as markers for cellular MR imaging*
Department of Radiology and Institute for Cell Engineering, Johns Hopkins University School of Medicine.
- [10] CHARM, S.E., KURLAND, G.S. 1974. *Blood flow and microcirculation*. John Wiley & Sons.

- [11] CHENG, H.D., JIANG, X.H., SUN, Y., WANG, J. 2001. "Color image segmentation: Advances and prospects". *Pattern Recognition*. 34:12. 2259-2281.
- [12] CYRUS T., ABENDSCHEIN DR., CARUTHERS SD., HARRIS TD., GLATTAUER V., WERKMEISTER JA. et al. 2006. *MR three-dimensional molecular imaging of intramural biomarkers with targeted nanoparticles*. Washington University School of Medicine.
- [13] DELUCIA, P.R., MATHER, R.D., GRISWOLD, J.A., SUNANDA MITRA 2006. "Toward the improvement of image-guided interventions for minimally invasive surgery: three factors that affect performance". *Human Factors*. 48:1. 23-38.
- [14] FEFLOUL, O. 2005. *Techniques de positionnement d'objets ferromagnétiques en irm*. Mémoire, Université de Montréal.
- [15] FELFOUL, O., MARTEL, S., BEAUDOIN, G., MATHIEU, J.-B. 2004. "Micro-device's susceptibility difference based MRI positioning system, a preliminary investigation". *Conference Proceedings - 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2004, Sep 1-5 2004*. San Francisco, CA, United States : Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ 08855-1331, United States. 26 II, P. 1140-1143.
- [16] FREIXENET, J., MUNOZ, X., RABA, D., MARTI, J., CUFI, X. 2002. "Yet another survey on image segmentation: region and boundary information integration". *Computer Vision - ECCV 2002. 7th European Conference on Computer Vision. Proceedings, Part III, 28-31 May 2002*. Copenhagen, Denmark : Springer-Verlag. P. 408-22.
- [17] GRAF, H., LAUER, U.A., BERGER, A., SCHICK, F. 2005-. "RF artifacts caused by metallic implants or instruments which get more prominent at 3 T: an in vitro study". *Magnetic Resonance Imaging*. 23:3. 493-9.
- [18] GRAF, H., LAUER, U.A., KLEMM, T., SCHNIEDER, L., SCHICK, F. 2003. "Artifacts in MRT caused by instruments and implants". *Zeitschrift fur Medizinische Physik*. 13:3. 165-70.
- [19] GRAF, H., STEIDLE, G., MARTIROSIAN, P., LAUER, U.A., SCHICK, F. 2006-. "Effects on MRI due to altered rf polarization near conductive implants or instruments". *Medical Physics*. 33:1. 124-7.
- [20] HAACKER, E.M., BROWN, R.W., THOMPSON, M.R., VENKATESAN, R. 1999. *Magnetic Resonance Imaging Physical principles and sequences design*. John Wiley & Sons, Inc.

- [21] HILLENBRAND, C.M., ELGORT, D.R., WONG, E.Y., REYKOWSKI, A., WACKER, F.K., LEWIN, J.S. et al. 2004-. "Active device tracking and high-resolution intravascular MRI using a novel catheter-based, opposed-solenoid phased array coil". *Magnetic Resonance in Medicine*. 51:4. 668-75.
- [22] HIROSE, M., FURUHASHI, H., ARAKI, K. 2001. "Automatic registration of multi-view range images measured under". *Virtual Systems and Multimedia, 2001. Proceedings. Seventh International Conference on*. P. 738-746.
- [23] HUANG, X., HILL, N.A., PETERS, T.M. 2005. "Ultrasound-based technique for intra-thoracic surgical guidance". *Medical Imaging 2005 - Visualization, Image-Guided Procedures, and Display, Feb 13-15 2005*. San Diego, CA, United States : International Society for Optical Engineering, Bellingham, WA 98227-0010, United States. 5744, P. 822-830.
- [24] JERZEWKI, A. , PATTYNAMA, PM., STEENDJIK, P., DOORNBOS, J., DE ROOS, A., BAAN, J. "Development of an MRI-compatible catheter for pacing the heart: initial in vitro and in vivo results.". *Journal of Magnetic Resonance Imaging*. 6 : 6. 948-9.
- [25] JINCHENG PAN, MINGLU LI 2003. "Segmentation of MR osteosarcoma images". *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003, 27-30 Sept. 2003*. Xi'an, China : IEEE Comput. Soc. P. 379-84.
- [26] KITWARE. <http://www.kitware.com/> (Page consultée le 24 2006)
- [27] LIANG, Z., LAUTERBUR, P.C. 1999. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley-IEEE Press.
- [28] LIGHT, E.D., IDRIS, S.F., SULLIVAN, K.F., WOLF, P.D., SMITH, S.W. 2005-. "Real-time 3D laparoscopic ultrasonography". *Ultrasonic Imaging*. 27:3. 129-44.
- [29] MACOVSKI, A. 1983. *Medical Imaging Systems*. Prentice-Hall.
- [30] MARTEL, S., MATHIEU, J.-B., FELFOUL, O., MACICIOR, H., BEAUDOIN, G., SOULEZ, G. et al. 2004. "Adapting MRI Systems to Propel and Guide Microdevices in the Human Blood Circulatory System". *26th Conference of the IEEE Engineering in Medicine and Biology Society* . IEEE.
- [31] MARTEL, S., MATHIEU, J., YAHIA, L., BEAUDOIN, G., SOULEZ, G. *Method and system for propelling and controlling displacement of a microrobot in a blood Vessel*. 10/417,475.

- [32] MATHIEU, J.-B., MARTEL, S., YAHIA, L., SOULEZ, G., BEAUDOIN, G. 2003. "MRI Systems as a Mean of Propulsion for a Microdevice in Blood Vessels ". *IEEE Engineering in Medicine and Biology Society* . 17-21 Sept 2003 : IEEE. P. 3419-3422.
- [33] MATHIEU, J.-B., MARTEL, S., YAHIA, L., SOULEZ, G., BEAUDOIN, G. Accepted. "Preliminary Investigation of the Feasibility of Magnetic Propulsion for Future Microdevices in Blood Vessels". *BioMedical Materials and Engineering*.
- [34] MATHIEU, J.-B. 2004. *Méthode de propulsion d'un corps ferromagnétique à l'aide d'un système d'imagerie par résonance magnétique* . Mémoire de Maîtrise, Université de Montréal.
- [35] MEDIC TINEDIC. MedicTinedic - Ballon Foley Catheter. <http://www.medictinedic.com/a1023.html> (Page consultée le 28 2006)
- [36] MUNOZ, X., FREIXENET, J., CUFI, X., MARTI, J. 2003-. "Strategies for image segmentation combining region and boundary information". *Pattern Recognition Letters*. 24:1-3. 375-92.
- [37] NEMA. DICOM Digital Imaging and Communication in Medicine. <http://medical.nema.org/> (Page consultée le 23 2006)
- [38] NOIRHOMME, Q., FERRANT, M., VANDERMEEREN, Y., OLIVIER, E., MACQ, B., CUISENAIRE, O. 2004. "Registration and real-time visualization of transcranial magnetic stimulation with 3-D MR images". *Biomedical Engineering, IEEE Transactions on*. 51:11. 1994-2005.
- [39] PAUCHARD, Y., SMITH, M.R., MINTCHEV, M.P. 2005-. "Improving geometric accuracy in the presence of susceptibility difference artifacts produced by metallic implants in magnetic resonance imaging". *IEEE Transactions on Medical Imaging*. 24:10. 1387-99.
- [40] PRESSMAN, R.S. 2001. *Software Engineering A practitioner'a approach*. New York : McGraw-Hill.
- [41] RORDEN, C. Introduction to DICOM medical image format. <http://www.sph.sc.edu/comd/rorden/dicom.html> (Page consultée le 23 2006)
- [42] ROTHENBERG, M.A. 1998. *Understanding X-Rays: A Plain English Approach*.

- [43] RUSHMER, R.F. 1972. *Structure and function of the cardiovascular system*, . S.W.B. Saunders Company.
- [44] SABRA, W., KHOUZAM, M., CHANU, A. 2005. "Use of 3D Potential Field and an Enhanced Breadth-first Search Algorithms for the Path Planning of Microdevices Propelled in the Cardiovascular System". *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference*. Shangai, China :
- [45] SCHMALZ, M.S., RITTER, G.X. 2004. "Region segmentation techniques for object-based image compression - A review". *Mathematics of Data/Image Coding, Compression, and Encryption VII, with Applications, Aug 4-5 2004*. Denver, CO, United States : International Society for Optical Engineering, Bellingham, WA 98227-0010, United States. 5561, P. 62-75.
- [46] SHEKHAR, R., ZAGRODSKY, V., GARCIA, M.J., THOMAS, J.D. 2004. "Registration of real-time 3-D ultrasound images of the heart for novel 3-D stress echocardiography". *Medical Imaging, IEEE Transactions on*. 23:9. 1141-1149.
- [47] SHELLEY MEDICAL IMAGING TECHNOLOGIES.
<http://www.simutec.com/Media/models/H+N-S-S-001-003%20hi%20res.pdf> (Page consultée le 24 2006)
- [48] SHENHAV, A., AZHARI, H. 2004-. "Gradient field switching as a source for artifacts in MR imaging of metallic stents". *Magnetic Resonance in Medicine*. 52:6. 1465-8.
- [49] STEREOTAXIS INC. Stereotaxis Home. <http://www.stereotaxis.com/index.php?section=1> (Page consultée le August 2006)
- [50] SURI, J.S., SINGH, S., REDEN, L. 2002. "Fusion of region and boundary/surface-based computer vision and pattern recognition techniques for 2-D and 3-D MR cerebral cortical segmentation. II. A state-of-the-art review". *Pattern Analysis and Applications*. 5:1. 77-98.
- [51] TAMAZ, S. 2006. "Conception et implémentation d'un contrôleur pour l'asservissement d'une bille ferromagnétique à l'intérieur du système cardiovasculaire à l'aide d'un appareil IRM". Anon. Notes: Mémoire à venir.
- [52] TAMAZ, S., CHANU, A., MATHIEU, M., GOURDEAU, R., MARTEL, S. "Real-time MRI-based Control of a Ferromagnetic Core for Endovascular Navigation ". *Transactions on Biomedical Engineering*. Notes: Soumis.

- [53] TAMAZ, S., GOURDEAU, R., MARTEL, S. "Bidimensional MRI-based Navigation System Using a PID controller". *Engineering in Medicine and Biology Society (EMBS)*. Notes: Accepté, en publication.
- [54] TAMAZ, S., MARTEL, S. 2005. "Impact of the MRI-based Navigation System Constraints on the Step Response Using a PID Controller". *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. P. 5073-5076.
- [55] VERCHENKO, A.P., KIRICHENKO, N.F., LEPEKHA, N.P. "Application of recurrent means of clusterization to image segmentation". *Journal of Automation and Information Sciences*. 52-9.
- [56] WANG, A.S., GIRISH NARAYAN, KAO, D., LIANG, D. 2005. "An evaluation of using real-time volumetric display of 3D ultrasound data for intracardiac catheter manipulation tasks". *Volume Graphics 2005, 20-21 June 2005*. Stony Brook, NY, USA : IEEE. P. 41-5.
- [57] WESTBROOK, C., KAUT ROTH, C. 2005. *MRI in practice*. Blackwell Publishing, Incorporated.
- [58] WINTER PM., CARUTHERS SD., KASSNER A., HARRIS TD., CHINEN LK., ALLEN JS. et al. 2003. *Molecular imaging of angiogenesis in nascent Vx-2 rabbit tumors using a novel alpha(nu)beta3-targeted nanoparticle and 1.5 tesla magnetic resonance imaging*. Washington University School of Medicine.
- [59] ZAGRODSKY, V., WALIMBE, V., CASTRO-PAREJA, C.R., JIAN XIN QIN, JONG-MIN SONG, SHEKHAR, R. 2005. "Registration-assisted segmentation of real-time 3-D echocardiographic data using deformable models". *Medical Imaging, IEEE Transactions on*.24:9.1089-1099.


```

ICE Shell
Searching v:\n4\x86\delivery\lib\IceWrapperLight.lib:
Searching v:\n4\x86\delivery\lib\IceWrapper.lib:
Searching v:\n4\x86\delivery\lib\IceObjectsLight.lib:
Searching v:\n4\x86\delivery\lib\IceObjects.lib:
Searching v:\n4\x86\delivery\lib\IceObjectsChest.lib:
Searching v:\n4\x86\delivery\lib\IceSDCLight.lib:
Searching v:\n4\x86\delivery\lib\IceSDC.lib:
Searching v:\n4\x86\delivery\lib\IceTransferObjects.lib:
Searching v:\n4\x86\delivery\lib\RegConf.lib:
Searching v:\n4\x86\delivery\lib\IceFeedback.lib:
Searching C:\MSUStudio6\Common\MSDev98\...\UC98\MFC\lib\mfcs42u.lib:
Searching C:\MSUStudio6\Common\MSDev98\...\UC98\MFC\lib\mfcs42u.lib:
Searching C:\MSUStudio6\Common\MSDev98\...\UC98\lib\msvert.lib:
Searching \SWF_extern\SDK\lib\kernel32.lib:
Searching \SWF_extern\SDK\lib\user32.lib:
Searching \SWF_extern\SDK\lib\gdi32.lib:
Searching \SWF_extern\SDK\lib\comdlg32.lib:
Searching \SWF_extern\SDK\lib\winspool.lib:
Searching \SWF_extern\SDK\lib\advapi32.lib:
Searching \SWF_extern\SDK\lib\shell32.lib:
Searching \SWF_extern\SDK\lib\comctl32.lib:
Searching \SWF_extern\SDK\lib\uuid.lib:

Done Searching Libraries
~~~~~
Installing file(s):
x86/IceMrSub21.dll -> U:\n4\x86\prod\bin/IceMrSub21.dll
...copied
x86/IceMrSub21.pdb -> U:\n4\x86\prod\bin/IceMrSub21.pdb
...copied
x86/IceMrSub21.lib -> v:\n4\x86\prod\lib/IceMrSub21.lib
...copied
x86/IceMrSub21.exp -> v:\n4\x86\prod\lib/IceMrSub21.exp
...copied
x86/IceMrSub21.dll -> v:\n4\x86\prod\bin/IceMrSub21.dll
...copied
x86/IceMrSub21.pdb -> v:\n4\x86\prod\bin/IceMrSub21.pdb
...copied
U:\n4\pkg\MrServers\MrIcePrograms\IceMrSub21>

```

Figure I.2 : Illustration de l'invite de commandes du ICE