

N° d'ordre : 909

Ecole Doctorale Mathématiques, Sciences de
l'Information et de l'Ingénieur

UdS

THÈSE

présentée pour obtenir le grade de

Docteur de l'Université de Strasbourg
Discipline : Sciences pour l'Ingénieur
Spécialité : Microélectronique

par

Qing SUN

**Etude et conception d'un processeur neuronal analogique
très faible consommation, application au pilotage
d'un pacemaker de nouvelle génération**

Soutenue publiquement le 27 janvier, 2011

Membres du jury

Directeur de thèse : M. Yannick HERVE	Maître de Conférences – HDR, UdS, Strasbourg
Rapporteur externe : M. Georges GIELEN	Professeur, KUL, Louvain, Belgique
Rapporteur externe : Mme Sylvie RENAUD	Professeur, IPB, Bordeaux
Rapporteur interne : M. Luc HEBRARD	Professeur, UdS, Strasbourg
Examineur : M. Patrick GARDA	Professeur, UPMC, Paris
Examineur : M. Jacques MICHEL	Maître de Conférences, UdS, Strasbourg

« Le hasard ne favorise que les esprits préparés »

Louis Pasteur

Remerciements

Je remercie tout d'abord **Monsieur Daniel MATHIOT**, le directeur de **l'Institut d'Electronique du Solide et des Systèmes (InESS)** de m'avoir accueilli au sein de son laboratoire.

Je remercie mon directeur de thèse **Monsieur Yannick HERVE** qui m'a donné une vue de très haut niveau du monde scientifique. Sa culture scientifique et ses conseils pertinents me sont toujours bénéfiques. Je le remercie également pour ses soutiens et ses connaissances qu'il m'a transmises au cours de cette thèse.

Je tiens à remercier tout particulièrement mon encadrant **Monsieur Jacques MICHEL** qui m'a ouvert la porte à une carrière de recherche scientifique. J'ai beaucoup profité de ses expériences solides sur la recherche scientifique notamment sur la conception analogique pour des systèmes complexes. Ma thèse ne verrait jamais le jour sans lui. Les soutiens qu'il m'a apportés dans le déroulement et dans la rédaction de ma thèse sont vraiment indispensables.

Je remercie aussi très fortement **Madame Sylvie RENAUD**, **Monsieur Georges GIELEN**, **Monsieur Patrick GARDA**, **Monsieur Luc HEBRARD** pour avoir accepté de faire partie du jury de ma thèse. Je suis très heureux de votre participation au jury et vous adresse mon profond respect.

Je voudrais aussi remercier mes collègues au sein du laboratoire InESS, **François SCHWARTZ**, **Mingchun TANG**, **Nicolas CHEVILLON**, **Dumitru ARMEANU**, **Kangwei WU**.

Enfin, sur un plan plus personnel, je souhaite aussi exprimer toute ma gratitude envers les personnes de mon entourage qui m'ont soutenu et plus particulièrement :

Mes parents, mes beaux-parents et ma chère épouse, **Rujun**, à qui je dédie ce mémoire.

Table des matières

LISTE DES CONSTANTES, SYMBOLES ET ABREVIATIONS	V
LISTE DES FIGURES	VII
LISTE DES TABLEAUX	IX
PREAMBULE	1
<i>Références</i>	<i>8</i>
CHAPITRE I INTRODUCTION AUX THERAPIES CARDIAQUES ET AUX PACEMAKERS.....	10
I.1 ETAT DE L'ART DES MALADIES CARDIAQUES	12
<i>I.1.1 L'insuffisance cardiaque et les traitements de la thérapie de resynchronisation cardiaque</i>	<i>12</i>
<i>I.1.2 Bref rappel biologique du cœur.....</i>	<i>13</i>
<i>I.1.3 Limitations des pacemakers actuels dans le cadre de la CRT.....</i>	<i>18</i>
I.2 ETAT DE L'ART DES PACEMAKERS.....	19
<i>I.2.1 Bref historique du pacemaker</i>	<i>19</i>
<i>I.2.2 Classification générale des pacemakers</i>	<i>23</i>
<i>I.2.3 Description des principaux blocs d'un pacemaker conventionnel.....</i>	<i>25</i>
<i>I.2.4 Marché des pacemakers.....</i>	<i>26</i>
I.3 SYNTHÈSE DES ORIENTATIONS FIXÉES AU PROCESSEUR NEURONAL.....	27
I.4 DÉTAILS DES DIVERS OBJECTIFS AU SEIN DU PROJET ADAPTER	29
<i>Conclusions</i>	<i>32</i>
<i>Références</i>	<i>33</i>
CHAPITRE II METHODOLOGIE DE CONCEPTION MIXTE BASEE SUR LE PROTOTYPAGE VIRTUEL FONCTIONNEL.....	38
II.1 INTRODUCTION.....	40
II.2 METHODE DE CONCEPTION CLASSIQUE EN V	41
<i>II.2.1 Description de la méthode de conception en V.....</i>	<i>41</i>
<i>II.2.2 Limites de la méthode de conception en V.....</i>	<i>42</i>
II.3 PROTOTYPAGE VIRTUEL FONCTIONNEL.....	43
<i>II.3.1 Le Prototypage Virtuel Fonctionnel et ses avantages.....</i>	<i>43</i>
<i>II.3.2 Flot de conception du Prototypage Virtuel Fonctionnel.....</i>	<i>46</i>
II.4 NOTRE APPROCHE DE CONCEPTION DANS CE PROJET	48
<i>II.4.1 Synoptique de notre approche de conception dans le cadre de ADAPTER</i>	<i>48</i>
<i>II.4.2 Description explicite de notre approche de conception.....</i>	<i>49</i>
<i>II.4.3 Le travail présenté dans ce manuscrit.....</i>	<i>54</i>
<i>Conclusion.....</i>	<i>56</i>
<i>Références</i>	<i>58</i>
CHAPITRE III DESCRIPTION GLOBALE DU CRT-P ADAPTATIF DU PROJET ADAPTER	60
III.1 LE CAHIER DES CHARGES INITIAL	62
III.2 PARTITION GLOBALE DU CRT-P ADAPTATIF DU PROJET ADAPTER	63

III.3	DESCRIPTION FONCTIONNELLE DES DEUX BLOCS CENTRAUX DU CRT-P.....	65
III.3.1	<i>Fonctionnalités du contrôleur général.....</i>	66
III.3.2	<i>Fonctionnalités de processeur neuronal.....</i>	66
III.4	LES MODES DE FONCTIONNEMENT DU PACEMAKER ADAPTATIF	68
Conclusion.....		71
Références		72
CHAPITRE IV	DESCRIPTION DU RESEAU DE NEURONES (SNN1).....	74
IV.1	OBJECTIFS DE CONCEPTION DU RESEAU DE NEURONES (SNN1)	76
IV.2	DESCRIPTION STRUCTURELLE DE SNN1	77
IV.2.1	<i>Description structurelle de trois modules de SNN1</i>	77
IV.2.2	<i>L'architecture du module « Integrate & Fire ».....</i>	80
IV.2.2.1	<i>Description de l'intégrateur analogique avec fuite (CRRC).....</i>	80
IV.2.2.2	<i>Réponse de l'intégrateur avec fuite recevant une seule impulsion en entrée</i>	82
IV.2.2.3	<i>Réponse de l'intégrateur avec fuite recevant deux impulsions en entrée</i>	84
IV.2.2.4	<i>Expressions de l'amplitude maximale en sortie du CRRC</i>	85
IV.2.2.5	<i>Choix de seuil de SNN1</i>	87
IV.2.2.6	<i>Simulation fonctionnelle de l'intégrateur avec fuite.....</i>	92
Conclusion.....		93
Références		94
CHAPITRE V	LES ALGORITHMES D'APPRENTISSAGE ADOPTES POUR LE SNN1	96
V.1	L'ALGORITHME DU MODE NON-ADAPTATIF : ALGORITHME D'APPRENTISSAGE DE HEBB (HL) ...	98
V.2	L'ALGORITHME DU MODE ADAPTATIF-DETECTE: ALGORITHME D'APPRENTISSAGE PAR RENFORCEMENT (RL)	100
V.2.1	<i>Choix de l'approche de l'algorithme RL.....</i>	100
V.2.2	<i>Description explicite de l'algorithme RL.....</i>	102
V.3	SIMULATION DES ALGORITHMES HL ET RL SOUS MATLAB.....	105
Conclusion.....		108
Références		109
CHAPITRE VI	SIMULATION DU SYSTEME COMPLET	110
VI.1	OBJECTIFS ET ATTENTES DE LA SIMULATION DU SYSTEME COMPLET	112
VI.2	MODELES SIMPLES DU CŒUR AVEC LE SCENARIO D'EXERCICE	113
VI.2.1	<i>Scénario d'exercice</i>	114
VI.3	MODELES COMPORTEMENTAUX DU CŒUR (SURFACES SV)	115
VI.3.1	<i>Surface 0</i>	115
VI.3.2	<i>Surface 1</i>	116
VI.3.3	<i>Surface 2</i>	117
VI.4	MODELE DU BRUIT INTEGRE DANS LA SIMULATION GLOBALE	119
VI.5	SIMULATIONS GLOBALES DU SYSTEME AVEC DIFFERENTS MODELES DU CŒUR SOUS VHDL-AMS..	121
VI.5.1	<i>Surface 0</i>	121
VI.5.2	<i>Surface 1</i>	124

VI.5.3	Surface 2	127
	Conclusion.....	131
	Références	132
CHAPITRE VII	LES ETUDES SUR LE MULTIPLIEUR ANALOGIQUE DANS LE SNN1	134
VII.1	NOTRE FLOT DE CONCEPTION EN APPROCHE TOP-DOWN.....	137
VII.1.1	Comment extraire les spécifications techniques de chaque « composant » système?	138
VII.1.2	Quelle importance accorder à un modèle théorique de niveau N si nous ne connaissons pas la structure du « composant » (la topologie du circuit) ?	142
VII.1.3	Comment choisir la meilleure topologie électrique pour un « composant » système?	143
VII.2	DEVELOPPEMENT DE NOTRE APPROCHE CONCEPTION AU NIVEAU « COMPOSANT ».....	145
VII.3	MODELES THEORIQUES DU MULTIPLIEUR ANALOGIQUE.....	148
VII.4	SELECTION DES TOPOLOGIES SELON L'ETUDE BIBLIOGRAPHIQUE	154
VII.5	BILAN DE PUISSANCE DU RESEAU DE NEURONES ANALOGIQUES (SNN1)	158
	Conclusion.....	164
	Références	166
	CONCLUSIONS GENERALES ET PERSPECTIVES.....	170
	PUBLICATIONS ET COMMUNICATIONS ASSOCIEES A CE TRAVAIL	179
	ANNEXES	180
ANNEXE I.	LES LANGAGES DE MODELISATION MIXTE MULTI-DOMAINES.....	181
ANNEXE I.1	VHDL-AMS	181
ANNEXE I.2	MAST.....	183
ANNEXE I.3	MATLAB/SIMULINK.....	183
ANNEXE I.4	MODELICA	185
ANNEXE I.5	VERILOG-AMS	186
ANNEXE I.6	COMPARAISON DES LANGAGES DE MODELISATION	187
ANNEXE II.	OUTILS DE SIMULATION VHDL-AMS	190
ANNEXE II.1	QUESTA ADMS (MENTOR GRAPHICS)	190
ANNEXE II.2	SYSTEMVISION (MENTOR GRAPHICS)	191
ANNEXE II.3	SABERHDL (SYNOPTIS)	191
ANNEXE II.4	SIMPLORER 8.1 (ANSOFT).....	192
ANNEXE II.5	PORTUNUS (GROUPE CEDRAT).....	192
ANNEXE II.6	SMASH 5.15 (DOLPHIN INTEGRATION)	193
ANNEXE III.	SYNTHESE BIBLIOGRAPHIQUE SUR LES NEURONES DE TYPE SNN (SPIKING NEURAL NETWORK)	195
ANNEXE III.1	PHYSIOLOGIE DES NEURONES BIOLOGIQUES	195
ANNEXE III.1.1	LE POTENTIEL D'ACTION.....	196
ANNEXE III.1.2	SYNAPSE	196
ANNEXE III.2	LES ASPECTS DE CALCUL ET DE MODELISATION.....	196
ANNEXE III.2.1	MODELES DETAILLES DE NEURONES	196
ANNEXE III.2.2	MODELES FORMELS DE NEURONES IMPULSIONNELS (SPIKING NEURON)	197

ANNEXE III.3	COMMUNICATION NEURONALE AVEC DES DECLENCHEMENTS DES IMPULSIONS (SPIKES)...	198
ANNEXE III.3.1	TAUX DE CODAGE.....	198
ANNEXE III.3.2	CODAGE TEMPORAL PAR DES DELAIS RELATIFS.....	199
ANNEXE III.3.3	PROPRIETES DE CALCUL DES NEURONES IMPLUSIONNELS	199
ANNEXE IV.	CODE SOURCE VHDL-AMS : MODELE COMPORTEMENTAL STRUCTUREL	201
ANNEXE IV.1	MODELE DU PACEMAKER	201
ANNEXE IV.2	MODELE DU CŒUR	203
ANNEXE IV.3	MODELE DU SNN	206
ANNEXE IV.4	MODELE DES ALGORITHMES HL ET RL	207
	<i>Références</i>	<i>219</i>

Liste des constantes, symboles et abréviations

Abréviations

Notation	Signification	Unités usuelles
C_{ox}	Capacité d'oxyde de grille	F
I_D	Courant de drain	A
L	Longueur du canal	m
V_T	Tension de seuil	V
W	Largeur du canal	m
V_{dsat}	Tension de saturation	V
V_d	Polarisation de drain	V
V_g	Polarisation de grille	V
V_s	Polarisation de source	V
μ	Mobilité des porteurs	$\text{cm}^2 \cdot \text{V}^{-1} \cdot \text{s}^{-1}$

Abréviations et acronymes

Nom	Signification	Traduction française
AV	Atrioventricular	Atrioventriculaire
AVB	Atrioventricular block	Blocage atrioventriculaire
AVN	Atrioventricular node	Nœud atrioventriculaire
CAO	Computer-aided design	Conception assistée par ordinateur
CHF	Congestive Heart Failure	Insuffisance cardiaque congestive
CRT	Cardiac resynchronization therapy	Thérapie de resynchronisation cardiaque
CRT-D	CRT Defibrillator	Défibrillateur de CRT
CRT-P	CRT Pacemaker	Pacemaker de CRT
DSP	Digital signal processor	Processeur de signal numérique
ECG	Electrocardiography	Electrocardiographie
FPGA	Field Programmable Gate Array	Réseau de portes programmables
GRL	Guided reinforcement learning	apprentissage par renforcement guidé
HCR	Hit control rate	Taux de contrôl cadré
HL	Hebb learning	Apprentissage de Hebb
HR	Heart rate	Rythme cardiaque
ICD	Implantable cardioverter-defibrillator	Défibrillateur automatique implantable
LV	Left ventricle	Ventricule gauche
MEMS	Microelectromechanical system	Mcirosystème électromécanique
NL	Noise level	Niveau de bruit

PSR	Post synaptic response	Réponse post-synaptique
PVF	Virtual functional prototypage	Prototypage Virtuel Fonctionnel
RA	Right atrium	Oreillette droite
RL	Reinforcement learning	Apprentissage par renforcement
RV	Right ventricle	Ventricule droit
SAB	Sinoatrial block	Blocage sinusal
SAN	Sinoatrial node (or sinus node)	Nœud sinusal
SNN	Spiking neural network	Réseau de neurones analogique
SV	Stroke Volume	Volume d'éjection systolique
VV	Interventricular	Interventriculaire

Liste des figures

Figure 0-1 Partenaires du projet ADAPTER.....	4
Figure I-1 Circulation sanguine.....	13
Figure I-2 Coupe du cœur	14
Figure I-3 Représentation d'un ECG normal	15
Figure I-4 Représentation du blocage auriculo-ventriculaire (AVB).....	16
Figure I-5 Représentation du blocage sino-atrial (SAB).....	16
Figure I-6 (Gauche) John A. HOPPS ; (Droit) M. HOPPS testant un des premiers prototypes du stimulateur cardiaque.	19
Figure I-7 Pacemakers électroniques externes des années 1950.....	20
Figure I-8 le premier transistor.....	20
Figure I-9 Le premier pacemaker portable.....	21
Figure I-10 Le premier pacemaker implanté.....	21
Figure I-11 Evolution des pacemakers d'ELA médical durant les 43 dernières années.....	22
Figure I-12 Description des principaux blocs d'un pacemaker conventionnel.....	26
Figure I-13 Prévision de la valeur de marché de CRT-D et de pacemaker dans les prochaines années	27
Figure II-1 Cycle de conception classique en V et ses boucles possibles de reprise	42
Figure II-2 Méthodologie renforcée de la conception mixte avec le « Prototypage Virtuel Fonctionnel ».....	45
Figure II-3 Evolution des modèles de différents niveaux hiérarchiques du système CRT-P ..	53
Figure III-1 Partition globale du CRT-P adaptatif du projet ADAPTER	64
Figure III-2 Description interne du processeur neuronal	68
Figure III-3 Trois modes de fonctionnement du CRT-P adaptatif.....	69
Figure IV-1 Synopsis de trois modules analogique de SNN1.....	78
Figure IV-2 Schéma de CRRC.....	81
Figure IV-3 la réponse du CRRC recevant deux impulsions unitaires : PSR_j^i ; $j \in \{1, 2\}$	84
Figure IV-4 Variation d'amplitude de réponse CRRC en recevant 5 impulsions de PSR_j^i unitaires consécutives.....	88
Figure IV-5 Evolution de l'asymptote $LI^{i(K)\max}(t)$ et de $LI^i(t_{pK}^i)$ en fonction de N	89
Figure IV-6 Définitions de l'« Ecart » et du « Reste »	90
Figure IV-7 Calcul de l'« Ecart» en fonction de N pour le choix du seuil.....	91
Figure IV-8 Déclenchement de T^i avoir reçu quatre impulsions de PSR_j^i unitaires consécutives	92

Figure V-1 Définition de l'« Etat du neurone I & F » et de l'« Etat synaptique ».....	99
Figure V-2 Simulation de l'algorithme HL et RL avec une surface en forme de « Cloche »	105
Figure V-3 Progression des pointeurs AV et VV avec l'algorithme HL et RL	106
Figure VI-1 Le scénario d'exercice en fonction de HR	114
Figure VI-2 Simulations de la surface 0 en fonction de HR	116
Figure VI-3 Simulations de la surface 1 en fonction de HR	117
Figure VI-4 Simulations de la surface 1 en fonction de HR	119
Figure VI-5 Intégration de modèle de bruit dans la simulation globale.....	120
Figure VI-6 Simulation du modèle de bruit aléatoire uniforme et son histogramme	121
Figure VI-7 Simulation de la surface 0 sans bruit (NL0).....	122
Figure VI-8 Zoom sur le signal SV^M de la surface 0 sans bruit.....	122
Figure VI-9 Simulation de la surface 0 avec le bruit de NL100	123
Figure VI-10. Simulation de la surface 0 avec le bruit de NL300	123
Figure VI-11. Simulation de la surface 0 avec le bruit de NL500	124
Figure VI-12. Simulation de la surface 1 sans bruit.....	125
Figure VI-13. Simulation de la surface 1 avec le bruit de NL100	126
Figure VI-14. Simulation de la surface 1 avec le bruit de NL300	126
Figure VI-15. Simulation de la surface 1 avec le bruit de NL500	127
Figure VI-16. Simulation de la surface 2 sans bruit.....	128
Figure VI-17. Simulation de la surface 2 avec le bruit de NL100	129
Figure VI-18. Simulation de la surface 2 avec le bruit de NL300	129
Figure VI-19. Simulation de la surface 2 avec le bruit de NL500	130
Figure VII-1 Modèles structurels comportementaux du réseau de neurones analogiques.....	137
Figure VII-2 Positionnement de notre démarche dans le cycle de conception d'une application	139
Figure VII-3 Etablissement des modèles théoriques des composants.....	140
Figure VII-4 Simulation du système complet avec des modèles théoriques de niveaux différents.....	141
Figure VII-5 Equivalence entre modèles théoriques et physiques.....	142
Figure VII-6 Flot de conception liant l'approche système et l'approche circuit	145
Figure VII-7 Méthodologie de notre étude sur le multiplieur analogique	147
Figure VII-8 Simulation du système complet avec $\varepsilon_1=8.5^e-2$	150
Figure VII-9 Simulation du système complet avec $\varepsilon_1 = -2.0^e-2$	151
Figure Annex I-1 L'architecture signal mixte/multi-domaine du langage Verilog-AMS [18]	187

Liste des tableaux

Tableau V-1 Variation de R_j^i avec l'algorithme d'apprentissage Hebb.....	99
Tableau V-2 Evolution de SV selon l'algorithme RL.....	104
Tableau VI-1 Evolution des délais AV_{MLV} et VV_{MLV} dans le scénario d'exercice	115
Tableau VI-2 Evolution de SV_{gmax} en fonction de HR dans la simulation de la surface 2	128
Tableau VII-1 Modèles abstraits du multiplieur analogique.....	148
Tableau VII-2 Gammes acceptables des coefficients $\varepsilon_1 - \varepsilon_7$ garantissant le bon fonctionnement du système	149
Tableau VII-3 Non-linéarités visées des modèles théoriques du multiplieur analogique....	155
Tableau VII-4 Comparaison des spécifications extraites dans les articles des multiplieurs analogique	156
Tableau VII-5 Comparaison des spécifications extraites dans les articles de multiplieurs analogiques (suite).....	156
Tableau VII-6 Comparaison entre le mode de courant et le mode de tension dans la conception analogique.....	157
Tableau Annexe I-1 Comparaison de cinq langages de modélisation mixte [7]-[9].....	188

Préambule

Contexte de recherche

Cette thèse s'inscrit dans le cadre des recherches sur l'intégration de solutions massivement parallèles au sein du laboratoire InESS de Strasbourg.

Cette thématique vise à proposer des architectures analogiques alternatives dans des applications aux exigences extrêmes. Les architectures massivement parallèles se construisent autour de composants élémentaires simples mais ayant une fonctionnalité souple. Elles peuvent donc résoudre des problèmes complexes de traitement de l'information dans un temps restreint. Une telle définition s'apparente aux architectures de cellules neuronales du cerveau. C'est donc assez logiquement que l'équipe s'est intéressée aux architectures de cellules neuronales sous la forme de circuits analogiques.

La rencontre avec la société ELA Medical (Sorin group) a permis de s'interroger sur la pertinence de telles solutions analogiques en comparaison de solutions purement numériques implantées sur des DSP ou des circuits de type FPGA. Une première évaluation du couple puissance consommée et surface de silicium, laisse présager de l'intérêt d'une solution analogique. C'est dans ce contexte que la société ELA Medical a proposé à l'équipe de concevoir une première solution analogique servant de démonstrateur. Le défi fut donc de s'appuyer sur une première solution numérique, d'en extraire les fonctionnalités et de les décliner dans une version analogique. Ce démonstrateur permettra ensuite, à fonctionnalité équivalente, de mesurer les réels bénéfices industriels d'une version analogique au regard de la version numérique.

Cette étude de faisabilité d'une transposition numérique vers analogique ne s'est pas départie des contraintes industrielles. C'est pourquoi le processus d'étude, d'analyse et de conception des circuits s'est appuyé sur des méthodologies de conceptions dont une partie est issue des travaux de l'équipe. Ces méthodes adressent le niveau système et le niveau conception de circuits. Ce travail a été l'occasion de montrer leurs intérêts et de pointer des lacunes dans le catalogue des méthodes disponibles.

Contexte socio-économique de l'application

Selon les études rapportées par la société ELA Medical, environ 14 millions de personnes en Europe souffrent d'insuffisance cardiaque (CHF : Congestive Heart Failure) et ce nombre atteindra 30 millions en 2020. En France, l'Etat dépense à peu près 200 millions d'euros chaque année pour les traitements de l'insuffisance cardiaque. Ce type de maladie

correspond à un état dans lequel une anomalie de la fonction cardiaque est responsable de l'incapacité du myocarde à assurer un débit cardiaque suffisant pour couvrir les besoins énergétiques de l'organisme. La thérapie de resynchronisation cardiaque (CRT) apparaît comme un traitement efficace pour ces patients. Elle permet non seulement d'améliorer les fonctions cardiaques, mais aussi de réduire considérablement le risque de mort subite.

A ce jour, les solutions conventionnelles sont basées sur des pacemakers délivrant des impulsions électriques au muscle cardiaque et permettant d'accélérer le cœur lorsqu'il est trop lent et parfois de resynchroniser des délais de contraction entre les deux ventricules [1]-[3].

Toutefois, comme les conditions physiologiques et les rythmes cardiaques dépendent des conditions physiques du patient et varient d'une personne à l'autre, les performances des pacemakers conventionnels, ne pouvant donner que des battements cardiaques fixes, sont souvent inadaptées aux besoins réels. Par exemple, un rythme de battement cardiaque correspondant au besoin cardiaque d'un patient au repos, ne lui suffit pas s'il veut faire des exercices physiques. Donc, ce patient est obligé de demander à un médecin d'ajuster son pacemaker « au mieux » pour l'ensemble de ses besoins variés. Sachant que la qualité de vie des patients CHF est souvent mauvaise, une réponse spécifique devient urgente. Le futur des pacemakers est donc de trouver une solution auto-adaptative pour délivrer des impulsions électriques correspondant aux besoins cardiaques de chaque patient en temps réel. Un processeur électronique devant pouvoir reconnaître les situations ou les pathologies du patient afin d'adapter ses conditions cardiaques est donc indispensable.

Contexte pluridisciplinaire du projet ADAPTER

Les travaux de cette thèse s'inscrivent dans un vaste projet pluridisciplinaire : le projet ADAPTER (En anglais : Advanced compatibilities for **Adaptive** CRT devices with ECG **R**econstruction for building a telemedicine system for congestive heart failure patients) qui a reçu le label EUREKA en avril 2006.

Le projet a réuni les partenaires suivants : la société ELA Medical (Sorin group) comme porteur du projet, la société AI Medical Semiconductor promoteur du processeur neuronal numérique, la société Systems'ViP contributeur au développement d'une plateforme unifiée de développement et de validation, le laboratoire LTSI de l'Université de Rennes pour la

modélisation du cœur humain et le laboratoire InESS de l'Université de Strasbourg pour son expertise en conception de circuits analogiques (Figure 0-1).

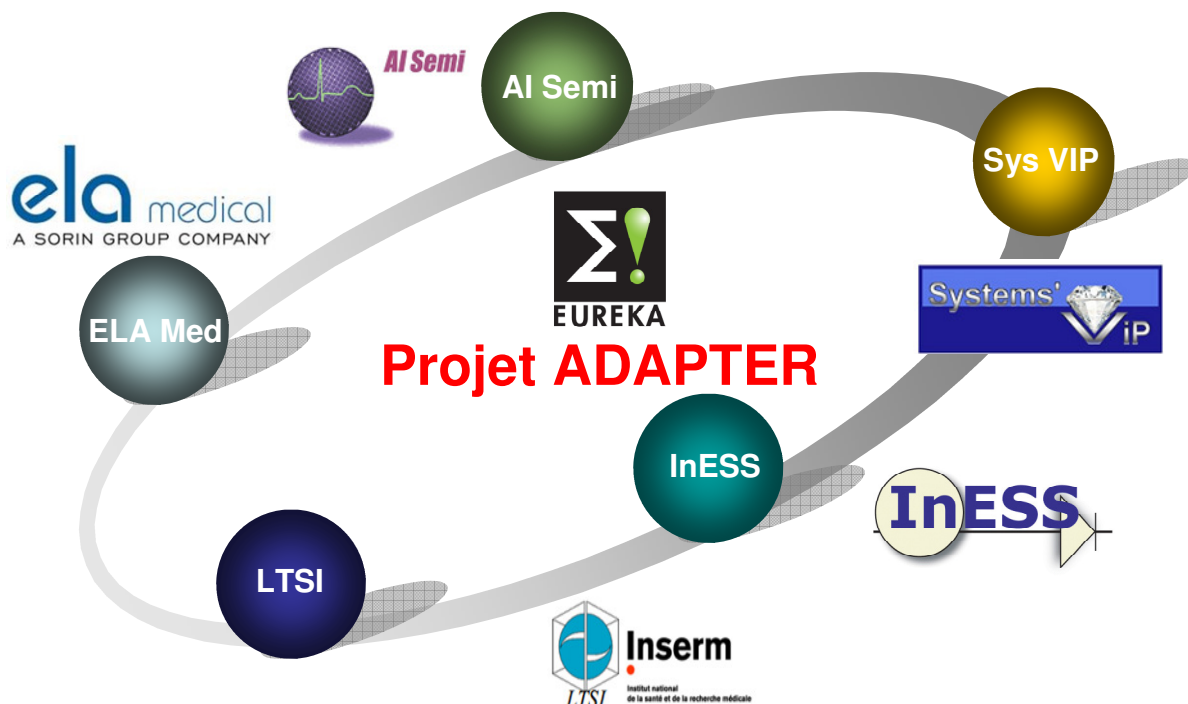


Figure 0-1 Partenaires du projet ADAPTER

Pour répondre aux attentes des patients, le pacemaker doit être adaptatif. Pour cela il doit pouvoir reconnaître les états physiologiques du patient à chaque instant ainsi que les principales pathologies cardiaques du malade. Cette reconnaissance nécessite 2 éléments complémentaires l'un de l'autre : un modèle physiologique du cœur et une unité d'analyse des paramètres cardiaques observés. Le premier élément fut développé conjointement par le laboratoire LTSI de Rennes et la société Systems'ViP. L'existence des modèles du cœur permet de répondre à 2 objectifs :

- formaliser les pathologies du cœur ce qui conduira à mieux les comprendre,
- construire artificiellement (donc rapidement) les bases de données d'apprentissage.

Le second élément est ici un processeur neuronal dont la première version (numérique) fut amenée par la société AI Semiconducteur. Une solution basée sur l'utilisation d'un processeur neuronal permet de répondre à la variabilité importante des situations pathologiques grâce à ses capacités de classifications des séquences observées, d'approximation d'une fonction quelconque et d'ajustement d'une consigne. Néanmoins, concevoir un réseau de neurones nécessite trois sous éléments :

- une base de données pour l'apprentissage et la validation du réseau,

- un algorithme d'ajustage des poids synaptiques (on parle aussi d'algorithme d'apprentissage),
- le choix d'une topologie du réseau et d'une architecture des neurones.

Le choix des architectures de neurones fait au sein du projet ADAPTER porte sur les neurones impulsionnels (spiking neuron). Le choix de la méthode d'apprentissage de ces neurones s'est arrêté sur un apprentissage « en ligne » (Online Learning). Ainsi, après une remise à zéro de sécurité, le processeur neuronal pourra immédiatement recommencer à reconstruire sa (ses) fonctionnalité(s).

Pour limiter le nombre d'interventions chirurgicales, l'autonomie du pacemaker doit être de 5 à 7 ans. La consommation énergétique devient alors une contrainte forte pour la conception du pacemaker. Si le principe de l'utilisation d'un processeur neuronale est admis, son implémentation hardware reste un challenge. La société ELA Medical a sollicité notre équipe afin de conduire une étude de faisabilité d'une alternative analogique afin d'évaluer le potentiel industriel d'une solution analogique comparativement à une solution numérique. Notre équipe s'est donc attachée à concevoir autant les architectures analogiques que les algorithmes spécifiques à ces architectures pour les phases d'apprentissage de la solution.

Structure de la thèse

Le **Chapitre I** commence par présenter l'état de l'art des maladies cardiaques, spécifiquement sur l'insuffisance cardiaque et la thérapie de resynchronisation cardiaque (CRT) pour ce type de maladie. Nous abordons ensuite un petit rappel biologique sur la structure et la fonctionnalité du cœur. Les limitations des pacemakers actuels dans le cadre CRT seront aussi montrées. Et puis une étude sur l'état de l'art des pacemakers va nous permettre de connaître les différents types de pacemakers au long de l'histoire de son développement et des principaux blocs dans un pacemaker bi-ventriculaire (dispositif CRT). Pour répondre aux besoins variés de chaque patient, nous justifierons notre orientation de recherche sur l'implémentation analogique des réseaux de neurone dans le dispositif CRT conventionnel. Les divers objectifs visés par le projet ADAPTER seront détaillés à la section suivante. A la fin de ce chapitre, nous allons introduire la synoptique de notre approche de conception utilisée dans le cadre du projet ADAPTER.

Le **Chapitre II** introduira la méthodologie de conception en VHDL-AMS que nous allons adopter dans cette thèse. Nous allons aussi décrire la méthodologie de « Prototypage Virtuel Fonctionnel » qui inspire la nôtre. Les langages et les outils de simulation seront présentés en annexe.

Le **Chapitre III** sera consacré à l'introduction globale du projet ADAPTER. Nous présenterons premièrement le cahier des charges et les objectifs initiaux. Nous décrirons la partition globale et l'architecture « maître-esclave » entre le contrôleur général du système et le processeur neuronal analogique. Ensuite, les trois modes de fonctionnement du système seront illustrés afin de montrer la fonctionnalité du processeur neuronal analogique.

Le **Chapitre IV** sera dédié à la description structurelle du processeur neuronal analogique. Nous analyserons en détail les blocs analogiques qui vont générer les délais cardiaques optimaux aux moments visés.

Le **Chapitre V** décrira les deux algorithmes d'apprentissage adoptés (Hebb & renforcement) pour les deux modes de fonctionnement (objectifs différents). Nous montrons aussi la première validation des algorithmes sous MATLAB®.

Le **Chapitre VI** correspondra à la simulation du système complet sous le langage VHDL-AMS. Pour tester la fonctionnalité du système, un modèle comportemental du cœur avec différents scénarios (constant / exercice) du battement cardiaque sera intégré dans la simulation. En plus, nous allons aussi y intégrer des modèles du bruit de différents niveaux (amplitude) pour valider les performances du système dans une situation plus réelle. Nous testerons l'ensemble du système en simulation ce qui validera d'une part l'architecture fonctionnelle et d'autre part notre approche de conception. Nous en attendons l'explicitation des plages dynamiques et des limites fonctionnelles qui serviront à l'élaboration du cahier des charges des circuits élémentaires.

Le **Chapitre VII** présentera notre approche de conception électronique pour les blocs élémentaires. Nous focalisons notre étude sur un bloc en particulier : le multiplieur analogique utilisé dans le processeur neuronal analogique. En appliquant notre méthodologie de conception présentée dans le chapitre II, nous décrirons comment extraire les spécifications de l'ensemble des simulations du système (chapitre VI) au niveau du « modèle structurel comportemental ». Une fois le cahier des charges extrait deux options s'ouvrent :

- Soit nous démarrons le dimensionnement depuis une topologie existante (c'est du ReUse),

- Soit nous commençons un cycle de conception complet avec sélection de topologie puis dimensionnement.

Références

- [1] Woollons, D.J.; , "To beat or not to beat: the history and development of heart pacemakers," *Engineering Science and Education Journal*, vol.4, no.6, pp.259-268, Dec 1995
- [2] Panescu, D.; , "Cardiac resynchronization therapy," *Engineering in Medicine and Biology Magazine, IEEE*, vol.24, no.2, pp. 22- 26, March-April 2005
- [3] Heart Rhythm Society, "Pacemaker" [online], Available : <http://www.hrspatients.org/patients/treatments/pacemakers.asp>

Chapitre I

Introduction aux thérapies cardiaques et aux pacemakers

Sommaire du chapitre I

CHAPITRE I	INTRODUCTION AUX THERAPIES CARDIAQUES ET AUX PACEMAKERS.....	10
I.1	ETAT DE L'ART DES MALADIES CARDIAQUES	12
I.1.1	<i>L'insuffisance cardiaque et les traitements de la thérapie de resynchronisation cardiaque</i>	<i>12</i>
I.1.2	<i>Bref rappel biologique du cœur.....</i>	<i>13</i>
I.1.3	<i>Limitations des pacemakers actuels dans le cadre de la CRT.....</i>	<i>18</i>
I.2	ETAT DE L'ART DES PACEMAKERS.....	19
I.2.1	<i>Bref historique du pacemaker</i>	<i>19</i>
I.2.2	<i>Classification générale des pacemakers</i>	<i>23</i>
I.2.3	<i>Description des principaux blocs d'un pacemaker conventionnel.....</i>	<i>25</i>
I.2.4	<i>Marché des pacemakers</i>	<i>26</i>
I.3	SYNTHESE DES ORIENTATIONS FIXEES AU PROCESSEUR NEURONAL.....	27
I.4	DETAILS DES DIVERS OBJECTIFS AU SEIN DU PROJET ADAPTER	29
	<i>Conclusions</i>	<i>32</i>
	<i>Références</i>	<i>33</i>

I.1 Etat de l'art des maladies cardiaques

I.1.1 L'insuffisance cardiaque et les traitements de la thérapie de resynchronisation cardiaque

L'insuffisance cardiaque est une appellation générique englobant de nombreux cas cliniques. C'est généralement une maladie grave qui survient lorsque le cœur ne fonctionne plus très bien, en étant plus capable de pomper suffisamment de sang pour répondre aux besoins en oxygène du corps. Petit à petit, les personnes qui en sont atteintes s'essoufflent, se fatiguent plus facilement, le cœur se pénalisant lui-même.

L'insuffisance cardiaque survient généralement chez des individus âgés dont la santé est fragilisée depuis plusieurs années par des troubles cardiaques ou respiratoires. Il s'agit d'une cause importante de mortalité chez les personnes âgées. L'âge moyen auquel survient l'insuffisance cardiaque est de 73,5 ans, avec 2/3 des patients ayant plus de 70 ans.

Aux Etats-Unis, on recense près de 5 millions de personnes atteintes d'insuffisance cardiaque, avec 500 000 nouveaux cas chaque année. Les traitements médicamenteux (par pharmacopées) sont inefficaces et les hospitalisations sont coûteuses. 12 à 15 millions de consultations par an et 6,5 millions d'hospitalisations y sont attribuables. Le gouvernement des Etats-Unis dépense ainsi 27 milliards de dollars américains sur les traitements de l'insuffisance cardiaque [1].

En France, le nombre d'insuffisants cardiaques s'élevait en 2001 à 500 000, avec 120 000 nouveaux cas chaque année. Il y a 3,5 millions de consultations et 150 000 hospitalisations pour insuffisance cardiaque par an, avec une durée moyenne de séjour de 11 jours. Plus de 32 000 décès annuels sont attribuables à l'insuffisance cardiaque. Les dépenses liées à l'insuffisance cardiaque représentent plus de 1 % des dépenses médicales totales de l'Etat français. Les frais de traitement de cette maladie sont estimés entre 109 à 208 millions d'euros chaque année [2].

Il devient urgent de chercher des solutions efficaces pour réduire, d'un point de vue humain le risque de mort et d'un point de vue économique les coûts des traitements.

I.1.2 Bref rappel biologique du cœur

Le cœur est un organe vital d'environ 250 à 350 grammes formé de trois parois qui sont de l'intérieur vers l'extérieur: l'endocarde, le myocarde (muscle proprement dit), et le péricarde. Il se compose de deux parties (droite, gauche) sans communication entre elles (voir Figure I-1). Chaque partie est constituée de deux cavités : une oreillette aux parois fines et un ventricule aux parois épaisses. Ces deux parties du cœur, bien qu'ayant le même fonctionnement, n'ont pas le même but. L'oreillette droite reçoit, grâce à deux veines caves, le sang pauvre en oxygène, provenant des organes. Elle se contracte et envoie ce sang appauvri dans le ventricule droit qui se contracte à son tour et l'envoie vers les poumons, grâce aux artères pulmonaires, afin qu'il se décharge en dioxyde de carbone et se recharge en oxygène. L'oreillette gauche reçoit, par quatre veines pulmonaires, le sang riche en oxygène. Elle se contracte et envoie ce sang dans le ventricule gauche qui se contracte à son tour et l'envoie vers les organes afin de les approvisionner, grâce à l'artère aorte. Un système de valves, appelées valvules, qui s'ouvrent et se ferment en fonction de la pression sanguine, sépare les oreillettes des ventricules d'une part et les ventricules des artères d'autre part. Ainsi la circulation du sang dans le cœur est à sens unique. C'est la fermeture des valvules qui est à l'origine du bruit des battements du cœur. Le sang accomplit l'aller-retour entre les parties droite et gauche du cœur plus de 100 000 fois par jour.

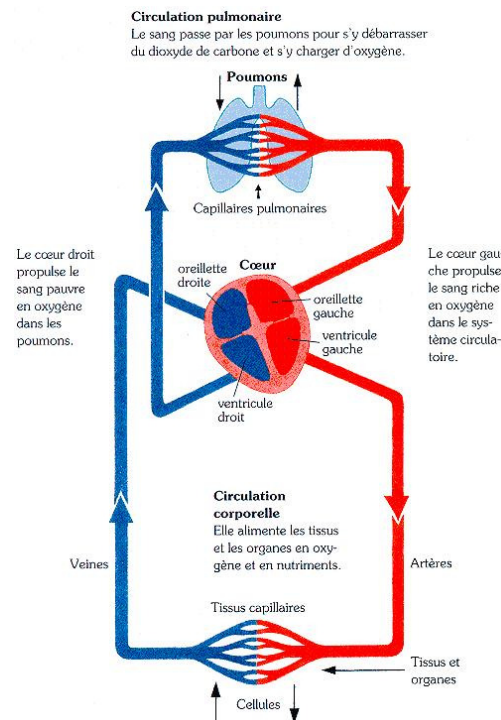


Figure I-1 Circulation sanguine

Cette succession de battements n'est pas provoquée par le cerveau. Le cœur comprend un système autonome et automatique d'influx nerveux qui assure chacun de ses battements (Figure I-2). Ce système de « pacemaker » interne est appelé le nœud sinusal ② (SAN), et se trouve dans la paroi de l'oreillette droite. Cette région produit une impulsion électrique ③ qui gagne une région centrale du cœur appelée nœud auriculo-ventriculaire ou auriculo-ventriculaire ④ (AVN). L'onde se propage le long des parois des oreillettes et provoque une contraction (en médecine on parle de dépolarisation) des oreillettes, c'est-à-dire l'expulsion du sang vers les ventricules. Le nœud AVN joue alors le rôle de régulateur et de modérateur des ondes de dépolarisation captées qu'il transmet ensuite aux ventricules. Les ondes parcourent alors les ventricules par des faisceaux appelés faisceaux de HIS ⑤ qui provoquent à leur tour, leur contraction et l'expulsion du sang hors du cœur. Enfin, le cerveau humain ① joue bien entendu un rôle important puisqu'il contribue à moduler le rythme cardiaque en agissant sur la fréquence d'émission des ondes de dépolarisation.

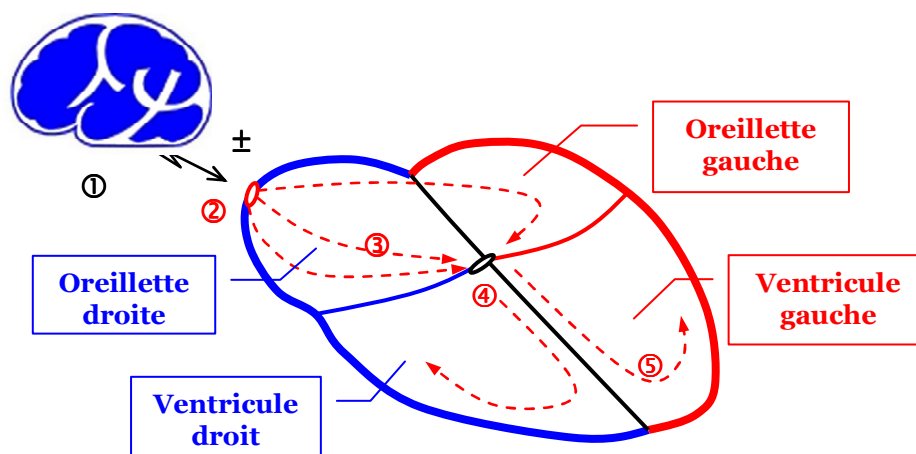


Figure I-2 Coupe du cœur

L'activité électrique du cœur peut être enregistrée par l'électrocardiogramme (ECG), qui montre si l'activité électrique du cœur est synchronisée. Cet examen est capital pour le diagnostic du type d'arythmie. L'ECG donne principalement trois tracés. Le tracé de l'onde P correspond à la dépolarisation (la contraction) des deux oreillettes (droite et gauche), et celui de l'onde QRS correspondant à celle des ventricules. Suit une onde T qui correspond à l'essentiel de la repolarisation (la relaxation) des ventricules (Figure I-3). Entre les battements, le système électrique se recharge. Entre deux contractions, le muscle cardiaque se décontracte et les oreillettes se remplissent à nouveau de sang.

Chapitre I

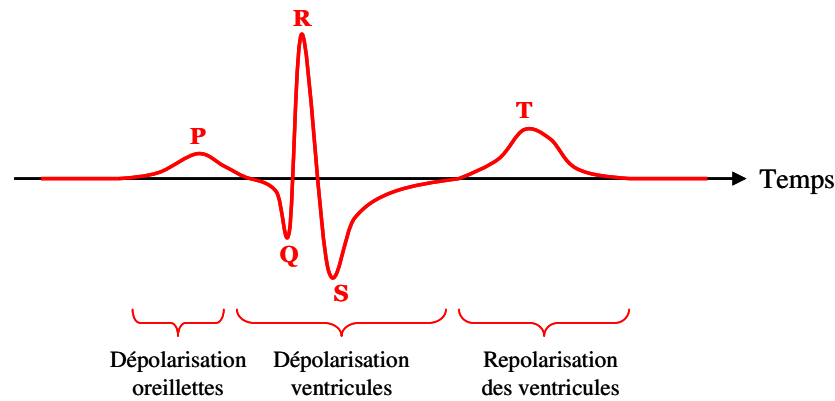


Figure I-3 Représentation d'un ECG normal

L'arythmie survient lorsque l'excitation électrique naît ailleurs que dans le nœud sinusal, ou le nœud auriculo-ventriculaire, ou le réseau de HIS ou bien que l'onde électrique ne suit plus les voies préférentielles de propagation. La contraction cardiaque qui en résulte s'écarte du rythme normal. Le battement supplémentaire qu'elle provoque se nomme extrasystole, par opposition à la contraction normale nommée systole.

Nous avons vu que la séquence normale d'activation du cœur comprend tout d'abord une activation des oreillettes (auriculaire), ensuite la contraction des grands muscles cardiaques des ventricules. Une extrasystole perturbe cette séquence. Souvent, elle est suivie d'une pause compensatrice ressentie comme une sorte d'interruption.

Une accélération soudaine, brutale et prolongée du rythme cardiaque est suivie d'une diminution de la pression artérielle, ce qui peut provoquer des vertiges et un sentiment de faiblesse ou de grande fatigue. La désynchronisation entre les oreillettes et les ventricules, enfin, entraîne également une sensation de palpitation. Cette sensation peut être perçue derrière ou à gauche du sternum, mais aussi dans le cou et même dans les oreilles ou dans le ventre.

Certains troubles sont plus graves que d'autres, surtout s'ils compromettent la fonction cardiaque. Dans certaines conditions pathologiques, un cercle vicieux entre fonction cardiaque et rythme anormal s'enclenche. Ainsi un cœur affaibli peut être la cause d'une arythmie, mais l'arythmie en soi peut affaiblir le cœur.

On classe les arythmies en fonction de leur lieu de formation et de leurs effets sur le rythme cardiaque. Si le trouble se déclare dans les oreillettes ou le nœud auriculo-ventriculaire, on parle d'arythmie supra ventriculaire; s'il apparaît dans les ventricules, on parle d'arythmie ventriculaire. Si une arythmie accélère le rythme cardiaque, on parle de

tachycardie; si elle le ralentit, on parle alors de bradycardie. Nous montrons ci-dessous quelques défauts du cœur.

- Blocage auriculo-ventriculaire (AVB) : La transmission des ondes par AVN est difficile, il en résulte un délai rallongé entre l'onde P et Q. Si le degré de AVB augmente, le stade suivant entraîne une dépolarisation des ventricules (ondes QRS) une fois sur deux jusqu'à disparition complète des ondes QRS.

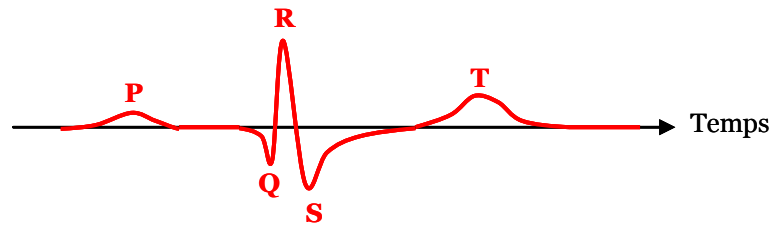


Figure I-4 Représentation du blocage auriculo-ventriculaire (AVB)

- Blocage sino-atrial (SAB) : Il n'y a plus de contraction des oreillettes.

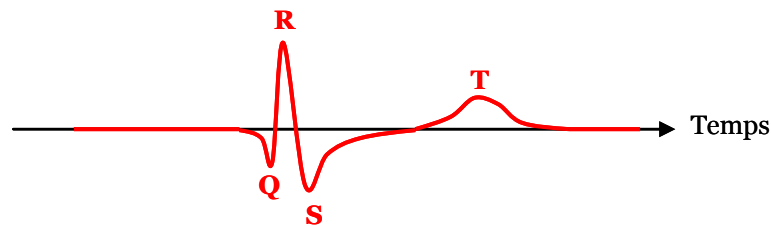


Figure I-5 Représentation du blocage sino-atrial (SAB)

- Blocage de HIS : Il n'y a plus de transmission des ondes à travers les faisceaux de HIS dans les ventricules. Les formes de l'ECG sont fortement asymétriques.

Plus généralement, les maladies cardiaques peuvent être structurelles ou fonctionnelles. Tout ce qui perturbe le cœur, son efficacité, sa capacité à se remplir et se contracter, et son approvisionnement en oxygène, vont altérer la coordination entre celui-ci, les reins et les vaisseaux et donc endommager non seulement le cœur mais aussi le reste du corps. Les maladies cardiaques peuvent provoquer :

- une embolie (caillot de sang formé à un endroit et se déplaçant jusqu'à boucher un vaisseau à distance) qui peut aboutir à une obstruction d'artères.
- une sténose (rétrécissement des vaisseaux arrivant au cœur)

Chapitre I

- une dilatation (extension des cavités provoquant un élargissement due à une augmentation de la pression)
- une hypertrophie (épaississement des parois du cœur causant une diminution de la taille des cavités et de la flexibilité du cœur)
- un reflux (retour en arrière de sang causant une augmentation de la pression sur les vaisseaux sanguins des poumons et du foie)
- un infarctus: mort de tissus (perte permanente de tissu cardiaque due à une carence en oxygène aboutissant à la cicatrisation)
- des arythmies (irrégularités des battements du cœur)
- une contraction insuffisante (le cœur ne se vide et ne se remplit pas complètement)
- une incapacité du cœur à faire face à la demande en oxygène et à l'élimination des déchets, comme par exemple durant une activité physique.
- des douleurs, fréquemment dues à l'ischémie (déficit en oxygène à cause de la diminution du flux sanguin), (angine de poitrine ou angor)

Les deux premières maladies sont externes au cœur. De la troisième à la sixième on peut les considérer comme d'origine structurelle (l'infarctus correspondant à l'arrêt de la structure). Les quatre dernières sont principalement d'origine fonctionnelle. On y relève l'importance du débit ou de la pression sanguine qui est le centre d'intérêt de l'application médicale. Parmi ces maladies fonctionnelles on y trouve l'insuffisance cardiaque. Les patients atteints d'insuffisance cardiaque ont une réduction de l'efficacité du ventriculaire gauche qui, pour 30% des patients, est causée par la mauvaise conduction électrique dans le faisceau de HIS du ventricule gauche. Cette mauvaise conduction désynchronise la contraction des ventricules droit et gauche et provoque une baisse considérable de la pression systolique (Blood Pressure = BP) et par conséquent du volume d'éjection systolique (Stroke Volume = SV).

Pour répondre à l'insuffisance cardiaque, la thérapie de resynchronisation cardiaque (CRT) a été développée [3]. Cette thérapie permet aussi de réduire considérablement le risque de mort subite [4]. Les pacemakers permettant la mise en œuvre de la thérapie CRT sont appelés dans cette thèse : CRT-P. Ils stimulent les ventricules droits et gauches à chaque battement cardiaque en accord avec le rythme cardiaque imposé par le nœud sinusal naturel et sont conçus pour resynchroniser les contractions du ventricule gauche et droit. Le délai auriculo-ventriculaire (AV) et le délai inter ventriculaire (VV) sont les deux paramètres

principaux qui déterminent l'efficacité de la thérapie. Nous pouvons les calculer en fonction des instants de contractions de l'oreillette droite (RA), du ventricule gauche (LV) et du ventricule droit (RV) au travers les équations suivantes :

$$AV = RV - RA \quad (I.1.1),$$

$$VV = LV - RV \quad (I.1.2).$$

I.1.3 Limitations des pacemakers actuels dans le cadre de la CRT

Les pacemakers actuels contribuant à la CRT présente quatre limitations majeures:

1. Les délais AV et VV, varient de patient à patient et doivent être déterminées par l'échocardiographie qui coûte cher, prend beaucoup du temps et dépend de l'opérateur (le médecin ou le clinicien).
2. Les délais AV et VV optimaux dépendent du rythme cardiaque de manière non triviale. Pour certains patients, le délai AV, par exemple, est plus court au cours de l'exercice alors que pour d'autres il est plus long.
3. Les positionnements des électrodes de stimulation sont très importants (surtout celui du ventricule gauche). Les « non-répondeurs » à la CRT peuvent devenir des « répondeurs » en modifiant ces positionnements.
4. Il reste cependant 30% des patients qui ne montrent aucune amélioration clinique avec l'application de la CRT.

Les deux premières limitations entrent directement dans le champ d'action du projet ADAPTER. Les deux dernières limitations restent des questions ouvertes.

I.2 Etat de l'art des pacemakers

I.2.1 Bref historique du pacemaker

Un pacemaker conventionnel est un dispositif médical qui est placé dans la poitrine ou l'abdomen d'un humain pour aider à contrôler les rythmes cardiaques anormaux. Cet appareil utilise des impulsions électriques pour forcer le cœur à battre à un rythme normal. Nous décrirons dans cette section l'histoire brève du pacemaker avec le développement de la technologie microélectronique afin de montrer la motivation de la recherche microélectronique pour l'amélioration du pacemaker.

Au début des années 1950, en tant qu'ingénieur électricien faisant des études d'hypothermie au Conseil National de Recherches du Canada, John A. Hopps (Figure I-6) a découvert accidentellement que le cœur "refroidi" pouvait être redémarré sans dommage à l'aide d'un stimulus électrique doux en se basant sur les observations du chirurgien cardiothoracique Wilfred Gordon Bigelow (Figure I-6) de l'Hôpital Général à Toronto [1]. Cette découverte conduit John A. Hopps à créer le premier stimulateur cardiaque artificiel (pacemaker) externe : un dispositif qui permettrait de sauver des milliers de vies.



**Figure I-6 (Gauche) John A. HOPPS ;
(Droit) M. HOPPS testant un des premiers prototypes du stimulateur cardiaque.**

Plusieurs photos, dont nous ne connaissons pas le régime de droit d'auteur, de ce document sont extraites des sites internet. En cas de réclamation, nous nous engageons à faire les modifications demandées

Le premier pacemaker fait 30 cm (environ un pied) de long, ce qui est beaucoup trop gros pour être implanté dans le corps. Nous pouvons voir quelques modèles de cette période sur la Figure I-7. Comme la plupart des dispositifs électroniques de cette période, ce pacemaker utilise des tubes à vide pour générer des impulsions électriques. Les impulsions électriques sont transmises via une électrode bipolaire à l'oreillette en utilisant une approche

transveineuse. La stimulation auriculaire a été facilement atteinte et le rythme cardiaque peut être contrôlé sans contractions inconfortables de la paroi thoracique. Mais les premiers pacemakers devaient être déposés sur des chariots pour les déplacements et branchés sur une prise murale pour obtenir leur puissance. Ils ne sont pas portables parce qu'ils ne peuvent pas être loin d'une prise électrique.



Figure I-7 Pacemakers électroniques externes des années 1950

Quelques années auparavant, en 1947, trois physiciens américains du laboratoire BELL : John Bardeen, William Shockley, et Walter Brattain ont inventé le premier transistor (Figure I-8), une des inventions les plus importantes du 20^{ième} siècle [7]. En effet, c'est cette invention primordiale qui va rendre possible et accélérer le développement du pacemaker. Le développement du transistor en silicium et de sa première disponibilité commerciale en 1956 a été un événement clé qui a conduit au développement rapide de pacemakers industriels.



Figure I-8 le premier transistor

Earl E. Bakken, ingénieur électricien et co-fondateur de Medtronic Inc. a produit le premier pacemaker portable à piles en 1957 (Figure I-9). Cette boîte en métal permet aux

Chapitre I

patients de vivre en la portant et ce pacemaker a été reconnu comme une des premières applications réussies de la technologie du transistor dans des dispositifs médicaux en lançant le nouveau domaine de recherche de « l'électronique médicale » [6].



Figure I-9 Le premier pacemaker portable

Un an après, en 1958, un pacemaker simple (fait à la main, voir Figure I-10) a été implanté pour la première fois dans la cage thoracique d'un homme, M. Arne LARSSON, un suédois de 43 ans à ce moment là. Cette implantation réussie, signifie l'arrivée de la nouvelle époque du pacemaker implantable. Avec seulement quelques transistors, ce premier prototype a commencé à améliorer la vie des patients cardiaques. À l'aide de pacemakers implantables, M. LARSSON a vécu encore 43 ans avant son décès en 2001 à l'âge de 86 ans [8].

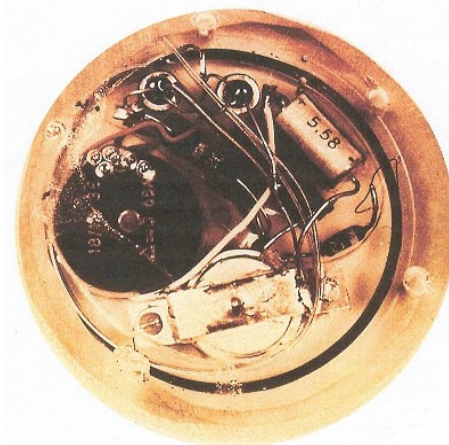


Figure I-10 Le premier pacemaker implanté

Au fil des ans, les progrès des batteries et des technologies du transistor ont permis aux pacemakers de devenir conjointement de plus en plus petits et de plus en plus performants, jusqu'à ce qu'ils puissent être implantés facilement dans le corps humain. La batterie au lithium-ion a été développée pour remplacer la pile zinc-oxyde de mercure. En conséquence,

la longévité des pacemakers a nettement été augmentée. L'avancement de la technologie RF permet aux médecins de communiquer avec le pacemaker et d'ajuster des paramètres sans contact. Si nous analysons les pacemakers des dernières années, nous trouvons qu'ils profitent toujours des fruits des dernières évolutions technologiques dans le domaine microélectronique. En ajoutant des capteurs supplémentaires et des microprocesseurs d'assistance, les pacemakers sont devenus très sophistiqués et sont capables de détecter et de stocker les événements en utilisant plusieurs algorithmes.

Nous prenons comme exemple l'évolution des pacemakers d'ELA Medical (Sorin groupe) (Figure I-11) [13]. Avec 2 transistors dans une boîte de 120 g, ELA Medical a fabriqué leur premier pacemaker commercial en 1964 qui avait quelques mois d'autonomie. Suivant la croissance rapide de la densité d'intégration, ils ont pu intégrer quelques 10 millions de transistors dans une boîte de 8 cc, qui pèse seulement 20 g (c'est 7 fois plus petit et 6 fois plus léger que le modèle de 1964). En plus, la durée de vie du pacemaker a été largement augmentée : neuf ans au lieu de quelques mois grâce à l'importante diminution de la consommation de puissance du transistor et au développement des batteries.

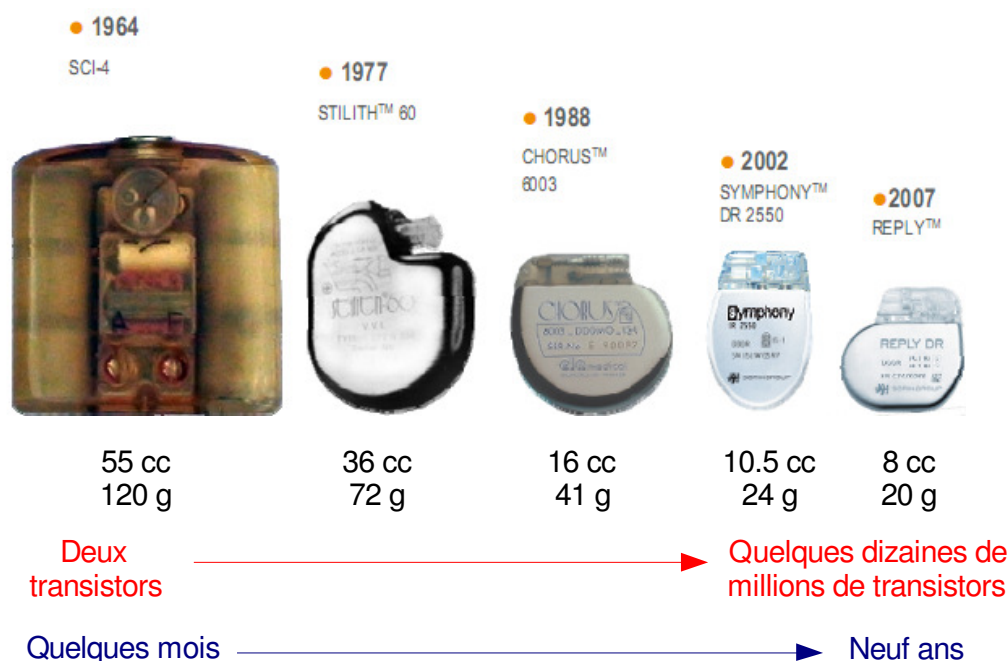


Figure I-11 Evolution des pacemakers d'ELA médical durant les 43 dernières années

Nous avons tracé ici une histoire incomplète du pacemaker artificiel en lien avec le développement de la technologie microélectronique. En coopérant intimement avec des médecins cliniciens (comme les pionniers dans ce domaine), les recherches en microélectronique peuvent largement contribuer à améliorer la qualité de vie des patients

Chapitre I

cardiaques en rendant le pacemaker plus petit, plus intelligent et plus performant. C'est la motivation qui porte les électroniciens dans leurs efforts de recherche.

I.2.2 Classification générale des pacemakers

Aujourd'hui, les pacemakers fonctionnent comme des ordinateurs miniaturisés qui sont chargés de surveiller des battements de cœur naturels. Si nécessaire, ils envoient une impulsion électrique via une électrode implantée dans un muscle (ou une paroi musculaire) générant une stimulation supplémentaire, qui contribue à réguler le rythme cardiaque. Selon la méthode de stimulation, nous pouvons déterminer la classification suivante [35]:

- la stimulation transcutanée (électro-choc externe),
- la stimulation épicaudique (temporaire),
- la stimulation transveineuse (temporaire),
- la stimulation permanente (interne).

Parmi celles-ci, la stimulation permanente s'adresse prioritairement aux cœurs qui battent trop lentement (**Bradycardie**). Le pacemaker peut stimuler un des muscles du cœur en y injectant une impulsion électrique (que l'on appelle aussi stimulation électrique). L'injection d'une impulsion électrique nécessite de placer l'électrode sur la paroi du muscle visé. Cette électrode est ensuite reliée galvaniquement (électriquement) au pacemaker par un cordon. Selon le placement et le nombre d'électrodes, nous trouvons trois types de pacemakers conventionnels.

- **Pacemaker mono chambre** : une électrode de stimulation avec son fil de liaison est placée dans une chambre du cœur : l'oreillette ou le ventricule.
- **Pacemaker double chambre** : deux électrodes de stimulation avec leur fil de liaison sont placées dans deux chambres du cœur; typiquement on en place un dans l'oreillette et l'autre dans le ventricule. Ce type de pacemaker reproduit la stimulation naturelle du cœur en aidant le cœur à coordonner la fonction entre les oreillettes et les ventricules.
- **Pacemaker bi-ventriculaire** : ce type de pacemaker peut stimuler à la fois les parois septale (paroi séparant les deux ventricules) et latérale du ventricule gauche. Les

trois électrodes sont placées respectivement dans l'oreillette droite et chacun des deux ventricules. Ainsi, les délais entre les contractions auriculaires et les parois latérales du ventricule droit, ainsi qu'entre les deux ventricules (via la contraction de la paroi septale) peuvent être ajustés pour obtenir la fonction cardiaque optimale. Ce type de pacemaker est appelé aussi le CRT-P (Pacemaker de **Thérapie de Resynchronisation Cardiaque**). Dans cette thérapie de resynchronisation, l'idée est de contracter en premier les parois du ventricule droit (externe et interne) avant de contracter le ventricule gauche. La contraction d'un muscle entraînant son durcissement, la contraction de la paroi externe du ventricule gauche s'appuiera alors sur un « arrière plan » durcis pour améliorer l'efficacité de l'éjection du sang.

Le sujet de cette thèse est basé sur le pacemaker bi-ventriculaire. Nous allons le décrire en détail dans la section I.2.3.

Soulignons qu'il est facile de provoquer la contraction artificielle d'un muscle décontracté. Il suffit d'injecter une stimulation électrique juste avant la stimulation naturelle; c'est le cas dans le CRT-P. Il en va tout autrement s'il faut « effacer » une contraction pour la recréer avec du retard. Ce cas de figure sera traité avec des défibrillateurs. Le défibrillateur automatique implantable (en anglais : implantable cardioverter defibrillator-ICD) est un dispositif conventionnel pour traiter les cœurs qui battent trop vite (Tachycardie). L'ICD détecte automatiquement les anomalies du rythme cardiaque rapide et les traite en libérant un choc électrique ou des stimulations rapides pour rétablir le rythme normal.

Nous résumons la classification des pacemakers selon leur application de thérapie :

- **Pacemaker simple** : il comprend les pacemakers monos chambres et les pacemakers doubles chambres pour les traitements de bradycardie.
- **CRT-P** : il s'agit de pacemaker bi-ventriculaire pour les traitements de CRT.
- **ICD** : c'est le défibrillateur pour les traitements de tachycardie.
- **CRT-D** : c'est un nouveau type de pacemaker (c'est-à-dire le CRT-Défibrillateur) qui est une combinaison du CRT-P et l'ICD afin de traiter la bradycardie, la CRT et la tachycardie en même temps. Dans les derniers rapports cliniques, nous constatons une amélioration importante sur des patients de l'insuffisance cardiaque en utilisant un CRT-D [32] [33].

I.2.3 Description des principaux blocs d'un pacemaker conventionnel

Afin de présenter les blocs fonctionnels d'un pacemaker conventionnel, nous considérons un CRT-D, qui comprend l'ensemble des fonctionnalités par rapport aux autres types de pacemaker. Le CRT-D est une combinaison du CRT-P et le ICD. En général, un CRT-D comprend les blocs décrits ci-dessous et illustrés dans la Figure I-12:

1. **Microprocesseur à faible consommation** : il fonctionne comme le cerveau du pacemaker. Il va analyser des données enregistrées par des capteurs analogiques et envoyer des commandes de stimulations électroniques aux électrodes pour stimuler le cœur aux moments optimaux en utilisant des algorithmes spécifiques. Afin de prolonger la durée de fonctionnement du pacemaker, la consommation énergétique du microprocesseur doit être très faible.
2. **ROM & RAM** : Elles contiennent respectivement les programmes informatiques (software) qui gouvernent les fonctions du pacemaker et les données captées ou calculées en temps réel.
3. **Batterie** : la source d'énergie du système complet qui doit assurer une autonomie d'au moins 5 ans. Il y a classiquement deux parties dans ce bloc batterie. Une première batterie dédiée à l'alimentation sous tension élevée du défibrillateur (ICD) et la seconde batterie pour le CRT-P à plus basse consommation.
4. **Entrées & Sorties analogiques**: le pacemaker reçoit des signaux du cœur via des capteurs analogiques et envoie ses différents signaux de stimulations (de tensions élevés pour l'ICD et de tensions faibles pour le CRT-P) aux électrodes via des générateurs.
5. **Interface de programmation RF (Wifi, GPS)** : l'interface de programmation RF permet aux médecins d'ajuster des paramètres du composant CRT et de récupérer des données cardiaques des patients sans « contact ». Au vu de l'évolution des vitesses de transferts et de traitements de l'informatique, il est possible d'envisager à l'avenir le développement de la télémédecine (consultations à distance) via internet. Lors d'une situation d'urgence, le centre d'appel reçoit les informations GPS qui vont permettre de localiser rapidement le patient pour gagner un laps de temps vital.

6. **Défibrillateur à tension élevée** : le défibrillateur doit envoyer un choc électrique lors de la détection d'une anomalie de rythme cardiaque (Tachycardie) pour stopper le cœur. Cette partie consomme une bonne part d'énergie car elle a besoin d'une tension élevée pour provoquer le choc.
7. **Transducteurs MEMS** : Les transducteurs MEMS sont placés dans les oreillettes ou les ventricules. Ils ont un double rôle : appliquer les stimuli et capter les paramètres hémodynamiques. L'application des stimuli se fait via les électrodes. La mesure des paramètres hémodynamiques se fait par les capteurs. Les informations captées permettent de fournir les données utiles au diagnostic et donc au pilotage du cœur.

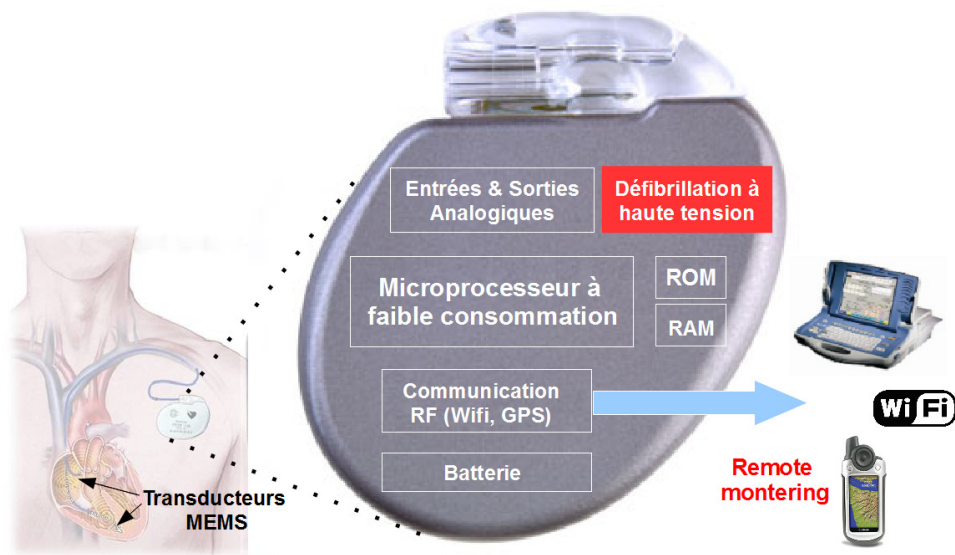


Figure I-12 Description des principaux blocs d'un pacemaker conventionnel

I.2.4 Marché des pacemakers

Les pacemakers et les défibrillateurs implantables ont connu une croissance considérable dans les dernières années : 50 000 pacemakers en France et plus de 800 000 unités sont implantés dans le monde entier chaque année. Avec l'augmentation du nombre des personnes âgées, avec la miniaturisation des circuits électroniques et avec les importants progrès des performances des pacemakers, le marché du pacemaker implantable va continuer à augmenter dans les prochaines années. Selon [34], on extrait la Figure I-13 qui montre la valeur du marché des CRT-D et celui du marché total des pacemakers en 2007 ainsi que la

Chapitre I

prévision pour 2012. Nous observons une croissance de 45 % pour les CRT-D et de 29 % pour l'ensemble des pacemakers.

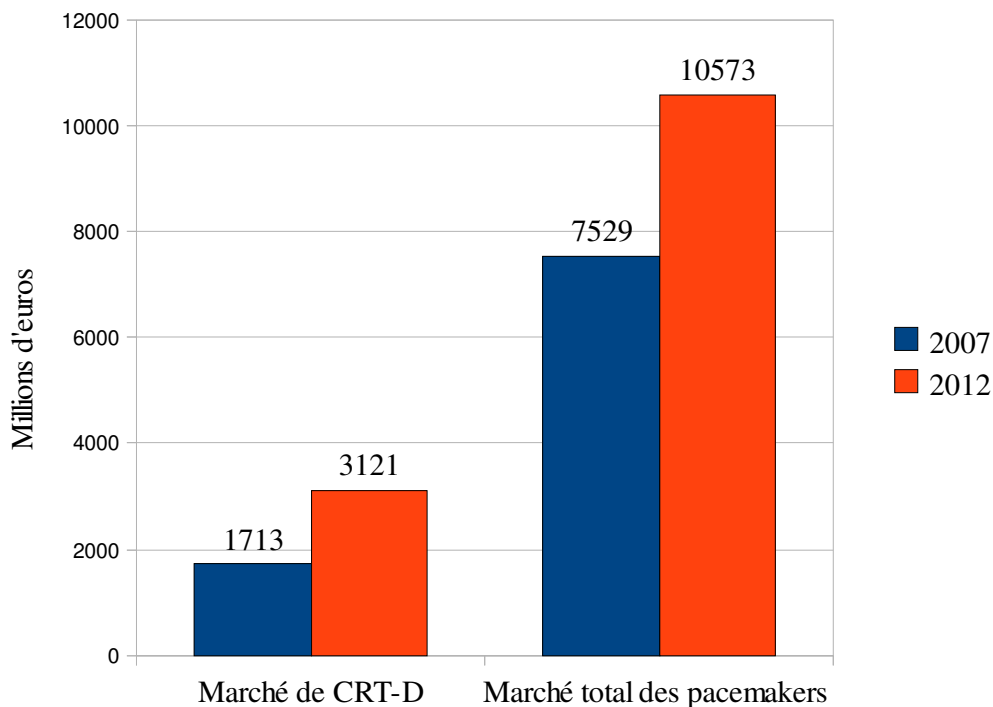


Figure I-13 Prévision de la valeur de marché de CRT-D et de pacemaker dans les prochaines années

I.3 Synthèse des orientations fixées au processeur neuronal

Une solution technique pouvant palier les limitations des CRT-P décrites au paragraphe I.1.3, doit être capable de s'adapter aux conditions physiques et physiologiques variées du patient implanté et optimiser en continu (en temps réel) les décalages temporels des contractions de l'oreillette puis de chaque ventricule (les délais AV et VV) pour chaque période cardiaque. La variabilité importante des conditions physiques et physiologiques du patient et les possibles variations à court ou long terme causées par le vieillissement ou la santé générale de ce dernier, suggère une solution basée sur l'utilisation des réseaux de neurones artificiels correspondant aux caractéristiques suivantes :

- Les lois de contrôle (de pilotage) qui permettraient de résoudre le problème sont inconnues ou très difficiles à formaliser ou à expliciter. Cependant, nous disposons d'un ensemble d'exemples qui correspondent à des entrées du problème et aux

solutions qui leurs sont données par des experts (cliniciens dans le cas de pacemaker) ;

- Le problème fait intervenir des données bruitées (les signaux issus des capteurs) ;
- Le problème peut évoluer : apparition d'une fièvre, etc. ;
- Le problème nécessite une grande rapidité de traitement : optimisation des délais AV et VV en temps réel;

Les autres facteurs importants attendus des réseaux des neurones artificiels sont : leur robustesse et leur tolérance aux pannes. La perte d'un neurone du réseau peut être compensée par le réajustement des autres neurones restant actifs. L'adoption de réseaux de neurones dans des applications du même type est en croissance comme le souligne une revue d'articles dans la communauté scientifique [36]–[58].

De plus, une implémentation matérielle massivement parallèle [18]–[20] peut répondre aux besoins ciblés de dimensions réduites et de très faible consommation d'énergie.

Les ajustements des délais AV et VV proposés par le processeur neuronal seront basés sur les données issues des capteurs hémodynamiques (les transducteurs dans la Figure I-12) implantés. Or, la localisation des électrodes sur le muscle cardiaque participe à l'efficacité d'une stimulation électrique par le pacemaker. Aussi, lors de l'implantation des transducteurs, le chirurgien va, via le pacemaker dans son mode basic, observer le débit cardiaque (c'est-à-dire le Stroke Volume : SV) durant trois à cinq minutes. Au regard des résultats observés, le clinicien va modifier (ou non) la position des électrodes. Il procèdera par itération pour trouver une efficacité relative du pacemaker dans son fonctionnement restreint qui deviendra de fait le régime de référence et par suite le régime de sécurité. Au final, le clinicien fixe les délais AV et VV qui, pour la localisation finale des électrodes, va lui permettre de trouver une efficacité du SV maximale dans l'état du patient (ici sous anesthésie). Ces délais de références seront appelés P_{AV} et P_{VV} .

Le processeur neuronal devra après chaque retour au régime de sécurité, réapprendre à proposer les valeurs de pacing (les délais) P_{AV} et P_{VV} . Après avoir satisfait à cette condition, il pourra basculer sur des modes plus complexes lui permettant de proposer des ajustements des délais plus appropriés pour maximiser le volume de sang éjecté (SV).

Les tâches d'amélioration du paramètre SV n'utilisent pas d'algorithmes d'optimisations au sens classique mais utilisent les capacités d'apprentissage, de classification et

Chapitre I

d'approximation de fonctions non explicites des réseaux de neurones. Les informations utilisées et les informations délivrées sont fonction du temps. C'est pourquoi, le choix s'est tourné vers la troisième génération de réseau de neurones artificiels. Un réseau de neurones impulsifs (Spiking Neural Network : SNN) utilise le moment du déclenchement des neurones impulsifs pour exprimer l'information analogique. Il offre naturellement la capacité de résoudre des problèmes dans les domaines continus telle que l'analyse des séries temporelles [21] [22]. Ainsi, même avec une complexité croissante de leur structure, les neurones impulsifs sont normalement plus faciles à implémenter dans un réseau de neurones de grande taille [23]. Au cours des dernières années, de nombreux articles discutant les implémentations analogiques/numériques de SNN ont été publiés dans la communauté scientifique [24]–[31], confortant l'orientation choisie pour cette recherche de nouveaux dispositifs CRT-P adaptatif.

I.4 Détails des divers objectifs au sein du projet ADAPTER

Le dispositif CRT-P visé dans le projet ADAPTER s'insère entre d'un côté le cœur qu'il doit surveiller et maîtriser et de l'autre côté les interfaces avec le milieu médical. En régime « normal », seul le couple cœur – CRT-P collabore. Lors de visites de contrôle médical, la troisième partie s'active.

Au delà du contenu de cette thèse, le projet ADAPTER comprend quatre autres objectifs industriels que nous souhaitons rappeler ici:

- **Objectif 1 (le cœur):** Le développement d'une nouvelle génération de pacemakers dédiés aux insuffisants cardiaques sévères (appelés CRT-P), incluant deux fonctionnalités originales :
 1. Une fonction d'optimisation et d'adaptation continue de la stimulation électrique (« pacing ») en fonction des besoins spécifiques du patient, des variations de son activité et de l'évolution de sa pathologie.
 2. Une fonction de reconstruction de l'ECG à partir des signaux électriques endocardiques captés par les transducteurs (l'EGM). Le laboratoire LTSI étudie une approche par calcul vectoriel 3D ou une approche par réseau de neurones dédié. Ce dernier ne faisant pas partie du processeur neuronal.

- **Objectif 2 (le dispositif CRT-P):** L'introduction d'un processeur neuronal doit permettre d'optimiser les délais AV/VV et à partir de l'ECG ou de l'EGM de détecter des évolutions pathologiques qui vont conditionner les lois de contrôle et d'optimisation des délais.
 1. Le processeur neuronal comporte un premier réseau neuronal (que nous appellerons SNN1) dédié à la prédiction des délais AV et VV. Ces prédictions doivent suivre trois modes. Le premier mode qui sera le mode sécurité servant de départ pour les modes suivants. Le second mode, vise à maximiser le volume de sang éjecté sans autre information qu'une estimation du SV issue d'un traitement des données des capteurs hémodynamiques (les transducteurs). Le dernier mode corrigera les règles d'optimisation à partir des informations issues de la surveillance de l'ECG.
 2. Au sein du processeur neuronal, un réseau spécifique (que nous appellerons SNN2) analyse les signaux de l'ECG pour y détecter des anomalies et peut-être reconnaître des pathologies spécifiques. Ces informations seront transmises vers le premier réseau de neurones (SNN1) dans son troisième mode de fonctionnement.
- **Objectif 3 (industriel):** dans ce contexte pluridisciplinaire, il importe de fédérer les travaux autour d'une plateforme de développement industriel unique. Cette plateforme permettra d'interfacer le cœur (par le biais de ses modèles), le personnel médical (via l'interface clinique RF) et les concepteurs du pacemaker (les électroniciens, les spécialistes de la batterie, les ingénieurs du packaging, etc.). Cette plateforme de simulation commune utilisera un langage fédérateur qui est à ce jour : VHDL-AMS. Ce langage ayant une standardisation, il est indépendant des outils de simulation (c'est quasiment vrai). L'intérêt d'une telle plateforme va au-delà des phases de développement. Elle peut servir à la validation des concepts industriels et contribuer à préparer les tests de production. Durant les phases de développement, la plateforme a permis, de tester plusieurs scénarii de fonctionnement du cœur humain en relation avec les prédictions du processeur neuronal. La convergence des résultats de simulations a permis de valider ou d'infirmer les modèles du cœur et/ou les fonctionnements du processeur dans les divers modes. Pour cela il a fallu développer des modèles des diverses parties (cœur, pacemaker, instruments médicaux) avec le langage VHDL-AMS. De plus, l'introduction du langage VHDL-AMS nous permet,

Chapitre I

à nous électroniciens, de glisser progressivement, et de manière sûre, d'une représentation abstraite (forme comportementale) vers une représentation réaliste du dispositif (forme structurelle). Elle a aussi imposé de redéfinir très clairement les entrées et sorties des différents modules. Dans le cadre pluridisciplinaire cela a permis d'harmoniser les échanges et assurer une cohérence des analyses et par suite des résultats.

- **Objectif 4 (Clinique):** L'introduction d'un canal de communication entre le composant CRT-P et les instances médicales permet d'envisager à moyen terme d'offrir un service de télésurveillance des patients de l'insuffisance cardiaque et réduire les coûts de suivi ou d'hospitalisation.

Conclusions

Ce chapitre d'introduction à la problématique du projet ADAPTER auquel ce travail de thèse appartient, a abordé premièrement un état de l'art de l'insuffisance cardiaque et sa thérapie actuelle (thérapie de resynchronisation cardiaque : CRT), deuxièmement une brève histoire du pacemaker. La tendance spécifique de recherche sur les dispositifs CRT, nous a montré le développement conjoint de pacemakers artificiels et de technologies microélectroniques ainsi que les limitations sur les dispositifs CRT conventionnels. Afin d'améliorer la qualité de vie de ces patients et de répondre à leurs besoins physiques en temps réel, une implémentation d'un processeur neuronal pour rendre le pacemaker auto-adaptatif a été proposée par le projet ADAPTER.

Plus précisément, nous avons choisi le réseau de neurones impulsionnels (en anglais : spiking neural network) qui offre naturellement la capacité de résoudre des problèmes dans le domaine continu telles que l'analyse des séries temporelles pour l'implémentation de ce processeur. Pour réaliser une étude de faisabilité industrielle, notre thèse est basée sur une approche analogique du processeur neuronal qui envisage de réduire la consommation d'énergie et la surface de silicium à fonctionnalité équivalente du pendant numérique.

La complexité d'un projet multi-domaine et multi-technologie comme le projet ADAPTER, nous a amené à adopter une approche de conception avancée pour augmenter la qualité et la fiabilité des produits avec une réduction sur les délais du projet. De ce fait, la méthodologie basée sur le Prototypage Virtuel Fonctionnel en utilisant le langage VHDL-AMS a été employée par les partenaires du projet ADAPTER pour établir une plate-forme commune de conception. En parallèle, pour la conception du niveau circuit, j'ai également proposé une méthode qui aide les concepteurs de circuits à extraire au mieux les spécifications de plus bas niveau et qui facilite la sélection de topologies potentiellement aptes à répondre aux exigences visées. Nous pouvons ensuite adopter la méthode ensembliste introduite dans la thèse de François SCHWARTZ [59] et permettant de les classer au sens de la robustesse. Finalement, on effectue la sélection topologique selon la conjonction de l'ensemble des critères.

Références

- [1] <http://www.emedicinehealth.com>, "Congestive heart failure" [online], Available: http://www.emedicinehealth.com/congestive_heart_failure/article_em.htm
- [2] F. Delahaye, G. de Gevigney, "Epidemiologie de l'insuffisance cardiaque," *Annales de Cardiologie et d'Angéiologie*, Vol. 50, Issue 1, pp. 6-11, Feb. 2001.
- [3] Panescu, D.;, "Cardiac resynchronization therapy," *Engineering in Medicine and Biology Magazine, IEEE*, vol.24, no.2, pp. 22- 26, March-April 2005
- [4] K. A. Ellenbogen, B. Wilkoff, G. N. Kay, Device therapy for congestive heart failure, Philadelphia, PA: Elsevier Inc., 2003, pp. 47–91.
- [5] Conseil national de recherches Canada, "Le génie au service de la qualité de vie" [online], Available: http://www.nrc-cnrc.gc.ca/fra/education/innovations/decouvertes/fauteuil_roulant.html
- [6] Lillehei CW., Gott VL., Hodges PC., Long D M., Bakken EE., ; "Transistor pacemaker for treatment of complete atrioventricular dissociation." *JAMA*, April 30, 1960; 172(18): 2006-2010.
- [7] "The Nobel Prize in Physics 1956". Nobelprize.org. 20 Oct 2010 http://nobelprize.org/nobel_prizes/physics/laureates/1956/.
- [8] Heart Rhythm Society, "Biography of Arne Larsson" [online], Available: <http://www.hrsonline.org/news/ep-history/notable-figures/arnelarsson.cfm>.
- [9] Woollons, D.J.; , "To beat or not to beat: the history and development of heart pacemakers," *Engineering Science and Education Journal* , vol.4, no.6, pp.259-268, Dec 1995
- [10] Acierno LJ. The History of Cardiology. London-Casteron-New York: Parthenon Publishing Group, 1994.
- [11] Haddad, S.A.P.; Houben, R.P.M.; Serdijin, W.A.; , "The evolution of pacemakers," *Engineering in Medicine and Biology Magazine, IEEE*, vol.25, no.3, pp.38-48, May-June 2006
- [12] Geddes, L.A.; , "Historical highlights in cardiac pacing," *Engineering in Medicine and Biology Magazine, IEEE* , vol.9, no.2, pp.12-18, Jun 1990
- [13] ELA medical, "Sorin group CRM presentation" [online], Available: <http://www.sorin-crm.com/uploads/Media/crmpres.pdf>
- [14] R. Rom, J. Erel, M. Glikson, K. Rosenblum, R. Ginosar, D. L. Hayes, "Adaptive cardiac resynchronization therapy device: a simulation report", *Pacing and Clinical Electrophysiology*. vol. 28, no. 11, pp. 1168–1173, Nov. 2005.
- [15] R. Rom, et al. "Adaptive cardiac resynchronization therapy device based on spiking neurons architecture", *IEEE Transaction on Neural Network*, vol. 18, no. 2, pp. 542–550, Mar. 2007.
- [16] F. Braunschweig, B. Kjellstrom, F. Gadler, C. Linde, "Optimization of cardiac resynchronization therapy by continuous hemodynamic monitoring: a case report", *Journal of Cardiovascular Electrophysiology*, vol. 15, pp. 94–96, Jan. 2004.

- [17] D. Odonnell, V. Nadurata, A. Hamer, P. Kertes, U. Mohammed, “Long term variations in optimal programming of cardiac resynchronization therapy devices”, *Pacing and Clinical Electrophysiology*, vol. 28, suppl. 1, pp. S24–S26, Jan. 2005.
- [18] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.
- [19] P. J. Antsaklis, “Neural networks for control systems”, *IEEE Transaction on Neural Network*, vol. 1, no. 2, pp. 242–244, Jun. 1990.
- [20] M. Mahowald and R. Douglas, “A silicon neuron,” *Nature*, vol. 354, pp.515-518, 1991.
- [21] W. Maass, “Networks of spiking neurons: The third generation of neural network models”, *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.
- [22] W. Maass, “Fast sigmoidal networks via spiking neurons,” *Neural Computation*, vol. 9, issue 2, pp. 279–304, Feb. 1997.
- [23] S. M. Bohte, J. N. Kok, H. La Poutre, “Error-back propagation in temporally encoded networks of spiking neurons,” *Neurocomputing*, vol. 48, no. 1–4, pp. 17–37, Oct. 2002.
- [24] G. Indiveri, “A low-power adaptive integrate-and-fire neuron circuit,” in *Proc. IEEE International Symposium on Circuits and Systems. IEEE*, May 2003, pp. IV-820-IV-823.
- [25] E. Farquhar and P. Hasler, “A bio-physically inspired silicon neuron,” *IEEE Transactions on Circuits and Systems*, vol.52, no. 3, pp. 477-488, Mar. 2005.
- [26] J. Wijekoon and P. Dudek, “Compact silicon neuron circuit with spiking and bursting behaviour,” *Neural Networks*, vol. 21, no. 2-3, pp. 524-534, March-April 2008.
- [27] M. Rastogi, V. Garg, and J. Harris, “Low power integrate and fire circuit for data conversion,” in *IEEE International Symposium on Circuits and Systems, ISCAS 2009, IEEE*, May 2009, pp. 2669-2672.
- [28] R. Sarpeshkar, “Analog versus digital: extrapolating from electroincs to neurobiology”, *Neural Computation*, vol. 1, no. 7, pp. 1601–1638, Oct. 1998.
- [29] L. Gatet, H. Tap-Béteille, F. Bony, “Comparison between analog and digital neural network implementations for range-finding applications”, *IEEE Transaction on Neural Network*, vol. 20, no. 3, pp. 460–470, Mar. 2009.
- [30] F. Anstotz, Y. Hu, J. Michel, J. L. Sohler, D. Lachartre, “Filtres: a 128 channels VLSI mixed front end read out electronic development for microstrip detectors”, *Nuclear Instruments and Methods in Physics Research, sec.A*, vol. 412, no. 1, pp. 123–134, Jun. 1998.
- [31] W. Maas, C. M. Bishop, *Pulsed neural networks*. Cambridge, MA: MIT Press, 1999.
- [32] JK. Ghali, J. Boehmer, AM. Feldman, LA. Saxon, T. Demarco, P. Carson, P. Yong, EG. Galle, J. Leigh, FL. Ecklund, MR. Bristow. “Influence of diabetes on cardiac resynchronization therapy with or without defibrillator in patients with advanced heart failure. ” *J Card*, Nov, 13(9): 769-73, Fail. 2007,.
- [33] RT. Journal, A1. Bradley, J. David, “Combining Resynchronization and Defibrillation Therapies for Heart Failure” *JF JAMA: The Journal of the American Medical Association JO JAMA YR 2003 FD May 28*
- [34] A. Ripart, “From Smart Pacemaker to Remote Monitoring of Cardiac Function,” *keynote of the IEEE International Symposium on Circuits and Systems*, Paris, France, Jun. 1, 2010.

Chapitre I

- [35] Wikipedia, "Artificiel Pacemaker" [online], Available: http://en.wikipedia.org/wiki/Artificial_pacemaker#cite_note-Mond_1982-2
- [36] Montalvo, A.J. et al., "An analog VLSI neural network with on-chip perturbation learning." *IEEE Journal of Solide State Circuits*, 32(4). pp. 535—543, April 1997.
- [37] Montalvo, A.J. et al. "Towards a general-pourpose analog VLSI neural network with on-chip learning" *IEEE Trans. On Neural Networks* 8(2), pp. 413-423, March 1997
- [38] Bo, G.M., Caviglia, D.D., Chible, H. and Valle, M., "A circuit architecture for analog on-chip back propagation learning with local learning rate adaptation." *Analog Integrated Circuits and Signal Processing* 18(2/3), pp.163-174, 1999
- [39] Lu, C., Shi,B.and Chen, L., "An on-chip BP learning neural network with ideal neuron characteristics and learning rate adaption" *Analog Integrated Cirucuits and Signal Processing* 31,pp.55-62,2002
- [40] Diotalevi, F., Valle.M. , Bo. G. M., Biglieri, E.and Caviglia, D.D. "An analog on-chip learning circuit architecture of the weight perturbation algorithm." *ISCAS'2000, IEEE International Symposium on Circuits and Systems*. Geneva, Switerland, 28-31 May, 2000, pp. II 717-II720. (ISBN: 0-7803-5485-0).
- [41] L. Bruzzoneetal., "A Technique for the Selection of Kernel-Function Parameters in RBF Neural Networks for Classification of Remote-SensingImages", *IEEE Tran. On Geoscience and Remote Sensing*, vol. 37, No. 2, pp. 1179-1184, 1999.
- [42] F.Diotalevi, etal., "An Analog on-chip Learning Circuit Architecture of the Weight Perturbation Algorithm", in *Proc. ISCAS2000*, Geneva (Switzerland), 28-31 May, 2000, pp.419-422.
- [43] K.Hirotsu and M.Brooke, "An analog neural network chip with random weight change learning algorithm," in *Proc. Int. Joint Conf. Neural Networks*, Oct.1993, pp.3031-3034.
- [44] M.Jabri and B.Flower, "Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks," *IEEE Trans. Neural Networks*, vol.3, Jan.1992
- [45] A.J.Montalvo, R.S.Gyurcsik, and J.J.Paulos, "An analog VLSI neural network with on-chip perturbation learning" *IEEE J.Solid-State Circuits*, vol.32, Apr.1997.
- [46] B.Burton, F.Kamran, R.G.Harley, T.G.Habetler, M.A.Brooke, and R.Poddar, "Identification and control of induction motor stator currents using fast on-line random training of a neural network," *IEEE Trans. Ind. Appicat.*, vol.32, May 1997
- [47] T.Morie and Y.Amemiya, "An all-analog expandable neural network LSI with on-chip backpropagation learning" *IEEE J.Solid-State Circuits*, vol.29, Sept.1994
- [48] C.R.Schneider and H.C.Card, "Analog CMOS deterministic Boltzmann circuits," *IEEE J.Solid-State Circuits*, vol.28, no.8, Aug.1993
- [49] Y.He and U.Cilingiroglu, "A charged-based on-chip adaptation Kohonen neural network," *IEEE Trans. Neural Networks*, July 1993, pp.849-852
- [50] Y.K.Choi and S.Y.Lee, "Subthreshold MOS implementation of neural networks with on-chip error back-propagation learning" in *Proc.Int.Joint Conf. Neural Network*, July 1993, pp.849-852

- [51] C.Schneider and H.Card, "Analog CMOS synaptic learning circuits adapted from invertebrate biology," *IEEE Trans.Circuits Syst.*, vol.38, Dec.1991
- [52] J.A.Lansner and T.Lehmann, "An analog Cmos chip set for neural network with arbitrary topologies," *IEEE Trans.Neural Network*, vol.4, May 1993
- [53] J.B.Lont and W.Guggenbuhl, "Analog CMOS implementation of a multilayer perceptron with nonlinear synapses," *IEEE Trans. Neural Networks*, vol.3, May 1992.
- [54] B.Linares-Barranco, E.Sanchez-Sinencio, A.Rodriguez-Vazquez, and J.L.Huertas, "A CMOS analog adaptive BAM with on-chip learning and weight refreshin," *IEEE Trans.Neural Networks*, vol.4, May 1993
- [55] T.Shima, T.Kimura, Y.Kamatani, T.Itakura, Y.Fujita, and T.Iida,"Neuro chips with on-chip back-propagation and/or Hebb learning," *IEEE J.Solid-State Circuits*, vol.27, Dec.1992.
- [56] P.W.Hollis and J.J.Paulos,"A neural network learning algorithm tailored for VLSI implementation," *IEEE Trans, Neural Networks*, vol.5, Sept.1994
- [57] T.Lehmann, E.Brunn, and C.Dietrich,"Mix analog/digital matrix-vector multiplier for neural network synapses," *Analog Integrated Circuits Signal Processin*, vol.9, 1996
- [58] P.W. Hollis and J.J.Paulos,"A neural network learning algorithm tailored for VLSI implementation," *IEEE Trans. Neural Networks*, vol.5, Sept. 1994.
- [59] F. SCHWARTZ, "Méthodologie de conception d'un système analogique massivement parallèle. Application à la conception de réseaux de neurones travaillant en mode courant." Ph.D. dissertation, Université de Strasbourg, 2010

Chapitre II

Méthodologie de conception mixte

basée sur le Prototypage Virtuel

Fonctionnel

Sommaire du chapitre II

CHAPITRE II	METHODOLOGIE DE CONCEPTION MIXTE BASEE SUR LE PROTOTYPAGE VIRTUEL FONCTIONNEL.....	38
II.1	INTRODUCTION.....	40
II.2	METHODE DE CONCEPTION CLASSIQUE EN V	41
II.2.1	<i>Description de la méthode de conception en V</i>	<i>41</i>
II.2.2	<i>Limites de la méthode de conception en V</i>	<i>42</i>
II.3	PROTOTYPAGE VIRTUEL FONCTIONNEL.....	43
II.3.1	<i>Le Prototypage Virtuel Fonctionnel et ses avantages.....</i>	<i>43</i>
II.3.2	<i>Flot de conception du Prototypage Virtuel Fonctionnel.....</i>	<i>46</i>
II.4	NOTRE APPROCHE DE CONCEPTION DANS CE PROJET	48
II.4.1	<i>Synoptique de notre approche de conception dans le cadre de ADAPTER</i>	<i>48</i>
II.4.2	<i>Description explicite de notre approche de conception</i>	<i>49</i>
	<i>Conclusion.....</i>	<i>56</i>
	<i>Références</i>	<i>58</i>

II.1 Introduction

A la suite de l'évolution conjointe de la technologie et des besoins du marché, de meilleures performances des systèmes sont souvent exigées dans un « time-to-market » de plus en plus strict. En outre, l'évolution vers les System-on-Chip (SoC) demande aux concepteurs de concevoir des circuits analogiques ou mixtes ayant des performances élevées sur une seule puce. Cela amène des problèmes variés comme l'hétérogénéité de technologies, la sensibilité des paramètres liée à la diminution géométrique des composants, des fonctions de plus en plus non-linéaires, etc [1]. Tous ces problèmes vont augmenter la difficulté de la conception tant analogique que mixte, en conséquence les industries prennent plus de risques. Dans ce contexte, certaines nouvelles méthodologies telles que la conception orientée objets au niveau du système et la co-conception matériel-logiciel sont en cours d'élaboration pour faciliter le flot de conception analogique et mixte [2]. Par ailleurs, plusieurs méthodologies produisant des composants de SoC réutilisables comme la réutilisation de propriété intellectuelle (IP) ou les composants virtuels [3]–[5], offrent des moyens intéressants afin d'accélérer la conception en vue de réduire le « time-to-market » pour les industries.

En général, l'approche dite « descendante » (« top-down » en anglais), comprenant la spécification du système, la décomposition fonctionnelle du système, la conception de cellules, le dessin des masques des cellules et l'assemblage des masques pour former le système [6], présente des avantages comme une meilleure chance de succès au premier *run* et permet de séparer les outils de front-end (indépendant de la technologie) et les outils de back-end. Ainsi, il permet de réduire le temps de conception et d'améliorer la conception globale du système. Toutefois, des connaissances profondes sont nécessaires lors de la modélisation des cellules aux différents niveaux d'abstractions. Dans les applications numériques, on décrit facilement les différents niveaux hiérarchiques en utilisant différents niveaux d'abstraction. Cependant, en analogique, toutes les caractéristiques analogiques devraient être prises en compte à tous les niveaux hiérarchiques, alors qu'elles sont encore inconnues. C'est à dire que les modèles de haut niveau devraient comprendre des détails électriques de bas niveau les plus complets possibles (par exemple la non-linéarité du circuit), et les spécifications de performance du système doivent pouvoir être vérifiées à chaque niveau d'abstraction. Mais en pratique, il est impossible de prendre en compte tous les effets de bas niveau dépendant de la technologie d'intégration choisie pendant la modélisation de haut niveau. C'est pourquoi les

Chapitre II

outils de CAO et des méthodologies spécifiquement faites pour la conception analogique et mixte, qui améliorent le flot de conception, sont fortement nécessaires.

Dans ce contexte, dans les premières phases de l'étude du système, nous nous sommes efforcés de mettre en œuvre une méthodologie basée sur le « Prototypage Virtuel Fonctionnel » (PVF) [7], renforçant toutes les étapes de conception par l'utilisation de modèles et de simulations. Dans les sections suivantes, nous commencerons par présenter des limites de la méthode de conception classique en V et puis nous décrirons en détail l'approche PVF et ses avantages.

II.2 Méthode de conception classique en V

II.2.1 Description de la méthode de conception en V

Actuellement, la méthode de conception classique en V est largement utilisée dans la conception des produits industriels (Figure II-1).

Nous pouvons observer clairement deux phases dans cette méthode : une phase descendante (top-down) d'analyse pour la conception et une autre phase ascendante (bottom-up) pour la réalisation et la validation. La Figure II-1 montre la propagation des détails techniques et la consommation moyenne du temps de cette méthode. Plus nous descendons dans les niveaux, et plus nous allons rencontrer de problèmes techniques explicites.

Dans la phase de conception, l'établissement du « cahier des charges » signifie le début de la conception. Selon l'« analyse fonctionnelle », nous pouvons commencer à décrire le système par ses comportements fonctionnels envisagés. Ensuite à partir d'une étude sur les caractéristiques globales, nous ne développons qu'une architecture fonctionnelle dans l'étape d'« exploration architecturale » parce qu'il serait très coûteux d'en développer plusieurs. Au niveau suivant (« analyse de fonctions de base »), les composants élémentaires sont étudiés afin de réaliser les fonctions de base du système. À la fin, tous les composants au plus bas niveau sont définis afin de réaliser le premier prototype physique dans l'étape « intégration de prototype ».

La phase de validation (bottom-up) débute à partir du premier prototype. Cette phase permet les tests et les validations de tous les niveaux d'intégration (« test de composant »,

« test de sous système », « test de système »). D'une manière hiérarchique, le système sera testé à partir des composants élémentaires jusqu'au niveau système. Finalement, la « validation paramétrique » va valider le système complet en accord avec les fonctionnalités du cahier des charges pour les performances attendues.

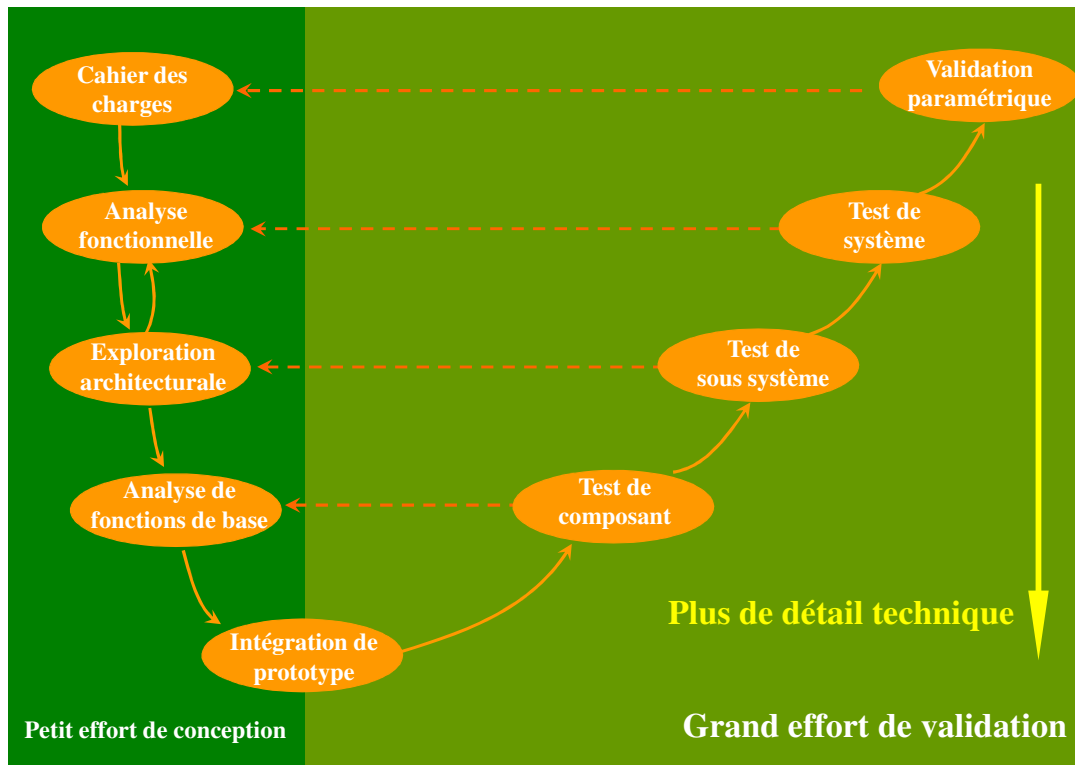


Figure II-1 Cycle de conception classique en V et ses boucles possibles de reprise

II.2.2 Limites de la méthode de conception en V

Dans le flot de conception classique en V, un prototype physique est censé être réalisé dans le milieu du flot de conception. Chaque composant de base est testé individuellement en premier lieu, puis il faut les assembler pour le test des sous-systèmes et le test final du système complet. La production de prototypes physiques et les tests hiérarchiques deviennent le goulot d'étranglement du processus de conception d'un produit parce qu'ils prennent beaucoup de temps et qu'ils sont souvent les principales sources d'erreurs et de dérive de coûts. S'il y a des erreurs dans ces étapes de validation, nous devons retourner à l'étape correspondante dans la phase de conception et recommencer le cycle (reprise de cycle). Comme montré dans la Figure II-1, la phase de validation classique prend typiquement 80% de l'effort de conception qui est quatre fois plus élevé que celui de la phase de définition. Un autre problème est la déconnexion entre la conception et le développement de la technologie.

Chapitre II

Considérant que les composants de base sont en interactions dans le niveau le plus bas, une détection précoce des défauts d'interconnexion est impossible. D'une manière générale, plus tard nous détectons les erreurs techniques lors des essais, plus élevés seront les coûts pour les corriger. L'exemple le plus criant est le rappel d'un produit industriel déjà dans le circuit commercial.

En raison des limitations du flot classique de conception en V mentionnées ci-dessus, beaucoup de problèmes potentiels ne peuvent pas y être efficacement prédits et ce flot de conception n'est plus adapté aux besoins industriels modernes [7]. En conséquence, réussir la conception du système au premier essai est assez difficile. Le problème « time-to-market » a déjà été soulevé. La demande d'une méthodologie renforcée de conception analogique et mixte établissant une communication étroite entre la conception (action du niveau abstrait) et la technologie (du niveau détail) est également un besoin. Pour répondre aux attentes changeantes des clients et du marché, un flot de conception flexible qui assure la robustesse du système et tolère l'interchangeabilité dans le même temps devient indispensable. Sous ces pressions, la conception basée sur la simulation et le prototypage virtuel, qui montrent sa capacité à assurer une meilleure performance du système avec une réduction du temps et du coût par rapport aux approches traditionnelles de l'« implémenter et puis tester » [8], nous fournit quelques solutions prometteuses. Dans la section suivante nous abordons la description explicite de cette méthodologie.

II.3 Prototypage Virtuel Fonctionnel

II.3.1 Le Prototypage Virtuel Fonctionnel et ses avantages

Le Prototypage Virtuel Fonctionnel (PVF), comme renforcement du processus de conception classique en V, a été formalisé par le Dr. Yannick HERVE (voir Figure II-2). Le principe de cette méthodologie est d'améliorer la conception d'un système par des simulations et des comparaisons d'un ensemble de modèles « prouvés » [7] [9]. Elle débute par l'expression des besoins (spécifications) et se termine par un prototype virtuel respectant les spécifications. Dans ce contexte, un modèle prouvé est un modèle qui respecte les

spécifications données par le modèle de plus haut niveau. Cette méthodologie peut être appliquée aux conceptions de systèmes complexes multi-domaines (multi-disciplines). Concernant la conception analogique ou mixte, nous accentuons plutôt sur les fonctionnalités des blocs au lieu des caractéristiques ou des paramètres non fonctionnels. Par exemple, les matériaux utilisés, le poids d'un composant ou sa forme sont des paramètres non fonctionnels parce qu'ils ne constituent pas en général un ensemble de paramètres pertinents et simulables entre les blocs pour le flot de conception analogique ou mixte. Virtuellement, le système complet peut être implémenté avec des blocs fonctionnels descriptibles et prévisibles. De cette façon, la fonctionnalité de l'ensemble du système peut être rapidement simulée et subir un ensemble de tests. Ainsi, le prototype d'un système complexe sous sa forme virtuelle, composé de plusieurs modèles de composants fonctionnels au lieu du prototype physique est adopté pour estimer les performances du produit et sa qualité. De ce fait, la phase de validation peut être notablement réduite et nous pouvons remettre l'accent sur l'étude de conception. Cette méthodologie nous permet de déplacer les efforts de conceptions/validations vers la conception. Selon le concept du PVF, les résultats de simulation du système complet pourraient fournir une première estimation globale de la performance du système et des fonctionnalités du système. L'attente du client est clairement perçue et démontrée. La faisabilité de l'ensemble du système peut également être menée pour réduire le risque de perte de temps pour un système irréalisable (au sens fonctionnel ou au sens de l'adéquation aux spécifications). Les modèles fonctionnels alternatifs nous permettent de facilement modifier les modalités de la réalisation. Comme il n'existe pas de prototype physique, le choix de la technologie n'a pas besoin d'être fait avant les dernières étapes de définitions, ce qui donne un avantage au profit de l'intégration éventuelle de la dernière évolution de la technologie. En outre, une fois que ces modèles fonctionnels sont validés, nous sommes en mesure d'établir un ensemble de bibliothèques afin de permettre la conception basée sur la réutilisation des conceptions. Cette automatisation peut accélérer encore le processus de la conception suivante. Dans le schéma du développement d'un produit, cette méthodologie profite de l'utilisation d'un outil CAO unique et d'un langage unifié afin de minimiser les problèmes de communication entre des ingénieurs de différents domaines dans un même projet. Les choix du langage de simulation et du simulateur seront examinés en Annexe I & II.

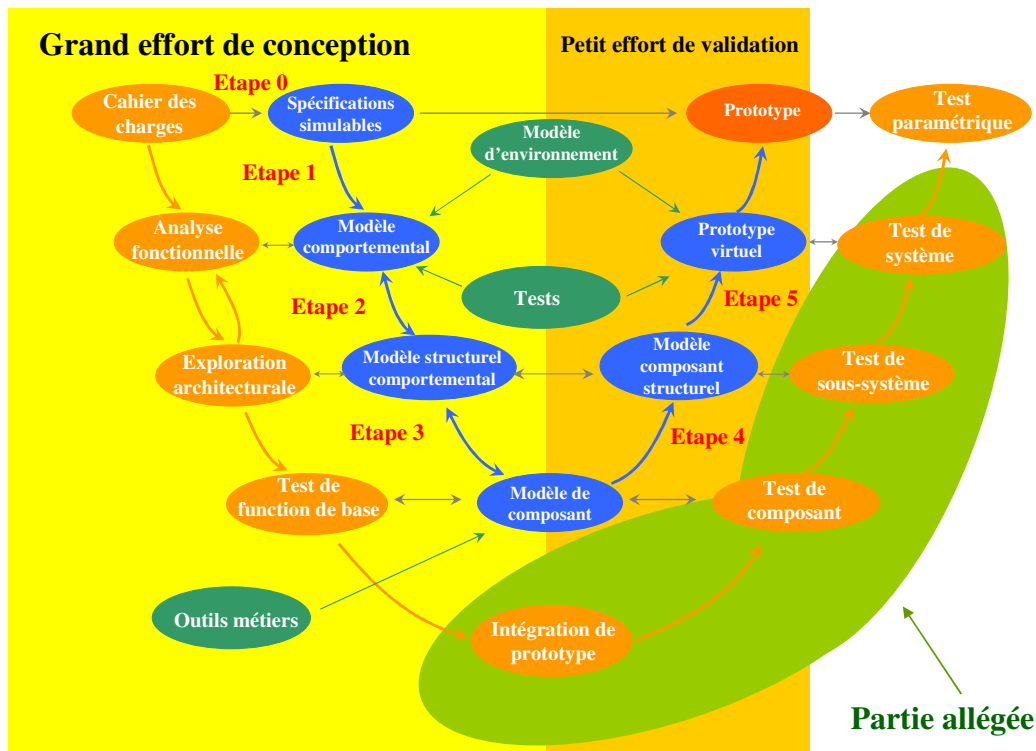


Figure II-2 Méthodologie renforcée de la conception mixte avec le « Prototypage Virtuel Fonctionnel »

Nous résumons ici les avantages apportés par le flot de conception du Prototypage Virtuel Fonctionnel :

- Réduction de la durée et du coût des projets,
- Réduction des risques liés au développement du produit,
- Réduction du coût de prototypage,
- Réduction des ressources humaines,
- Capacité de porter l'effort plus sur l'innovation que sur la vérification,
- Capacité de gérer les ruptures technologiques,
- Vérification simultanée du matériel et du logiciel,
- Capacité d'écarter le prototypage matériel du chemin critique,
- Formalisation des savoirs,
- Facilitation des échanges en interne,
- Travail facilité avec les partenaires, fournisseurs et sous-traitants,
- Augmentation de la qualité et de la fiabilité des produits,
- Passage du savoir formalisé des sphères privées à l'entreprise,

- Capitalisation et réutilisation des connaissances formalisées,
- Optimisation au niveau système.

II.3.2 Flot de conception du Prototypage Virtuel Fonctionnel

La tâche principale du Prototypage Virtuel Fonctionnel (PVF) est de compléter le flot de conception classique en V dans un environnement de CAO. Grâce à son environnement de travail (outil CAO) et le faible coût des solutions d'étude, le PVF pourrait réduire largement le temps et le coût du processus de conception [10]. A partir de l'établissement du cahier de charges jusqu'à la validation du prototype virtuel prêt à fabriquer, cinq étapes principales de conception sont nécessaires pour générer un flot de conception du PVF (voir Figure II-2):

Etape 0 – Etablissement des spécifications simulables: à partir de l'expression informelle des besoins du client, nous pouvons établir le cahier de charges sous la forme d'un document rédigé. Les principales fonctionnalités du système visé, du point de vue de l'utilisateur (spécifications externes), doivent être définies pour guider les tâches suivantes. Des ingénieurs ou des chercheurs traduisent ensuite ces spécifications externes en description de spécifications formelles éventuellement simulables (spécifications internes) pour la conception du système visé.

Etape 1 - Modèle comportemental de haut niveau : selon les spécifications simulables (spécifications internes) obtenues dans l'étape précédente, un modèle comportemental de haut niveau ayant des comportements idéaux sera utilisé pour décrire la fonction globale du système. En même temps, nous écrivons aussi un modèle de haut niveau de l'environnement pour simuler le comportement du système et des modèles de test permettant de tester les performances du système complet (système visé dans son environnement). Avec ces spécifications internes, la performance de l'ensemble de ces modèles comportementaux doit satisfaire les spécifications externes requises. A ce niveau, il n'y a notion ni de structure, ni de technologie.

Etape 2 - Modèle structurel comportemental : elle consiste à partitionner le système complet en sous-systèmes différents selon leur fonctionnalité et à définir des liens internes. Dans chaque sous-système et sous-bloc, la description des modèles comportementaux

Chapitre II

employés est en général de haut niveau. En outre, les modèles comportementaux peuvent tester les limites des paramètres connus de la future structure à réaliser grâce à des simulations globales du système.

Etape 3 - Modèle de composant de bas niveau : une fois obtenues les limites des paramètres de chaque structure, une étude de sélection topologique sera entreprise. Selon les critères des spécifications, les structures qui répondent à la spécification du premier ordre seront choisies. Ensuite, des spécifications d'ordres supérieurs seront les critères suivants pour affiner la sélection de la topologie de chaque structure. En revanche, si l'effet de premier ordre ne peut pas satisfaire des spécifications, alors il sera inutile d'utiliser des effets de deuxième ordre et il faudra alors créer une nouvelle topologie afin de satisfaire les besoins fonctionnels et les spécifications visées. Nous pouvons aussi simplifier les modèles composants en utilisant directement des résultats calculés par des outils métiers. Les appels aux outils métiers ad-hoc nous permettent d'économiser largement le temps de simulation. De cette façon, les modèles composants du niveau transistor peuvent être implémentés pour tous les blocs de base du système.

Etape 4 - Modèle composant au niveau structurel : à partir du modèle composant de niveau transistor, avec une méthodologie de vérification de bas en haut, les modèles structurels composants et des modèles complets seront progressivement assemblés et simulés avec les mêmes tests que dans la partie descendante. Ce niveau de modèle nous donne non seulement la vue globale des comportements du système, mais aussi la fonction de chaque composant, ainsi que les interactions entre eux.

Etape 5 – Prototypage virtuel : Enfin, un « prototype virtuel » basé sur le prototype réel sera formalisé avec tous les modèles composants au niveau structurel établis dans l'étape précédente (voir la Figure II-2). Pour optimiser les temps de calculs et en fonction de l'objectif recherché, les modèles des différents niveaux d'abstraction sont « mélangeables » dans le système complet (système visé + son environnement). Comme les validations et tests de fonctionnement avec les modèles de différents niveaux d'abstraction développés dans le « prototype virtuel » sont effectuées dans chaque étape du PVF pour respecter les performances requises du système visé, nous pouvons bien alléger les étapes de validation dans le cycle de conception classique en V (voir Figure II-2).

Après la première étape, si le résultat de la simulation avec les modèles comportementaux de haut niveau ne peut pas atteindre les performances requises, nous

pouvons conclure à la non-faisabilité. Les modèles de différents niveaux d'abstraction offrent également des possibilités de réutilisation dans d'autres projets pour améliorer la productivité. Basé sur ce flot de conception du PVF, les principales phases de notre approche de conception sont décrites en détails dans la section suivante pour présenter un flot de conception complet.

II.4 Notre approche de conception dans ce projet

II.4.1 Synoptique de notre approche de conception dans le cadre de ADAPTER

Compte tenu de l'historique des développements d'ELA Medical, le couple cœur – pacemaker a été initialement abordé par des analyses théoriques utilisant le formalisme Matlab[®]. L'introduction du langage VHDL-AMS en cours de projet a nécessité de convertir l'ensemble des existants (programmes C ou Matlab[®]) en matériaux VHDL-AMS. Or toute transposition de modèle doit subir une phase de validation. Lors de cette étape nous nous sommes aidés, en ce qui concerne la partie analogique, d'une procédure de validation introduite par François Schwartz dans sa thèse [12]. Les transpositions s'étant faites progressivement, nous avons dû jongler avec les deux formalismes comme cela apparaîtra dans ce document.

Bien que notre travail concernait les circuits analogiques, nous ne pouvions faire abstraction du système dans sa globalité et ceci pour plusieurs raisons. La première tient aux caractéristiques propres des réseaux de neurones. Les réseaux de neurones sont des structures adaptatives. Ceci signifie que si l'on dispose d'une réalisation (software ou hardware), celle-ci peut s'adapter à diverses applications sans changement topologique ou dimensionnel. Bien sûr il y a des limites à cette souplesse entraînant de nouvelles questions : comment définir les spécifications d'un réseau ? Veut-on élargir la souplesse pour plus d'universalité ? Quelle précision peut-on alors attendre d'un réseau trop souple ? Peut-on connaître la plage de variation des poids synaptiques conférant au réseau sa souplesse d'adaptation ?

Ces questions ne peuvent trouver de réponses qu'au niveau système puisqu'elles dépendent de l'application. Par conséquent, s'il est difficile de fixer des spécifications

Chapitre II

dynamiques, le travail du concepteur de circuit analogique agissant aux niveaux hiérarchiques inférieurs sera flou donc complexe voir impossible.

Dans le contexte de « time to market », nous devons donc aborder cette étude avec parcimonie. Nous avons donc utilisé deux approches méthodologiques différentes mais complémentaires. Pour que deux méthodes distinctes contribuent à augmenter l'efficacité globale, il faut nécessairement qu'elles communiquent. L'une et l'autre doivent offrir des réponses qui conduisent à des prises de décisions utiles pour la convergence de l'ensemble.

Au niveau système nous avons donc adossé notre approche à la méthode du prototypage virtuel fonctionnel (PVF). Au niveau circuit, nous utiliserons une approche ensembliste qui offre une aide à la sélection des topologies électriques selon un critère de robustesse de fabrication. La première méthode fut initiée par Dr. Yannick HERVE [15] et la seconde par Dr. Jacques MICHEL [16] [17]. Dans cette thèse j'ai également proposé une méthode qui guide les concepteurs à fixer au mieux les spécifications de plus bas niveau et qui contribue à la présélection de topologies potentiellement aptes à répondre aux exigences visées. La méthode ensembliste permettra de les classer au sens de la robustesse. La sélection topologique finale résultera ensuite d'une conjonction des critères.

II.4.2 Description explicite de notre approche de conception

Dans cette section, en nous basant sur le prototypage virtuel fonctionnel, nous décrirons de manière plus explicite notre méthodologie renforcée de la conception analogique et mixte en nous appuyant sur le projet Adapter. Partant du système complet, nous recentrerons notre effort sur un seul « composant » qui est le multiplieur synaptique. Néanmoins la méthodologie peut s'appliquer aux autres « composants ». Les six étapes que nous avons suivies sont donc :

Etape 0

Généralement, un projet innovant vient d'une nouvelle idée ou d'un nouveau besoin, ce qui signifie que les connaissances dans ce domaine sont faibles voire inexistantes. Cela nous amène à une synthèse de conception de type descendant (approche « Top-Down »). Par conséquent, plusieurs études de faisabilité telles que la faisabilité économique, la faisabilité

du planning (time-to-market) et la faisabilité technique doivent être menées avant tout démarrage industriel du projet. Dans le domaine scientifique, nous nous concentrons plutôt sur la faisabilité technique. La première tâche d'une étude de faisabilité technique est d'exprimer les besoins réels sous la forme des fonctionnalités. Il faut traduire des spécifications informelles du point de vue du client ou de l'utilisateur (spécifications externes) en description de spécifications formelles éventuellement simulables (spécifications internes) du point de vue concepteur. « Quel est l'objectif de ce système » et « quelles sont les fonctionnalités dont nous avons besoin » sont des questions à débattre avant tout. Il est possible que la première version des spécifications requises du projet soit très approximative, plusieurs itérations doivent être réalisées.

Etape 1

Comme un projet industriel moderne associe souvent plusieurs disciplines physiques, la première tâche est en réalité plus complexe qu'il n'y paraît. La difficulté principale réside dans la transcription des besoins requis par un expert du domaine A sous forme de quelques fonctionnalités compréhensibles. Ensuite ces fonctionnalités doivent être comprises par un expert du domaine B. La question qui reste ouverte est de savoir comment assurer l'exactitude et la cohérence de la transcription par rapport aux besoins.

Le « retour arrière » est en effet indispensable pour confirmer le démarrage correct du développement. Cette validation, dans notre approche, peut être quantifiée en comparant les résultats de simulation des premiers modèles établis selon les spécifications internes du projet. Mais ces simulations doivent être accessibles aux deux spécialistes pour que les analyses soient concluantes. Pour cela, nous employons un langage commun et fédérateur pour décrire l'ensemble du système avec un haut niveau d'abstraction. Nous avons choisi le langage VHDL-AMS en raison de ses multiples capacités. Dans une approche « Top-Down », commençant par les modèles du système requis et de son environnement, les premiers modèles comportementaux de haut niveau d'abstraction avec les spécifications internes préliminaires peuvent être établis pour lancer les premières simulations globales. Grâce à des discussions entre les partenaires du projet, les exigences du système peuvent être exprimées plus clairement, de manière plus explicite.

Etape 2

Selon les exigences fonctionnelles des divers sous ensembles déduites des premières discussions, et en tenant compte de l'état de l'art de chaque domaine de compétence, plusieurs

Chapitre II

approches architecturales seront proposées au consortium. Avec l'expression des spécifications essentielles, les partenaires peuvent démarrer une analyse plus critique des stratégies architecturales. Des sous-blocs fonctionnels seront définis (ou virtuellement conçus) pour garantir les performances du système (spécifications de performance requises). Mais le partage des tâches de conception soulève le problème des signaux d'échanges entre les sous blocs. Ceux-ci doivent être définis de manière exhaustive : leur nature, leur plage de variation, leurs caractéristiques. Si deux sous blocs appartiennent à deux domaines différents (par exemple, électronique et biologique comme dans ce projet), une discussion entre les experts des différents domaines devra confirmer les propriétés de ces signaux. Ces discussions ne pourront être fructueuses et efficaces que si l'on partage entre ces domaines une même plateforme d'analyse et de simulation. Les deux experts pourront, ensemble, analyser les simulations du système de haut niveau et valider conjointement les propriétés des interfaces au vu des simulations globales. Cette méthode se concentre sur le développement de modèles comportementaux de haut niveau décrivant l'architecture du système à réaliser (modèle abstrait au sens des détails technologiques liés à la réalisation finale) sans entrer dans les détails techniques. Cette déconnexion de la technologie permet d'adapter la réalisation finale à ses évolutions. Ces modèles de haut niveau d'abstraction favoriseront l'exploration architecturale du dispositif étudié. Si ensuite, tous les tests de haut niveau sont validés, le consortium pourra entériner tant la partition du système en sous blocs que les spécifications relatives à chaque sous bloc. Les partenaires peuvent alors dans leur domaine de compétences respectif progresser dans leurs développements selon leurs méthodologies propres en adéquation avec les objectifs fonctionnels du système.

Etape 3

A l'issue de l'étape 2, un concepteur électronique dispose d'un ensemble de spécifications lui permettant de progresser dans l'étude de conception de signal mixte. Selon l'approche descendante, la description de haut niveau des blocs fonctionnels manque de précision pour attaquer une sélection architecturale trop dépendante de la technologie de fabrication. Il lui faudra affiner les détails électriques en introduisant de nouveaux modèles de plus bas niveau (moins abstrait). La méthode PVF suggère de hiérarchiser ces modèles comportementaux en partant des modèles de haut niveau créés dans la phase précédente qui décrivent les sous-blocs avec des conditions techniques idéales (donc abstraites) c'est à dire avec des équations mathématiques simples. Les modèles suivants, toujours encore comportementaux, intègrent les effets de second ordre, puis ceux de troisième ordre, etc. Ceci

suggère d'introduire successivement les imperfections des réalisations futures (plage dynamique, non-linéarité, intermodulation, bruit, dissipation thermique locale, etc.) en fonction de technologies de fabrication possibles. Grâce à l'utilisation du langage VHDL-AMS (voir Annexe I), ces nouveaux modèles peuvent être créés comme des « architectures » supplémentaires dans les modèles de sous-bloc. Cela nous permet de noter l'évolution des modèles de différents niveaux hiérarchiques du système CRT-P (Figure II-3). En raison de la simplicité des expressions sous la forme d'équation mathématique, les modifications des paramètres peuvent être rapidement effectuées pour re-simuler le système complet avec les nouveaux modèles hiérarchiques. De cette façon, les effets des détails techniques de plus en plus réalistes sont pris en compte. Les limites de ces effets réels seront testées par les simulations globales avec ces sous-blocs. Ces simulations peuvent établir les spécifications techniques des sous-blocs, qui sont utiles à l'implémentation des circuits au plus bas niveau. Un concepteur de systèmes optimise le système visé par les re-simulations du système complet avec les modèles améliorés et ajuste ensuite les spécifications techniques des sous-blocs en liaison avec le concepteur circuit (pour les parties électroniques). Cette phase constitue un pont entre les conceptions descendantes et ascendantes, c'est aussi le canal de communication indispensable pour la convergence des efforts de conceptions dans une collaboration multi-disciplinaire. Sans cette communication, les erreurs apparaissent plus tard et l'objectif crucial de « time to market » ne pourra être satisfait. Via le flot de PVF et une plateforme sous VHDL-AMS, les spécifications des performances initiales évoluent et les spécifications techniques des sous blocs émergent. De plus, l'introduction des modèles hiérarchiques conduit à l'établissement des modèles structurels comportementaux de niveau de cellule (multiplicateur, OTA, miroir de courant, etc.) avec leurs ensemble de spécifications fonctionnelles et techniques. Le concepteur peut alors prendre en compte la technologie et regarder la sélection de topologies électriques avec plus de parcimonie.

Dans la même étape, l'amélioration de la prédiction permet de régler la spécification de performance visée pour chaque bloc et éviter tout excès de spécifications qui conduit souvent les concepteurs de bas niveau à des compromis excessifs. Relâcher des contraintes donne plus de liberté au bas niveau ce qui inévitablement accroît la tolérance d'un circuit aux dispersions de fabrication et par suite améliore le rendement de production. Un tel système devient donc nécessairement plus fiable au regard des dérives des caractéristiques durant sa durée de vie. Enfin, même dans les premières phases d'études, à un très haut niveau d'abstraction, cela renforcera la crédibilité sur la faisabilité du projet.

Chapitre II

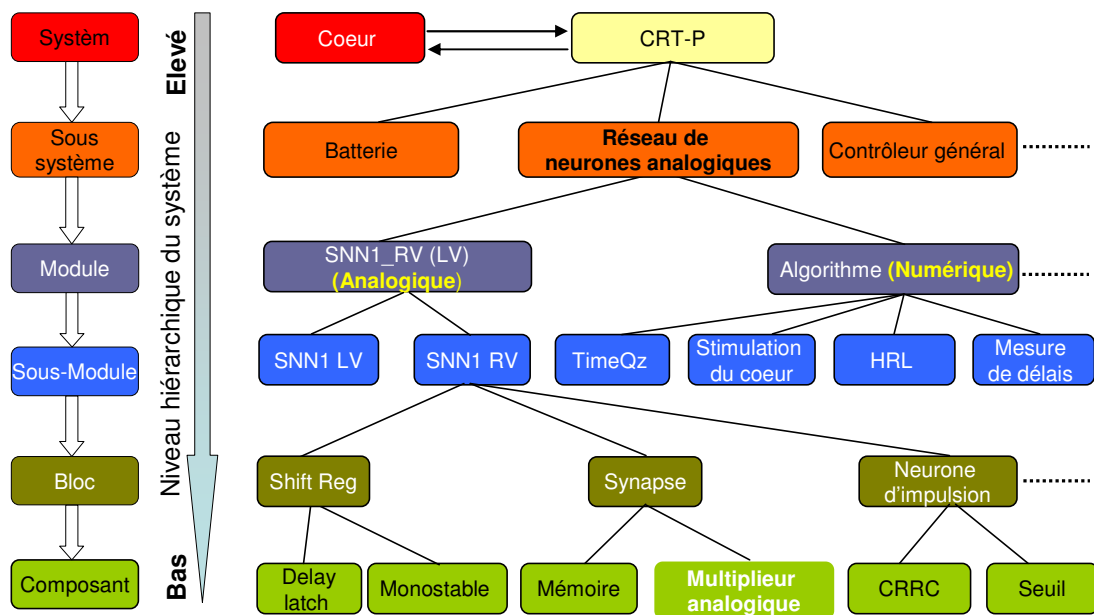


Figure II-3 Evolution des modèles de différents niveaux hiérarchiques du système CRT-P

Etape 4

Cette étape aborde le travail classique de la conception, c'est à dire opérer le basculement de l'expression des besoins vers la réalisation physique du dispositif. En électronique, la conception d'un circuit commence par la sélection d'une topologie de circuit électrique. Ensuite, cette topologie est dimensionnée et ses caractéristiques sont analysées par des simulations électriques (typiquement des simulations utilisant le programme SPICE ou équivalent). La validation de la sélection et du dimensionnement des topologies se fait en comparant les caractéristiques simulées (ou extraites des simulations) et les spécifications (fonctionnelles et techniques) de la cellule. Enfin c'est le tracé des masques de productions (les layouts). Ce dernier n'est pas des moindres car il conditionne l'introduction des éléments parasites comme les courants de fuites, les capacités, les couplages électromagnétiques et thermiques, etc. L'étape 4 doit donc permettre de guider le concepteur vers la validation des composants en orientant ces choix de conception et en préparant les tests de validations. Or la préparation de ces derniers permet aussi de s'interroger sur la pertinence de certaines spécifications.

Etape 5

Après avoir réalisé les conceptions de chaque composant et les modules dans le système visé, les tests de validations peuvent être effectués afin de valider les performances du système complet. En utilisant la plateforme sous VHDL-AMS, le concepteur de système peut

faire les simulations du système complet basé sur les modèles de niveaux mixtes (modèles de haut niveau d'abstraction et modèles de niveau transistor) pour économiser le temps de simulations. De ce fait, les performances du système complet peuvent être rapidement évaluées et validées. Un prototypage virtuel (voire un prototypage) satisfaisant les spécifications du système complet peut donc être réalisé.

II.4.3 Le travail présenté dans ce manuscrit

Même si nous avons travaillé sur les six étapes précédentes, le travail exposé dans cette thèse, est volontairement limité aux étapes 0 à 3 de notre méthode.

Les étapes 0 à 2 nous ont permis, au sein d'une collaboration, de développer une structure fonctionnelle. Ce développement inclut le partitionnement en « bloc » et « composants » mais aussi le dimensionnement fonctionnel de ces entités. Ce partitionnement et ce dimensionnement font l'objet des chapitres III à V.

Pour établir la base d'échange système vers domaine, l'étape 3 doit fournir un ensemble de spécifications. Cet ensemble comprend des spécifications du niveau hiérarchique système ainsi que des spécifications du niveau hiérarchique « composants ». Ces dernières constituent une part importante du cahier des charges des concepteurs domaines. Le chapitre VI y est consacré.

L'articulation entre l'étape 3 et l'étape 4 comporte une rupture entre les domaines de compétences et les niveaux de description et d'analyses. C'est le passage du domaine système vers le domaine métier. Ce manuscrit n'aborde pas à proprement parler l'étape 4. Néanmoins, nous voulions montrer comment préparer le travail de conception de l'étape 4 qui est le travail de conception typique en électronique analogique ou mixte. Partant du constat que cette étape débute avec la sélection des topologies des « composants », il apparaît évident que les étapes suivantes (les dimensionnements de topologies et les layouts) découlent directement de celle-ci. Le bon choix topologique permettra de limiter le temps de conception conformément au critère de « time to market ». Choisir une topologie suppose que nous disposions d'un ensemble de topologies potentiellement adaptées à nos spécifications du sous-bloc. Plusieurs approches sont possibles pour créer cet ensemble. On peut disposer d'une bibliothèque de circuits ayant déjà été réalisés [11]. C'est le principe du « Re-Use » qui vise à capitaliser autant l'expertise de conception que le temps de conception [1]. Pour les développements de

Chapitre II

sujets originaux, une approche similaire consiste à fouiller dans les bases de données publiques, c'est à dire dans les publications scientifiques et techniques. Une autre approche en pleine évolution dans les laboratoires de recherches est la génération « spontanée » de topologies [13] [14]. N'ayant pas accès aux générateurs précédents, notre approche fût la fouille de données publiques. Après la constitution d'un ensemble de topologies possibles, nous introduisons un critère de robustesse pour aider à la sélection. Ce critère de robustesse est issu d'une exploration de l'espace de conception et retourne une estimation de la dimension de l'espace solution. Plus cette dimension est grande, plus la réalisation sera robuste pour un point nominal au centre de cet espace solution [12]. L'ensemble de ces approches nous permet de proposer une sélection de topologie plus avisée. Cette démarche sera détaillée dans le chapitre VII sur un « composant » qui est le multiplieur analogique générant la pondération synaptique dans un neurone.

Conclusion

Nous avons introduit dans ce chapitre la méthodologie de conception analogique et mixte que nous avons suivie tout au long de cette thèse. La méthode s'appuie sur le Prototypage Virtuel Fonctionnel (PVF). Grâce à sa structuration en plusieurs étapes, elle permet de construire un lien avec les méthodes plus spécifiques à la conception de circuits. Ce lien et les améliorations possibles au plus bas niveau de conception ont été introduits ici.

En premier lieu, nous avons abordé une présentation des attentes urgentes des entreprises sur une méthodologie renforcée de conception analogique et mixte pour non seulement augmenter la qualité et la fiabilité des produits, mais aussi avoir une réduction de la durée et du coût des projets. De ce fait, la méthodologie de conception PVF a apporté des modifications et des renforcements sur la méthodologie de conception classique (dans les projets industriels) en V. L'objectif de cette méthodologie est d'appuyer toutes les phases de conception et de vérification sur l'utilisation d'une plateforme de simulation et une hiérarchie de modèles de différents niveaux d'abstraction. Ces derniers permettent de propager pas à pas des spécifications simulables en pouvant valider chaque phase par des simulations globales du système. Les avantages de cette méthodologie et les principales phases de conception ont été décrits dans ce chapitre. Nous avons également présenté nos attentes plus spécifiques dans le cadre de la conception des « composants » dans le domaine de l'électronique analogique et mixte. Nous y avons souligné l'importance de la sélection topologique qui suggère de disposer d'un maximum de spécifications extraites du système ainsi qu'une estimation de la robustesse technologique d'une topologie. Durant cette étape, le concepteur établit le lien entre l'abstrait et la technique. Selon notre philosophie méthodologique, ce passage doit être le mieux assisté.

Le besoin d'échanges entre spécialistes de différents domaines et entre les concepteurs système et cellules sont l'essence même de la synergie souhaitée dans le développement de tels projets. L'utilisation d'une plateforme de simulation commune est donc incontournable. La plateforme doit donc utiliser le même outil de simulation pour les divers domaines. Par suite le langage de description des ensembles et sous ensembles doit être compatible avec l'outil. Cela justifie notre choix du langage de modélisation (VHDL-AMS). Nous proposons en annexe (Annexe I) une comparaison des cinq langages de modélisation mixte les plus utilisés sur le marché : VHDL-AMS, MAST, Modelica, Matlab/Simulink, Verilog-AMS. Le

Chapitre II

langage VHDL-AMS offre l'avantage d'être normalisé. Il nous permet de simuler les systèmes multi-physiques et multi-niveau d'abstraction en temps continu et discret. Ses multi-capacités correspondent bien aux besoins de notre méthodologie et du projet ADAPTER.

Références

- [1] R. Castro-Lopez & all, Reuse-based methodologies and tools in the design of analog and mixed-signal integrated circuits, springer publishers, 2006
- [2] G.G.E Gielen, "CAD tools for embedded analogue circuits in mixed-signal intergrated systems on chip", IEE Proceedings online of Computer Digital Techniques; vol.152, n°3, May 2005, pp.317-332
- [3] M. Keating and P. Bricaud, Reuse methodology manual, 2nd ed. Boston: Kluwer Academic Publishers, 1999
- [4] Virtual components design reuse, R. Seepold and N. Martinez Eds. Boston: Kluwer Academic Publishers, 2001
- [5] Analog/Mixed-signal VSI extension specification 1 version 2.0 (AMS 1 2.0), Analog/Mixed-Signal Development Working Group (VSI AllianceTM), 1999
- [6] K. A. Ellenbogen & all, "Device Therapy for Congestive Heart Failure", Philadelphia, Pennsylvania, Elsevier Inc., 2004, pp 47-91.
- [7] HERVE Y. "Functional virtual prototyping design flow and VHDL-AMS", in *Proc. Of Forum on specification & Design Languages (FDL'06)*, Darmstadt (Germany), September 19-22, 2006, Proc. pp. 69-76
- [8] Ryan, R., Digital testing in the context of digital engineering – Functional Virtual Prototyping, Mechanical Dynamics, Inc., Michigan, USA, 1999
- [9] A. Legendre, Y. Hervé, "Functional Virtural Prototyping Applied to Medical Device Development", Systems VIP, Strasbourg, France, 2008
- [10] M. Popovic, Z. Jugovic, R. Slavkovic, "The Concept of Fonctional Virtural Prototyping in the Design of Excavator Cutting Teeth" Tribology in industry, Volume 31, No. 3&4, 2009
- [11] T. LEVI, "Méthodologie de développement d'une bibliothèque d'IP-AMS en vue de la conception automatisée de systèmes sur puces analogiques et mixtes" Ph.D. dissertation, Université Bordeaux I, 2007
- [12] F. SCHWARTZ, "Méthodologie de conception d'un système analogique massivement parallèle. Application à la conception de réseaux de neurones travaillant en mode courant." Ph.D. dissertation, Université de Strasbourg, 2010
- [13] McConaghy, T.; Palmers, P.; Steyaert, M.; Gielen, G.G.E.; , "Variation-Aware Structural Synthesis of Analog Circuits via Hierarchical Building Blocks and Structural Homotopy," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.28, no.9, pp.1281-1294, Sept. 2009
- [14] Palmers, P.; McConnaghy, T.; Steyaert, M.; Gielen, G.; , "Massively multi-topology sizing of analog integrated circuits," *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE '09.* , vol., no., pp.706-711, 20-24 April 2009
- [15] Y. HERVE., "Functional virtual prototyping design flow and VHDL-AMS", *Forum on specification & Design Languages (FDL'06)*, Darmstadt (Germany), September 19-22, 2006, Proc. pp. 69-76

Chapitre II

- [16] F. SCHWARTZ, Q. SUN, J. MICHEL, “An OTA sizing method using interval analysis”, *XXIIIth Conference on Design of Circuits and Integrated Systems (DCIS'08)*, Grenoble (France), November 12-14, 2008, Proc. on USB key
- [17] F. SCHWARTZ, Q. SUN, J. MICHEL, Y. HERVE, “A robustness-oriented design tool for the topology selection in analog synthesis,” in *Proc. International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD)*, Tunis-Gammarth, Tunisia, Oct. 5-6, 2010,

Chapitre III

Description globale du CRT-P

adaptatif du projet ADAPTER

Sommaire du chapitre III

CHAPITRE III	DESCRIPTION GLOBALE DU CRT-P ADAPTATIF DU PROJET ADAPTER	60
III.1	LE CAHIER DES CHARGES INITIAL	62
III.2	PARTITION GLOBALE DU CRT-P ADAPTATIF DU PROJET ADAPTER	63
III.3	DESCRIPTION FONCTIONNELLE DES DEUX BLOCS CENTRAUX DU CRT-P.....	65
<i>III.3.1</i>	<i>Fonctionnalités du contrôleur général.....</i>	<i>66</i>
<i>III.3.2</i>	<i>Fonctionnalités de processeur neuronal.....</i>	<i>66</i>
III.4	LES MODES DE FONCTIONNEMENT DU PACEMAKER ADAPTATIF	68
	<i>Conclusion.....</i>	<i>71</i>
	<i>Références</i>	<i>72</i>

III.1 Le cahier des charges initial

Le point fondamental du cahier des charges est l'objectif fonctionnel visé par le pacemaker. Nous rappelons que l'objectif du dispositif (CRT-P : Cardiac Resynchronisation Therapy - Pacemaker) est la thérapie de resynchronisation cardiaque. Celle-ci agit sur les instants de contraction des deux ventricules en référence à celle de l'oreillette droite [1] [2]. Le dispositif prévu dans le projet ADAPTER est d'asservir ces contractions ventriculaires, en contrôlant les délais artio-ventriculaire (AV) et inter-ventriculaire (VV), afin de maximiser la pression systolique (en anglais : Blood Pressure) via le volume d'éjection systolique (en anglais : Stroke Volume = SV). Tous les délais AV et VV utilisés dans le dispositif sont calculés en fonction des instants RA, LV et RV (voir I.1.2) avec les équations suivantes :

$$AV = RV - RA \quad (III.1.1),$$

$$VV = LV - RV \quad (III.1.2),$$

$$SV = C_0 * BP \quad \text{avec } C_0: \text{ une constante} \quad (III.1.3).$$

Le volume d'éjection systolique SV est une grandeur extraite de mesures captées par les capteurs hémodynamiques. Pour chaque période cardiaque, on « mesure » la valeur obtenue pour SV en fonction des délais AV et VV appliqués (les délais naturels ou ceux préconisés par le CRT-P) au cœur. On peut donc considérer que la valeur de SV est issue d'une relation avec le couple (AV, VV) appliqué : $SV = F_{NL}(AV, VV)$. Selon les observations cliniques faites par ELA Medical, les plages dynamiques maximales des délais AV et VV sont respectivement [40, 240] ms et [-100, 100] ms.

L'objectif secondaire du développement de cette solution technique du projet ADAPTER, est d'améliorer le confort de vie du patient. Pour ce faire, il est souhaité de minimiser le nombre d'interventions chirurgicales. Cet objectif impose donc de prévoir une autonomie du pacemaker implanté d'environ 5 à 7 ans de fonctionnement sans intervention technique. Toujours selon le même objectif de confort, le dispositif doit avoir un volume aussi réduit que possible, si possible plus petit que les versions antérieures. Les défis techniques sont donc un fort ratio énergie stockée / énergie consommée et une surface de silicium minimale. Finalement, l'objectif de confort introduit le besoin d'adaptabilités. On souhaite que le dispositif s'adapte aux situations de vie du patient, mais aussi à l'évolution de ses conditions physiologiques et par suite à chaque patient. Parmi les options techniques possibles

Chapitre III

pour créer un dispositif auto adaptatif, une solution reposant sur un processeur neuronal a été choisie. La question qui se pose porte sur l'apprentissage qui configure le processeur pour solutionner la fonctionnalité d'ajustement des délais. L'adaptation à chaque patient et aux évolutions physiologiques à long terme nécessite d'étudier la plasticité de ces algorithmes et la sécurité globale du système.

III.2 Partition globale du CRT-P adaptatif du projet

ADAPTER

Le CRT-P prend place entre le cœur et les instruments de surveillance médicale. Lors de l'implantation, ou lors d'un contrôle, les médecins peuvent analyser le comportement du cœur au travers du CRT. En fonctionnement normal, le CRT-P ne communique qu'avec le cœur. La Figure III-1 présente le cas complet mis en œuvre lors d'une analyse médicale [3]. Le cœur est représenté (décrit) au travers de modèles électromécaniques qui communiquent avec le CRT-P adaptatif. Plusieurs modèles peuvent être implémentés pour décrire des comportements cardiaques différents et aussi pour tester les fonctionnalités du pacemaker adaptatif dans des conditions cardiaques différentes. Le CRT-P adaptatif échange avec le cœur les signaux de trois électrodes (atrium droit : RA, ventricule droit : RV et ventricule gauche : LV), et les signaux des deux capteurs hémodynamiques. Les trois signaux RA, RV et LV sont du type « entrée-sortie » alors que les signaux de mesures hémodynamiques sont seulement de type « entrée » pour le CRT-P. Le CRT-P adaptatif peut envoyer, si nécessaire, une série d'impulsions aux électrodes RV & LV pour stimuler la dépolarisation des ventricules via les deux électrodes LV et RV.

Le dispositif CRT-P se compose d'un ensemble de modules fonctionnels internes qui prennent chacun en compte une mission spécifique. Ainsi le CRT-P adaptatif laisse apparaître 6 modules principaux décrits dans la (Figure III-1). On peut y voir la liste des signaux échangés entre les entités (cœur, CRT-P, instruments médicaux) ainsi que leur sens de propagation.

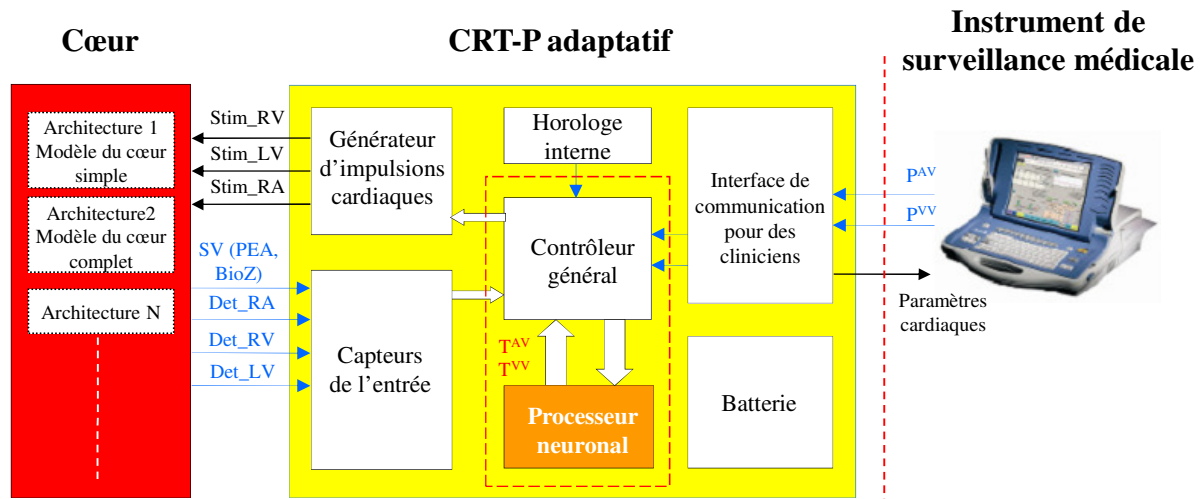


Figure III-1 Partition globale du CRT-P adaptatif du projet ADAPTER

Générateur d'impulsions cardiaques :

Ce bloc comprend le dispositif pour générer une série d'impulsions électriques vers les électrodes (Stim_RV, Stim_LV et Stim_RA) et provoquer la dépolarisation des divers muscles. Par ce biais, on peut forcer une contraction absente et par suite ajuster soit le rythme cardiaque soit le volume d'éjection.

Capteurs d'entrée :

Cette interface avec les capteurs récupère les signaux analogiques permettant au CRT-P de reconstruire sa vision du cœur et établir sa réponse de stimulation. Ces entrées sont les signaux de contractions cardiaques, détectés par les capteurs (Det_RA, Det_RV et Det_LV) et les signaux des capteurs hémodynamiques. Le premier sert de référence temporelle à chaque période cardiaque. Les suivants servent au processeur pour construire sa réponse optimale. La « mesure » du volume de sang éjecté s'appuie sur le signal de l'un des capteurs hémodynamiques. Deux capteurs sont envisagés au début du projet : soit un capteur « Bio Impedance Sensor » (*BioZ*) de ELA, soit un capteur « Peak Endocardial Accelerometer Sensor » (*PEA*).

Contrôleur général (maître) :

Le centre névralgique du CRT-P est le contrôleur général qui fonctionne comme un contrôleur maître. En échangeant des données avec le processeur neuronal, ce bloc surveille l'état physique du patient, il décide quel mode de fonctionnement sera utilisé de manière à assurer les meilleures performances ainsi que la sécurité globale et enfin décide si les

Chapitre III

prévisions du processeur neuronal sont cohérentes. La section III.4 présentera les modes de fonctionnement en détail.

Processeur neuronal (esclave) :

Le processeur neuronal travaille en arrière plan et agit en esclave du contrôleur général. Il reçoit l'ensemble des signaux d'entrées et reconstruit sa « vision » du cœur. Il ajuste ses poids synaptiques pour proposer les meilleurs délais RV & LV selon les objectifs visés. Ces délais optimaux (RV et LV) sont proposés au contrôleur général pour chaque battement cardiaque qui décide ou non de les utiliser. Une description plus détaillée de cette partie sera introduite dans le chapitre IV.

Interface de communication pour des cliniciens :

Via cette interface, un médecin peut « observer » les paramètres du cœur en récupérant les mesures de l'ECG. Lors de l'implantation, ou lors d'une visite, il peut régler manuellement les « meilleures » valeurs des délais AV et VV pour chaque patient. Ces valeurs seront stockées dans le contrôleur général comme les délais AV et VV par défaut (P^{AV} et P^{VV}) et permettront d'assurer la sécurité globale en cas de problème.

Batterie :

Elle doit fournir l'énergie au CRT-P pour assurer un fonctionnement autonome pour une durée de 5 à 7 ans.

Dans la section suivante, nous allons décrire plus en détail les fonctionnalités des deux blocs centraux du CRT-P formant le couple Maître-Esclave : le Contrôleur général et le Processeur neuronal.

III.3 Description fonctionnelle des deux blocs centraux du CRT-P

Le fonctionnement général du dispositif CRT - P est conditionné par celui du couple (contrôleur général, processeur neuronal). Ces deux blocs travaillent ensemble pour fournir la meilleure stimulation du cœur et accroître le bien-être du patient. L'utilisation du processeur

neuronal se justifie par la volonté de s'adapter à toutes les situations mais aussi pour répondre dans les plus brefs délais (à chaque battement) et pour une énergie minimale. Le contrôleur général doit assurer la sécurité globale du patient mais aussi les différents modes de fonctionnements. On peut dissocier le fonctionnement du CRT-P lors des contrôles médicaux et les fonctionnements en autonomie. La multiplicité des modes de fonctionnement en autonomie fournir une meilleure polyvalence au processeur neuronal face à la diversité des cas cliniques des patients (entre autre les « répondeurs » et les « non-répondeurs »).

III.3.1 Fonctionnalités du contrôleur général

Nous listons des fonctionnalités non-exhaustives du contrôleur général ci-dessous :

- Il décide et génère les stimulations si les délais optimaux AV et VV suggérés par le processeur neuronal sont appropriés ou acceptables,
- Il surveille le statut du mode de fonctionnement et change de mode de fonctionnement si nécessaire,
- Il assure la sécurité globale en surveillant l'état physique du cœur,
- Il génère des séquences temporelles internes.

Le contrôleur général travaille comme un contrôleur maître du CRT-P adaptatif. Il supervise l'ensemble des sous-blocs et décide d'activer ou de désactiver la fonction d'un sous-bloc. Via l'interface de communication, le contrôleur général reçoit les valeurs des délais AV et VV par défaut, et puis les conserve comme les valeurs sûres. Il prend en compte la proposition du processeur neuronal et va décider ensuite quel mode sera utilisé. Dans le même temps, il garde un œil sur l'ensemble du système. Il sera prêt à redémarrer le système si le rythme cardiaque varie brutalement.

III.3.2 Fonctionnalités du processeur neuronal

Nous résumons les fonctionnalités principales du processeur neuronal :

- L'objectif principal est de générer les délais optimaux d'AV & VV afin de maximiser le SV.

Chapitre III

- Il va analyser le comportement de l'ensemble des paramètres cardiaques mesurés sur des périodes de plusieurs périodes cardiaques et contribuer à augmenter la plasticité du réseau de neurones qui propose les délais AV et VV.
- Il doit surveiller les paramètres du cœur pour détecter (voir diagnostiquer) les évolutions à moyens et long termes.

Pour répondre à ces différents objectifs, il a été proposé d'utiliser plusieurs réseaux de neurones. Chacun se focalisant sur une tâche en particulier. Les deux premiers objectifs étant des contrôles du temps, elles furent confiées à des réseaux de type SNN [1] [2]. Dans l'approche actuelle, au début de cette thèse, la première fonctionnalité fut dédiée au réseau SNN1 (Figure III-2) [3]. Ce réseau comporte deux neurones impulsionnels (SN^{AV} et SN^{VV}) pour pronostiquer respectivement les délais AV et VV visés. Un neurone de type SNN s'active à un instant T lorsque qu'une condition de seuil est satisfaite, c'est pourquoi il génère à cet instant T une impulsion qui marque son estimation du délai. Nous appelons les impulsions fournies par ces deux neurones : T^{AV} et T^{VV} . L'évolution de la réponse d'un neurone dépend des poids synaptiques qui conditionnent le passage du seuil de déclenchement. Aussi, tout neurone (et plus généralement tout réseau de neurones) utilise un algorithme pour ajuster ses poids synaptiques. L'algorithme est spécifique autant au neurone, qu'à la fonctionnalité attendue. Pour le réseau SNN1 nous utilisons deux algorithmes différents qui seront décrits dans la section III. 4. Pour améliorer la stabilité des pronostiques en présence de bruit et de forte fluctuation des paramètres physiologiques, il faut préserver la plasticité du réseau neuronal (ici SNN1). Pour cela, un second réseau de neurones impulsionnels (SNN2) a été proposé qui va moduler les taux d'apprentissage des algorithmes de SNN1.

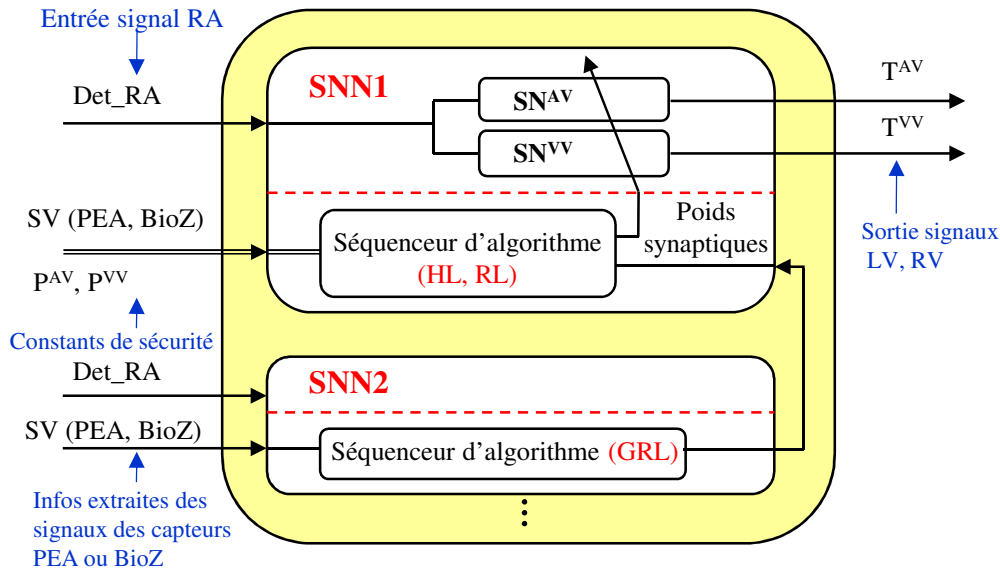


Figure III-2 Description interne du processeur neuronal

III.4 Les modes de fonctionnement du pacemaker adaptatif

Dans son régime de fonctionnement autonome, le CRT-P adaptatif possède trois modes de fonctionnement [2] : le mode Non-adaptatif, le mode Adaptatif-déecté et le mode Adaptatif-optimal (Figure III-3).

Mode Non-Adaptatif

C'est le mode par défaut et aussi le mode sécuritaire du système. L'algorithme d'apprentissage de Hebb sera utilisé dans ce mode. L'évolution des poids synaptiques des neurones SN^{AV} et SN^{VV} va permettre, en partant de n'importe quelle condition initiale, de proposer au contrôleur général des délais de AV et VV équivalents aux valeurs par défaut (P^{AV} et P^{VV}) programmés par le clinicien. Lorsque le réseau SNN1 a appris à reproduire les délais AV et VV par défaut ($T^{AV} = P^{AV}$, $T^{VV} = P^{VV}$), il se fige sur ces valeurs synaptiques. Compte tenu des imperfections (bruit sur les poids synaptiques, sur le seuil de déclenchement, etc.), le contrôleur général va vérifier que les propositions de SNN1 restent dans une fourchette voisine de ces valeurs par défaut. Si les propositions de SNN1 sont correctes durant N périodes cardiaques consécutives, il considèrera que l'apprentissage de SNN1 est abouti et basculera dans le mode suivant : Adaptatif-déecté.

Chapitre III

Mode Adaptatif-déecté

Dans ce mode, le réseau SNN1 reçoit comme objectif de maximiser la valeur de SV. En se basant exclusivement sur les observations consécutives des valeurs prises par le signal SV, le réseau doit proposer un nouveau couple de délais (AV, VV) pour chaque période cardiaque. Pour chaque proposition de couple (AV, VV), le réseau SNN1 ajuste ses poids synaptiques sous le contrôle d'un algorithme d'apprentissage avec renforcement (RL). Les ajustements se font donc à chaque période cardiaque permettant de conserver une « vigilance » face à des évolutions physiologiques. C'est un algorithme « On-line » qui offre la propriété de « suivre » les évolutions de la relation $SV = F_{NL}(AV, VV)$.

Mode Adaptatif-optimal

Un deuxième réseau de neurones impulsionnels (SNN2) a été proposé pour une amélioration future du système. Le réseau SNN2 adopte un algorithme d'apprentissage avec renforcement guidé (GRL). Sa fonction sera de proposer un taux d'apprentissage pour l'algorithme de RL pilotant SNN1. En réduisant le taux d'apprentissage, on limite l'impact d'une variation des données d'entrées sur la variation de la sortie : on réduit le gain entre les entrées et la sortie. Dans le cas contraire on augmente ce gain et les corrections sont plus nettes. Si le trajet vers le point optimal est long, cela réduira le temps de convergence. Mais en arrivant au voisinage de l'optimum, les variations risquent d'être chahutées. L'ajustement de tels comportements revient à parler de stabilité et la capacité du système à se corriger revient à préserver sa plasticité.

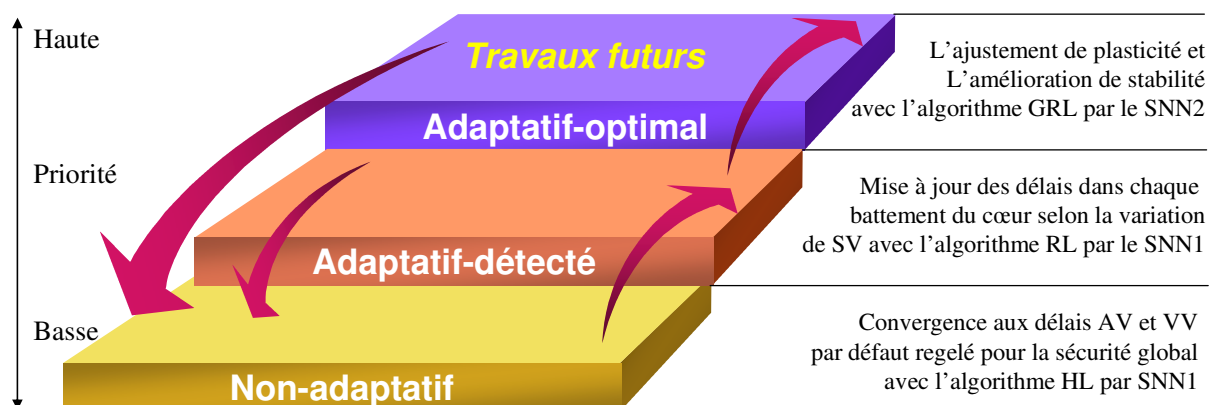


Figure III-3 Trois modes de fonctionnement du CRT-P adaptatif

Parmi ces trois modes de fonctionnement, le mode non-adaptatif a la priorité la plus basse tandis que le mode adaptatif-optimal a la plus haute (voir la Figure III-3). Donc, le système commence toujours dans le mode Non-adaptatif, où le processeur neuronal utilise

Description globale du CRT-P adaptatif du projet ADAPTER

SNN1 avec l'algorithme HL afin d'atteindre des délais AV et VV définis par les cliniciens comme valeurs par défaut. Une fois que cet apprentissage est approuvé par le contrôleur général, celui-ci bascule en mode Adaptatif-déecté. Si tout se passe bien, le CRT-P va arriver progressivement dans le mode suivant (Adaptatif-optimal), et le SNN2 va permettre d'ajuster le taux d'apprentissage de l'algorithme RL afin d'améliorer la stabilité et de préserver la plasticité du processeur neuronal. En cas d'échec, le CRT-P adaptatif peut retourner au mode Non-adaptatif à tout moment pour assurer un état de sécurité.

Conclusion

Dans ce chapitre, nous avons présenté le cahier des charges initial du projet ADAPTER et le synoptique complet de l'ensemble : Cœur – CRT-P – Instruments de surveillance médicale. Seul le fonctionnement en autonomie est pris en compte dans cette thèse. Les blocs internes du CRT-P sont détaillés avec leurs descriptions fonctionnelles.

L'architecture interne du CRT-P comporte deux blocs principaux que sont le contrôleur général et le processeur neuronal. Le contrôleur général prend en compte la gestion basique et systématique du dispositif. Le processeur neuronal travaille lui en arrière plan et offre des estimations qui peuvent s'adapter aux différentes situations dans la vie d'un patient soumis à la thérapie CRT.

L'originalité de ce travail est d'insérer un processeur neuronal, fonctionnant en temps réel avec son apprentissage « On-line », dans une chaîne instrumentale de contrôle. Avec trois modes de fonctionnement dont deux utilisent la même architecture de réseau de neurones (SNN1) notre objectif de conception vise à proposer une solution techniquement réalisable avec une relative robustesse de fabrication, une relative robustesse de fonctionnement, et ceci pour une surface de silicium minimaliste et une très basse consommation énergétique. Le chapitre suivant va présenter en détails le réseau SNN1.

Références

- [1] R. Rom, J. Erel, M. Glikson, K. Rosenblum, R. Ginosar, D. L. Hayes, “Adaptive cardiac resynchronization therapy device: a simulation report”, *Pacing and Clinical Electrophysiology*. vol. 28, no. 11, pp. 1168–1173, Nov. 2005.
- [2] R. Rom, et al. “Adaptive cardiac resynchronization therapy device based on spiking neurons architecture”, *IEEE Transaction on Neural Network*, vol. 18, no. 2, pp. 542–550, Mar. 2007.
- [3] Q. SUN, J. MICHEL, “Global partition of the ADAPTER project”, internal report of the ADAPTER project, May. 2008

Chapitre IV

Description du réseau de neurones

(SNN1)

Sommaire du chapitre IV

CHAPITRE IV	DESCRIPTION DU RESEAU DE NEURONES (SNN1)	74
IV.1	OBJECTIFS DE CONCEPTION DU RESEAU DE NEURONES (SNN1)	76
IV.2	DESCRIPTION STRUCTURELLE DE SNN1	77
<i>IV.2.1</i>	<i>Description structurelle de trois modules de SNN1</i>	<i>77</i>
<i>IV.2.2</i>	<i>L'architecture du module « Integrate & Fire »</i>	<i>80</i>
<i>IV.2.2.1</i>	<i>Description de l'intégrateur analogique avec fuite (CRRC)</i>	<i>80</i>
<i>IV.2.2.2</i>	<i>Réponse de l'intégrateur avec fuite recevant une seule impulsion en entrée</i>	<i>82</i>
<i>IV.2.2.3</i>	<i>Réponse de l'intégrateur avec fuite recevant deux impulsions en entrée</i>	<i>84</i>
<i>IV.2.2.4</i>	<i>Expressions de l'amplitude maximale en sortie du CRRC</i>	<i>85</i>
<i>IV.2.2.5</i>	<i>Choix de seuil de SNN1</i>	<i>87</i>
<i>IV.2.2.6</i>	<i>Simulation fonctionnelle de l'intégrateur avec fuite</i>	<i>92</i>
	<i>Conclusion</i>	<i>93</i>
	<i>Références</i>	<i>94</i>

IV.1 Objectifs de conception du réseau de neurones (SNN1)

L'objectif de conception du réseau de neurones (SNN1) est double. D'une part nous souhaitons utiliser le même réseau neuronal analogique pour le fonctionnement des deux premiers modes fonctionnels du CRT-P adaptatif [1] [2]. Cette option permettra de réduire la surface silicium. D'autre part le réseau doit satisfaire aux objectifs fonctionnels à savoir produire des impulsions décalées dans le temps de l'instant de dépolarisation de l'oreillette (RA). Les impulsions produites sont T^{AV} et T^{VV} qui contrôlent les contractions des ventricules en accord avec les objectifs du mode.

En mode Non-Adaptatif, SNN1 doit apprendre à reproduire les délais AV & VV fixés par le clinicien ($T^{AV} = P^{AV}$ & $T^{VV} = P^{VV}$). Dans ce mode, l'algorithme d'apprentissage est de type supervisé: HL (Hebb Learning rules). En mode Adaptatif, on bascule sur un algorithme de type « Online Learning » (non-supervisé): RL (Reinforcement Learning rules). L'objectif de ce mode est de maximiser SV (Stroke Volume) en ajustant en temps réel les délais AV & VV.

L'objectif fonctionnel étant de contrôler les délais temporels entre la contraction de l'oreillette (RA) et l'émission par SNN1 des impulsions T^{AV} & T^{VV} conduisant à la contraction des ventricules RV et LV, il nous faut définir une référence temporelle. Le temps est découpé en période cardiaque (une systole). L'instant T_0 correspond à une impulsion captée par l'électrode RA. Ensuite, dans chaque période cardiaque, le temps est redécoupé au pas de 5ms (Time Step). Les délais AV et VV sont alors définis par les équations (III.1.1) et (III.1.2).

Sur la base d'observations cliniques, ELA Medical a montré que les délais RV et LV doivent couvrir l'intervalle de temps de [40ms, 240ms]. Sachant que le délai RV est typiquement d'environ 140ms, on en déduit que le délai AV sera aussi de 140ms avec l'équation (III.1.1). Par suite de la relation (III.1.2), il apparaît que VV peut varier de -100ms à +100ms par rapport à la valeur centrale de AV. Ces valeurs découlent de l'hypothèse que la période cardiaque sera toujours supérieure ou égale à 250ms (c'est-à-dire avec un rythme cardiaque inférieur ou égal à 240 battements/mn). Pour simplifier les études dans cette approche originale, les plages dynamiques des délais AV et VV ont été uniformisées à 200ms. Ainsi les deux neurones composant SNN1 (SNAV & SNVV) auront les mêmes caractéristiques extrêmes. Une version industrielle futur pourra, sans conséquence sur le

Chapitre IV

fonctionnement, réduire les plages des délais ce qui réduira le nombre de synapses et donc la surface et la puissance.

La structure interne du réseau de neurones de type impulsif (SNN) se décompose en un nombre de synapses. Chaque synapse s'attache à pondérer l'information d'un intervalle de temps. Pour que l'analyse du neurone couvre une plage temporelle, il doit donc comporter un nombre de synapse égale à : $\text{plage temps} / \text{intervalle temps}$. Dans notre application l'intervalle de temps a été fixé à 5ms. Ce sera notre pas temporel d'analyse. Le choix de la valeur de 5ms est un compromis entre la finesse de réglage des délais et le couple : surface silicium & puissance consommée. Dans le mode adapté, le pas temporel nous limitera dans l'objectif d'atteindre le maximum du SV car celui-ci peut être localisé dans l'intervalle de temps et donc inaccessible. Nous devons accepter de rester dans son voisinage (voir chapitre VI).

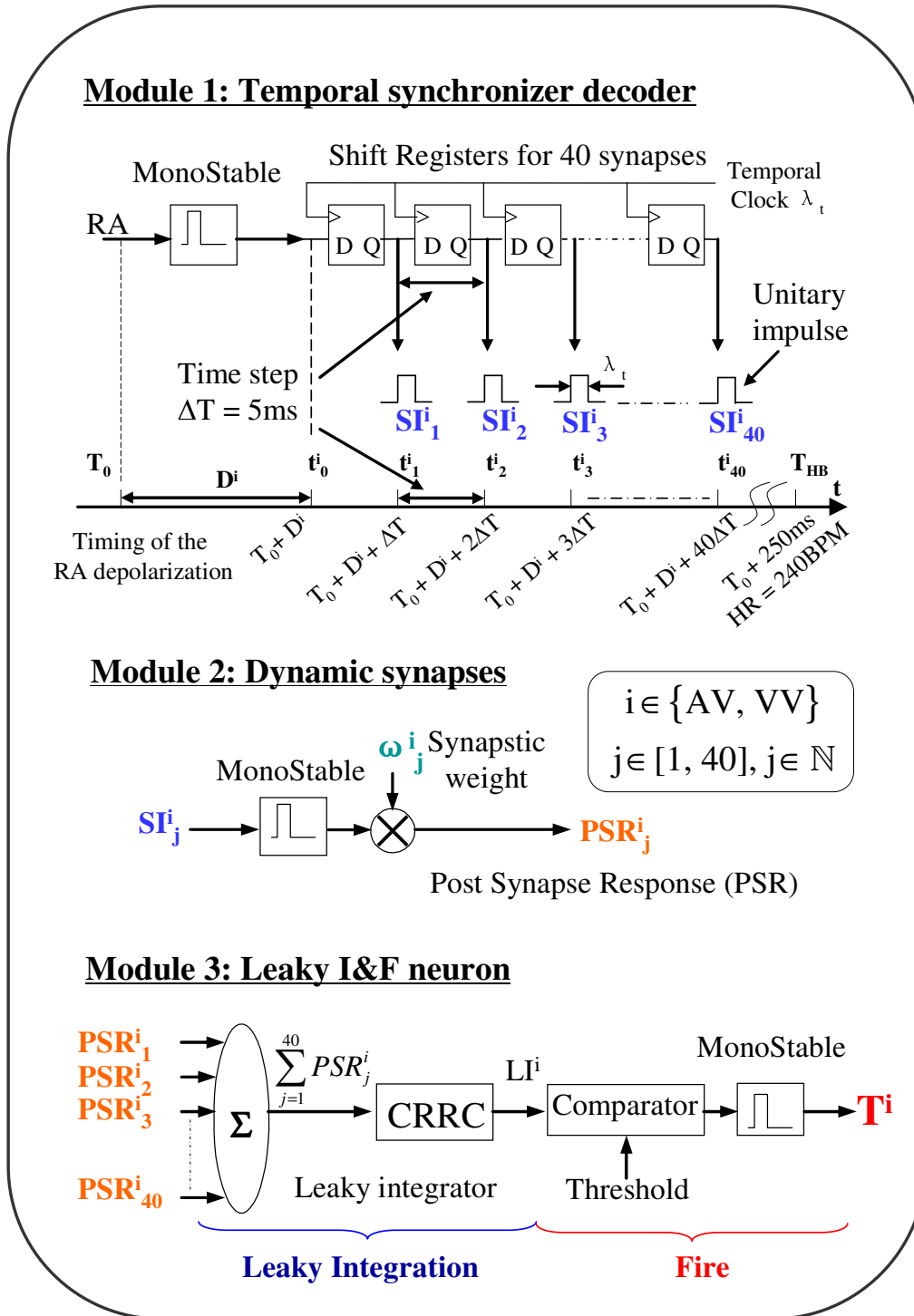
Chaque synapse va pondérer une information d'excitation par un poids synaptique ω . Ce poids synaptique est modifié par l'algorithme d'apprentissage. Après apprentissage, on peut figer ces poids et pour une séquence d'excitations (une suite d'information dans les intervalles de temps de la plage d'analyse), l'ensemble des synapses fournira une nouvelle séquence d'excitations pondérées. Cette séquence conduira, ou non, le neurone à « déclencher » une impulsion en sortie (Firing). L'instant du déclenchement du neurone sera donc conditionné par la séquence pondérée. Chaque neurone du réseau SNN1 fournira donc une impulsion appelée respectivement T^{AV} et T^{VV} . Ici le T représente le « Triggering ». En annexe de cette thèse, on trouvera une synthèse bibliographique sur les neurones de type SNN.

IV.2 Description structurelle de SNN1

IV.2.1 Description structurelle de trois modules de SNN1

Dans le SNN1, à chaque pas de temps, une impulsion unitaire (en anglais : Synaptic Impulse = SI) excite une synapse (voir Figure IV-1). Selon les poids des synapses, les signaux SI seront transmis au module de l'I & F (Integration & Fire) après la pondération. Ainsi, chaque neurone dispose de 40 synapses dynamiques pour couvrir la plage de réglage des délais (200 ms). Pour distinguer ces SI, nous les appelons SI_j^i avec i , la propriété des neurones $i \in \{AV, VV\}$ et j , l'indice pour chaque synapse $j \in [1, 40]$, $j \in \mathbb{N}$. Les structures internes des

deux neurones (SN^{AV} et SN^{VV}) sont identiques de sorte que nous ne décrivons que SN^{AV} , qui peut être divisé en trois modules listés ci-dessous (Figure IV-1).



- Le premier module est un « temporal synchronizer decoder » synchronisé sur le signal RA. Après la détection de la dépolarisation RA, le bloc monostable génère un délai prédéfini D^i pour atteindre la plage du délai AV ou VV. Par exemple, pour AV

Chapitre IV

$\in [40, 240]$ ms, D^{AV} est fixé à 35 ms afin de générer le SI^{AV}_1 à $T_0 + 40$ ms avec un écart de $\Delta T = 5$ ms. Ce point $t^{AV}_0 = 35$ ms ($t^{AV}_0 = T_0 + D_{AV}$) est reconnu comme le début du traitement synaptique. Puis l'horloge temporelle contrôle 40 registres pour générer 40 impulsions unitaires (SI^i_j) d'une durée de $\lambda_t \ll \Delta T$. Dans la pratique, T_0 peut être considéré comme l'origine des temps pour une étude d'une période : $T_0 = 0$. Nous avons :

$$SI^i_j(t) \triangleq \delta(t - t_j^i), \quad t_j^i = T_0 + D^i + j \cdot \Delta T \quad (IV.2.1)$$

où $\delta(t)$ est la fonction de Dirac. Par conséquent, les impulsions d'excitation SI^i_j sont toutes unitaires (avec une valeur arbitraire de 1).

- Le deuxième module est composé des 40 synapses par neurone. Chaque synapse, recevant un SI^i_j du premier module, émet une impulsion appelée le « Post Synapse Réponse » (PSR^i_j) vers le troisième module du neurone. Le PSR^i_j est une impulsion pondérée par le poids synaptique ω^i_j , qui est réglé par les différents algorithmes. Le PSR^i_j peut être exprimé comme :

$$PSR^i_j \triangleq \omega^i_j \cdot SI^i_j = \omega^i_j \cdot \delta(t - t_j^i) \quad (IV.2.2)$$

comme SI^i_j est unitaire et $\omega^i_j \in [0, 1]$, on aura donc $PSR^i_j \in [0, 1]$.

- Le troisième module I & F comprend le bloc « leaky Intregation » et le bloc « Fire » du neurone. Le bloc « leaky Intregation » commence par additionner les 40 signaux PSR^i_j du second module afin de générer le signal $\sum_{j=1}^{40} PSR^i_j$ et ensuite intègre ce signal au travers du « leaky integrator » (aussi appelé CRRC) pour créer le signal (LI^i). L'expression de LI^i est donnée ci-dessous:

$$\begin{aligned} LI^i(t) &= h(t) * \sum_{j=1}^{40} PSR^i_j(t) \\ &= h(t) * \sum_{j=1}^{40} \omega^i_j \cdot \delta(t - t_j^i) \\ &= \sum_{j=1}^{40} \omega^i_j \cdot h(t - t_j^i) u(t - t_j^i) \end{aligned} \quad (IV.2.3)$$

où (*) est le produit de convolution. Le signal $h(t)$ est la réponse impulsionnelle du groupe CRRC et $u(t)$ est la fonction de Heaviside. La seconde partie du troisième module est un comparateur à seuil réglable. Si l'énergie cumulée par le CRRC est supérieure à ce seuil (c'est-à-dire : $LI^i(t) > \text{seuil}$), un déclenchement sera effectué en

sortie du neurone, c'est à dire que l'on génère une impulsion unitaire à l'instant T^i correspondant au premier dépassement du seuil. Après ce déclenchement, tous les « shift registers » sont remis à 0 et aucun autre déclenchement ne pourra être émis dans cette période cardiaque.

En considérant les exigences de faible consommation d'énergie, le premier module peut être conçu avec des cellules numériques standards. Les parties critiques du neurone pour déterminer l'instant du déclenchement (donc l'instant d'émission de l'impulsion T^i par le neurone) sont les deux derniers modules. Dans la première phase de l'étude d'implémentation, le multiplieur synaptique utilisé dans le second module est considéré avec une architecture classique de multiplieur analogique que nous allons considérer comme idéal. Les caractéristiques réelles du multiplicateur vont affecter les performances du système complet et sa conception sera discutée plus en détail dans le chapitre VII. Le troisième module est la partie la plus importante du neurone car elle contrôle directement le délai. C'est cette partie qui va être détaillé ci-dessous.

IV.2.2 L'architecture du module « Integrate & Fire »

IV.2.2.1 Description de l'intégrateur analogique avec fuite (CRRC)

L'objectif de conception de l'intégrateur analogique avec fuite (leaky Integrator) est d'avoir une fonction de transfert temporel de la forme $h(t)_{\text{target}} = t \cdot e^{-t/\tau} \cdot u(t)$. A l'aide de la transformée de Laplace, on obtient la fonction de transfert:

$$H(P)_{\text{target}} = \tau^2 \cdot \frac{1}{(1 + \tau p)^2} \quad (\text{IV.2.4})$$

Dans la pratique, pour l'implémentation de l'intégrateur analogique avec fuite du troisième module, nous avons choisi une cellule assez classique dans les expérimentations de physiques des particules de haute énergie [1] et appelé CRRC. Le signal d'entrée de cette architecture collecte l'ensemble des réponses synaptiques $\sum_{j=1}^{40} PSR_j^i$, soit la séquence temporelle des 40 excitations de la première partie pondérées par les poids synaptiques de la

Chapitre IV

seconde partie. Le circuit CRRC construit le signal LI^i qui sera présenté au comparateur à seuil. Les raisons motivant la sélection de cette architecture sont les suivantes:

- L'idée d'accumulation des excitations est également associée au concept de l'intégration énergétique;
- Le CRRC présente une réponse impulsionnelle qui s'étend au fil du temps. Il permet les additions des énergies des excitations d'entrée par l'empilement des amplitudes associées suivie de la fuite de la résultante de l'accumulation.
- Le CRRC utilise une entrée en courant et une sortie en tension de sorte que la sommation des signaux PSR_j^i (des courants) peut facilement être réalisée sans une consommation d'énergie supplémentaire. Le signal de sortie en tension (LI^i) peut aussi être directement utilisé dans un comparateur de seuil classique.

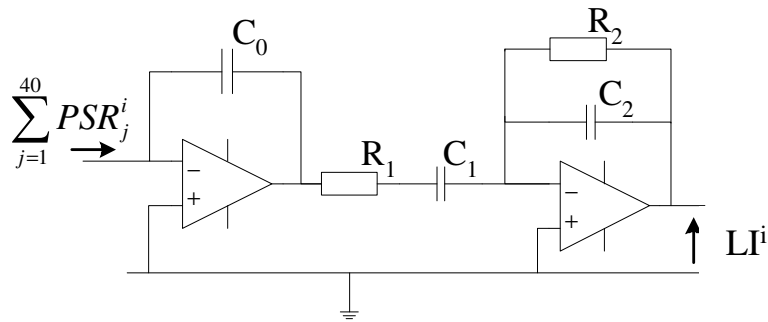


Figure IV-2 Schéma de CRRC

La Figure IV-2 présente le schéma du circuit CRRC qui est capable de produire un signal de sortie ressemblant à un potentiel d'action d'un neurone de type I & F [4]. La fonction de transfert de ce circuit est donnée ci-dessous :

$$H(P)_{CRRC} = \frac{LI^i}{\sum_{j=1}^{40} PSR_j^i} = \left(\frac{R_2 C_1}{C_0} \right) \cdot \frac{1}{(1 + R_1 C_1 p)(1 + R_2 C_2 p)} \quad (IV.2.5)$$

Afin d'atteindre l'objectif de conception fixé par l'équation (IV.2.4), nous spécifions :

$$R_1 C_1 = R_2 C_2 = RC = \tau \quad (IV.2.6)$$

$$\Rightarrow H(P)_{CRRC} = \frac{\tau}{C_0} \cdot \frac{1}{(1 + \tau p)^2} \quad (IV.2.7)$$

$$\Rightarrow h(t)_{CRRC} = \frac{\tau}{C_0} \cdot t \cdot e^{-\frac{t}{\tau}} \cdot u(t) = \left(\frac{\tau}{C_0} \right) \cdot h(t)_{target} \quad (IV.2.8)$$

Compte tenu de cette simplification, la fonction de transfert $h(t)_{\text{CRRC}}$ (IV.2.8) du CRRC est proportionnelle à la fonction de transfert cible $h(t)_{\text{target}}$. L'introduction d'une constante de proportionnalité n'aura que peu d'effet dans la fonctionnalité globale, elle constituera tout au plus un facteur de gain dans la chaîne instrumentale. La formulation adoptée pour cette fonction de transfert lors de son implémentation en VHDL-AMS est notée ci-dessous :

$$H(P) = \frac{a_0 + a_1 P}{b_0 + b_1 P + b_2 P^2} \quad (\text{IV.2.9})$$

$$a_0 = \frac{1}{\tau \cdot C_0}, a_1 = 0, b_0 = \frac{1}{\tau^2}, b_1 = \frac{2}{\tau}, b_2 = 1, \tau = RC \quad (\text{IV.2.10})$$

IV.2.2.2 Réponse de l'intégrateur avec fuite recevant une seule impulsion en entrée

L'étude de l'évolution temporelle du signal de sortie LI^i est fonction de la séquence d'entrée $\sum_{j=1}^{40} PSR_j^i$ de ce troisième module. Cette étude temporelle est fondamentale pour déterminer l'instant du déclenchement du neurone, c'est à dire l'instant où le signal LI^i dépassera le seuil. Or la séquence d'entrée de ce module dépend directement des poids synaptiques du second module. Nous définissons donc une séquence d'entrée spécifique du troisième module en décrivant la « séquence d'actions » des poids synaptiques sur chaque signal PSR_j^i composant la séquence d'entrée. Pour cela, nous définissons une séquence spécifique qui portera le label « K » par les relations suivantes:

$$\begin{cases} \omega_j^i = 1, & j \in [1, K], \\ \omega_j^i = 0, & j \in \mathbb{N} - [1, K], \end{cases} \quad K \in \mathbb{N}, K \leq 40 \quad (\text{IV.2.11}).$$

Une telle séquence présentera donc une suite de K impulsions unitaires consécutives, c'est à dire au pas de $\Delta T = 5\text{ms}$.

La réponse du circuit CRRC étant spécifique à cette séquence d'entrée portant le label « K », aussi noterons nous le signal de sortie pour cette séquence spécifique $LI^{i(K)}(t)$:

$$LI^{i(K)}(t) \triangleq \sum_{j=1}^{40} \omega_j^i \cdot h(t - t_j^i) \cdot u(t - t_j^i) = \sum_{j=1}^K LI_j^i(t) \quad (\text{IV.2.12})$$

Chapitre IV

En vertu du théorème de superposition, le signal de sortie LI^i est la somme des réponses à chaque impulsion PSR_j^i . Or la réponse du CRRC à l'impulsion PSR_j^i reçue à l'instant t_j^i est $LI_j^i(t) = \omega_j^i \cdot h(t - t_j^i) \cdot u(t - t_j^i)$.

Afin d'introduire progressivement les notations, nous allons traiter successivement des séquences comportant une seule impulsion, puis deux impulsions avant de généraliser la mise en équation de la réponse temporelle LI^i .

Pour commencer, nous analysons l'évolution temporelle de la réponse lorsque le circuit CRRC ne reçoit qu'une seule impulsion d'entrée. Soit, selon (IV.2.12), $LI^{i(1)}(t)$:

$$LI^{i(1)}(t) = \frac{\tau}{C_0} \cdot (t - t_1^i) \cdot e^{-\frac{(t-t_1^i)}{\tau}} \cdot u(t - t_1^i) \quad (\text{IV.2.13})$$

Comme l'objectif est de tester le dépassement d'un seuil, nous allons nous focaliser sur l'amplitude maximale que peut atteindre la réponse $LI^{i(K)}$. Pour $t \geq t_1^i$, nous notons $t^{i(K)\max}$ l'instant où le signal $LI^{i(K)}(t)$ atteint son maximum (voir Figure IV-3). Pour $K = 1$, nous avons :

$$\begin{aligned} \text{Max} \{ LI^{i(1)}(t) \} &\triangleq LI^{i(1)\max}(t) \triangleq LI^{i(1)}(t^{i(1)\max}) \\ \Rightarrow t^{i(1)\max} &= t_1^i + \tau, \quad LI^{i(1)}(t^{i(1)\max}) = \frac{\tau^2}{C_0 \cdot e} \end{aligned} \quad (\text{IV.2.14})$$

Le temps nécessaire pour que la réponse du CRRC à une impulsion d'entrée unitaire atteigne son amplitude maximale est τ (appelé le délai du maximum). Pour la simplicité, l'amplitude de la réponse de CRRC à une impulsion d'entrée unitaire est normalisée à 1.

$$LI^{i(1)}(t^{i(1)\max}) = \frac{\tau^2}{C_0 e} \triangleq 1 \Rightarrow C_0 = \frac{\tau^2}{e} \quad (\text{IV.2.15})$$

Cette normalisation conditionne l'expression de la réponse à une impulsion unitaire arrivant à l'instant t_j^i (PSR_j^i) comme suit:

$$LI_j^i(t) = \omega_j^i \cdot \frac{e}{\tau} \cdot (t - t_j^i) \cdot e^{-\frac{(t-t_j^i)}{\tau}} \cdot u(t - t_j^i) \quad (\text{IV.2.16})$$

IV.2.2.3 Réponse de l'intégrateur avec fuite recevant deux impulsions en entrée

Après le cas trivial d'une séquence à une seule impulsion, il nous faut analyser le cas d'une séquence présentant deux impulsions unitaires consécutives. Le caractère consécutif est fondamental car il permettra d'éviter des déclenchements intempestifs en cas de bruits électroniques dans les signaux PSR_j^i (bruits issus soit de la mémorisation des poids synaptiques soit des disparités des multiplieurs soit d'autres sources encore). Ce point sera discuté au paragraphe IV.2.2.5. Dans ce paragraphe, nous étudions le cas d'une réception d'une séquence ($K = 2$) à l'entrée de CRRC (Figure IV-3). L'expression du signal de sortie en réponse de cette séquence est $LI^{i(2)}(t)$. Elle peut être établie à partir de (IV.2.12):

$$LI^{i(2)}(t) = LI_1^i(t) + LI_2^i(t) \quad (IV.2.17)$$

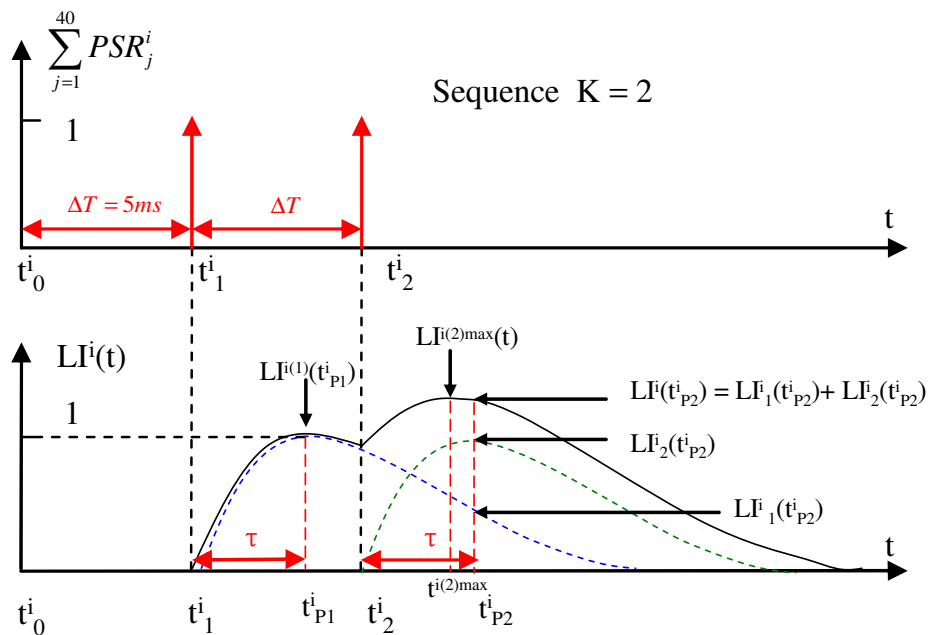


Figure IV-3 la réponse du CRRC recevant deux impulsions unitaires : $PSR_j^i, j \in \{1, 2\}$

Plusieurs observations sont à faire sur la figure IV.3. La première est que comme nous l'annoncions, l'amplitude de sortie résulte de la superposition des amplitudes des signaux respectifs. L'effet de « fuite » de l'intégrateur vient du décalage temporel ΔT séparant les impulsions d'entrée. Il devient évident que si ce décalage augmente, l'effet d'empilement va diminuer et l'amplitude résultante va diminuer. Par suite il devient facile de généraliser l'étude à une séquence K quelconque. La seconde observation est que l'amplitude maximale est atteinte avant l'instant du maximum de la seconde réponse individuelle. Pour calculer

Chapitre IV

l'amplitude maximale de la réponse $LI^{(K)}(t)$, il nous faut donc déterminer l'instant de ce maximum : $t^{i(2)\max}(t)$.

Au paragraphe suivant nous allons exprimer l'amplitude maximale atteinte en sortie du circuit CRRC pour une séquence de K impulsions unitaires consécutives.

IV.2.2.4 Expressions de l'amplitude maximale en sortie du CRRC

Dans ce paragraphe nous proposons deux approches pour évaluer le maximum du signal LI^i en réponse à une séquence de K impulsions unitaires consécutives. La première approche est grossière mais simple et nous a permis de mettre en place les jeux d'équations. La seconde approche est le calcul exact de l'amplitude maximale.

Approche grossière

Nous étudions l'amplitude cumulée de CRRC à ces deux instants ($t_{P_1}^i$ et $t_{P_2}^i$).

$$\begin{aligned} LI^i(t_{P_1}^i) &= LI_1^i(t_{P_1}^i) = LI_1^i(t^{i(1)\max}) = 1 \\ LI^i(t_{P_2}^i) &= LI_1^i(t_{P_2}^i) + LI_2^i(t_{P_2}^i) = LI_1^i(t_{P_2}^i) + 1 \end{aligned} \quad (\text{IV.2.18})$$

En première approche, nous pouvons considérer que l'amplitude résultante est proche de la somme des amplitudes à l'instant du dernier maximum : c'est l'instant $t_{P_2}^i$ dans la Figure IV-3. Pour le cas d'une séquence K, nous aurons l'instant du dernier maximum (celui relatif à la réponse à la dernière impulsion) $t_{PK}^i = t_K^i + \tau$, et par suite l'expression approchant l'amplitude du signal $LI^i(t)$ en calculant son amplitude à l'instant $t_{P_K}^i$:

$$\begin{aligned} LI^i(t_{P_K}^i) &= \sum_{k=1}^{40} LI_k^i(t_{P_K}^i) \\ &= \sum_{k=1}^K \omega_j^i \left(1 + (j-k) \frac{\Delta T}{\tau} \right) e^{\frac{-(j-k)\Delta T}{\tau}} u(t_{P_K}^i - t_k^i) \\ &= \sum_{k=1}^K \omega_j^i C_{j-k}^i(t_{P_K}^i) \end{aligned} \quad (\text{IV.2.19})$$

$$\text{avec } C_{j-k}^i(t_{P_K}^i) = \left(1 + (j-k) \frac{\Delta T}{\tau} \right) e^{\frac{-(j-k)\Delta T}{\tau}} u(t_{P_K}^i - t_k^i) \quad (\text{IV.2.20})$$

Dans l'expression de C_{j-k}^i , k est une variable itérative muette.

Approche exacte

Selon la Figure IV-3, le maximum de la réponse du CRRC recevant une séquence ($K = 2$), est attendu à l'instant $t^{i(2)\max} < t_{p2}^i$. L'instant du maximum sera obtenu en dérivant l'expression du signal de sortie au voisinage de t_{p2}^i :

$$LI^{i(2)}(t) = e^{\frac{t-t_1^i}{\tau}} e^{-\frac{(t-t_1^i)}{\tau}} + e^{\frac{t-t_2^i}{\tau}} e^{-\frac{(t-t_2^i)}{\tau}} \quad (IV.2.21)$$

$$\frac{d\left(LI^{i(2)}\left(t^{i(2)\max}\right)\right)}{dt} = 0 \Rightarrow t^{i(2)\max} = t_1^i + \tau + \frac{\Delta T}{1 + e^{-\frac{\Delta T}{\tau}}} \quad (IV.2.22)$$

$$\Rightarrow t_2^i < t^{i(2)\max} < t_2^i + \tau \triangleq t_{p2}^i \quad (IV.2.23)$$

Ainsi, $t^{i(2)\max}$ est toujours inférieur à t_{p2}^i de sorte que le maximum cumulé arrive un peu plus tôt que le deuxième (le $K^{\text{ième}}$ en général) « point du maximum » t_{p2}^i .

Dans le cas d'une séquence K quelconque, les expressions respectives de l'instant du maximum $t^{i(K)\max}$ et la valeur de l'amplitude maximale $LI^{i(K)\max}(t)$ sont données par :

$$t^{i(K)\max} = t_1^i + \tau + \frac{b_K}{a_K} \quad (IV.2.24)$$

$$LI^{i(K)\max}(t) = LI^i\left(t^{i(K)\max}\right) = a_K e^{-\left(\frac{b_K}{a_K \tau}\right)} \quad (IV.2.25)$$

où $a_K = \sum_{j=0}^{K-1} e^{\frac{j\Delta T}{\tau}}$, $b_K = \sum_{j=1}^{K-1} j\Delta T e^{\frac{j\Delta T}{\tau}}$

La conclusion de cette étude est qu'en contrôlant l'énergie d'excitation du circuit, on peut contrôler l'amplitude de sortie du CRRC. Or l'énergie d'excitation vient de la somme des impulsions d'entrée, c'est à dire des impulsions unitaires après pondérations par les poids synaptiques. En contrôlant ces poids synaptiques, on contrôle d'une part, la séquence d'impulsions et de d'autre part, l'énergie cumulée. Par conséquent, l'algorithme d'apprentissage qui contrôle les poids synaptique doit adapter ces derniers de manière à présenter au circuit CRRC la séquence ad hoc pour provoquer le dépassement du seuil dans l'intervalle de temps idoine afin de satisfaire aux objectifs fonctionnels du réseau de neurone SNN1 selon son mode de fonctionnement.

IV.2.2.5 Choix de seuil de SNN1

Rappelons que l'objectif du troisième module est de délivrer une impulsion qui servira de référence temporelle pour la contraction de l'un des ventricules. Or l'instant d'apparition de cette impulsion doit correspondre au retard voulu sur la contraction de l'oreillette RA : le retard AV ou VV. Par ailleurs, rappelons que toute séquence K est référencée à la dernière contraction RA captée. Ainsi, le signal de sortie du CRRC LI^i doit dépasser le seuil dans l'intervalle de temps contenant le délai idoine : AV ou VV. On admet donc que les poids synaptiques conduisent à générer une séquence de K impulsions unitaires et consécutives telle que la $K^{ième}$ impulsion provoque le dépassement du seuil et déclenche donc le neurone. L'idée d'imposer que le neurone ne déclenche qu'après K impulsions consécutives est une simplification de l'idée que le déclenchement du neurone ne doit survenir que si un minimum de densité d'énergie par unité de temps a été présenté en entrée du circuit CRRC. Cela revient à considérer qu'un bruit (ou processus aléatoire) n'a que peu de chance de provoquer le déclenchement et donc que seul l'ajustement ad hoc des poids permettra de contrôler ce déclenchement.

La question qui se pose maintenant est : quelle valeur de K peut-on prendre ? Avant de répondre à cette question regardons ce qui se passe lorsque le nombre K croît.

La Figure IV-4 montre le cas d'une séquence $K = 5$. Dans cette simulation, nous avons pris une constante de temps τ est égal au pas temporel $\Delta T = 5ms$. On devine que si le pas temporel est plus court que la constante de temps, l'effet de « mémoire » sur les amplitudes des précédentes réponses sera renforcé et l'accumulation d'énergie (l'augmentation de l'amplitude résultante) sera plus rapide. Dans le cas contraire, l'effet de l'accumulation d'énergie sera moins important. On peut remarquer par ailleurs que l'accroissement de l'amplitude résultante diminue. S'il faut introduire un seuil entre deux maxima successifs du signal LI^i , alors il vaut mieux fixer ce seuil après les deux à trois premiers maxima. De ce fait les écarts sont plus importants et des variations d'amplitudes (bruit électronique) seront moins pénalisantes. Par suite pour répondre à notre question précédente, il nous faut analyser l'évolution des maxima en fonction du nombre K ainsi que l'influence du rapport $\Delta T / \tau$ qui conditionne à la fois l'amplitude maximale de la réponse du CRRC et l'instant d'apparition de ce maximum.

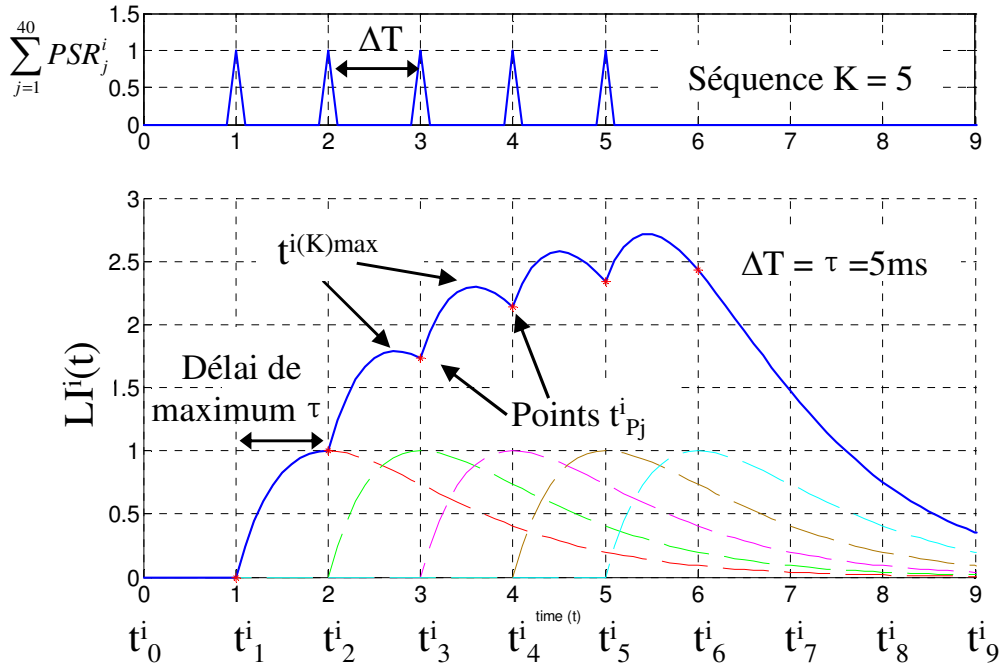


Figure IV-4 Variation d'amplitude de réponse CRRC en recevant 5 impulsions de PSR_j^i unitaires consécutives

Pour cette étude, nous définissons le coefficient $N = \Delta T / \tau$ que nous introduisons dans l'équation (IV.2.25). Nous traçons Figure IV-5 l'évolution des maxima en fonction de la longueur K de la séquence d'impulsions unitaires consécutives. Cette évolution est faite pour diverses valeurs du paramètre N . On visualisera aussi sur la Figure IV-5, l'évaluation de l'amplitude selon l'équation approchée (IV.2.19).

Nous pouvons constater que l'évolution des maxima avec l'augmentation de K tend vers des valeurs asymptotiques de l'amplitude maximale. La vitesse de convergence vers ces asymptotes diffère avec la valeur de N . Ces valeurs asymptotiques obtenues pour des valeurs de K infini, seront appelées respectivement $LI_{\infty}^{i(K)\max}$ et $LI_{pK\infty}^i$. Avec la définition de N et l'équation (IV.2.19), l'expression de $LI_{pK\infty}^i$ est décrite ci-dessous:

$$LI^i(t_{P_K}^i) \Big|_{t_{P_K}^i \rightarrow \infty} = \sum_{k=0}^{\infty} (1 + kN) \beta^k, \quad \beta = e^{-N} \quad (IV.2.26)$$

si $|\beta| < 1$, la série $\sum_{k=0}^{\infty} (a + kN) \beta^k$ est convergente

$$\sum_{k=0}^{\infty} (a + kN) \beta^k \rightarrow \frac{a}{1 - \beta} + \frac{\beta N}{(1 - \beta)^2} \quad (IV.2.27)$$

avec $\alpha = 1$, $N > 0$, $\beta = e^{-N} < 1$,

$$LI_{p_K \rightarrow \infty}^i = LI^i(t_{p_K}^i) \Big|_{t_{p_K}^i \rightarrow \infty} = \frac{1 + (N-1)e^{-N}}{(1-e^{-N})^2} \quad (IV.2.28)$$

De la même manière, nous calculons le $LI_{\infty}^{i(K)\max}$:

$$LI_{\infty}^{i(K)\max} = LI^i(t^{i(K)\max}) \Big|_{t^{i(K)\max} \rightarrow \infty} = \frac{e^{\frac{N}{1-e^{-N}}}}{e^N - 1} \quad (IV.2.29)$$

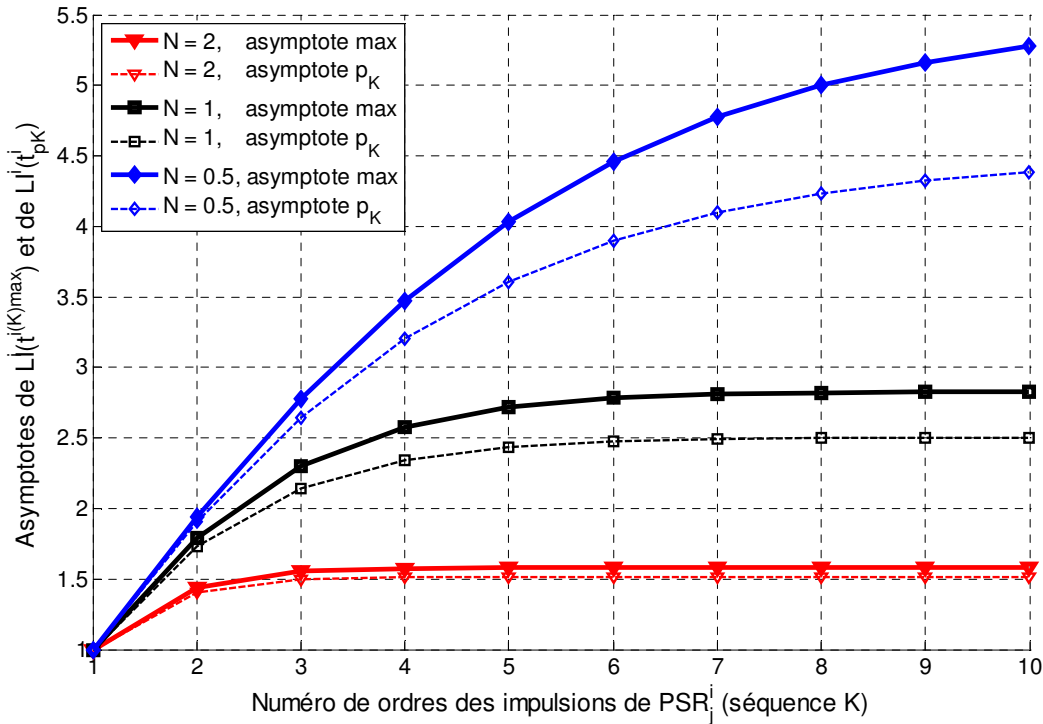


Figure IV-5 Evolution de l'asymptote $LI^{i(K)\max}(t)$ et de $LI^i(t_{pK}^i)$ en fonction de N

Pour répondre à la question sur le nombre d'impulsions, nous allons introduire une contrainte additionnelle. Supposons que le déclenchement du neurone à l'instant T^i exige m impulsions unitaires consécutives, alors toute séquence $K \geq m$ provoquera le même déclenchement, mais les $K-m$ impulsions sont inutiles. La contribution énergétique de toutes les impulsions PSR_j^i au-delà de $K-m$ sera nommée le « Reste » (avec K pouvant devenir infini pour cette étude).

$$\ll \text{Reste} \gg = LI_{\infty}^{i(K)\max}(t) - LI^{i(m)\max}(t) \quad (IV.2.30)$$

La contrainte additionnelle est d'imposer que le déclenchement du neurone ne peut se produire qu'avec m impulsions unitaires consécutives et qu'aucun déclenchement ne peut être obtenu en supprimant l'impulsion m (la dernière). Cela signifie que la contribution des $K-m$ impulsions qui peuvent suivre ne peut compenser la perte de la dernière impulsion. Une autre

interprétation est de dire que la contribution en amplitude du « Reste » ne peut compenser la perte d'une impulsion unitaire et consécutive dans la séquence m. Selon cette condition additionnelle, il devient clair que le déclencheur exclusif sera une séquence de m impulsions unitaires consécutives et que par conséquent le seuil de déclenchement sera placé sous l'amplitude maximale atteinte par le signal $LI^{i(m)\max}$.

La Figure IV-6 illustre la contrainte additionnelle. On y fait apparaître « l'Ecart » qui représente le manque dans l'amplitude résultante si l'impulsion m est absente. L'interprétation de l'apport, dans l'amplitude maximale, du « Reste » y est aussi représentée. Enfin la position du seuil, en accord avec notre contrainte, y est matérialisée. Afin d'accroître la tolérance aux fluctuations d'amplitude, le seuil sera choisit au milieu de l'« Ecart », avec une valeur inférieure à $LI^{i(m)\max}(t)$. La figure IV-6 repose sur les équations suivantes :

$$\begin{cases} LI^{i(m)\max}(t) > seuil \\ LI^{i(m-1)\max}(t) + \ll Reste \gg < seuil \end{cases} \quad (IV.2.31)$$

$$\Rightarrow \ll Ecart \gg = LI^{i(m)\max}(t) - LI^{i(m-1)\max}(t) - \ll Reste \gg > 0 \quad (IV.2.32)$$

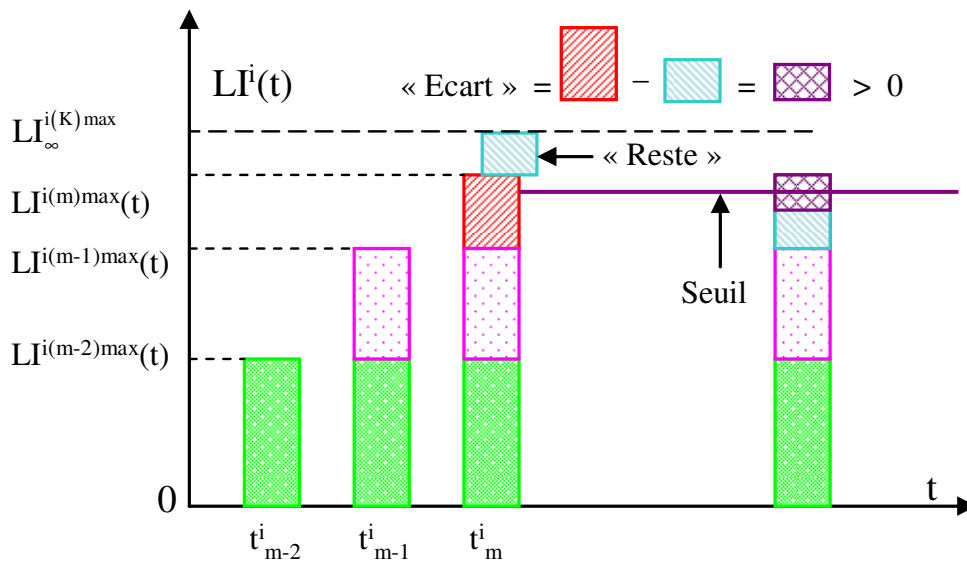


Figure IV-6 Définitions de l'« Ecart » et du « Reste »

La Figure IV-6 s'est concentrée sur l'influence du nombre K. Revenons donc à l'influence du rapport N. Selon l'évolution du maximum $LI^{i(K)\max}(t)$ dans la Figure IV-5, la valeur asymptotique peut quasiment être atteinte après deux impulsions unitaires consécutives pour une valeur $N = 2$. Pour cette valeur de N, on devine que « l'Ecart » sera très faible dès lors que $m > 2$. Lorsque $N = 0,5$, l'amplitude maximale augmente rapidement mais la valeur asymptotique ne peut être atteinte qu'après plusieurs impulsions (> 10 impulsions). L'instant du déclenchement ne pourra donc pas être proche de RA : le délai minimum entre RA et le

Chapitre IV

déclenchement du neurone sera donc supérieur à $10 \times \Delta T = 50 \text{ms}$. Ceci est en contradiction avec les spécifications initiales. Toutefois, lorsque $N = 1$, l'amplitude maximale s'approche de la valeur asymptotique après seulement de 3 ou 4 impulsions, ce qui correspond justement à la spécification de la contrepartie numérique (AI Semiconductor). En outre, l'« Ecart » est plus grand dans ce cas que dans le cas de $N = 2$. En conséquence la robustesse est également améliorée.

Dans la Figure IV-7, nous traçons la valeur de l'« Ecart » selon (IV.2.32) en fonction de N . Nous recherchons le nombre (m) d'impulsions unitaires consécutives indispensables pour satisfaire à notre contrainte additionnelle et aux spécifications initiales du projet. On observe que l'« Ecart » passe d'une valeur négative à une valeur positive dans le cas de $N = 1$ et $N = 0,5$. Selon notre contrainte, seule une valeur positive de l'« Ecart » est compatible avec notre situation. Par conséquent les plus petites valeurs de m sont 4 pour $N = 1$ et 9 quand $N = 0,5$. De toute évidence, le cas $N = 1$ avec $m = 4$ est le choix approprié pour l'implémentation de SNN1.

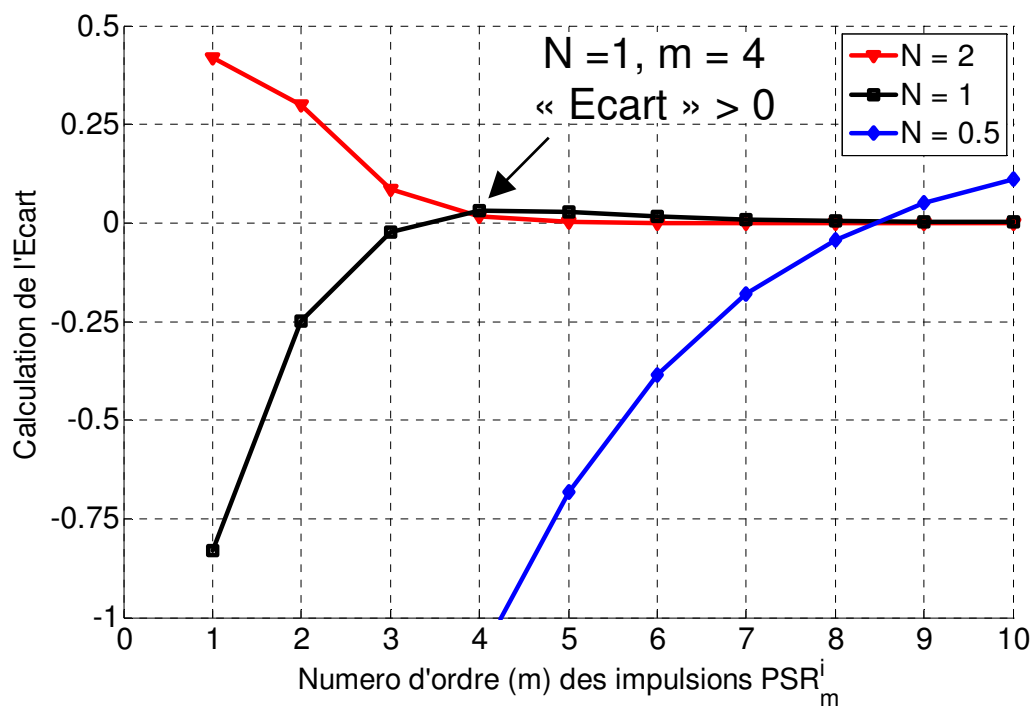


Figure IV-7 Calcule de l'« Ecart » en fonction de N pour le choix du seuil

IV.2.2.6 Simulation fonctionnelle de l'intégrateur avec fuite

Pour valider la fonctionnalité et la robustesse de la structure de l'intégrateur avec fuite, les simulations fonctionnelles sont effectuées tout d'abord sous MATLAB, puis sous VHDL-AMS. La simulation de l'architecture CRRC sous VHDL-AMS est illustrée dans la Figure IV-8. Les impulsions PSR_j^i y sont idéalement bornées à $[0, 1]$. La valeur optimale de N est fixée à 1 selon l'étude ci-dessus. Le seuil est choisi avec soin entre les $LI^{i(3)\max}(t)$ et le $LI^{i(4)\max}(t)$. Le résultat de la simulation (Figure IV-8) montre que lorsqu'il manque dans la séquence la quatrième impulsion PSR_j^i unitaire consécutive, l'amplitude cumulée ne dépasse pas le seuil et qu'il faut, quelque soit la séquence précédente, attendre que quatre impulsions unitaires consécutives soient appliquées en entrée du CRRC pour générer un déclenchement de la sortie du neurone I & F. En d'autres termes, la quatrième impulsion sera discriminante pour le déclenchement du délai optimal de AV ou VV (T^i).

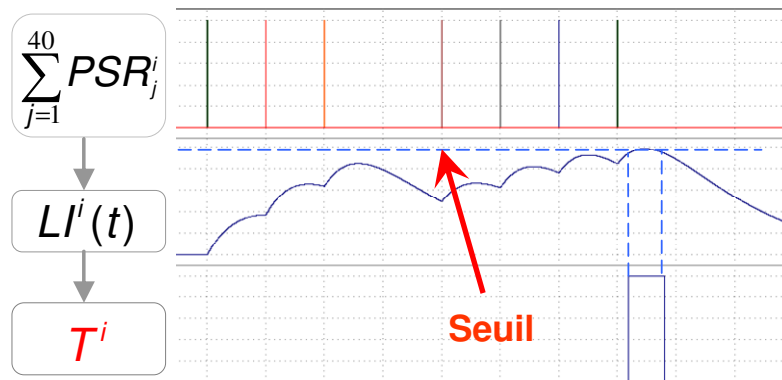


Figure IV-8 Déclenchement de T^i avoir reçu quatre impulsions de PSR_j^i unitaires consécutives

Conclusion

Une implémentation analogique du réseau de neurones impulsionnels (appelé SNN1) a été décrite dans ce chapitre. L'objectif du SNN1 est de générer le déclenchement du neurone I & F dédié au délai optimal de AV/VV (T^i , $i \in \{AV, VV\}$) au moment où le signal de l'intégrateur avec fuite $LI^i(t)$ est égal au seuil choisi. Nous utilisons le bloc CRRC en tant qu'intégrateur avec fuite. L'étude portant sur le choix du nombre d'impulsions indispensables et du seuil est détaillée dans ce chapitre. Le rôle attendu des algorithmes d'apprentissage y est souligné. Dans le chapitre suivant, nous allons décrire les deux algorithmes utilisés qui contrôlent T^i pour les deux objectifs différents.

Références

- [1] R. Rom, J. Erel, M. Glikson, K. Rosenblum, R. Ginosar, D. L. Hayes, “Adaptive cardiac resynchronization therapy device: a simulation report”, *Pacing and Clinical Electrophysiology*. vol. 28, no. 11, pp. 1168–1173, Nov. 2005.
- [2] R. Rom, et al. “Adaptive cardiac resynchronization therapy device based on spiking neurons architecture”, *IEEE Transaction on Neural Network*, vol. 18, no. 2, pp. 542–550, Mar. 2007.
- [3] F. Anstotz, Y. Hu, J. Michel, J. L. Sohler, D. Lachartre, “Filtres: a 128 channels VLSI mixed front end read out electronic development for microstrip detectors”, *Nuclear Instruments and Methods in Physics Research, sec.A*, vol. 412, no. 1, pp. 123–134, Jun. 1998.
- [4] W. Maas, C. M. Bishop, *Pulsed neural networks*. Cambridge, MA: MIT Press, 1999.

Chapitre V

Les algorithmes d'apprentissage

adoptés pour le SNN1

Sommaire du chapitre V

CHAPITRE V	LES ALGORITHMES D'APPRENTISSAGE ADOPTES POUR LE SNN1	96
V.1	L'ALGORITHME DU MODE NON-ADAPTATIF : ALGORITHME D'APPRENTISSAGE DE HEBB (HL) ...	98
V.2	L'ALGORITHME DU MODE ADAPTATIF-DETECTE: ALGORITHME D'APPRENTISSAGE PAR RENFORCEMENT (RL)	100
V.2.1	<i>Choix de l'approche de l'algorithme RL.....</i>	<i>100</i>
V.2.2	<i>Description explicite de l'algorithme RL.....</i>	<i>102</i>
V.3	SIMULATION DES ALGORITHMES HL ET RL SOUS MATLAB.....	105
	<i>Conclusion.....</i>	<i>108</i>
	<i>Références</i>	<i>109</i>

V.1 L'algorithme du mode Non-adaptatif : algorithme d'apprentissage de Hebb (HL)

En mode Non-Adaptatif, l'algorithme d'apprentissage de Hebb (HL) [3] vise à obtenir le déclenchement du délai optimal AV/VV T^i qui correspond à la valeur de P^i pré-réglée par le clinicien. En contrôlant les amplitudes des impulsions unitaires par les poids synaptiques, nous pouvons contrôler l'énergie cumulée dans le CRRC. Idéalement, le contrôle du poids permet de régler avec précision le moment du déclenchement (T^i) lorsque le signal LI^i dépasse le seuil. Nous pouvons ajuster les poids synaptiques ω_j^i avec (V.1.1).

$$\omega_j^i(b+1) = \omega_j^i(b) + \eta \cdot R_j^i \quad (\text{V.1.1})$$

Où η est le taux d'apprentissage et b est la période de battement de cœur. En fait, la modification du poids synaptique dépend du paramètre R_j^i déterminée par cet algorithme, qui est une évolution de [5]. La différence principale est que les « shift registers » seront remis à zéro une fois que le déclenchement du neurone I & F de sortie aura été généré. Le SNN1 ne prend en compte que le passé (pas le futur) des énergies cumulées. Cet algorithme agit comme une machine à états finis [1] [2] qui propose l'utilisation de deux variables d'état: « Etat de neurone I & F » et « Etat synaptique » (Figure V-1). La mesure du décalage ($P^i - T^i$) est utilisée pour définir l'état du neurone I & F. Si le déclenchement T^i est proche de P^i , c'est à dire $|P^i - T^i| < 3 \cdot \Delta T$, l'« Etat du neurone I & F » récupère la valeur « Proche ». Dans tout autre cas, T^i est loin de P^i et il aura la valeur « Loin ». Pour adresser les synapses adéquates, l'algorithme mesure le délai ($TPSR_j^i - T^i$) entre l'instant du début de chaque signal PSR_j^i ($TPSR_j^i$) et l'instant de déclenchement du neurone (T^i). Cette mesure définit l'« Etat de synapse » de la synapse (i, j). Cet état peut être trouvé dans un des quatre états différents: Out, PreHebb, Hebb et PostHebb, selon les définitions ci-dessous. L'état synaptique (i, j) dépend de la position d'une fenêtre de temps spécifique ($\nabla = 20$ ms: 4 impulsions consécutives). L'état synaptique (i, j) est donné par :

- $(TPSR_j^i - T^i) \in [-2\nabla, -\nabla]$, \Leftrightarrow PostHebb;
- $(TPSR_j^i - T^i) \in [-\nabla, 0]$, \Leftrightarrow Hebb;
- $(TPSR_j^i - T^i) \in [0, \nabla]$, \Leftrightarrow PreHebb;
- $(TPSR_j^i - T^i) \notin [-2\nabla, \nabla]$, \Leftrightarrow Out.

Chapitre V

Une synapse dans l'état « Out » n'est pas destinée à contribuer au déclenchement T^i du neurone et son poids synaptique doit être faible (idéalement 0). Les synapses dans l'état « Hebb » sont les quatre impulsions indispensables qui contribuent au déclenchement de T^i et elles doivent avoir un poids synaptique élevé (idéalement 1). Les poids des synapses dans les états « PreHebb » ou « PostHebb » seront soit augmentés ($R_j^i = +1$), soit baissés ($R_j^i = -1$) en fonction de la valeur du taux d'apprentissage η (Tableau V-1). Ces variations des poids synaptiques permettent de faire glisser progressivement le groupe de poids élevés vers la position induisant un déclenchement $T^i = P^i$.

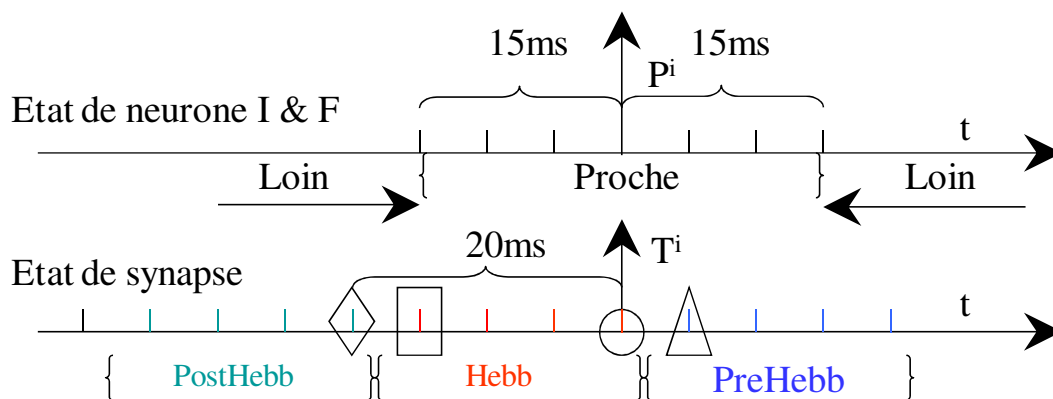


Figure V-1 Définition de l'« Etat du neurone I & F » et de l'« Etat synaptique »

Pour la simplicité du tableau ci-dessous, nous fixons le taux d'apprentissage η à 1 pour avoir toujours un déclenchement dans chaque période cardiaque. Nous pouvons aussi modifier ce taux d'apprentissage pour avoir une optimisation plus fine. Le SNN2 proposé va ajuster ce taux d'apprentissage dans le mode Adaptatif-Optimal. En réduisant le taux d'apprentissage, le réseau SNN1 ne produira pas toujours de déclenchement. Néanmoins dans le mode non-adapté, les prédictions du processeur neuronal ne sont pas appliquées au cœur. Dans ce mode le processeur neuronal est en cours d'acquisition des valeurs sécuritaires de délais. L'absence de déclenchement n'est donc pas handicapante pour le patient.

Tableau V-1 Variation de R_j^i avec l'algorithme d'apprentissage Hebb

R_j^i	Etat de neurone I & F	Etat de synapse		
		<i>PostHebb</i>	<i>Hebb</i>	<i>PreHebb</i>
$T^i < P^i$	Proche		$-1 \square^*$	$+1 \Delta^*$
$T^i > P^i$	Proche	$+1 \diamond^*$	$-1 \circ^*$	

R_j^i	Etat de neurone I & F	Etat de synapse		
		<i>PostHebb</i>	<i>Hebb</i>	<i>PreHebb</i>
$T^i < P^i$	Loin		-1	+1
$T^i > P^i$	Loin	+1	-1	

* correspond à l'impulsion spécifique de l'état de synapse sélectionné dans Figure V-1

Lorsque les synapses sont dans l'état « Loin », les poids synaptiques des quatre impulsions indispensables pour le déclenchement T^i vont être changés en même temps vers la direction de P^i dans chaque période cardiaque. En mettant les poids synaptiques des quatre impulsions dans la zone « Hebb » à 0 et ceux dans les zones « PostHebb » ou « PreHebb » à 1, l'algorithme HL va déplacer le T^i vers le P^i . Donc, le déclenchement de T^i dans l'état « Loin » varie de quatre pas de temps ($4\Delta T = 20ms$) dans chaque période cardiaque afin d'atteindre la zone « Proche » plus rapidement. Quand le déclenchement arrive dans la zone « Proche », l'algorithme va affiner l'optimisation. Le déclenchement de T^i dans l'état « Proche » ne varie que d'un pas de temps (5 ms) dans chaque période cardiaque afin d'atteindre le P^i plus précisément.

V.2 L'algorithme du mode Adaptatif-Déecté: algorithme d'apprentissage par renforcement (RL)

V.2.1 Choix de l'approche de l'algorithme RL

Tout d'abord, rappelons que l'objectif dans ce mode est de maximiser le « Blood Pressure » (BP), via la mesure du paramètre « Stroke Volume » (SV), en adaptant les délais AV et VV.

L'étude clinique, faite par Dr. Whinnett [6], montre une relation entre la BP et les délais AV et VV. Cette relation est non explicitée dans la publication. Néanmoins elle est décrite comme une courbe en cloche « Bell Shape ». La mesure du « Blood Pressure » (BP) est envisagée soit par un capteur « hemodynamic Bio Impedance Sensor » (BioZ) de ELA Medical, soit par un capteur « Peak Endocardial Accelerometer Sensor » (PEA).

Chapitre V

Afin d'étudier les effets hémodynamiques des ajustements AV et VV, nous introduisons une relation mathématique explicite décrivant $SV = f(AV, VV)$. En considérant les deux variables AV et VV comme indépendantes, nous pouvons considérer le couple (AV, VV) comme un vecteur d'un espace de dimension 2. Ainsi la relation $SV = f(AV, VV)$ peut être interprétée dans une représentation en 3D comme une surface. Chaque point de cette surface associe une valeur de la grandeur SV à une valeur du vecteur (AV, VV), c'est à dire à un point du plan (AV x VV). La surface courbe suggérée par Dr. Whinnett ne présente qu'un seul maximum de SV dans tout le plan (AV x VV).

La mise en œuvre d'un algorithme de recherche de maxima présente une alternative : soit l'on dispose de l'expression explicite de la relation, soit l'on ne dispose que d'une mesure instantanée de la solution. Le premier cas induit une approche prédictive, le second cas une approche déductive.

Dans l'approche prédictive, la relation mathématique non-linéaire entre les délais AV et VV, et le signal SV de sortie, doit être connu explicitement. Les nouvelles valeurs (AV, VV) se déduiront de la mesure de SV en inversant la relation précédente. Cela pose deux problèmes: la relation non-linéaire, $SV = F_{NL}(AV, VV)$, est dépendante du patient et très difficilement explicitable, sa résolution fait partie des « problèmes inverses » rarement soluble. En outre, ces relations évoluent aussi au cours du temps avec l'état de santé du patient.

Dans le cas de l'approche déductive, l'évolution du SV suite au changement des délais AV et VV est observée. La précédente modification des délais sera conservée si la valeur de SV augmente. Au contraire, la dernière modification sera corrigée si la valeur de SV diminue. Les corrections se font continuellement après analyse de l'évolution du système, une telle approche permet donc d'envisager un apprentissage «online». Les principaux avantages d'une telle approche font: que l'on n'a pas besoin de connaître explicitement F_{NL} , que l'optimalité subsiste même si la relation évalue lentement au cours du temps, que le calcul différentiel intensif n'est pas nécessaire et enfin que l'algorithme RL s'applique à la même architecture analogique de SNN1 conçue pour l'algorithme HL [3]. Ainsi, la consommation d'énergie ainsi que la surface de silicium du dispositif CRT adaptatif peuvent être drastiquement réduites.

Basé sur les avantages ci-dessus, nous choisissons l'approche connexionniste pour notre étude actuelle.

V.2.2 Description explicite de l'algorithme RL

Pour expliquer l'algorithme d'apprentissage de renforcement (RL) de manière simple, on suppose que F_{NL} est une simple formule quadratique formant une surface en « cloche » conformément aux références [1] [2]:

$$SV = SV_{\max} - \alpha (AV - AV_{opt})^2 - \beta (VV - VV_{opt})^2 \quad (V.2.1)$$

où SV_{\max} est le maximum global de la surface SV . Le point (AV_{opt}, VV_{opt}) est le point pour lequel $SV = SV_{\max}$ et constitue le point de coordonnées AV et VV de valeurs optimales. Les coefficients α et β permettent de contrôler la « forme » de la surface. Dans le cas réel, tous ces coefficients ci-dessus sont inconnus et souvent variables, ce qui justifie notre choix d'approche connexionniste. Pour simplifier les écritures, nous définissons:

$$\begin{cases} \Delta AV = AV - AV_{opt} \\ \Delta VV = VV - VV_{opt} \end{cases} \quad (V.2.2)$$

$$SV = SV_{\max} - \alpha (\Delta AV)^2 - \beta (\Delta VV)^2 \quad (V.2.3)$$

L'algorithme évolue au rythme des battements cardiaques. A chaque battement, un seul délai est modifié. Chaque valeur d'un délai sera donc indexée par un indice: « m » pour AV et « n » pour VV . Après deux battements cardiaques, les indices seront respectivement « m + 1 » et « n + 1 ». Les relations pour AV deviennent:

$$\Delta AV_m = AV_m - AV_{opt} \quad (V.2.4)$$

$$\Delta AV_{m+1} = AV_{m+1} - AV_{opt} \quad (V.2.5)$$

$$\Rightarrow \frac{dAV}{dm} = AV_{m+1} - AV_m = \frac{d\Delta AV}{dm} \quad (V.2.6)$$

$$\Rightarrow AV_{m+1} - AV_m = \Delta AV_{m+1} - \Delta AV_m \quad (V.2.7)$$

De la même manière, nous avons l'expression de VV :

$$VV_{n+1} - VV_n = \Delta VV_{n+1} - \Delta VV_n \quad (V.2.8)$$

Comme AV et VV changent alternativement et afin d'enlever toutes ambiguïtés, nous déclarons les indices de SV en fonction des pas antérieurs:

$$\frac{\partial SV}{\partial m} = SV_{m,n} - SV_{m-1,n} \quad (V.2.9)$$

Chapitre V

L'équation ci-dessus montre l'impact de la dernière modification de délai AV sur SV lorsque le délai VV reste inchangé. Dans l'algorithme RL, pour un compromis entre rapidité et précision, le pas de variation des délais AV et VV est fixé à $R_m = R_n = 5\text{ms}$. Ce pas de variation est identique à celui de l'algorithme HL [3] en mode Non-adaptatif. La variation du délai AV est indiquée ci-dessous:

$$AV_m - AV_{m-1} = \pm R_m \quad (\text{V.2.10})$$

La modification du délai AV opérée par l'algorithme RL est donnée par:

$$\begin{cases} AV_{m+1} = AV_m + \text{sign} \cdot R_m \\ \text{sign} = \text{sign}[SV_{m,n-1} - SV_{m-1,n-1}] \end{cases} \quad (\text{V.2.11})$$

$$\text{si } \left. \frac{\partial SV}{\partial AV} \right|_m > 0, \Rightarrow R_{m+1} = R_m, \text{ sinon, } R_{m+1} = -R_m \quad (\text{V.2.12})$$

La variation du délai VV s'opère de manière similaire:

$$\begin{cases} VV_{n+1} = VV_n + \text{sign} \cdot R_n \\ \text{sign} = \text{sign}[SV_{m,n} - SV_{m,n-1}] \end{cases} \quad (\text{V.2.13})$$

Nous définissons un indice temporel k indiquant la séquence des battements cardiaques (Hb : Heart Beat). L'évolution du SV sera donnée pour chaque battement cardiaque par:

$$\frac{\partial SV}{\partial k} = \begin{cases} -\alpha(\Delta AV_{k+1}^2 - \Delta AV_k^2) \text{ changement de délai AV} \\ -\beta(\Delta VV_{k+1}^2 - \Delta VV_k^2) \text{ changement de délai VV} \end{cases} \quad (\text{V.2.14})$$

Conformément aux connaissances médicales actuelles sur une courbe en « cloche », les coefficients α et β ont des valeurs positives. La convergence de l'algorithme vers le maximum peut être analysée avec le délai AV:

$$\begin{aligned} \text{Si } \frac{\partial SV}{\partial k} > 0 \\ \Rightarrow SV_{k+1} > SV_k \\ \Rightarrow \Delta AV_{k+1}^2 - \Delta AV_k^2 < 0 \\ \Rightarrow \Delta AV_k^2 > \Delta AV_{k+1}^2 > 0 \end{aligned} \quad (\text{V.2.15})$$

Quel que soit l'écart $\Delta AV_k = AV_k - AV_{\text{opt}}$, l'algorithme RL réduit l'écart entre AV_{k+1} et AV_{opt} . En d'autres termes, la projection du couple (AV, VV) sur la surface de SV tendra asymptotiquement vers SV_{max} . Cet algorithme RL peut être décrit par une matrice "historique"

de SV mesuré après chaque variation des délais AV ou VV. L'évolution de SV en fonction des délais est détaillée dans le Tableau V-2.

Tableau V-2 Evolution de SV selon l'algorithme RL

SV de sortie	Pas principal	Petit pas		Conditions	Hb (k)
		<i>m</i>	<i>n</i>		
SV(0,0)	0	0	0	Initialisation de SV(0,0)	0
SV(1,0)	1*	1	0	Premier pas aléatoire d'AV	1
SV(1,1)		1	1	Premier pas aléatoire de VV	2
SV(2,1)	2	2	1	$\frac{\partial SV}{\partial m} = SV(1,0) - SV(0,0)$	3
SV(2,2)		2	2	$\frac{\partial SV}{\partial n} = SV(1,1) - SV(1,0)$	4
SV(3,2)	3	3	2	$\frac{\partial SV}{\partial m} = SV(2,1) - SV(1,1)$	5
SV(3,3)		3	3	$\frac{\partial SV}{\partial n} = SV(2,2) - SV(2,1)$	6

* Avec le démarrage aléatoire, la performance initiale pourrait ne pas être favorable

Chaque pas principal (sauf pour le pas initial 0) comporte deux petits pas individuels modifiant AV ou VV respectivement. Chaque petit pas est traité dans un battement cardiaque. La direction des premiers petits pas sont aléatoires afin de générer une première variation de SV. Au cours des pas suivants, les variations des délais AV ou VV dépendront de l'évolution antérieure de SV. Par exemple, la variation durant le petit pas $(m, n) = (1, 1)$ à $(2, 1)$ dépend de la dernière modification du délai AV durant le petit pas de $(0, 0)$ à $(1, 0)$. Dans chaque pas principal, les délais AV et VV sont changés alternativement. Si une modification permet d'améliorer SV mesuré, alors « $\text{sign} \cdot R_m$ » aura une valeur positive, sinon il sera négatif. Ainsi, nous avons besoin de mémoriser les derniers effets des délais AV et VV sur SV mesuré pour la prochaine modification des délais. De cette manière, des délais AV et VV seront modifiés pour atteindre la valeur maximale de SV. Les poids synaptiques des neurones seront modifiés en conséquence pour provoquer un déclenchement en sortie des neurones. Ainsi, à chaque période cardiaque, le SNN1 enverra les nouveaux délais AV et VV au cœur pour maximiser le SV. Pour simplifier l'architecture analogique de SNN1, l'évolution des poids synaptiques reprendra de la même implémentation analogique que pour l'algorithme HL.

V.3 Simulation des algorithmes HL et RL sous MATLAB

Afin de valider les performances des algorithmes HL et RL, ils ont été tout d'abord implémentés sous MATLAB®, puis sous VHDL-AMS. La Figure V-2 illustre la progression de la projection du couple (AV, VV) à la fois dans le mode Non-adaptatif et dans le mode Adaptatif-déecté sur la surface SV de forme de « cloche ». La valeur maximale possible de SV (i.e. SV_{max}) est fixée à 1000 pour la clarté. On peut y observer la convergence du point SV vers le maximum. Le point de départ du mode Non-adaptatif est aléatoire pour la première fois afin de démarrer le processus d'apprentissage. Après cela, un point de départ de sécurité sera choisi par le clinicien en fonction de la condition physiologique de chaque patient. Ce point sera utilisé comme le point de redémarrage du mode Non-adaptatif pour la sécurité globale. La distance entre les points de départ et le point des délais AV et VV par défaut permet d'estimer le temps nécessaire pour accomplir le mode Non-adaptatif. L'algorithme HL peut avancer de $4 \cdot \Delta T$ durant chaque cycle cardiaque (CC) dans le mode « Loin » ou de $1 \cdot \Delta T$ dans le mode « Proche ». Dans le pire des cas (distance de départ-par défaut = 195 ms), il faut 12 CC pour arriver aux délais AV et VV prédéfini par défaut. Si nous prenons en compte le temps de validation du contrôleur général, le mode Non-adaptatif peut être accompli en moins de 30 CC. Comme on peut le remarquer dans Figure V-2, la progression vers les valeurs cliniques par défaut sous le contrôle de l'algorithme HL est rapide.

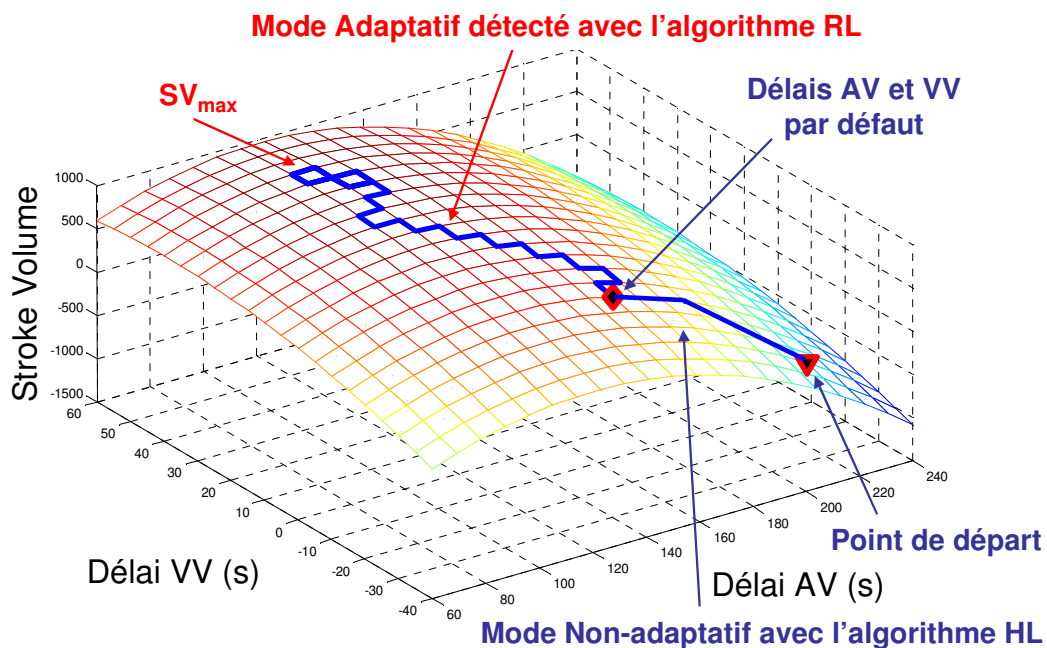


Figure V-2 Simulation de l'algorithme HL et RL avec une surface en forme de « Cloche »

Après avoir atteint ces délais et obtenu la confirmation du contrôleur général, le processeur neuronal bascule en mode Adaptatif-déTECTÉ et exécute l'algorithme RL qui va piloter le couple (AV, VV) pour emmener SV vers son maximum. Le temps de convergence du mode Adaptatif-déTECTÉ dépend de la distance entre le point des délais AV et VV par défaut et le point SV_{max} . Cette distance dépend de l'état physique des patients et des réglages au cours de la dernière visite médicale. La première étape de ce mode commence par un déplacement du point (AV, VV) dans une direction aléatoire (pas sûr d'être la direction optimale) pour initier une première série de variations de SV en fonction du couple (AV, VV). Dans ce mode, les délais AV et VV changent alternativement un battement sur deux. Leurs déplacements cherchent à faire progresser SV vers le point culminant de la courbe : SV_{max} . Le mécanisme de recherche du maximum est une recherche à l'aveugle. De ce fait, le point SV ne peut rester figer sur le point SV_{max} s'il l'atteint. Même si le point SV_{max} est atteint, l'algorithme continuera à chercher si un autre maxima existe dans le voisinage, ce qui aura pour conséquence que le point SV décrit comme un 8 centré sur SV_{max} . Ce comportement « hiératique » autour de SV_{max} peut être vu comme une limitation. Dans notre cas nous le voyons comme un avantage car si la courbe évolue, le point SV_{max} se déplacera. Dans ce cas notre algorithme en perpétuelle quête d'un meilleur maxima, poursuivra le déplacement de SV_{max} pour s'y repositionner. L'algorithme RL suivra toute variation de SV_{max} s'il n'y a qu'un seul maximum et que la vitesse de variation de SV_{max} est inférieure à la vitesse de convergence de l'algorithme RL.

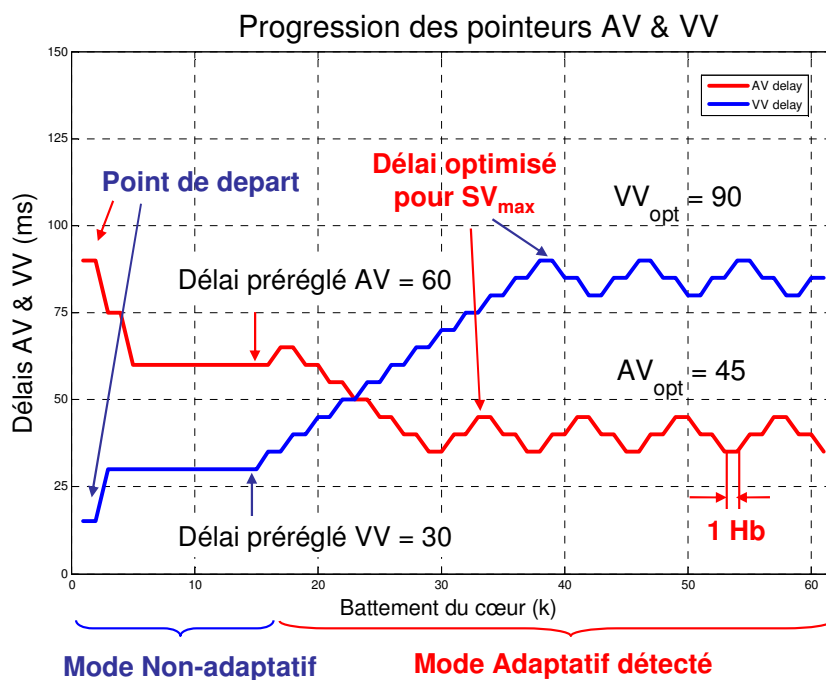


Figure V-3 Progression des pointeurs AV et VV avec l'algorithme HL et RL

Chapitre V

La Figure V-3 illustre la progression du couple (AV, VV) pour un rythme cardiaque de 60 Hb/min (soit 1Hb/s). Nous pouvons y voir la progression des délais dans les deux modes de fonctionnement. Dans le mode Non-adaptatif, le couple de délais va apprendre à reproduire les délais préconisés par les cliniciens. Le maintien de ce délai permet au contrôleur général de s'assurer de l'apprentissage afin d'autoriser le basculement en mode Adaptatif-déecté. Dans ce second mode, les délais AV et VV convergent vers leurs valeurs optimales. Les changements alternatifs des délais AV et VV avec la tendance d'une oscillation autour des délais optimisés peuvent être clairement observés.

Conclusion

Dans ce chapitre V, nous avons abordé les deux algorithmes utilisés par le réseau SNN1 dans ces deux premiers modes de fonctionnement. Le processeur neuronal commence toujours dans le mode Non-adaptatif, où le réseau SNN1 utilise l'algorithme d'apprentissage Hebb (HL) afin d'apprendre à reproduire des délais AV et VV égaux aux délais préconisés par les cliniciens. L'algorithme HL modifie les poids synaptiques pour laisser apparaître en entrée du neurone I&F une séquence de 4 impulsions consécutives dont la dernière va provoquer le déclenchement T_i . La localisation des synapses se base sur la mesure de la distance entre le T^i produit avec ces poids synaptiques et la cible P^i , $i \in \{AV, VV\}$. Une fois que cet apprentissage est approuvé par le contrôleur général, celui-ci bascule en mode Adaptatif-déecté avec l'algorithme d'apprentissage par renforcement (RL). Le processeur neuronal reçoit alors le signal SV des capteurs «hémodynamiques» qu'il transmet au réseau SNN1. Ce dernier sous le contrôle de l'algorithme RL ajuste les délais AV & VV pour chaque battement cardiaque afin de maximiser le signal SV. Afin de suivre la variation de la surface SV, l'algorithme RL va changer alternativement les délais AV/VV et guider en permanence le SNN1 à la recherche du maximum de SV. Cette approche permet au réseau SNN1 de rester vigilant à toute transformation de la relation $SV = f_{NL}(AV, VV)$ et toujours proposer des délais maximisant SV, sous l'hypothèse d'un maxima unique.

Références

- [1] R. Rom, J. Erel, M. Glikson, K. Rosenblum, R. Ginosar, D. L. Hayes, “Adaptive cardiac resynchronization therapy device: a simulation report”, *Pacing and Clinical Electrophysiology*, vol. 28, no. 11, pp. 1168–1173, Nov. 2005.
- [2] R. Rom, et al. “Adaptive cardiac resynchronization therapy device based on spiking neurons architecture”, *IEEE Transaction on Neural Network*, vol. 18, no. 2, pp. 542–550, Mar. 2007.
- [3] Q. Sun, F. Schwartz, J. Michel, R. Rom, “Implantation study of an analog spiking neural network in an auto-adaptive pacemaker”, in *Proc. Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference (NEWCAS-TAISA’08)*, Montréal, Canada, Jun. 22-25, 2008, pp. 41–44.
- [4] D. O. Hebb, *The organization of Behavior*. New York: Wiley, 1949.
- [5] A. H. Klopff, “A neuronal model of classical conditioning”, *Psychobiology*, vol. 16, no. 2, pp. 85–125, Jun. 1988.
- [6] Z. I. Whinnett et al., “Haemodynamic effects of changes in atrioventricular and interventricular delay in cardiac resynchronisation therapy show a consistent pattern: analysis of shape, magnitude and relative importance of atrioventricular and interventricular delay”, *Heart journal*, vol. 92, no. 11, pp. 1628–1634, Nov. 2006.

Chapitre VI

Simulation du système complet

Sommaire du chapitre VI

CHAPITRE VI	SIMULATION DU SYSTEME COMPLET	110
VI.1	OBJECTIFS ET ATTENTES DE LA SIMULATION DU SYSTEME COMPLET	112
VI.2	MODELES SIMPLES DU CŒUR AVEC LE SCENARIO D'EXERCICE	113
VI.2.1	<i>Scénario d'exercice</i>	<i>114</i>
VI.3	MODELES COMPORTEMENTAUX DU CŒUR (SURFACES SV)	115
VI.3.1	<i>Surface 0</i>	<i>115</i>
VI.3.2	<i>Surface 1</i>	<i>116</i>
VI.3.3	<i>Surface 2</i>	<i>117</i>
VI.4	MODELE DU BRUIT INTEGRE DANS LA SIMULATION GLOBALE	119
VI.5	SIMULATIONS GLOBALES DU SYSTEME AVEC DIFFERENTS MODELES DU CŒUR SOUS VHDL-AMS	121
VI.5.1	<i>Surface 0</i>	<i>121</i>
VI.5.2	<i>Surface 1</i>	<i>124</i>
VI.5.3	<i>Surface 2</i>	<i>127</i>
	<i>Conclusion.....</i>	<i>131</i>
	<i>Références</i>	<i>132</i>

VI.1 Objectifs et attentes de la simulation du système complet

La simulation du système complet répond à plusieurs objectifs et attentes.

Le premier objectif est de vérifier puis valider, selon la méthode PVF, le comportement cohérent du couple cœur – CRT-P. La collaboration a pu vérifier d'une part que les divers modèles du cœur donnaient des résultats compatibles avec les objectifs de pilotage par le pacemaker et d'autre part que le CRT-P rendait des excitations qui, selon les modèles du cœur, amenaient ce dernier à maximiser son volume d'éjection SV. Cette part de vérifications et de validations a donc implicitement validé la plateforme de simulation construite autour du langage VHDL-AMS. Ce fut alors le point de départ pour chaque équipe d'entrer dans leurs domaines respectifs, sachant que lors des réunions projet, les résultats pouvaient être compris de tous et donner lieu à des échanges de modèles. Ainsi nous avons pu proposer des modèles simplifiés du cœur qui ne décrivaient que les relations entre le stroke volume SV et les délais AV et VV.

Le second objectif fut de construire un partitionnement du dispositif CRT-P. Ce partitionnement a permis de séparer les domaines de compétences techniques comme la partie stockage énergétique (la batterie), les interfaces radio fréquences, les parties numériques et analogiques, etc. Ce partitionnement par secteur a aussi permis d'établir les connexions physiques d'échanges entre les parties et assurer une cohérence en vue de la construction de l'architecture finale du CRT-P.

La première attente est bien évidemment de pouvoir analyser en détails l'impact des choix techniques concernant le processeur neuronal sur le fonctionnement du cœur. C'est dans cet esprit que la collaboration a remis en question l'hypothèse du Dr. Whinnett et que de nouvelles relations $SV = F_{NL}(AV, VV)$ ont été proposées. Ces nouvelles relations ont par ailleurs démontré l'intérêt d'une approche neuronale car elle a su s'adapter à ces changements sans modification architectural ou algorithmique. Ces analyses sont abordées dans ce chapitre.

La seconde attente, après l'acceptation des choix techniques, fut de pouvoir investiguer plus en détails les limites des grandeurs électriques et algorithmiques du processeur neuronal. Cela nous a permis d'extraire des spécifications par partie, puis par module et enfin par bloc

Chapitre VI

élémentaire. Ces spécifications ont ensuite permis d'affiner les choix architecturaux des blocs électroniques. Cette étude de conception sera développée dans le chapitre suivant.

VI.2 Modèles simples du cœur avec le scénario d'exercice

Selon une publication du Dr. Whinnett [1] et les observations cliniques du projet Adapter [2], la valeur des délais optimisés varie avec la fréquence cardiaque (en anglais : heart rate = HR). Les changements des valeurs optimales sont particulièrement prononcés lorsque la fréquence cardiaque augmente. Pour étudier ce phénomène, nous introduisons un scénario de modification de la fréquence cardiaque pour simuler l'évolution du rythme cardiaque d'un patient virtuel entre une situation de repos et une situation d'exercice physique. Le maximum de la surface SV dépend maintenant des délais AV et VV et du HR.

Suite aux observations citées ci-dessus, nous avons pu constater qu'une surface $SV = F_{NL}(AV, VV, HR)$ pouvait comporter des maxima locaux fixes et un maximum local variable en fonction du rythme cardiaque HR. Pour analyser plusieurs situations, la collaboration a introduit trois surfaces d'études qui résument divers cas cliniques observés. La surface que nous appellerons « **Surface 0** » ne comporte qu'un seul maximum local, c'est donc aussi le maximum global. Si ce maximum est fixe on se retrouve dans une situation équivalente à celle qui a servi de base d'analyse dans les chapitres précédents. Pour refléter les nouvelles observations décrites précédemment, ce maximum local et global est rendu variable en fonction de HR. Les coordonnées de ce maximum local variable sont appelées AV_{MLV} et VV_{MLV} et sont définies par les relations suivantes [2]:

$$\begin{aligned} AV_{MLV} &= 160 - 1.1 \cdot (HR - 60) & 60\text{BPM} < HR < 180\text{BPM (Beat Per Minute)} \\ VV_{MLV} &= 0.3325 \cdot (HR - 60) - 40 & 60\text{BPM} < HR < 180\text{BPM} \end{aligned} \quad (\text{VI.2.1})$$

Soulignons que dans les trois surfaces, le maximum local variable sera toujours le même, c'est à dire celui décrit ci-dessus par l'équation (VI.2.1).

La surface que nous appellerons « **Surface 1** » comporte un maximum local fixe et un maximum local variable. Nous donnons une plus grande importance au maximum local variable ce qui fait qu'il restera le maximum global de la surface. L'objectif visé est de tester si le processeur neuronal peut éviter le piègeage sur un maximum local.

La surface que nous appellerons « **Surface 2** » comporte deux maximums locaux fixes et un maximum local variable. Ici nous ne donnons plus la plus grande importance au maximum local variable. L'un des deux maxima locaux fixes sera dominant en amplitude. Il portera le maximum global tant que le maximum local variable en reste éloigné. Lors du trajet suivi par le maximum local variable, celui-ci se rapproche du maximum local fixe dominant et la résultante (somme des contributions) va générer un maximum global qui se placera entre les deux maxima locaux. L'objectif visé est similaire à celui de la surface 1. Pour décrire les coordonnées de ce maximum global variable, nous introduirons le couple (AV_{GM}, VV_{GM}) .

VI.2.1 Scénario d'exercice

Pour tester le comportement du processeur neuronal face à des changements de conditions physiologiques du cœur, le scénario d'exercice est conçu pour ressembler à un exercice de relaxation-exercice-relaxation du patient (Figure VI-1).

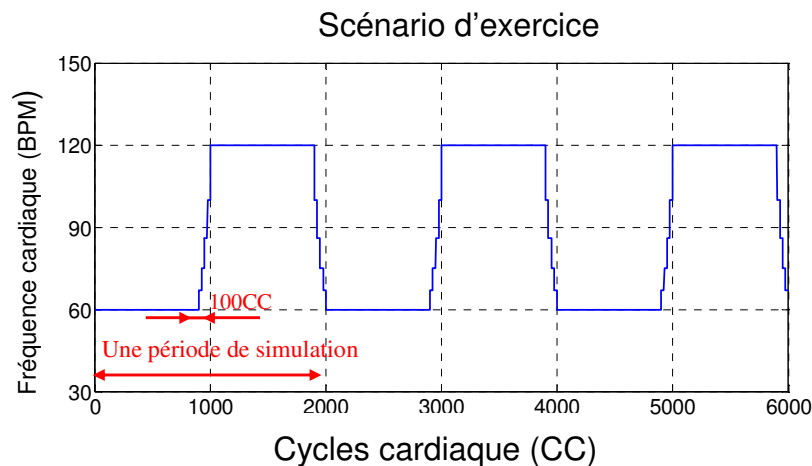


Figure VI-1 Le scénario d'exercice en fonction de HR

Le scénario d'exercice va faire varier le rythme cardiaque de 60 BPM (la situation de relaxation du patient) à 120 BPM (situation d'exercice du patient) et retourner ensuite à la situation de relaxation. La période d'un exercice est de 2000 cycles cardiaques (CC). Dans les premiers 900 CC de chaque période d'exercice, le HR reste à 60 BPM, puis elle augmente tous les 25 CC (Tableau VI-1) jusqu'à atteindre 120 BPM. Après un palier d'exercice physique de 900 CC à 120 BPM, le rythme cardiaque va redescendre à 60 BPM à la même vitesse que la montée.

Chapitre VI

La simulation du couple Cœur – CRT-P comportera 3 périodes d'exercices, soit une simulation totale de 6000 cycles cardiaques.

Tableau VI-1 Evolution des délais AV_{MLV} et VV_{MLV} dans le scénario d'exercice

HR (BPM)	60	67	75	86	100	120
AV_{MLV} (ms)	160	152.3	143.5	131.4	116	94
VV_{MLV} (ms)	-40	-37.6	-35.0	-31.3	-26.7	-20.0

VI.3 Modèles comportementaux du cœur (Surfaces SV)

Pour cette série d'analyses, nous n'avons pas utilisé le modèle le plus complet et détaillé du cœur. Les simulations seraient longues et fourniraient une masse inutile de données pour l'objectif de cette étude. Pour focaliser l'étude sur le couple cœur – CRT-P, des modèles simplifiés du cœur ont été proposés. Cette étude utilise donc ces modèles simplifiés décrivant le lien entre le stroke volume SV et les délais AV et VV préconisés par le processeur neuronal. Notre étude préliminaire, exposée aux chapitres précédents, a été réalisée avec la surface SV en forme de « cloche » et avec un rythme cardiaque constant. Nous avons pu montrer que l'algorithme RL atteint le maximum de la courbe SV et pouvait probablement « suivre » l'évolution de cette surface SV tant qu'elle ne présente qu'un seul maximum. Le premier objectif de ces surfaces est de confirmer que le processeur neuronal peut « suivre » un déplacement de la position du maximum au cours du temps. Toutefois, une surface SV en forme de « cloche » n'étant pas systématique [2], nous avons introduit plusieurs nouvelles surfaces SV (Surface 1 et Surface 2) avec un ou deux maxima locaux afin d'améliorer la réalité de notre étude et démontrer les capacités d'adaptation de la solution neuronale à des situations diverses et évolutives. Afin d'harmoniser les surfaces, il a été décidé de fixer arbitrairement la valeur du maximum de la surface SV (pour toute valeur de HR) à 4000.

VI.3.1 Surface 0

La surface 0 est une surface quadratique, similaire à la courbe en « cloche », mais où le maximum va évoluer en fonction de HR comme illustrée dans la Figure VI-2. La localisation du maximum dans le plan (AV, VV) correspond au point de coordonnée (AV_{MLV} , VV_{MLV}) dont les valeurs vont évoluer avec le rythme cardiaque. Par conséquent, la localisation du

maximum évoluera au cours du temps en corrélation avec le rythme cardiaque HR. Cette surface sert de surface standard pour analyser le processeur et doit valider son aptitude à « suivre » une évolution du maximum au cours du temps. Nous définissons :

$$\Delta AV = AV - AV_{OPT} \quad (\text{VI.3.1})$$

$$\Delta VV = VV - VV_{OPT} \quad (\text{VI.3.2})$$

L'expression de la surface 0 se trouve ci-dessous :

$$SV = 4000 - \frac{2(\Delta AV)^2 + 2(\Delta AV)(\Delta VV) + (\Delta VV)^2}{20} \quad (\text{VI.3.3})$$

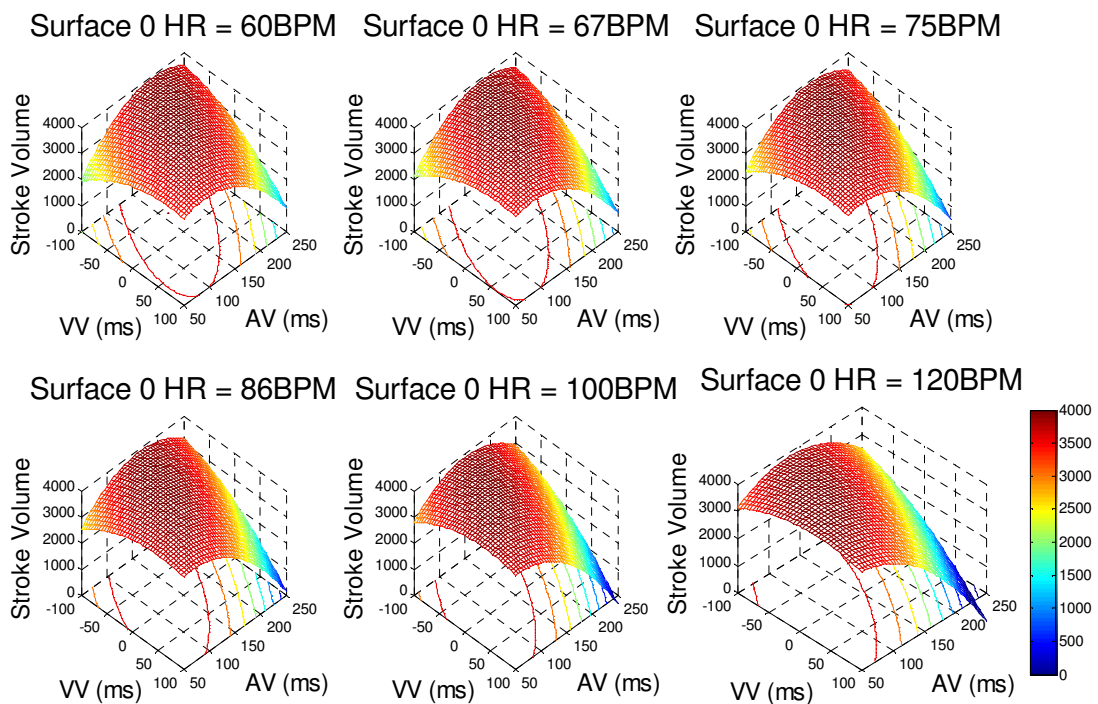


Figure VI-2 Simulations de la surface 0 en fonction de HR

VI.3.2 Surface 1

La surface 1 est présentée Figure VI-3 et doit permettre de vérifier si l'algorithme d'apprentissage par renforcement RL peut atteindre le maximum global de la courbe SV. Au delà de la question de la capacité à « suivre » tout déplacement du maximum, nous voulions tester le processeur face au problème de piégeage sur un maximum local ou dans son voisinage. Pour cela la surface 1 possède un maximum local fixé et un maximum global

Chapitre VI

variable en fonction de HR. Un autre point à voulu être analysé : comment se comporte le SV si la surface présente une forte différence de pente ? Sur la Figure VI-3, on peut observer qu'il y a une région presque plate, conséquence du fait que les deux maxima ont des valeurs assez proches. Cependant lorsque le délai AV est supérieur à 200 ms, la pente varie très fortement. L'expression de la surface 1 se trouve ci-dessous :

$$SV = 1.156 \cdot \left(-1 \cdot e^{-5} \cdot VV^4 + \frac{7}{1500} VV^3 - \frac{197}{250} VV^2 + \frac{288}{5} VV + \frac{9284}{5} - \frac{(\Delta AV + \Delta VV)^2}{20} \right) \quad (\text{VI.3.4})$$

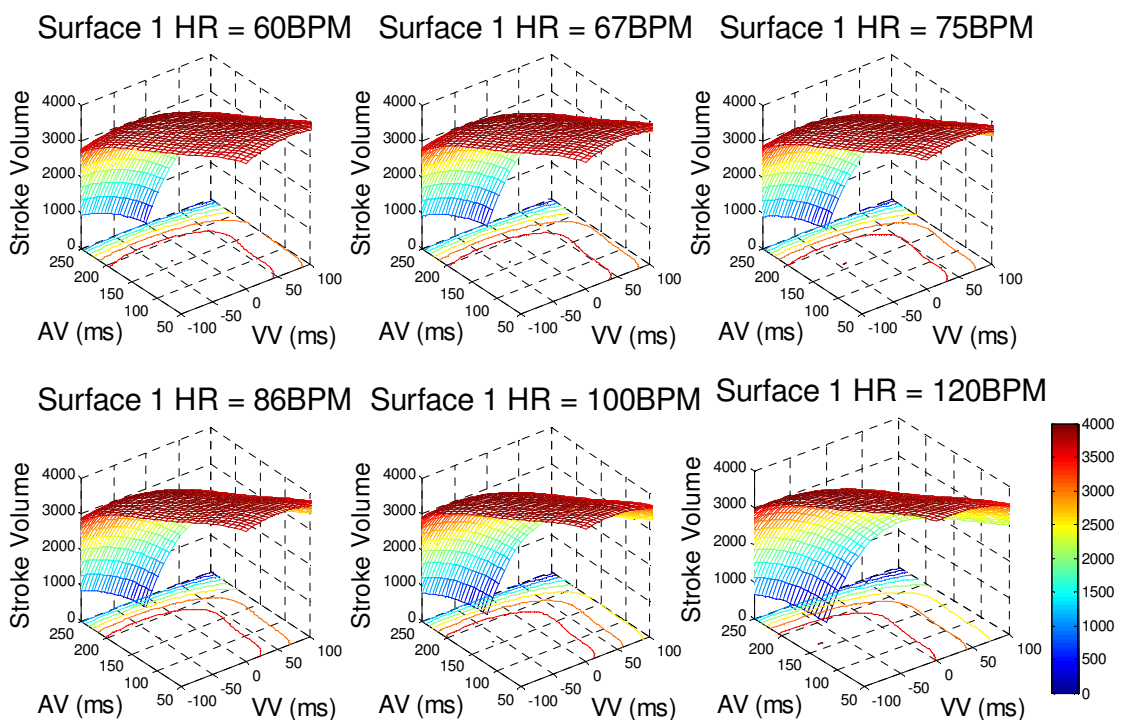


Figure VI-3 Simulations de la surface 1 en fonction de HR

VI.3.3 Surface 2

La surface 2 (Figure VI-4) est la surface la plus complexe utilisée dans les simulations. Elle comporte deux maxima locaux fixes et un maximum local variable. Ce dernier est localisé au point de coordonnées (AV_{MLV}, VV_{MLV}) dont les valeurs respectives dépendent de HR qui lui-même évolue au cours du temps durant le scénario d'exercice. Etant donné que ce maximum local variable est à proximité de l'un des deux maxima locaux fixes et que d'autre part ces deux maxima ont des élévations identiques, le maximum global (appelé SV_{GM}) peut quitter le maximum local fixe pour se placer entre les deux maxima locaux. L'amplitude maximale atteinte par SV n'est pas, ici, de 4000. Car une modulation de SV_{max} en fonction de

HR y est introduite. La valeur maximale de SV sera donnée pour le couple AV_{GM} , $VV_{GM} = (100, -5)$, voir Tableau VI-2. Néanmoins, compte tenu de la grille temporelle (AV et VV évolue au pas de 5ms) la valeur maximale observable en simulation restera inférieure à 4000. La surface 2 peut être considérée comme un cas particulier pour vérifier les performances du processeur neuronal dans des conditions diverses avec ou sans bruit (voir la section suivante). Comme l'expression de la surface 1 est compliquée, nous montrons le modèle de la surface 2 implémenté sous Matlab® [2]. Le modèle est basé sur les délais LV et RV qui ont des relations simples avec des délais AV et VV selon (III.1.1) et (III.1.2). Puisque RA est considéré comme le début d'une période cardiaque, nous pouvons noter simplement : $RV = AV$ et $LV = AV + VV$. Les équations des trois courbes dans le modèle correspondent aux trois maxima sur la surface 2:

```
clear;

LV=linspace(50,250,41);
RV=linspace(50,250,41);
[x,y]=meshgrid(LV,RV);
pA=10; %coefficient de variation

a1=1e-3;    b1=1e-4;    c1=-3e-4;    %paramètres de la courbe 1
x1_0=165;   y1_0=155;    %centre de la courbe 1
a2=5e-3;    b2=3e-3;    c2=1e-4;    %paramètres de la courbe 2
x2_0=210;   y2_0=210;    %centre de la courbe 2
a3=5e-3;    b3=3e-3;    c3=1e-4;    %paramètres de la courbe 3
x3_0=100;   y3_0=100;    %centre de la courbe 3

HR=60; %Fréquence cardiaque (heart rate)
RV0= 160 - 1.1.*(HR-60); %RVOPT en fonction de HR
LV0= 120 - 0.7675.*(HR-60); %LVOPT en fonction de HR

F1=1.0./(1.0+a1*(x-LV0).^2+b1*(y-RV0).^2+c1*(x-LV0).*(y-RV0));
%equation de la courbe 1
F2=1.0./(1.0+a2*(x-x2_0).^2+b2*(y-y2_0).^2+c2*(x-x2_0).*(y-y2_0));
%equation de la courbe 2
F3=1.0./(1.0+a3*(x-x3_0).^2+b3*(y-y3_0).^2+c3*(x-x3_0).*(y-y3_0));
%equation de la courbe 3

A=(LV0-(x1_0-pA))/pA/2.0; %coefficients entre 0 et 1

Surface2=floor((1000.0*(F1+0.35*F2+0.45*F3)*(0.8+0.2*(1.0-A))+100.0)*1.7);
```

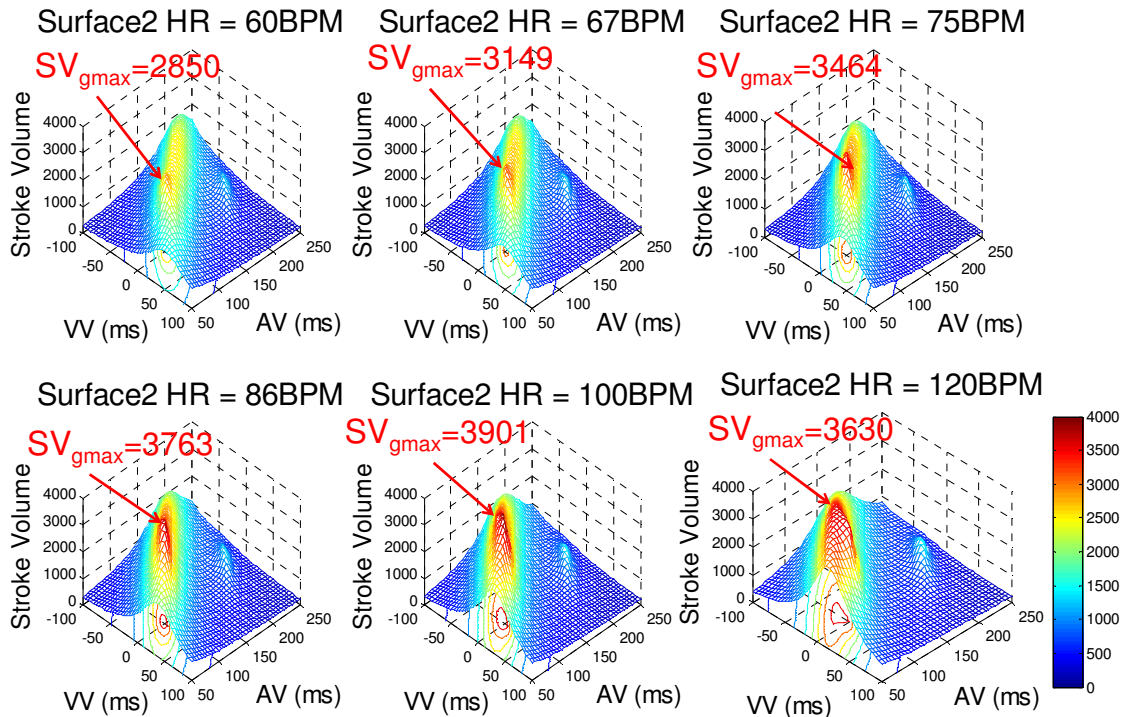


Figure VI-4 Simulations de la surface 2 en fonction de HR

VI.4 Modèle du bruit intégré dans la simulation globale

Les signaux d'entrée du CRT-P servant au pilotage des contractions sont : RA, RV, LV et une mesure de la pression sanguine (BP). Ces mesures proviennent de capteurs implantés dans les parois du cœur. Ces capteurs sont sensibles aux mouvements des tissus (pour RA, RV et LV), c'est à dire soit à l'accélération pour le capteur PEA soit à l'impédance électrique pour le BioZ. Dans tous les cas, les vibrations mécaniques externes au cœur se superposent aux informations utiles. Entre autre, le mouvement respiratoire des poumons introduit des signaux additionnels totalement asynchrones par rapport au mouvement du cœur. Ces contributions annexes peuvent être considérées comme du bruit ajouté aux signaux « utiles » du cœur. Le CRT-P devra donc travailler sur la base de signaux acquis avec du bruit additionnel nécessitant sa prise en compte dans la simulation du système complet. A défaut d'information sur la distribution statistique des amplitudes du bruit, nous introduirons dans les simulations un bruit ayant une distribution statistique uniforme. Nous supposons le processus d'introduction du bruit comme additif. Le signal reçu par le processeur neuronal sera donc le signal « mesuré » $SV^M = SV^P \pm \text{Bruit}$ où SV^P est le signal sans bruit résultant de la projection

du point (AV, VV) proposé par le processeur sur la surface SV du cœur : le volume d'éjection systolique projetée résulte du calcul $SV^P = F_{NL}(T_{AV}, T_{VV}, HR)$. La Figure VI-5 présente notre modèle d'introduction du bruit.

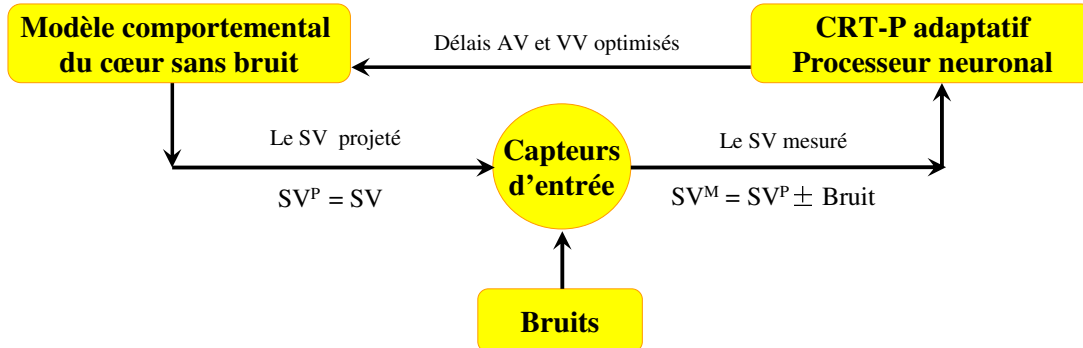
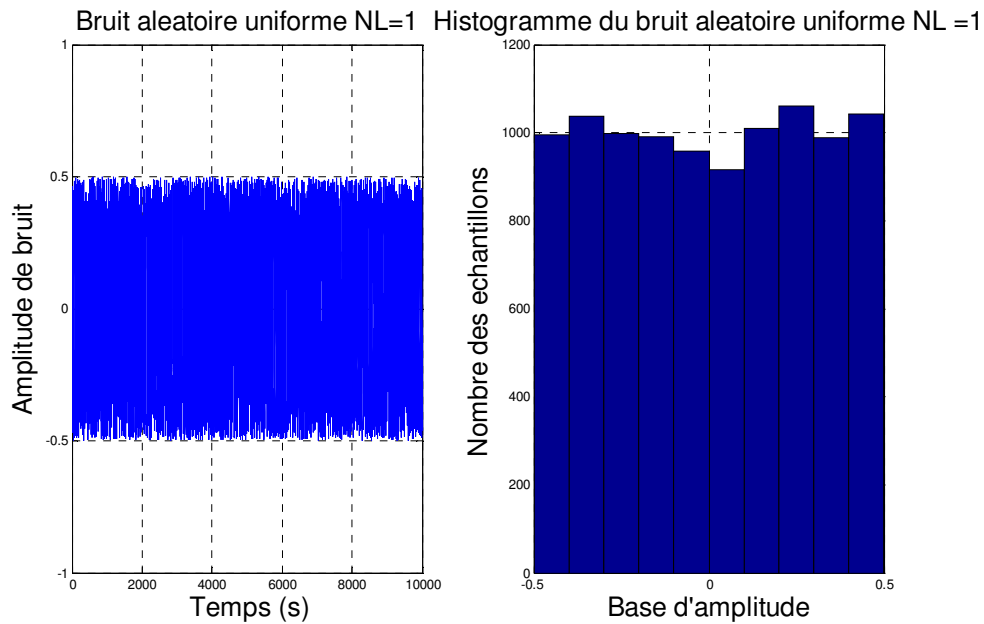


Figure VI-5 Intégration de modèle de bruit dans la simulation globale

Notre signal de bruit est construit sur la base d'un générateur de bruit aléatoire borné sur $[0, 1]$. Nous ramenons sa valeur moyenne à zéro par: $\text{Bruit} = (\text{aléatoire} - 1/2)$. Dans la mesure où le signal SV varie de 0 à 4000, nous allons moduler l'amplitude de notre bruit et ainsi contrôler le rapport signal à bruit. L'équation de notre bruit modulé en amplitude est donnée par : $\text{Bruit} = (\text{aléatoire} - 1/2) * NL$. La grandeur NL est donc directement l'amplitude du signal de bruit. Cette routine de génération de bruit a été développée en VHDL-AMS et une rapide vérification des caractéristiques de notre générateur est présentée pour le cas $NL = 1$ Figure VI-6. Pour chaque modèle comportemental du cœur (surface SV), nous simulons l'ensemble du système avec quatre niveaux de bruit différents, $NL = 0$ (sans bruit), 100, 300 et 500.

Rappelons que l'amplitude maximale du volume d'éjection est $SV_{\max} = 4000$. La mesure du volume d'éjection en fonction du couple (AV, VV) appliqué au cœur donne la valeur SV^P nécessairement inférieure ou égale à SV_{\max} . Le processus additif du bruit conduit donc le signal SV^M à dépasser la valeur maximale SV_{\max} . Le rapport signal à bruit en entrée du processeur neuronal sera donc donné par : $SNR = (SV^P)^2 / (\sigma)^2$. Dans le meilleur des cas à NL donné, ce rapport sera de $SNR_{\max} = (SV_{\max})^2 / (\sigma)^2$. L'écart type du bruit uniformément distribué est donné par : $\sigma^2 = NL^2 / 12$. Avec un niveau de bruit $NL = 500$, on aura $(\sigma) = 144,34$ et par suite $SNR_{\max} = 767,97$, soit un rapport signal de 28,85 dB. Si le signal SV^P diminue, le rapport signal sur bruit réel diminuera aussi, mais il faudra considérer l'évolution de SV^P au cours du temps pour pouvoir évaluer le rapport réel.



VI.5 Simulations globales du système avec différents modèles du cœur sous VHDL-AMS

VI.5.1 Surface 0

Les simulations relatives à la surface 0 sont présentées dans les figures VI-7 à VI-11. Dans la Figure VI-7, nous pouvons observer, le signal SV mesuré (SV^M) sans bruit qui est donc identique au signal SV projeté (SV^P). Sur cette figure on peut distinguer l'évolution du signal SV depuis le démarrage du processeur neuronal ; c'est à dire une situation typique qui surviendrait après un retour forcé par le contrôleur générale au mode de sécurité qui est le mode non- adapté. Dans ce mode non adapté, le processeur neuronal apprend à reproduire les délais AV et VV par défaut et ses prédictions sont ignorées. Le cœur reçoit durant cette période les valeurs préconisées par les cliniciens (P^{AV} et P^{VV}). La Figure VI-8 présente un zoom sur les 200 premiers CC. Nous pouvons constater que le mode Non-adaptatif est achevé en moins de 30 CC et que le processeur bascule alors sur le mode adapté – détecté. Les premiers pas de l'algorithme RL dans ce mode sont aléatoires et non forcément optimaux comme indiqué dans le chapitre précédent. On peut également y observer le comportement de l'algorithme autour du point maximum (SV_{max}). La valeur de SV n'est jamais constante car l'algorithme continue

de scruter le voisinage de ce maximum. De fait, sa valeur ne dépassera jamais la valeur maximale possible de SV (4000) tout en restant proche expliquant les ondulations de SV.

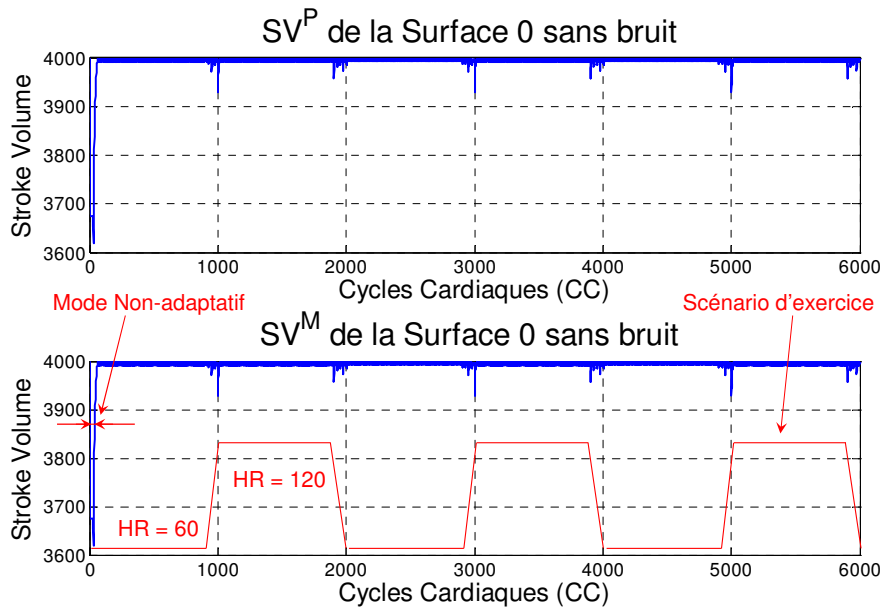


Figure VI-7 Simulation de la surface 0 sans bruit (NL0)

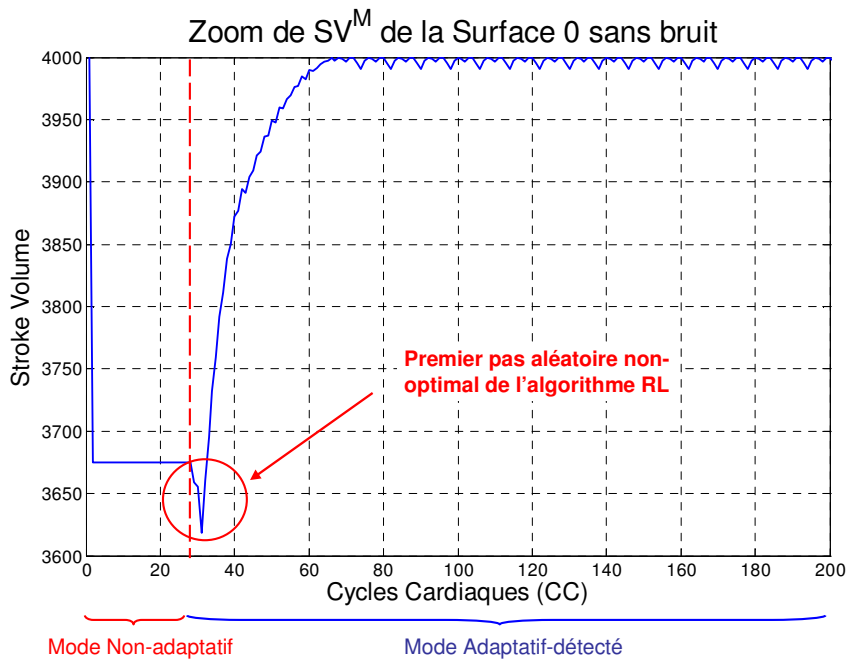


Figure VI-8 Zoom sur le signal SV^M de la surface 0 sans bruit

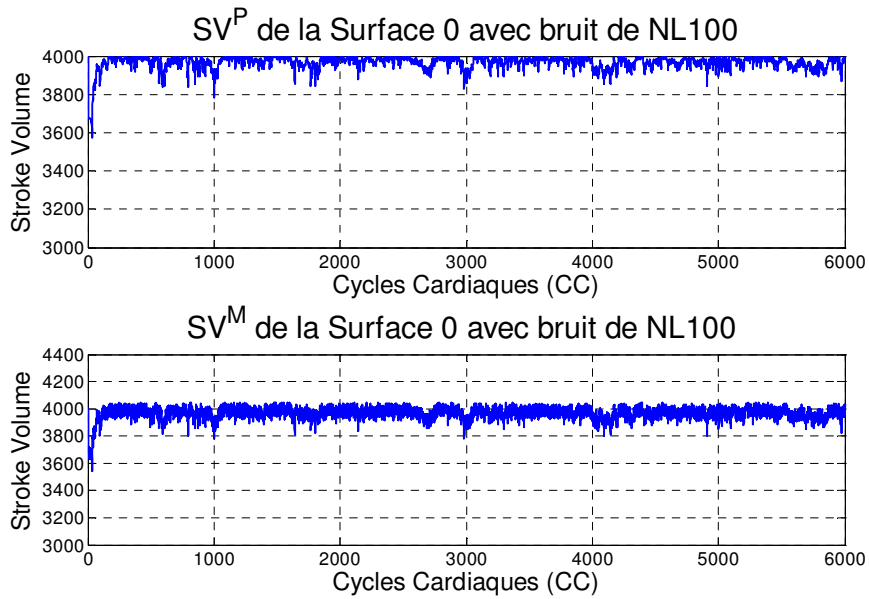


Figure VI-9 Simulation de la surface 0 avec le bruit de NL100

La Figure VI-9 présente la simulation de la surface 0 avec l'introduction d'un bruit d'amplitude $NL = 100$. Comme on a déjà pu le voir sur la Figure VI-8, les changements du rythme cardiaque entraînent une diminution du volume d'éjection SV. Cela est dû au fait que le lieu du maximum s'éloigne du point (AV, VV) courant. Il faut donc un certain temps à l'algorithme RL pour rejoindre le nouveau maximum. L'introduction du bruit relativise cette chute de SV.

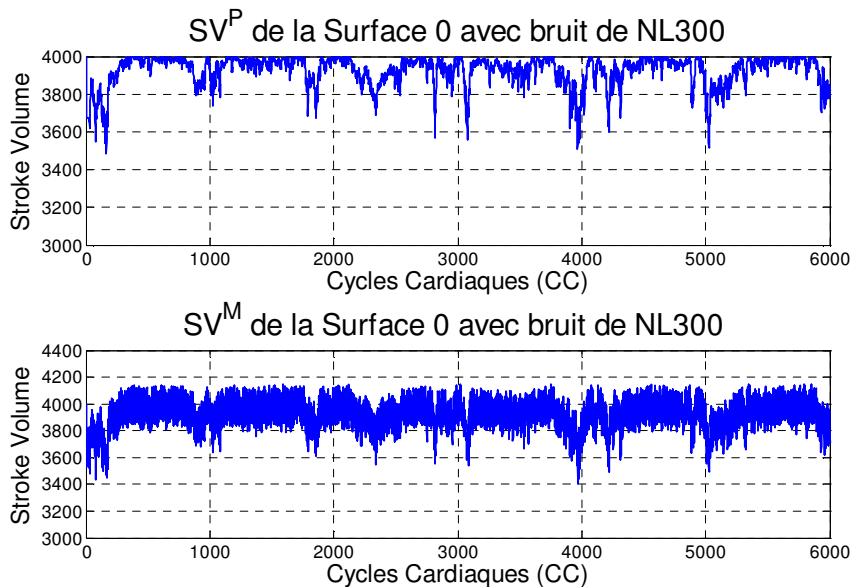


Figure VI-10. Simulation de la surface 0 avec le bruit de NL300

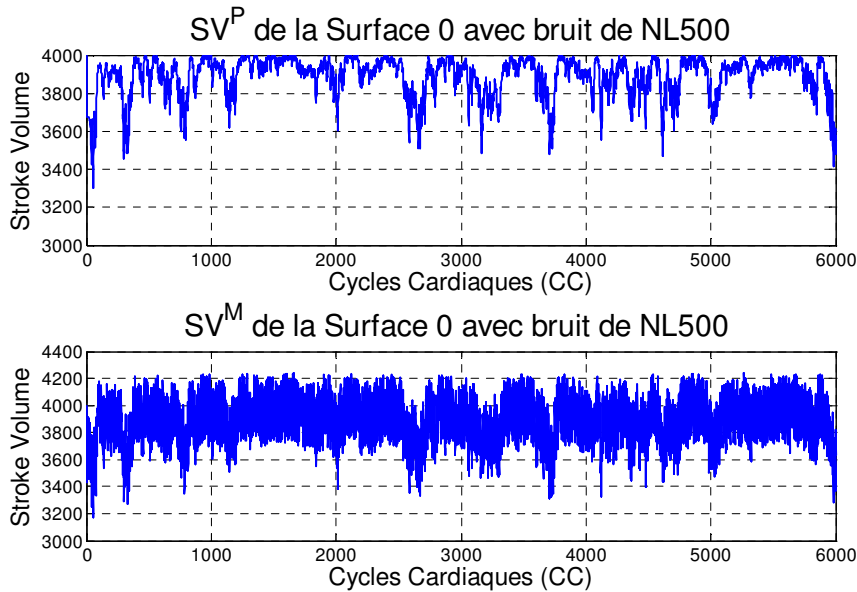


Figure VI-11. Simulation de la surface 0 avec le bruit de NL500

Les figures VI-10 et VI-11 présentent respectivement les situations en présence de bruit d'amplitudes respectives $NL = 300$ et $NL = 500$. Avec l'augmentation des niveaux de bruit, le signal SV mesuré (SV^M) sera plus perturbé et par conséquent les prévisions du processeur plus fausses. On observera une perte dans le volume éjecté avec des valeurs observées d'amplitude du signal SV^P de 3400. Cependant, la valeur moyenne de SV reste élevée démontrant que le processeur a un comportement stable et même stabilisant. L'idée de calculer une valeur moyenne de SV sur une fenêtre temporelle afin de corriger le taux d'apprentissage de RL doit permettre de prendre en compte le bruit dans la mesure où ses variations sont plus rapides qu'un changement physiologique. Cette idée a conduit à regarder les apprentissages de type Q-Learning. Cette alternative n'est pas développée dans cette thèse est fait partie du futur du projet.

VI.5.2 Surface 1

Les simulations relatives à la surface 1 sont présentées dans les figures VI-12 à VI-15. La surface 1 possède une région plate et une région à forte pente. En l'absence de bruit (Figure VI-12), on observe moins d'agitation sur SV durant les moments d'exercices physiques ($HR = 120BPM$). Dans ces moments, le maximum global est proche du maximum local et l'algorithme RL, toujours à la recherche d'un meilleur maximum, va élargir son terrain d'exploration entraînant de plus grandes variations de SV. L'influence d'une pente

Chapitre VI

raide apparaît en fin du mode non – adapté. On entre dans le mode adapté – détecté en partant de la localisation des valeurs préconisée P^{AV} et P^{VV} . Si celles-ci sont proches du maximum local fixe et dans la pente raide, l’algorithme RL va guider le point de fonctionnement sur ce maximum local sans laisser entrevoir le second maximum. Dans la Figure VI-13, nous pouvons observer qu’au début de la simulation, les délais AV et VV par défaut sont définis proches du maximum local et le processeur neuronal va le reconnaître comme l’unique maximum. Le volume SV reste au voisinage de ce point jusqu’au prochain changement de HR (passage d’un état de relaxation à un état d’exercice du patient ou l’inverse). Lors du changement, le maximum local variable va s’approcher du maximum fixe et l’algorithme peut « l’apercevoir ». L’ajout du bruit va entrainer le point (AV, VV) à élargir la zone de recherche ce qui peut conduire l’algorithme à « apercevoir » le maximum global et y glisser. A contrario, si le niveau de bruit augmente, des trajets aller-retours entre deux maxima peuvent s’installer. La probabilité de l’existence de ces trajets dépendra de la profondeur de la vallée par rapport au niveau de bruit.

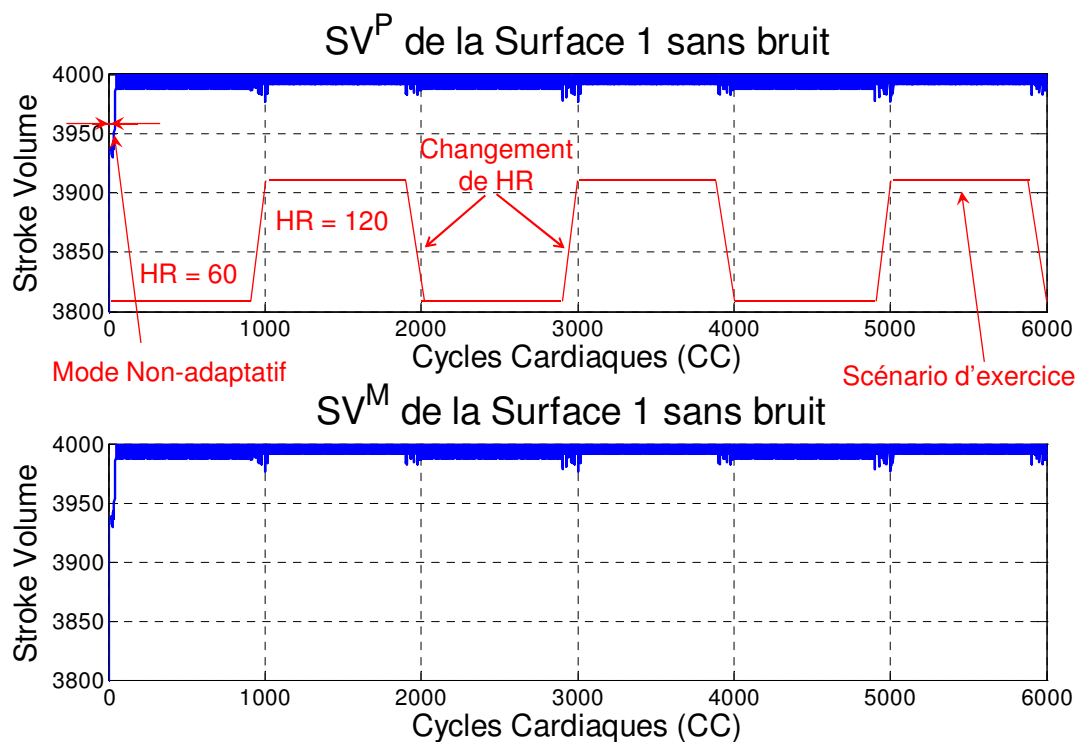


Figure VI-12. Simulation de la surface 1 sans bruit

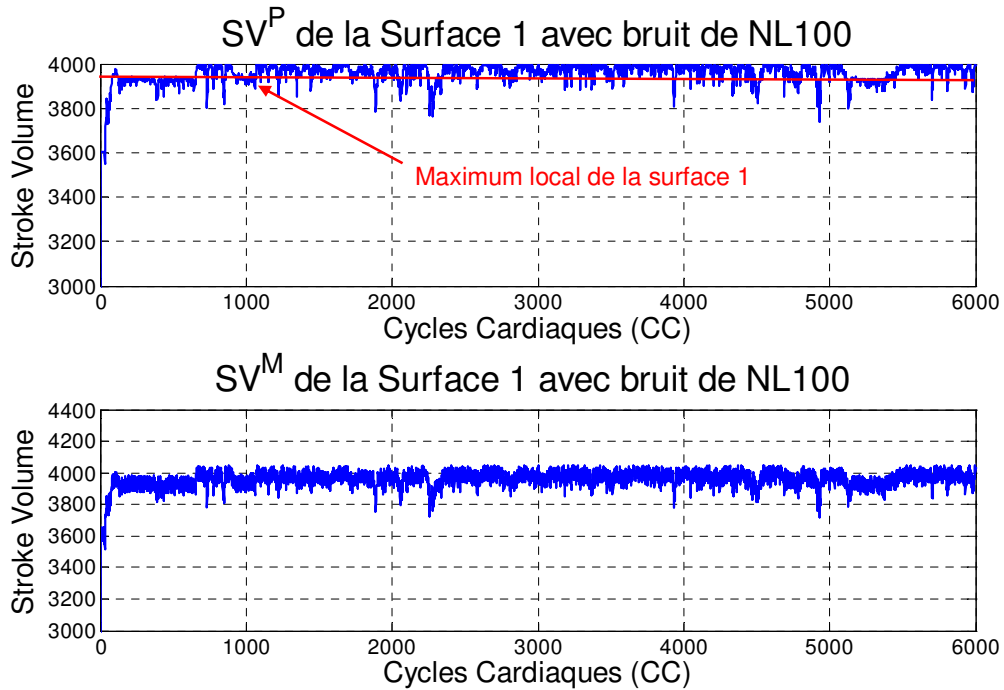


Figure VI-13. Simulation de la surface 1 avec le bruit de NL100

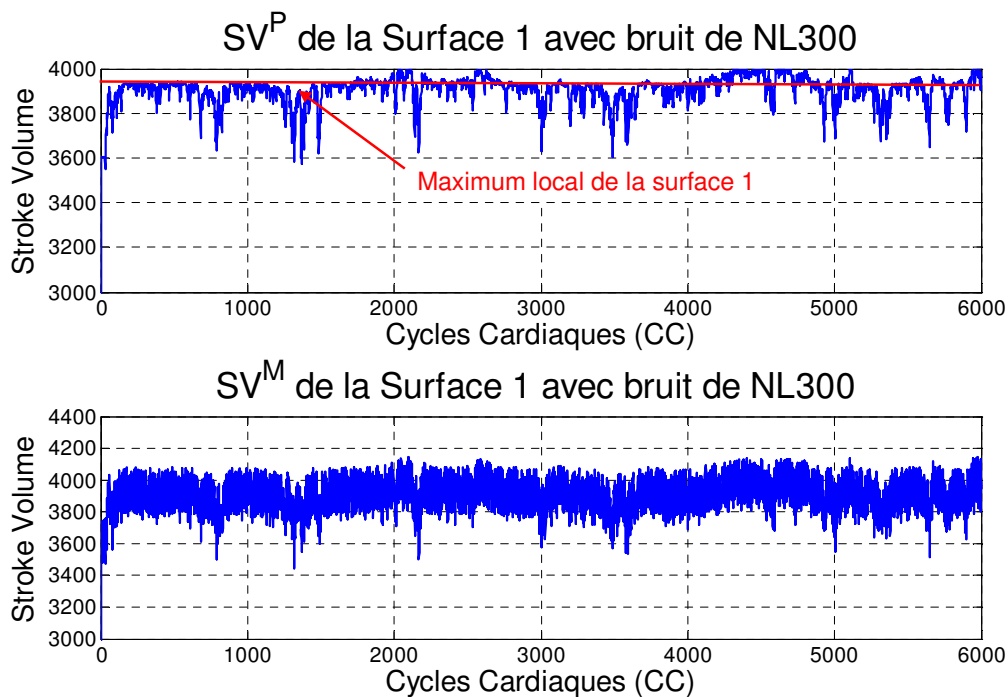


Figure VI-14. Simulation de la surface 1 avec le bruit de NL300

On remarquera aussi que le piégeage sur le maximum local est plus probable durant les phases de relaxations (0 à 1000, 2000 à 3000, 4000 à 5000) car les maxima sont proches.

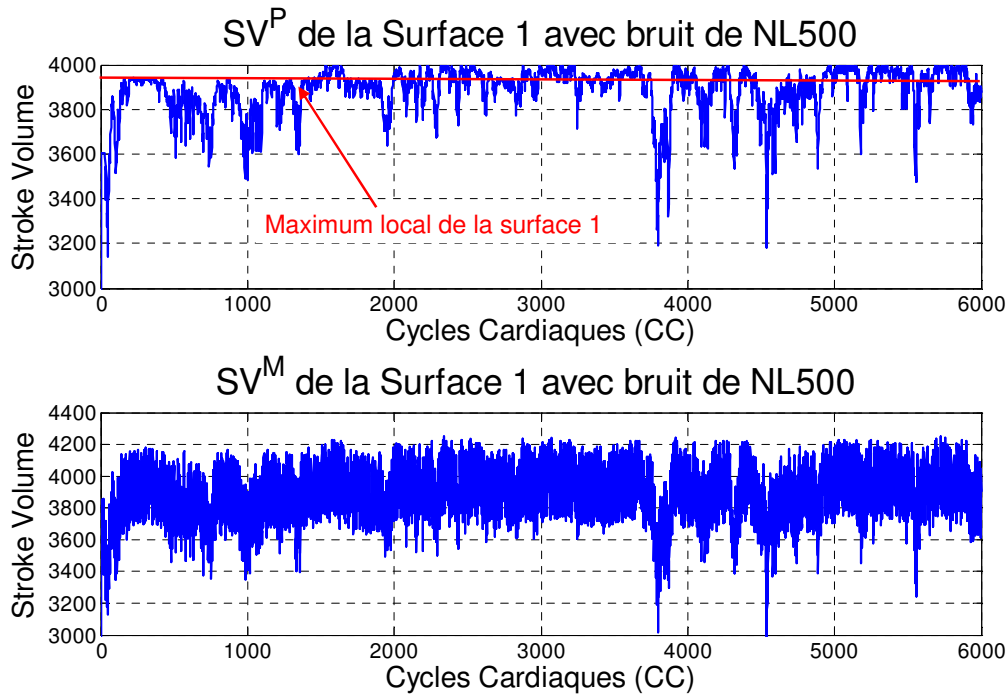


Figure VI-15. Simulation de la surface 1 avec le bruit de NL500

VI.5.3 Surface 2

Les simulations relatives à la surface 2 sont présentées dans les figures VI-16 à VI-19. Cette surface combine plusieurs difficultés et s'en trouve donc être la plus complexe à analyser. Elle comporte trois maxima (deux maxima locaux fixes et un maximum local variable en fonction de HR). En outre, le point de coordonnées (AV_{MLV}, VV_{MLV}) dont les valeurs sont fonction de HR ne porte pas le maximum global. Les coordonnées de ce maximum global (AV_{GM}, VV_{GM}) se situent d'abord aux coordonnées de l'un des maxima locaux fixes dont la coordonnée est $(AV_{MLF1}=100, VV_{MLF1}=0)$. Lorsque le maximum local variable (AV_{MLV}, VV_{MLV}) évolue, il se rapproche du maximum local fixe précédent et le point associé au maximum global se déplace entre les deux maxima locaux. Le Tableau VI-2 retrace l'évolution des coordonnées (AV_{GM}, VV_{GM}) du maximum global et les coordonnées du maximum local variable (AV_{MLV}, VV_{MLV}) en fonction de HR. Lorsque le HR est égal à 60 BPM, le maximum global prend la valeur $SV_{gmax} = 2850$ et se localise aux coordonnées $(AV_{MLF1}=100, VV_{MLF1}=0)$. Lorsque le HR est égal à 100 BPM, le maximum global prend la valeur $SV_{gmax} = 3901$ et se localise aux coordonnées $(AV_{GM}=100, VV_{GM}= -5)$. C'est à cette

valeur de rythme cardiaque que le volume d'éjection maximum SV_{gmax} sera le plus proche de 4000.

Tableau VI-2 Evolution de SV_{gmax} en fonction de HR dans la simulation de la surface 2

HR (BPM)	60	67	75	86	100	120
SV_{gmax}	2850	3149	3464	3763	3901	3630
AV_{MLV} (ms)	160	152.3	143.5	131.4	116	94
VV_{MLV} (ms)	-40	-37.6	-35.0	-31.3	-26.7	-20.0
AV_{GM} (ms)	100	100	100	100	100	100
VV_{GM} (ms)	0	0	0	0	-5	-20

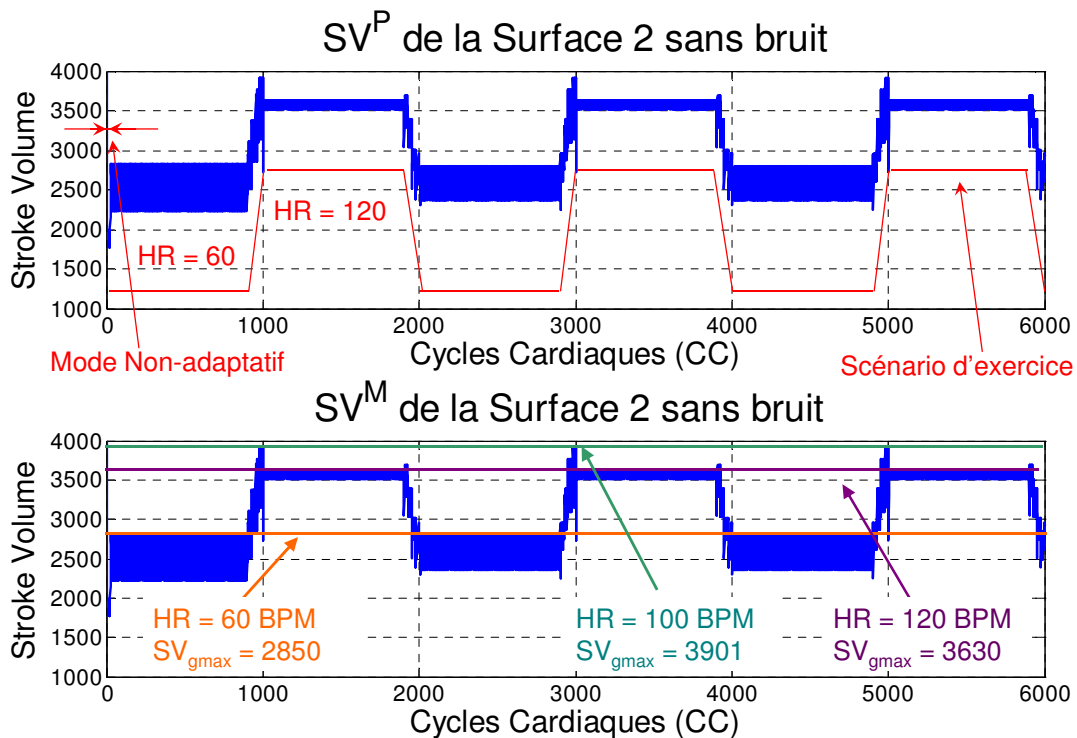


Figure VI-16. Simulation de la surface 2 sans bruit

La simulation de la surface 2 sans le bruit est illustrée à la Figure VI-16. Nous pouvons y faire plusieurs observations. La première observation est que la valeur SV_{gmax} atteint sa valeur maximale lorsque $HR = 100$ BPM durant les accélérations cardiaques. La seconde est que l'on observe clairement les changements de rythme cardiaque car les valeurs de SV_{gmax} , aux deux valeurs extrêmes de HR, sont notablement différentes et découlent directement de la forme de la surface 2. La troisième observation est que les variations d'amplitudes de SV sont plus importantes lorsque $HR = 60$ BPM que dans le cas de $HR = 120$ BPM. Ceci est lié au fait, qu'à ce rythme cardiaque ($HR = 60$ BPM), le point SV se localise au voisinage du maximum local fixe où les pentes de la surface sont raides (voir Figure VI-4).

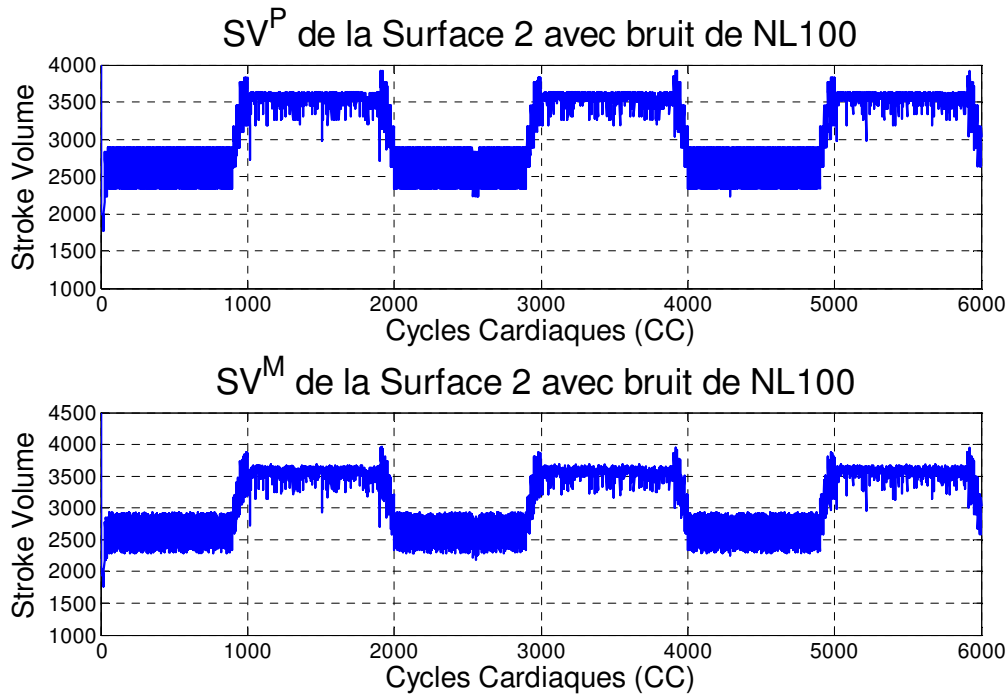


Figure VI-17. Simulation de la surface 2 avec le bruit de NL100

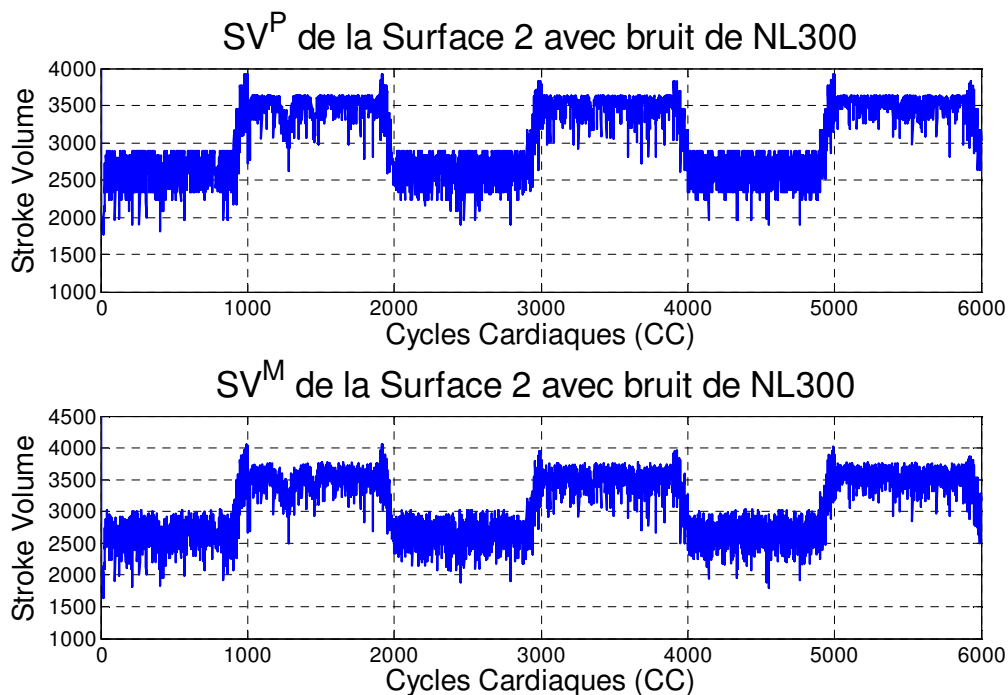


Figure VI-18. Simulation de la surface 2 avec le bruit de NL300

Avec la croissance du bruit ($NL = 100$ et $NL = 300$), Figure VI-17 et Figure VI-18 respectivement, nous observons plus de perturbations sur le signal SV mesuré ainsi que sur le signal SV projeté, mais ceux-ci restent comparables aux cas sans bruit. On peut donc en déduire que le processeur neuronal reste parfaitement fonctionnel en présence de bruit et même sur une surface aussi complexe, et « chahuté » que la surface 2. On remarquera que le

signal SV atteint sa valeur maximale pour HR = 100 à chaque changement de rythme cardiaque, tant en monté qu'en descente.

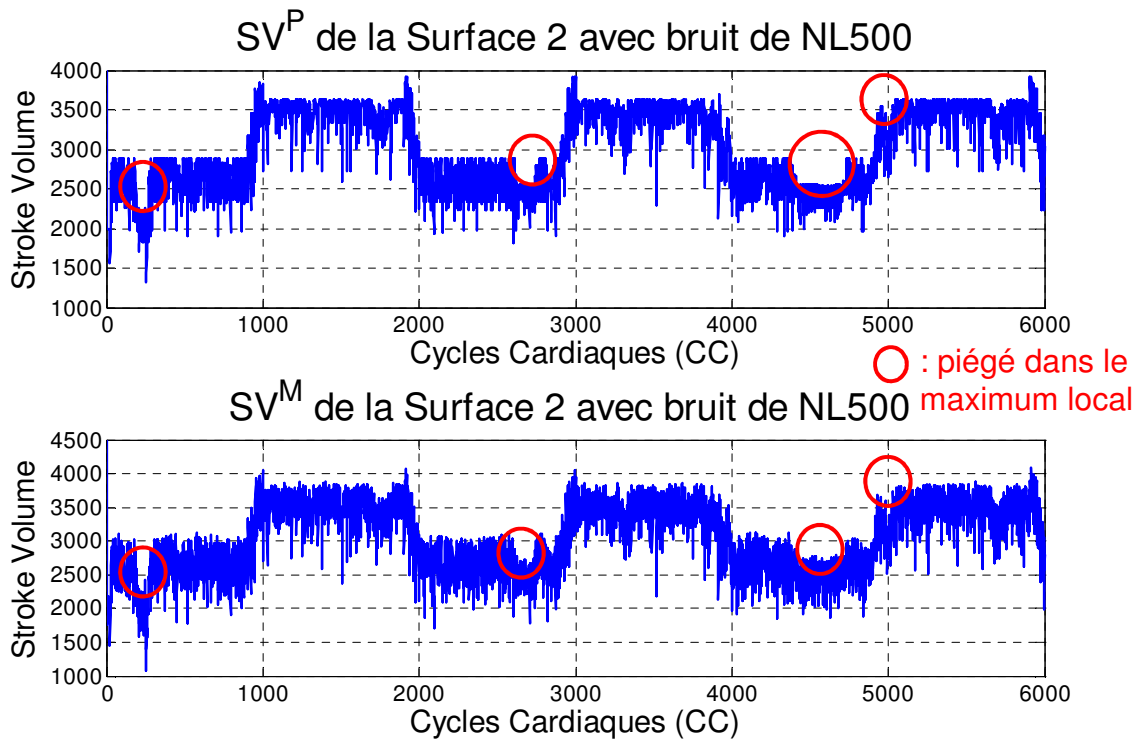


Figure VI-19. Simulation de la surface 2 avec le bruit de NL500

La simulation (Figure VI-19) avec un niveau de bruit élevé : NL500, montre des cas de piégeages sur des maximaux locaux. Les cercles rouges montrent les instants où le processeur neuronal est piégé dans un maximum local. D'autre part, au voisinage des cycles cardiaques 4000 CC, la valeur de SV n'atteint pas sa valeur maximale de 3901. Ceci vient du fait que cette phase, où HR vaut 100 BPM, ne dure que 25 CC. Or il vient, juste avant le changement de HR, de se faire piéger dans un maximum local et ne peut corriger les poids pour passer par le vrai maximum.

Conclusion

Dans ce chapitre VI, nous avons effectué un ensemble assez large de simulations du système complet (cœur et CRT-P) afin de tester, d'analyser et de valider les fonctionnalités du réseau SNN1 qui est donc le sous bloc principal du processeur neuronal. La validation du réseau SNN1 valide à la fois l'architecture analogique et les deux algorithmes associés (apprentissage de Hebb et par renforcement). Pour conforter les partenaires du projet les simulations ont pris en compte des réponses complexes du cœur à la sollicitation par des délais AV et VV proposées par le processeur neuronal. Un scénario d'exercice a été proposé afin de prendre en compte des variations de battement du cœur qui entraîne des variations de la surface de réponse du cœur à la mesure du volume éjecté SV. Pour améliorer le réalisme du système, nous avons ajouté un bruit uniforme de différents niveaux sur les signaux des capteurs d'entrée.

Selon les résultats des simulations du système complet, notre processeur neuronal analogique répond à l'ensemble des attentes fonctionnelles et apporte une réponse validée au problème de maximisation du volume éjecté SV. Les temps de correction des poids synaptiques le rendent apte à travailler en temps réel.

Ces simulations vont maintenant pouvoir prendre une nouvelle orientation. Si la fonctionnalité est avérée, il nous reste à analyser les conditions aux limites. Quelles sont les latitudes dont dispose le concepteur pour réaliser son circuit électronique ? Cette question revient à s'interroger sur les spécifications techniques à imposer au concepteur. C'est l'objectif du chapitre suivant.

Références

- [1] Z. I. Whinnett et al., “Haemodynamic effects of changes in atrioventricular and interventricular delay in cardiac resynchronisation therapy show a consistent pattern: analysis of shape, magnitude and relative importance of atrioventricular and interventricular delay”, *Heart journal*, vol. 92, no. 11, pp. 1628–1634, Nov. 2006.
- [2] R. Rom, “Adaptive CRT - Guidelines and Scenarios” *Internal report of the ADAPTER project*, Mar. 2009.

Chapitre VII

Les études sur le multiplieur

analogique dans le SNN1

Sommaire du chapitre VII

CHAPITRE VII	LES ETUDES SUR LE MULTIPLIEUR ANALOGIQUE DANS LE SNN1	134
VII.1	NOTRE FLOT DE CONCEPTION EN APPROCHE TOP-DOWN	137
<i>VII.1.1</i>	<i>Comment extraire les spécifications techniques de chaque « composant » système?</i>	<i>138</i>
<i>VII.1.2</i>	<i>Quelle importance accorder à un modèle théorique de niveau N si nous ne connaissons pas la structure du « composant » (la topologie du circuit) ?</i>	<i>142</i>
<i>VII.1.3</i>	<i>Comment choisir la meilleure topologie électrique pour un « composant » système?</i>	<i>143</i>
VII.2	DEVELOPPEMENT DE NOTRE APPROCHE CONCEPTION AU NIVEAU « COMPOSANT »	145
VII.3	MODELES THEORIQUES DU MULTIPLIEUR ANALOGIQUE	148
VII.4	SELECTION DES TOPOLOGIES SELON L’ETUDE BIBLIOGRAPHIQUE	154
VII.5	BILAN DE PUISSANCE DU RESEAU DE NEURONES ANALOGIQUES (SNN1)	158
	<i>Conclusion</i>	<i>164</i>
	<i>Références</i>	<i>166</i>

Ce chapitre s'intéresse à l'étape 3 de notre méthodologie de conception décrite au chapitre II paragraphe 4.2. Cette étape doit préparer le passage de l'analyse système vers l'analyse domaine (et métier). Ce passage est un passage difficile et délicat dans le contexte industriel du fait des contraintes « Time To Market » et « First Time Success ». Nous avons donc décomposé autant l'étape 3 que l'articulation entre les étapes 3 et 4 en sous objectifs méthodologiques.

Le premier objectif est d'illustrer la mise en application de notre approche de conception depuis le plus haut niveau d'abstraction du système jusqu'aux vues structurelles des divers « composants » du système. Nous formalisons ainsi la démarche qui a conduit aux résultats présentés au chapitre VI. La finalité de cette approche est double : la première est de pouvoir confirmer que le partitionnement du système en accord avec une fonctionnalité globale et la seconde est d'extraire les spécifications pour chaque « composants ». Avec cette dernière finalité, le concepteur système peut établir le cahier des charges de chaque « composants » et le transmettre au concepteur domaine. Dans notre cas le domaine est l'électronique et l'illustration choisie porte sur le « composant » multiplieur analogique de chaque synapse.

Le second objectif méthodologique est d'assister au mieux la première opération du concepteur dans l'étape 4 qui est la sélection de topologies. Sachant que le dimensionnement d'un circuit sous entend de connaître sa topologie, tous les outils métiers en CAO électronique spécifiques à l'étape 4 (voir II.4.2) ne démarrent qu'avec la saisie du schéma. En l'absence de méthode ad hoc, le concepteur doit faire appelle à une forte heuristique, c'est à dire que ce passage est délicat pour un concepteur peu expérimenté. Or notre objectif de recherche est d'aider ces derniers pour permettre aux entreprises d'améliorer leur capacité d'innovation. En ce sens, nous avons donc repositionné cette opération dans notre flot de conception pour essayer de la rationaliser autant que peut aujourd'hui. Nous n'aborderons pas dans cette thèse le dimensionnement et la génération des layouts, car ces étapes peuvent s'appuyer sur des outils d'aides existants même si des améliorations peuvent leurs être apportées.

VII.1 Notre flot de conception en approche Top-Down

En suivant la méthodologie de conception basée sur le Prototypage Virtuel Fonctionnel introduite dans le chapitre II, le système formé par le couple Cœur – CRT-P du projet ADAPTER a été décomposé en plusieurs entités allant des sous systèmes aux composants. La priorité durant cette décomposition est l'adéquation fonctionnelle des entités avec la fonctionnalité générale. Une des difficultés est que cette fonctionnalité générale est rarement totalement explicitable. C'est pourquoi l'approche PVF nous semble incontournable.

Le partitionnement du système, nous permet d'identifier chaque entité avec un modèle structurel comportemental décrivant sous VHDL-AMS la fonctionnalité de l'entité avec un haut niveau d'abstraction. Nous montrons dans la Figure VII-1 une fraction de la décomposition du système. Elle illustre la décomposition du réseau de neurones SNN1 au sein du processeur neuronal dans notre système de CRT-P. Dans les chapitres III à V, nous avons détaillé et argumenté le partitionnement du système complet, puis du réseau de neurones (SNN1) et enfin des algorithmes utilisés par SNN1. Dans le contexte multidisciplinaire du projet, l'utilisation de la plateforme commune basée elle aussi sur le langage VHDL-AMS a permis de valider notre décomposition et d'aboutir à l'étape industrielle fondamentale du « GO / NO GO ».

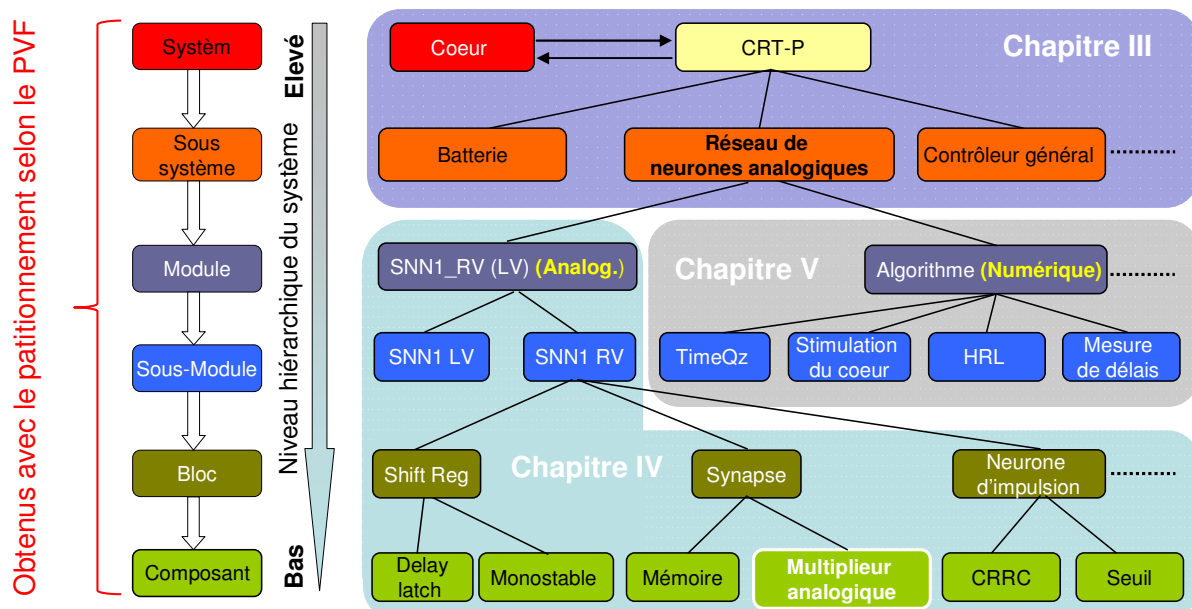


Figure VII-1 Modèles structurels comportementaux du réseau de neurones analogiques

Après la décision de poursuivre les études, débute le travail des concepteurs métiers. Le concepteur en électronique reçoit la liste des « composants » (composants au sens du système

évidement) à réaliser avec leurs spécifications. Les étapes classiques qui suivent sont donc : la sélection d'une topologie – le dimensionnement de cette topologie – la génération du dessin des masques du « composants ».

On constatera que cette articulation entre l'approche système et l'approche circuit s'effectue avec le passage des spécifications et une sélection de topologie. Ce passage est souvent extrêmement arbitraire et souvent fondée sur des heuristiques : c'est l'expertise du concepteur de circuit qui solutionne les dilemmes. Aussi pour conceptualiser cette étape et offrir des pistes à des non experts, nous avons cherché à répondre aux diverses questions qui suivent.

VII.1.1 Comment extraire les spécifications techniques de chaque « composant » système?

Pour répondre à cette question, il faut connaître la structuration des approches typiques des développements de projets industriels. Au début du projet on dispose d'une idée. Celle-ci sans lien métier est analysée avec des outils généralistes comme Matlab[®], Scilab ou Mathcad, etc. La description du système est purement mathématique c'est à dire avec un très haut niveau d'abstraction physique et technique. Le premier travail des concepteurs (systèmes et métiers) est de transposer l'idée et la problématique sur une plateforme d'analyse commune. Pour notre projet le langage VHDL-AMS est un excellent candidat du fait de ses capacités multi domaines (restriction aux domaines à flux conservatifs). C'est un point essentiel pour tirer le maximum de profit de l'approche PVF. Suivant cette approche, le partitionnement du système conduit aux interfaces métiers avec la définition des modèles structurels comportementaux des « composants » systèmes. Pour progresser vers les limites fonctionnelles, il faut maintenant introduire progressivement les limitations et contraintes induites par les choix technologiques. Pour cela une hiérarchisation des modèles comportementaux des « composants » est nécessaire. C'est le second avantage offert par le langage VHDL-AMS. Chaque entité (ici les « composants ») peut être décrite par des modèles différents (on parle d'architecture dans la norme VHDL). Ces modèles peuvent contenir des descriptions plus ou moins raffinées au sens de la physique. Le troisième avantage du langage VHDL-AMS est de pouvoir, en plus, associer à la fois des descriptions fonctionnelles (les architectures comportementales) et les descriptions structurelles (les

Chapitre VII

architectures structurelles). Ces dernières descriptions permettent d'établir un lien avec l'outil de validation ultime des concepteurs de circuits électroniques : l'outil SPICE. Le processus de conception que nous venons de décrire est illustré dans la Figure VII-2. L'utilisation du langage VHDL-AMS autorise les itérations entre les différents niveaux hiérarchiques (système vers « composants ») et ensuite entre les différents niveaux d'abstractions (architecture comportemental vers architecture structurelle ou netlist SPICE). Le point fort de cette approche est l'homogénéité des simulations faites du système. Elle permet de collaborer entre les partenaires, de valider les principes et de guider les phases de conception jusqu'au plus près de la technologie.

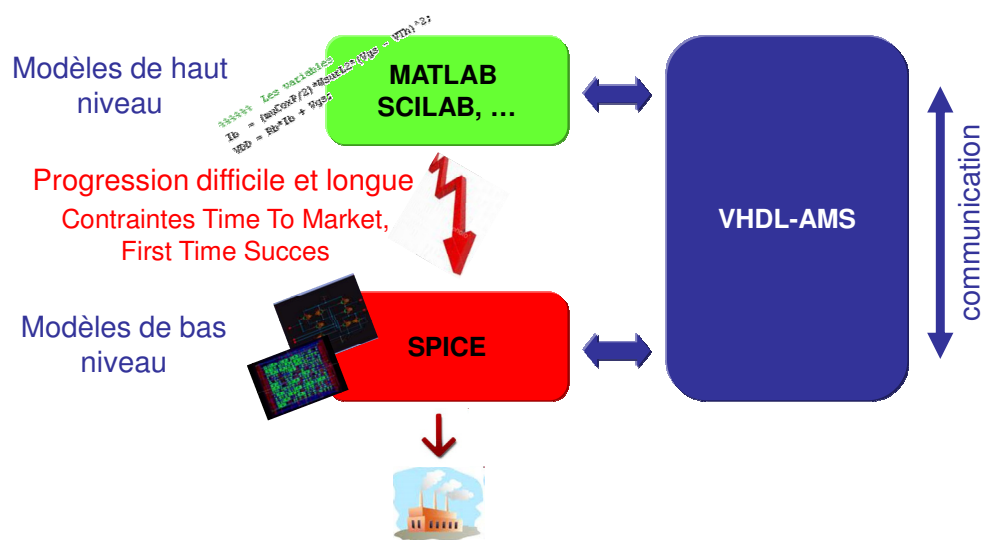


Figure VII-2 Positionnement de notre démarche dans le cycle de conception d'une application

L'entité « composant » dans le partitionnement Figure VII-1 est donc une description structurelle comportementale du système. Mais chaque entité « composant » aura maintenant une description comportementale pour le domaine métier. Ainsi la Figure VII-3 présente une série des modèles comportementaux (que nous appellerons modèles théoriques dans le domaine métier) de niveaux d'abstraction différents pour les « composants » élémentaires du réseau SNN1.

Pour chaque « composant » le concepteur de circuits peut créer N modèles théoriques. Ces modèles sont « théoriques » car ils ne prennent en compte aucune réalisation particulière. Par contre ils peuvent décrire des imperfections du « composant ». Ces imperfections sont issues d'une analyse très synthétique des comportements obtenus lors de réalisations précédentes. Le caractère théorique exprime ici le caractère possible de l'existence d'une telle imperfection sans pouvoir affirmer qu'elle existera après fabrication. Cette incertitude sur

l'existence d'une imperfection vient du fait qu'aucun choix (physique ou technique) n'est réalisé à ce stade de l'étude de la conception. L'idée de l'approche PVF est entre autre de repousser au plus tard l'introduction de la technologie. Cela permet aussi de capitaliser l'effort de conception en la rendant plus générique. Les modèles théoriques comportent un numéro de niveau d'abstraction. Le niveau 0 correspond au niveau de description idéal, c'est à dire le plus abstrait. Le niveau N correspond à une description détaillée du comportement du « composant » en intégrant le maximum d'imperfections possibles et imaginables (Figure VII-3).

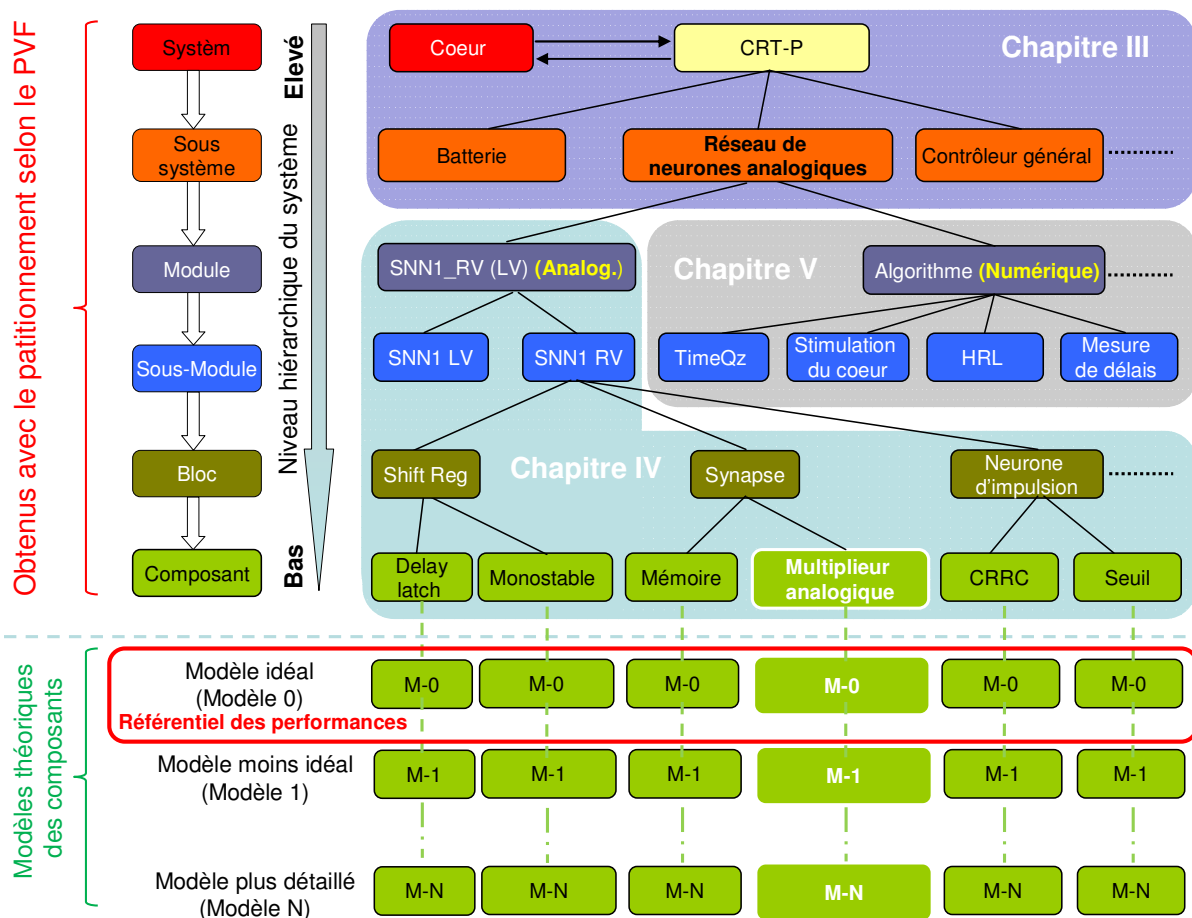


Figure VII-3 Etablissement des modèles théoriques des composants

Avec la plateforme en VHDL-AMS, nous pouvons maintenant intégrer progressivement des modèles de moins en moins abstraits dans le système complet afin d'en mesurer les impacts. La mesure des impacts doit permettre d'établir les premières spécifications que l'on imposera aux concepteurs métiers pour garantir le bon fonctionnement du système après fabrication. Néanmoins, comme dans un système complexe les divers paramètres sont certainement couplés, il convient de progresser de manière cohérente pour identifier les sources de limitations et établir leurs spécifications. Pour cela nous effectuons plusieurs

Chapitre VII

simulations du système (voir chapitre VI) en y introduisant graduellement les modèles théoriques. La Figure VII-4 illustre plusieurs combinaisons de niveaux d'abstraction pour chaque simulation du système. Ces simulations ont besoin d'un référentiel pour construire la mesure de l'impact des imperfections. Notre référentiel des performances sera donné par une simulation du système avec l'ensemble de ses modèles théoriques au niveau 0. Cette simulation fixe en fait le point de fonctionnement NOMINAL pour le système et pour chaque « composant ». Les simulations suivantes (introduisant au moins 1 modèle théorique à un niveau n supérieur à 0) permettront de mesurer l'écart au point nominal. L'observation d'une violation d'une des spécifications du niveau système conduira à fixer une des limites de spécifications pour le paramètre « composant » à l'origine de la violation. C'est cette identification qui nécessite d'introduire progressivement les modèles théoriques de niveau $n > 0$.

	Delay latch	Monostable	Mémoire	Multiplieur analogique	CRRC	Seuil
Simulation du système complet 1	{ M-0	M-0	M-0	M-1	M-0	M-0 }
Simulation du système complet 2	{ M-0	M-0	M-0	M-2	M-0	M-0 }
Simulation du système complet 3	{ M-0	M-0	M-1	M-0	M-0	M-0 }
Simulation du système complet 4	{ M-1	M-1	M-1	M-1	M-1	M-1 }

Figure VII-4 Simulation du système complet avec des modèles théoriques de niveaux différents

Par ce processus, on construit progressivement les spécifications techniques de chaque « composant » système sur la base des modèles théoriques. Ce qui fait que certaines spécifications adresseront des imperfections qui n'existeront pas lors de la réalisation. Mais donner une spécification sur un paramètre d'imperfection est toujours une spécification centrée sur la valeur nominale « 0 ». Par conséquent une imperfection qui n'existe pas, aura par définition une valeur nominale nulle. Elle satisfait donc, par définition même, à cette spécification. Il restera à se poser la question si cela est « utile » d'extraire des spécifications qui ne serviront pas. C'est une question que nous laissons ouverte mais qui nous semble compatible avec l'idée de capitalisation de l'étude générique. Une version future du même système utilisera peut-être de nouvelles technologies pour lesquelles autant les topologies que les caractéristiques laisseront apparaître des imperfections théoriques absentes dans la première version.

VII.1.2 Quelle importance accorder à un modèle théorique de niveau N si nous ne connaissons pas la structure du « composant » (la topologie du circuit) ?

Dans un cas idéal, une fonctionnalité élémentaire est associée à un « composant » système et celle-ci est associée de façon unique à une structure physique. Pour un électronicien cela voudrait dire que la fonction multiplieur analogique 4 quadrants est associée de façon unique à la topologie du multiplieur de Gilbert. Imaginons que ce soit vrai. Dans ce cas nous pourrions, avec une approche « Bottom – Up », construire des représentations de son comportement en progressant de la description la plus exacte vers une description la plus idéale. Chaque description de son comportement physique sera appelée : modèle physique de niveau P. Le niveau le plus exact sera le niveau M, et le niveau le moins exact mais le plus idéal sera le niveau 0. Selon cette affirmation, les deux descriptions (le modèle théorique et le modèle physique) de niveau 0 doivent nécessairement être équivalentes. On aura donc une passerelle entre les deux modèles au niveau 0.

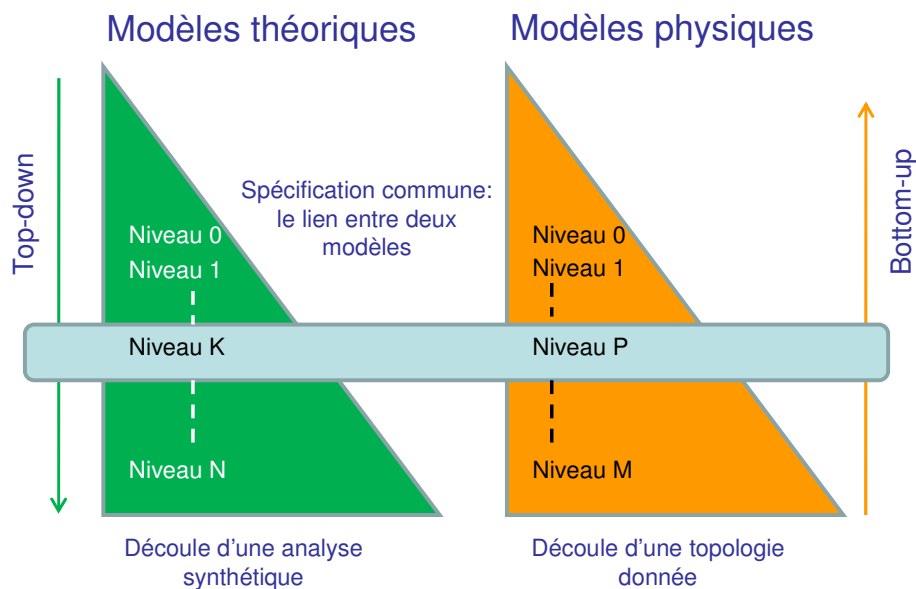


Figure VII-5 Equivalence entre modèles théoriques et physiques

Quittons maintenant ce cadre idéal et reprenons nos deux ensembles de modèles : les $N+1$ modèles théoriques et les $M+1$ modèles physiques. La question est : existe-t-il un niveau K et un niveau P entre les modèles théoriques et physiques respectivement différent de 0 où il y a équivalence ? Si ces niveaux existent alors les spécifications du « composants » pourront

Chapitre VII

être transposées avec une meilleure efficacité dans le domaine métier (Figure VII-5). Une idée que nous proposons est de comparer les hypothèses d'existence des imperfections qui ont servies à la construction des N niveaux théoriques et les hypothèses de simplifications qui ont permis de construire les M niveaux des modèles physiques. Malheureusement cette proposition n'a pas pu être évaluée durant cette thèse et restera ouverte.

Sans cette transposition, les spécifications resteront plus abstraites et le choix d'une topologie sur la seule base de ces spécifications sera plus discutable. Cette réflexion pose alors la troisième question.

VII.1.3 Comment choisir la meilleure topologie électrique pour un « composant » système?

Conceptuellement, faire un choix nécessite un ensemble contenant les éléments sur lesquels porte le choix et un ensemble de critères qui vont conditionner le choix. Regardons plus en détails comment construire ces deux ensembles. Pour un concepteur de circuits électroniques, le premier ensemble contient des topologies qui peuvent être potentiellement intéressantes pour la réalisation de la fonctionnalité assignée au « composant ». Une première méthode pour construire cet ensemble est basée sur la réutilisation de topologies analogiques déjà validées pour des objectifs similaires [47]. C'est une approche de type "Re-Use". Une alternative à cette méthode qui est spécifiquement adaptée aux réseaux de neurones analogiques fut proposée par T. LEVI dans sa thèse [35]. Il a généré puis utilisé une bibliothèque de IP (Intellectual Property) AMS afin de faciliter et d'automatiser la recherche de topologies potentiellement intéressantes. Une approche plus générique est d'utiliser des moteurs de créations automatiques de topologies comme ceux décrit dans [37] [38]. Mais ces derniers sont à ce jour des objets de laboratoire. Pour un projet industriel innovant, la notion de « Re-Use » n'existe pas, et l'expertise des concepteurs n'est pas acquise. Pour rester dans l'optique de notre démarche, je propose une étude bibliographique (qui peut être associée à du Data Mining) dans les publications techniques ou scientifiques. La sélection des éléments de l'ensemble des topologies, se fait sur la fonctionnalité et les plages de spécifications « composant ». C'est une sélection que l'on peut qualifier de « heuristique ». Pour ne pas limiter la qualité du choix final, nous retiendrons le plus grand nombre de topologies qui répondent aux spécifications. Cette « présélection » est délicate car il ne faut pas pré-juger des

réelles possibilités d'une topologie. C'est pour cela que nous acceptons des topologies dont les performances s'approchent de nos spécifications.

Le second ensemble nécessaire comprend tout naturellement les spécifications techniques du « composant ». Mais pour améliorer la qualité de la sélection topologique, nous devons introduire une estimation des possibilités de robustesse de cette topologie en regard des dispersions technologiques possibles. Cette estimation a été introduite dans la thèse de mon équipier François SCHWARTZ [36].

La sélection topologique finale se fera donc sur le choix des topologies potentiellement intéressantes issues de l'analyse bibliographiques et, en accord avec les spécifications techniques, selon un critère de maximum de robustesse à la fabrication.

Le flot complet de conception pour un électronicien est décrit ci-dessous dans la Figure VII-6. Nous pouvons voir que cette méthode établit un pont entre la simulation de niveau système (bloc gris) et la conception analogique (blocs bleus). Grâce à la plate-forme en VHDL-AMS, nous pouvons extraire des plages de spécifications acceptables en garantissant les performances du système. L'approche se concentre ensuite sur l'articulation la plus critique qui est la sélection topologique. Elle reste parfaitement compatible avec les approches métiers classiques (tout au moins en électronique pour cette thèse). Ces outils métiers peuvent ensuite guider la conception vers l'outil de validation final qui est SPICE avec une post simulation finale du « composant » système. Cette simulation finale permet de mesurer les écarts entre les simulations électriques de bas niveaux d'abstraction et les attentes exprimées par les spécifications « composants ». La démarche est compatible avec l'objectif du PVF qui est d'éviter la réalisation d'un prototype physique.

Pour illustrer la mise en œuvre de notre approche de conception, nous allons l'appliquer à un « composant » système en particulier. Nous avons retenu le multiplieur analogique qui est un composant élémentaire critique pour le bon fonctionnement d'un neurone de type « Integer & Fire », c'est à dire un neurone d'un réseau de type SNN (Spiking Neural Network). Le neurone est celui du réseau SNN1 du processeur neuronal analogique.

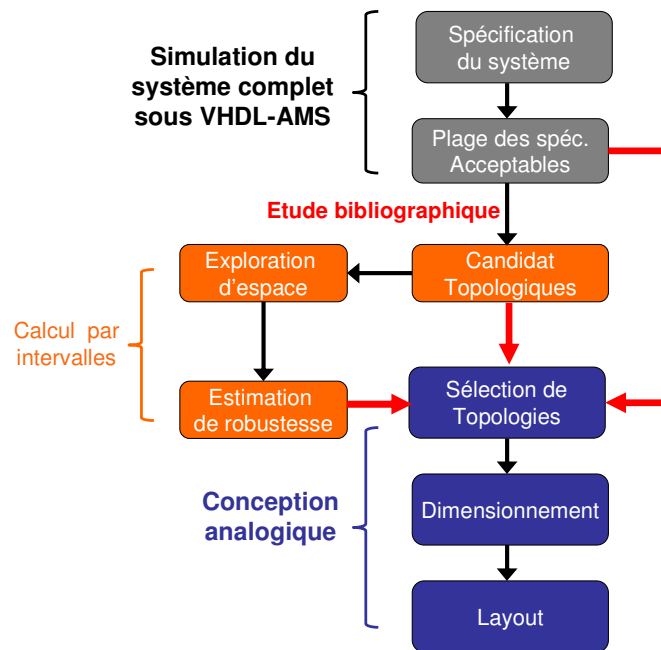


Figure VII-6 Flot de conception liant l'approche système et l'approche circuit

VII.2 Développement de notre approche conception au niveau « composant »

Rappelons que l'objectif fonctionnel du réseau SNN1 est de proposer l'instant de déclenchement des contractions des deux ventricules. Ces instants appelés T^i , $i \in \{AV, VV\}$ sont établis en regard des délais optimisés AV et VV. Les parties critiques des neurones constituant le réseau SNN1 sont les deux derniers modules (voir la Figure IV-1): le module « Synapses dynamiques » et le module « Leaky integration ». Le multiplieur analogique effectue la multiplication des impulsions unitaires générées par les shift-registers dans le module précédant (voir la Figure IV-1) avec les poids synaptiques gérés par l'algorithme d'apprentissage. Le multiplieur module donc l'intensité (l'amplitude) des différentes impulsions excitant le module « Leaky intégration ». C'est précisément les valeurs de ces impulsions qui conditionnent le déclenchement du neurone et fixe la valeur de l'instant T^i . Les performances du multiplieur analogique conditionnent donc l'instant de déclenchement et par suite les délais AV et VV séparant les contractions respectives des ventricules.

Dans ce paragraphe nous allons détailler les fondements de notre approche de conception au niveau circuit. Cette approche sera illustrée sur un seul « composant » qui est le multiplieur analogique. Dans la première phase de notre étude d'implémentation d'un neurone

analogique, le multiplicateur analogique utilisé dans le bloc « Synapse » fut considéré comme une entité idéale. Le modèle du multiplieur utilisé lors des phases de validation fonctionnelle fut le modèle théorique idéal, c'est à dire le modèle théorique de niveau 0. Le modèle théorique de niveau 0 utilise simplement une expression mathématique décrivant la fonctionnalité d'un multiplieur idéal :

$$I_{out} = C \cdot V_x \cdot V_y \quad (\text{VII.2.1})$$

où V_x et V_y sont les deux signaux d'entrée en tension (le poids synaptique ω_j^i et l'impulsion synaptique SI_j^i , respectivement), C un coefficient de gain de conversion et I_{out} le courant de sortie.

On peut immédiatement constater l'impact de ce choix de fonction mathématique sur une future sélection topologique. L'ensemble des contraintes de sélection augmentera de trois critères pour la sélection qui sont : les entrées sont des tensions et la sortie sera un courant. Ces mêmes critères sont déjà codés dans les modèles théoriques en langage VHDL-AMS par la déclaration de type des grandeurs d'échange entre les entités. On comprendra que la sélection topologique est une étape difficile à coder car influencée par de nombreux facteurs souvent intuitifs pour le concepteur. Si la sélection topologique ne peut trouver de topologie adéquate, il faudra modifier l'équation mathématique et donc modifier les interfaces entre les entités. Cela veut dire qu'il faudra introduire des étages de conversion courant – tension, ou bien carrément changer les entités en relation ce qui peut provoquer un effet domino, c'est à dire reconsidérer tous les « composants » du système pour retrouver une cohérence globale sur les grandeurs d'échanges. Pour éviter l'effet domino, les concepteurs privilégieront l'introduction d'entité de conversion laissant le reste du système inchangé.

Nous allons considérer l'équation (VII.2.1) comme notre objectif de conception. Pour préparer la sélection de topologie, il nous faut extraire du système les spécifications techniques du multiplieur. Pour cela nous allons développer le modèle théorique sur un plus grand nombre de niveaux d'imperfections. Ces différents modèles permettront de mesurer l'impact de ces imperfections sur le système puis en extraire les limites acceptables pour le système et qui deviendront les spécifications pour le multiplieur. Ces spécifications seront ajoutées à l'ensemble des contraintes de sélection. Ces modèles théoriques d'imperfections s'appuient sur une vision très théorique d'un multiplieur et ne considère pas les aspects particuliers d'une technologie précise. Pour illustrer ces propos, considérons le coefficient C . Il est clair que sa valeur numérique n'est pas contrôlable dans l'absolu par le procédé de

Chapitre VII

fabrication. Une première imperfection peut être introduite en posant $C = C_{\text{Nom}} + \varepsilon_C$ où ε_C est l'écart à la valeur nominale C_{Nom} . En seconde approche, on doit s'attendre à ce que le gain de conversion soit influencé par les amplitudes des signaux. Dans ce cas on peut écrire que $C = C_{\text{Nom}} + GV_x$ où G traduit l'impact de la tension d'entrée V_x sur le gain de conversion C . Mais une telle relation conduit aussi à la relation mathématique $I_{\text{Out}} = C_{\text{Nom}}V_xV_y + GV_x^2V_y$. Voici un exemple de construction des modèles théoriques qui ne fait référence à aucune technologie particulière. Cela n'exclut pas le fait qu'un concepteur puisse orienter ces modèles s'il connaît la technologie, mais selon notre philosophie méthodologique, cela réduira la « réutilisabilité » de l'analyse si une autre technologie est utilisée plus tard. Au delà des imperfections fonctionnelles, la méthode permet aussi d'affiner en parallèle, la consommation énergétique. Ce point peut s'ajouter à l'ensemble des contraintes pour la sélection de topologie.

On peut donc constater que le nombre d'éléments de l'ensemble des contraintes de sélection augmente avec le niveau d'imperfections que l'on utilise pour l'analyse du système. Il devient évident que si l'on lance la recherche bibliographique en se basant uniquement sur le modèle de niveau 0, le nombre de topologies potentiellement intéressantes est grand dans la mesure où le nombre de critères de sélection est faible. A contrario, si le niveau des modèles augmente, le nombre de contraintes augmente et le nombre de topologies intéressantes diminue. C'est ce que nous illustrons dans la Figure VII-7. Ainsi, nous appelons « gamme » la liste des topologies potentiellement intéressantes associée au niveau des imperfections utilisé pour les critères de sélections.

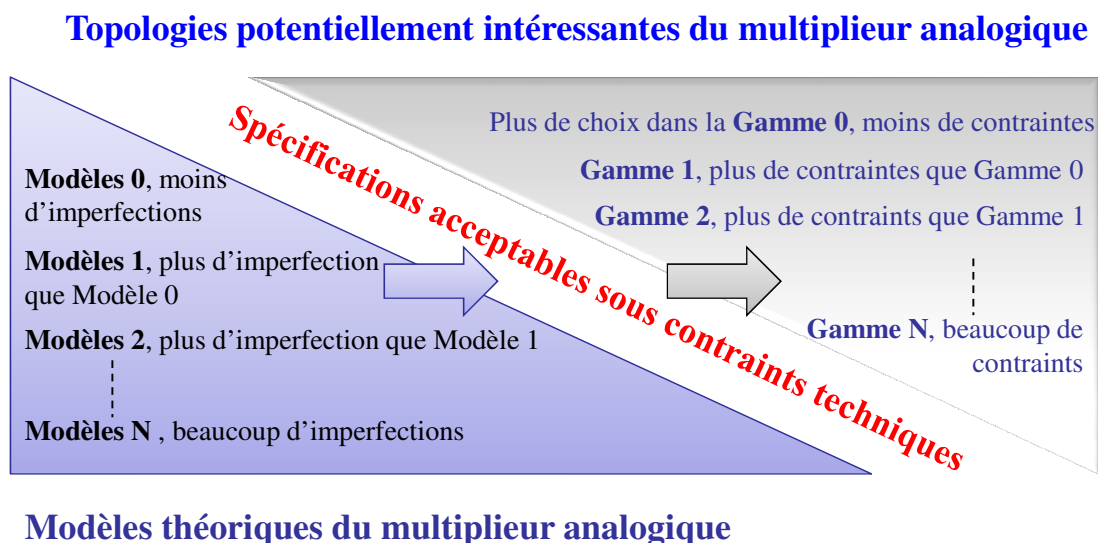


Figure VII-7 Méthodologie de notre étude sur le multiplieur analogique

VII.3 Modèles théoriques du multiplieur analogique

Mesurer l'impact des imperfections sur le système nécessite aussi de définir les critères de non fonctionnement. Nous devons donc durant les phases de validations de spécification « composant » établir une liste des critères à maîtriser. Ces critères participent naturellement à la prise de décision du « Go / No Go ». Pour ce qui concerne le fonctionnement du réseau de neurones SNN1, les valeurs de poids synaptiques ne doivent jamais être négatives. C'est une des hypothèses conceptuelles qui conditionnent le fonctionnement et l'algorithme d'apprentissage. Donc, notre premier critère pour la spécification du module SNN1 est qu'il n'y a pas de poids synaptiques négatifs. Si nous observons l'apparition de poids synaptiques négatifs, alors nous pouvons considérer que les valeurs prises par les imperfections dans notre modèle de niveau n constituent une borne pour les spécifications techniques du « composants ». Un autre critère pour SNN1 est d'assurer le déclenchement de chaque neurone uniquement par suite d'une excitation de l'étage CRRC par 4 impulsions consécutives de forte amplitude (voir le Chapitre IV).

En considérant le modèle de niveau 0 du multiplieur analogique défini dans (VII.2.1), nous avons développé les modèles théoriques de moindre niveau d'abstraction donnés au Tableau VII-1.

Tableau VII-1 Modèles abstraits du multiplieur analogique

Modèle 0	$I_{out} = C_{Nom} \cdot V_x \cdot V_y$	Modèle idéal
Modèle 1	$I_{out} = (1 + \varepsilon_1) \cdot C_{Nom} \cdot V_x \cdot V_y$	ε_1 représente l'écart du gain de conversion C
Modèle 2	$I_{out} = (C_{Nom} + \varepsilon_2 \cdot V_x) \cdot V_x \cdot V_y$	ε_2 représente l'impact de la tension d'entrée V_x sur le gain de conversion C
Modèle 3	$I_{out} = (C_{Nom} + \varepsilon_3 \cdot V_y) \cdot V_x \cdot V_y$	ε_3 représente l'impact de la tension d'entrée V_y sur le gain de conversion C
Modèle 4	$I_{out} = C_{Nom} \cdot V_x \cdot V_y + \varepsilon_4 \cdot V_x$	ε_4 représente l'impact de la tension d'entrée V_x sur l'offset du courant de sortie I_{out}
Modèle 5	$I_{out} = C_{Nom} \cdot V_x \cdot V_y + \varepsilon_5 \cdot V_y$	ε_5 représente l'impact de la tension d'entrée V_y sur l'offset du courant de sortie I_{out}
Modèle 6	$I_{out} = C_{Nom} \cdot V_x \cdot V_y + \varepsilon_6 \cdot V_x^2$	ε_6 représente l'impact du carré de la tension d'entrée V_x sur l'offset du courant de sortie I_{out}
Modèle 7	$I_{out} = C_{Nom} \cdot V_x \cdot V_y + \varepsilon_7 \cdot V_y^2$	ε_7 représente l'impact du carré de la tension d'entrée V_y sur l'offset du courant de sortie I_{out}

Chapitre VII

Avec le modèle 0, nous avons pu fixer les valeurs nominales des différentes grandeurs. Ainsi C_{Nom} prend la valeur de $1mA/V^2$, V_x (poids synaptique ω_j^i) vaut $1mV$ et V_y (impulsion synaptique SI_j^i) vaut $1V$. Les spécifications préciseront donc les intervalles d'acceptation des valeurs réalisées, c'est à dire la plage de valeurs obtenues après fabrication qui garantissent le bon fonctionnement du système. Nous lançons ensuite une série de simulations du système complet pour établir ces plages d'acceptations, c'est à dire les spécifications techniques. De cette façon, nous obtenons les plages des divers coefficients d'imperfections selon les critères de bon fonctionnement du système. Ces plages sont données au tableau suivant :

Tableau VII-2 Gammes acceptables des coefficients $\varepsilon_1 - \varepsilon_7$ garantissant le bon fonctionnement du système

ε_1	$[-1.5e-2, 8.0e-2]$
ε_2	$[-9.0e-2, 1.5e-2]$
ε_3	$[-1.9e-5, 8.0e-5]$
ε_4	$[-9.4e-8, 4.0e-7]$
ε_5	$[-5.5e-8, 1.0e-8]$
ε_6	$[-1.5e-4, 2.5e-5]$
ε_7	$[-1.5e-8, 4.5e-8]$

Pour illustrer la méthode, nous avons effectué deux simulations du système complet avec deux valeurs incompatibles avec le « bon fonctionnement » du système. Cette illustration est faite avec le modèle théorique de niveau 1 qui introduit le paramètre ε_1 .

Pour notre première simulation du système considérons le cas $\varepsilon_1=8.5e-2$. La Figure VII-8 représente le signal $LI^i(t)$ d'un neurone (SN^{AV} ou SN^{VV}). Lors d'un fonctionnement normal, seule une suite de quatre impulsions unitaires consécutives peut conduire le signal $LI^i(t)$ à dépasser le seuil interne du neurone et provoquer l'émission en sortie d'une impulsion. Si le gain du multiplieur s'accroît, les amplitudes des impulsions PSR_j^i s'accroissent et peuvent devenir supérieur à l'unité. En fait, comme l'écart du gain ε_1 est positif, le gain de conversion C augmente entraînant l'augmentation du courant de sortie I_{out} . La question est alors : une suite d'impulsions non unitaires peut-elle conduire au dépassement du seuil ? (voir paragraphe IV.2.2.5) . Dans la simulation de la Figure VII-8, nous n'avons pas quatre impulsions unitaires consécutives mais la séquence d'impulsions $[0.5, 1, 1, 0.5]$. Mais l'accroissement du courant I_{out} conduit à dépasser le seuil de déclenchement. En détectant ce déclenchement, l'algorithme d'apprentissage va reconnaître cette séquence comme une séquence de quatre impulsions unitaires consécutives $([1, 1, 1, 1])$ et va procéder au

changement des poids synaptiques de cette séquence. Cela conduit l'ensemble à faire apparaître progressivement des poids synaptiques négatifs, invalidant le bon fonctionnement du système. Du fait de cette limitation fonctionnelle, nous trouvons la borne supérieure du paramètre ϵ_1 .

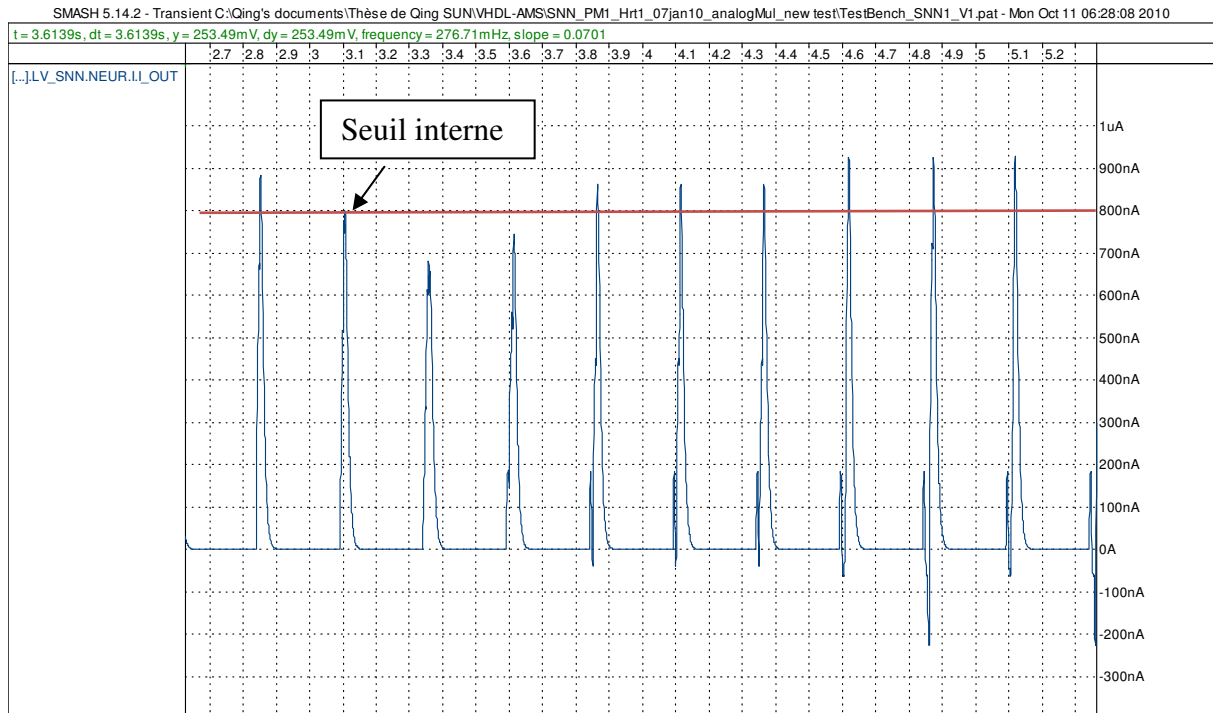


Figure VII-8 Simulation du système complet avec $\epsilon_1=8.5 \times 10^{-2}$

La seconde simulation (Figure VII-9) s'effectue avec le cas contraire. Avec $\epsilon_1 = -2.0 \times 10^{-2}$ nous sommes sous la borne inférieure. Le courant de sortie devient plus petit avec la réduction du gain de conversion C. Nous n'aurons plus de déclenchement après une bonne séquence d'impulsions ([1, 1, 1, 1]). Dans le mode Non Adaptative (voir le chapitre III.4), nous observons qu'il n'y a pas de déclenchement dans l'emplacement normal du premier déclenchement. Cela va aussi nous poser des problèmes parce que l'algorithme d'apprentissage adopté (ici l'algorithme d'apprentissage de Hebb) va continuer à augmenter les poids synaptiques en l'absence de déclenchement. Comme précédemment, cette valeur de ϵ_1 n'est pas compatible avec le bon fonctionnement du système.

L'analyse comportementale du système permet, en modifiant ces paramètres d'imperfections, d'extraire progressivement les différentes plages de valeurs acceptables. Dans cette illustration, la plage acceptable ($[-1.5 \times 10^{-2}, 8.0 \times 10^{-2}]$) du coefficient ϵ_1 au sens de la fonctionnalité du système est établie. Elle constitue donc une spécification de conception pour cette imperfection.

Chapitre VII

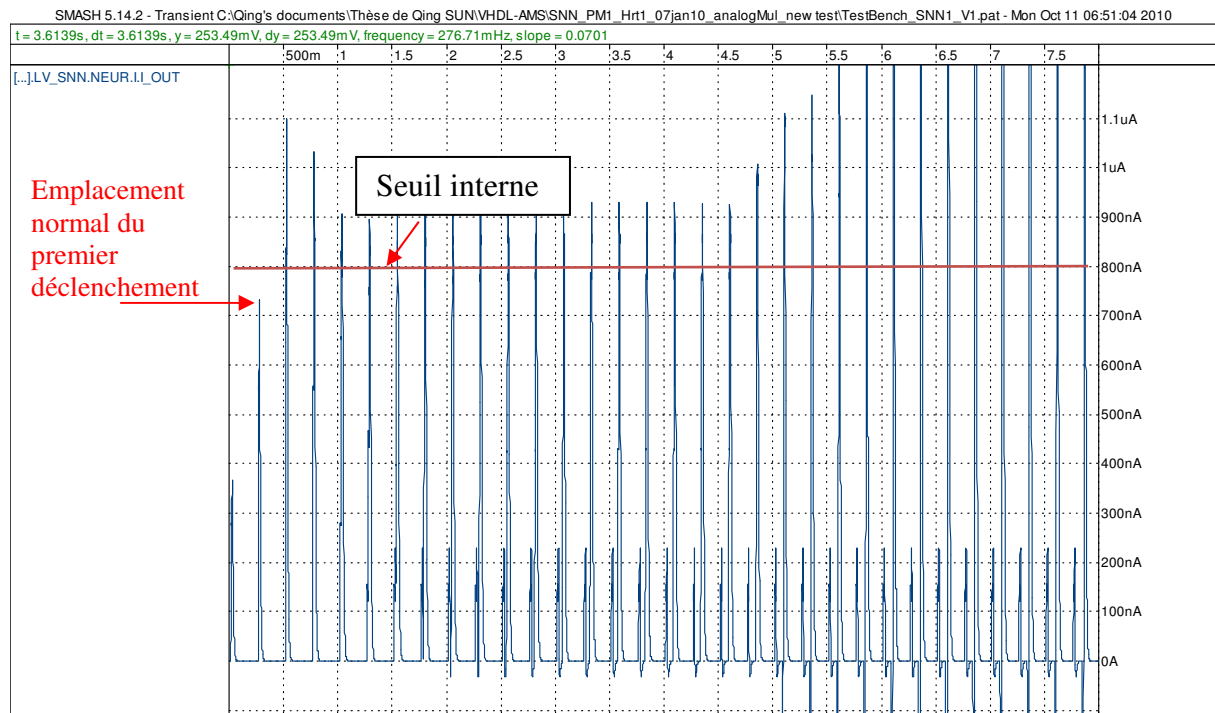


Figure VII-9 Simulation du système complet avec $\epsilon_1 = -2.0^{-2}$

Rappelons que les imperfections utilisées sont de nature théorique et peuvent ne pas apparaître après fabrication. C'est une situation que nous admettons et que nous ne considérons pas comme limitative. Cependant cette approche « purement théorique » ne permet pas de converger « rapidement » vers la solution industrielle car l'impact des choix technologiques y est soit absent soit trop décorrélé. Au sens de l'efficacité, il serait préférable de s'appuyer sur les modèles physiques qui permettent alors de focaliser ces simulations sur les imperfections « existantes ». Mais cela suppose la sélection d'une topologie.

Une alternative méthodologique est de s'appuyer sur une topologie générique. L'idée est alors d'identifier les paramètres les plus critiques. Au travers de ces paramètres nous pourrions introduire l'impact du choix de la technologie. Dans le paragraphe suivant nous allons illustrer comment appliquer la même démarche méthodologique mais en y intégrant le choix technologique qui a pu être fait dès la définition du projet. Ce fut en fait notre cas dans le cadre du projet Adapter. Nous considérons toujours que notre entité multiplieur s'appuie sur le modèle théorique de l'équation (VII.2.1). Pour introduire l'impact d'une technologie il nous faut donc définir la topologie de référence. Le choix de cette topologie est critique car elle conditionnera la validité des spécifications lors de recherche d'autres topologies candidates. Comme topologie générique, nous prenons un multiplieur analogique de type Gilbert [10] . Dès lors il devient possible d'explicitier divers paramètres des modèles théoriques. En fait, cette réorientation de notre approche, suggère que nous utilisons une

identification de niveaux d'équivalence entre des modèles théoriques et physiques comme nous l'avons suggéré au paragraphe VII.1.2 et dans la Figure VII-5.

Ainsi le gain de conversion C du multiplieur est corrélé avec le coefficient K classiquement utilisé dans l'équation caractéristique I-V d'un transistor MOS en régime de saturation. L'expression au premier ordre de la relation courant – tension d'un transistor MOS est donnée par :

$$I_d = \frac{\mu C_{ox}}{2} \frac{W}{L} (V_{gs} - V_t)^2 = K (V_{gs} - V_t)^2, \quad K = \frac{\mu C_{ox}}{2} \frac{W}{L} \quad (\text{VII.3.1})$$

où μ est la mobilité des porteurs, C_{ox} est la capacité d'oxyde de grille, W est la largeur du canal, L est la longueur du canal et V_t est la tension de seuil. Selon [10], la relation fonctionnelle d'un multiplieur analogique en tension de type Gilbert est :

$$I_{out} = \sqrt{2} K V_x V_y, \quad \text{avec } K = \frac{\mu C_{ox}}{2} \cdot \frac{W}{L} \quad (\text{VII.3.2})$$

Donc, le gain de conversion C du multiplieur peut être exprimé en utilisant l'équation caractéristique I-V d'un transistor MOS en régime de saturation :

$$C = \sqrt{2} K = \frac{\sqrt{2} \mu C_{ox}}{2} \cdot \frac{W}{L} \quad (\text{VII.3.3})$$

Si nous supposons que le gain de conversion C reste indépendant des grandeurs électriques V_x et V_y , la variation du gain C sera décrite par le paramètre d'imperfection ϵ_1 . Celui-ci dépendra du terme μC_{ox} et du ratio W/L. Le premier terme introduit directement la technologie. Le second terme introduit l'impact du dimensionnement. Or l'équation VII.3.1 suppose l'utilisation de transistors à canaux longs. Cette observation souligne que l'introduction des hypothèses devra nécessairement participer à la définition des niveaux des modèles physiques et théoriques. Une telle liste permettra d'identifier des niveaux d'équivalences entre théorique et physique

Contrairement à l'hypothèse précédente, si le dimensionnement privilégie des transistors à canaux courts, nous devons prendre en compte la variation de mobilité due au champ électrique critique. La saturation de la vitesse liée au champ critique est décrite par le paramètre technologique (θ). Selon [45] [46], la caractéristique I-V simplifiée a été modifiée comme ci-dessus :

$$I_d = \frac{K (V_{gs} - V_t)^2}{1 + \theta (V_{gs} - V_t)} \quad (\text{VII.3.4})$$

Chapitre VII

A l'aide du logiciel Maxima[®], nous reprenons la même architecture pour le cas de (VII.3.2) dans [10] en remplaçant (VII.3.1) par (VII.3.4). Nous recalculons symboliquement le courant de sortie (I_{out}) du multiplieur analogique dans [10] et nous obtenons l'expression prenant en compte l'effet de la saturation de la vitesse des porteurs décrite par θ . Cette description conduit au modèle théorique 8:

$$I_{out} = \frac{NUMERATEUR}{DENOMINATEUR} \quad (VII.3.5)$$

où

$$\begin{aligned} NUMERATEUR = & 48\theta^4 V_x^3 V_y^5 K + 48\theta^2 V_x V_y^5 K \\ & - 96\theta^4 V_x^5 V_y^3 K - 224\theta^2 V_x^3 V_y^3 K \\ & - 256 V_x V_y^3 K + 48\theta^4 V_x^7 V_y K \\ & + 176\theta^2 V_x^5 V_y K + 512 V_x^3 V_y \end{aligned} \quad (VII.3.6)$$

et

$$\begin{aligned} DENOMINATEUR = & \theta^6 V_y^8 - 32\theta^6 V_x^2 V_y^6 \\ & - 104\theta^4 V_y^6 + 286\theta^6 V_x^4 V_y^4 \\ & + 840\theta^4 V_x^2 V_y^4 + 144\theta^2 V_y^4 \\ & - 480\theta^6 V_x^6 V_y^2 - 1816\theta^4 V_x^4 V_y^2 \\ & - 1184\theta^2 V_x^2 V_y^2 - 256 V_y^2 \\ & + 225\theta^6 V_x^8 + 1080\theta^4 V_x^6 \\ & + 1296\theta^2 V_x^4 \end{aligned} \quad (VII.3.7)$$

Nous faisons ensuite la simulation du système complet en utilisant ce modèle du multiplieur analogique en tension de type Gilbert sous l'impact du paramètre technologique θ [45] [46]. Avec les mêmes critères d'acceptation que les modèles théoriques 1-7, nous extrayons la plage acceptable de ce coefficient $\theta < 0.058$, sachant que normalement θ est compris entre 0.001 et 0.1 (V^{-1}). Ce modèle théorique nous montre qu'en intégrant des imperfections réelles de la technologie, nous pouvons créer des modèles théoriques servant à extraire la plage des coefficients acceptables d'un modèle physique. Une telle approche nous donne un critère important lors de la sélection de la technologie. Mais ce point n'est pas développé dans cette thèse.

On notera que les spécifications des coefficients partagés par les modèles théoriques et physiques constituent un pont pour traverser le fossé entre ces deux types de modèles différents. Les outils de calcul symbolique seront très utiles pour générer les modèles théoriques en prenant en compte des imperfections techniques complexes.

VII.4 Sélection des topologies selon l'étude bibliographique

A la suite des simulations de système global avec les modèles théoriques du multiplieur analogique, nous avons extrait les plages des spécifications acceptables dans le Tableau VII-2. Afin de créer le lien avec les spécifications consultables dans les articles scientifiques ou technologiques, nous avons calculé la non-linéarité pour chaque modèle théorique.

Prenons le modèle 1 comme exemple. Nous définissons l'expression analytique de l'erreur de non-linéarité NL:

$$I_{out1} = \underbrace{C_{Nom} \cdot V_x \cdot V_y}_{\text{Modèle idéal}} + \underbrace{\varepsilon_1 \cdot C_{Nom} \cdot V_x \cdot V_y}_{\text{Termes de non-linéarité}} \quad (\text{VII.4.1})$$

Le premier terme de cette équation est juste le modèle théorique de niveau 0 sans imperfection. Le second terme introduit la non-linéarité. Nous définissons la non-linéarité comme l'écart du réalisé sur le nominale en normalisant cet écart à la valeur nominale; ainsi $NL = (I_{réel} - I_{nom})/I_{nom} = (\varepsilon I_{nom})/I_{nom} = \varepsilon$. Comme l'écart peut être positif ou négatif on doit considérer sa valeur absolue. Comme les bornes ne sont pas forcements symétriques, nous prendrons la borne dont la valeur absolue est minimale afin de construire un critère plus restrictif car lui seul peut garantir le bon fonctionnement. Rappelons immédiatement que notre objectif est d'aider à la sélection des topologies candidates et non le dimensionnement à proprement parler. C'est pourquoi, considérer des contraintes restrictives, est plus une façon d'affiner la présélection de topologies candidates. Si, sur la base de ces restrictions, l'ensemble des topologies candidates est vide, on pourra adoucir cette contrainte. L'idée initiale est tout de même de limiter le temps de développement, travailler sur peu de topologies est certainement plus efficace, mais n'avoir aucune topologie est bloquant. Un compromis devra être fait par les concepteurs.

Les non-linéarités de chaque modèle théorique peuvent être calculées par l'équation suivante :

$$NL_i = \frac{\min |\text{Termes de non-linéarité}_i|}{I_{outi}} \quad i \in [1, 2, 3, 4, 5, 6, 7] \quad (\text{VII.4.2})$$

Les résultats de chaque modèle théorique sont montrés dans le Tableau VII-3. Selon les résultats, la non-linéarité du modèle 3 est dominante. Par contre, les modèles 4 et 6 génèrent des non linéarités de très faibles valeurs. Autrement dit, les impacts de la tension d'entrée V_x

Chapitre VII

(le poids synaptique ω_j^i) sur l'offset du courant de sortie sont très sensibles pour un bon fonctionnement du système.

Tableau VII-3 Non-linéarités visées des modèles théoriques du multiplieur analogique

Modèle 1	$NL_1 = \frac{\min \varepsilon_{NL_1} }{I_{out1}} = \frac{\min \varepsilon_1 C_{Nom} V_x V_y }{(1 + \varepsilon_1) C_{Nom} V_x V_y}$	$NL_1 = 1.5\%$
Modèle 2	$NL_2 = \frac{\min \varepsilon_{NL_2} }{I_{out2}} = \frac{\min \varepsilon_2 V_x^2 V_y }{C_{Nom} V_x V_y + \varepsilon_2 V_x^2 V_y}$	$NL_2 = 1.5\%$
Modèle 3	$NL_3 = \frac{\min \varepsilon_{NL_3} }{I_{out3}} = \frac{\min \varepsilon_3 V_x V_y^2 }{C_{Nom} V_x V_y + \varepsilon_3 V_x V_y^2}$	$NL_3 = 1.9\%$
Modèle 4	$NL_4 = \frac{\min \varepsilon_{NL_4} }{I_{out4}} = \frac{\min \varepsilon_4 V_x }{C_{Nom} V_x V_y + \varepsilon_4 V_x}$	$NL_4 = 0.0094\%$
Modèle 5	$NL_5 = \frac{\min \varepsilon_{NL_5} }{I_{out5}} = \frac{\min \varepsilon_5 V_y }{C_{Nom} V_x V_y + \varepsilon_5 V_y}$	$NL_5 = 1.0\%$
Modèle 6	$NL_6 = \frac{\min \varepsilon_{NL_6} }{I_{out6}} = \frac{\min \varepsilon_6 V_x^2 }{C_{Nom} V_x V_y + \varepsilon_6 V_x^2}$	$NL_6 = 0.0025\%$
Modèle 7	$NL_7 = \frac{\min \varepsilon_{NL_7} }{I_{out7}} = \frac{\min \varepsilon_7 V_y^2 }{C_{Nom} V_x V_y + \varepsilon_7 V_y^2}$	$NL_7 = 1.5\%$

L'introduction des paramètres de non linéarités est nécessaire pour notre exploration bibliographique, mais d'autres critères peuvent (et doivent) être introduits pour affiner la présélection des topologies. La tension d'alimentation et les bruits sont d'autres critères importants. Dans notre projet, ce circuit analogique est conçu pour un pacemaker, donc la consommation d'énergie et la surface du circuit sont aussi à considérer.

Notre exploration bibliographique recherche des articles publiés dans la communauté scientifique ou industrielle sur le dimensionnement de multiplieurs analogiques à sortie en courant. La sortie en courant est imposée par l'opération de sommation en entrée de la fonction d'activation (paragraphe IV.2). Les spécifications et les contraintes extraites de ces publications seront filtrées selon les niveaux d'imperfections correspondant à nos objectifs. Au fur et à mesure, nous construisons un tableau des spécifications des différentes structures publiées. Même si les spécifications dans les articles sont souvent incomplètes, nous pouvons faire une première comparaison afin de trouver les structures les plus pertinentes pour notre projet. Une telle bibliographie permet de vérifier que les paramètres dominants sont en adéquation avec nos modèles théoriques. Ensuite, elle permet d'actualiser les ordres de

grandeurs de ces paramètres dominants. Ceci permettra aussi d'alimenter des bases de données bibliographiques d'une manière plus générale si l'on souhaite utiliser dans le futur une approche de type « ReUse ». Dans le pire cas, si l'ensemble des topologies candidates est vide, soit on réduit les contraintes sélectives, soit on corrige le système pour adoucir les spécifications « composants », soit finalement on demandera l'aide concepteurs pour une nouvelle topologie. Cette dernière option devra être le choix ultime car il pénalisera le temps de développement sans garantie de résultats.

Tableau VII-4 Comparaison des spécifications extraites dans les articles des multiplieurs analogique

N° Publication.	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]
Power Supply ±(V)	0.75	2.5	1.25	4	1.5	5	2.5	3	3	1.5	1.2	
DC current (µA) Bias				200							2300	
Non Linearity	2%	20%		0.5%				2.47%	1.5%	2%	0.89%	0.2%
Input ±	0.125uA	0.1-20uA	250nA	1.5mA	7µA	200mA	0.6V	12uA	20nA	0.8 V	250m V	5V
THD				10%			0.8			2%	1.1%	0.5%
BW -3dB (MHz)			1	3	3		13.3		5	5	2.2	12
Area (mm2)			0.0016				0.077		9.3 ^c -4			0.05
Consumption (µW)		8.0			495		5	409 ^c 3	0.1			1.1 ^c 3
Techno (µm)	0.8	0.5	0.8	2	0.5	2		2.4		0.8	0.8	2

Tableau VII-5 Comparaison des spécifications extraites dans les articles de multiplieurs analogiques (suite)

N° Publication	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]
Power Supply±(V)	5	5	5	5	5	5	3		0.65
DC current (µA)									0.005
Non Linearity	1%	1%	2%	2%	1%	1%	0.44%	0.44%	0.44%
Input ±	1V	3V	0.8V	1.2V	0.8V	2.5V	V	A	5nA
THD	1%	2.5 ^c %	2%	0.05 ^c %	0.1%	1%	0.59%	0.59%	-30dB
BW -3dB (MHz)	7	20	4.5	0.45		1.2	5	5MHz	5
Area (mm2)			0.01067	4.4	0.18	210	5.6 ^c -5	500	
Consumption (µW)			1 ^c 3		6 ^c 3		10 ^c 3	10 ^c 3	0.0124
Techno (µm)	2	3	2	3	2	3	3	5	0.13

Les résultats issus de notre étude bibliographie du multiplieur analogique sont présentés dans les tableaux : Tableau VII-4 et Tableau VII-5. Sur un ensemble de vingt articles trouvés dans la communauté scientifique [11]–[32], les treize papiers [14], [19], [21]–[31] correspondent au critère de non-linéarité. En considérant les autres critères, on peut réduire l'ensemble de présélection à deux articles [19], [31]. Ces topologies affichent une très faible consommation d'énergie (100nW et 12.4nW) et utilisent des transistors en régime de faible inversion. De plus ces deux circuits utilisent le mode de courant en entrée qui semble plus avantageux pour l'implémentation dans le projet ADAPTER. Avec une faible tension d'alimentation, le mode courant nous donne plus de liberté pour la plage dynamique (Dynamics range : DR). Celle-ci conditionne la plage de variation des coefficients synaptiques. Plus cette plage est large plus le circuit aura de liberté pour s'adapter à une

Chapitre VII

problématique. Il offrira donc une solution topologique plus générale au sens des fonctionnalités que peut reproduire le neurone (ou le réseau de neurones). L'approche courant dans les neurones offre des avantages résumés dans le Tableau VII-6.

Bien que les modèles théoriques du multiplieur analogique soient initialisés avec des entrées en mode tension, cette recherche bibliographique montre qu'une solution topologique ad hoc doit privilégier les entrées en mode courant. Ce point est illustratif de l'intérêt de l'approche PVF. Cette constatation est obtenue très tôt dans le processus de développement du produit. L'impact économique est faible car aucun prototype n'a été nécessaire. Le concepteur domaine (électronique) doit maintenant redéfinir ces « composants » et reconstruire (partiellement) les spécifications pour le nouveau « composant » sans nécessiter l'intervention des autres partenaires. Le temps du dimensionnement est lui aussi épargné sachant que le dimensionnement final d'une topologie analogique est long et délicate.

Tableau VII-6 Comparaison entre le mode de courant et le mode de tension dans la conception analogique

Mode de courant	Mode de tension
haute vitesse à faible consommation d'énergie, à faible coût, haute précision	Conceptions exclusifs, strictement définis, de manière limitée
Logique dans la conception, le mode en courant lorsque leur distribution considérée	le mode en tension logique compte tenu des signaux à la porte des interfaces
Faible couplage inductif et non limité par la faible tension d'alimentation	Pire taux de signal/ bruit et augmentation sur les bruits de couplage
Dont l'état du signal est totalement et définitivement défini par les courants	Par ses tensions des nœuds

Comme nous l'avons illustré au travers de cet exemple, notre démarche méthodologique nous a permis de développer une structure fonctionnelle d'un réseau de neurones analogique. Celle-ci a ensuite fait l'objet d'une première sélection de topologies candidates. Les corrections structurelles ont pu être effectuées et au final, un cahier des charges plus précis avec des solutions topologiques peut être transmis aux concepteurs domaines pour progresser vers l'implémentation. Via la plateforme sous VHDL-AMS, nous assurons la performance du système et nous pouvons également prouver la faisabilité technique de chaque « composant » système.

VII.5 Bilan de puissance du réseau de neurones analogiques (SNN1)

Notre démarche méthodologique nous a permis d'établir une décomposition structurelle fonctionnelle assez fine et surtout totalement cohérente pour accroître les chances de réussite au premier prototype et surtout d'éviter les corrections systèmes révélées en fin de développement qui pénaliseraient le temps de développement. En nous basant sur les topologies candidates, nous pouvons maintenant effectuer un bilan de puissance consommée par un réseau de neurones analogiques (SNN1), ce critère étant l'un des critères fondamentaux du projet Adapter. Rappelons que notre étude devait, entre autre, permettre à l'industriel (ELA Medical) de choisir entre une implémentation digitale ou une implémentation analogique du processeur neuronale. Il nous incomber de pouvoir offrir une estimation justifiable de notre consommation énergétique et de la comparer à l'estimation initiale du prototype digitale de la société AI Semiconductor.

Nous récapitulons ici l'analyse en consommation énergétique de la solution digitale de AI Semiconductor. Ce processeur numérique fonctionne avec une fréquence interne de 1kHz. La technologie retenue est une technologie CMOS 0,13 μ m alimentée sous une tension de $V = 1,2V$. Une estimation des capacités de charge conduit à une estimation de la puissance moyenne consommée : $P = CV^2f$. Les capacités de charge présentent une valeur de 5 nF (1 million de capacités de 5 fF) ce qui donne une puissance moyenne de 7,2 μ W. L'activité moyenne du processeur numérique est de 10%, la puissance moyenne est alors de 0,72 μ W et l'énergie consommée sera donc de 22,7 Joules par an ($31,54 \times 10^6$ secondes/an). Pour une autonomie de 7 ans, la batterie devra avoir une capacité énergétique de 158,9 Joules. Selon [39]-[42], la densité volumique d'énergie des batteries lithium-manganèses [43] est de 1,98-2,09 kJ/ml, celle des batteries lithium-ions conventionnelles est au moins de 0,83 kJ/ml. Ainsi, moins d'un millilitre d'une batterie conventionnelle (lithium-ions) peut satisfaire le besoin énergétique du processeur neuronal. Selon [44], la batterie dans les pacemakers conventionnels est d'environ 5-8ml, ce qui devrait permettre de satisfaire une autonomie de 5-7 ans. On souligne que l'excitation des muscles cardiaques consomme la plus grande part d'énergie.

La consommation d'un neurone analogique résulte de la somme des énergies consommées par chaque module du neurone SN. Nous présentons ci-dessous les estimations de consommation pour chaque bloc du neurone en référence avec la figure IV-1. Ces

Chapitre VII

estimations sont conditionnées par des hypothèses que nous rappellerons cas par cas et que nous mettrons en perspectives avec les analyses précédentes. Les différentes grandeurs qui ont été introduites dans les chapitres précédents furent des grandeurs normalisées. Lors du passage à la description physique, nous transformons ces grandeurs normalisées en grandeurs physiques par les appellations du type : Var (nom de la grandeur normalisée), I_Var (nom de la grandeur physique avec dans ce cas l'indication qu'il s'agit d'un courant).

i. Shift Register

C'est l'étage d'entrée du neurone dans le SNN1. Il génère les 40 impulsions Pré Synaptiques. On considère ici le bloc shift register (SR) comportant 50 bascules D pour couvrir la plage temporelle de 250 ms au pas de 5ms. Chaque bloc opère donc

à la fréquence maximale $f_{SR} = \frac{1}{250ms} = 4Hz$. Ce calcul se base sur le pire cas, c'est

à dire pour un cœur qui bat au rythme de 240BPM, soit un battement du cœur tout les 250ms. Chaque bascule comporte 5 portes élémentaires dont la charge capacitive est estimée à un total de 2pF. La charge synaptique est plus importante qu'une simple porte et estimée à 1pF. Le monostable aura une charge plus grande que l'on peut estimer à 5pF. La consommation énergétique maximale de cette partie peut s'exprimer comme ci-dessous :

$$P_{SR} = C_{SR} V^2 f_{SR} \quad \text{avec } C_{SR} = 50 \cdot (2pF + 1pF + 5pF) = 400pF \quad (\text{VII.4.3})$$

Les grandeurs délivrées par le bloc sont les impulsions SI_j dont les valeurs sont comprises entre 0 et 1. Ces grandeurs sont traduites en courants $I_{SI_j} = SI_j * DR_S$ où DR_S est la plage dynamique du courant.

ii. Synapses (multiplieur analogique et mémoire)

Le bloc synaptique fonctionne avec des grandeurs analogiques et contribue au contrôle de l'amplitude du signal LI avant le déclenchement. De ce fait, nous sommes partie sur une hypothèse de forte linéarité des fonctions synaptiques. La seconde hypothèse est que les poids synaptiques ainsi que les impulsions pré et post synaptiques sont strictement positives.

Le fonctionnement proposé du bloc synaptique introduit un gain global maximum de 1, c'est à dire que l'amplitude de sortie est égale à l'amplitude d'entrée. Nous postulerons que les différentes entrées sont en mode courants et que la plage

dynamique de ces courants est DR_S . Ainsi le courant d'entrée des shift registers est I_{SI_j} , le courant d'entrée des poids synaptiques est $I_w = w_j \cdot DR_S$ et finalement le courant de sortie est $I_{PSR_j} = PSR_j \cdot DR_S$. Pour assurer une bonne linéarité du multiplieur analogique et de la mémoire analogique, nous fixons les courants de polarisation à $I_{Bias} = 3 \cdot DR_S$. Les puissances consommées par le multiplieur et par la mémoire seront données par:

$$\begin{cases} P_{multip} = V \cdot I_{Bias} = 3 \cdot V \cdot DR_S \\ P_{mémoire} = V \cdot I_{Bias} = 3 \cdot V \cdot DR_S \end{cases} \quad (VII.4.4)$$

iii. CRRC

Le troisième module du neurone comporte deux parties : l'intégrateur avec fuite et le comparateur à seuil. Nous considérons tout d'abord l'intégrateur avec fuite (le CRRC). Le courant d'entrée du CRRC ($I_{\Sigma PSR}$) est une suite d'impulsions de courants dont les amplitudes sont dans la gamme $[0, DR_S]$. La tension de sortie du CRRC est donnée par $V_{LI}(t)$. Du fait que les impulsions de courants sont positives, la tension de sortie restera strictement positive; on prévoit donc d'utiliser des étages amplificateurs polarisés en classe AB. Ainsi la puissance consommée sera dominée par la puissance associée au signal qui sera proportionnelle à l'intégrale du signal $V_{LI}(t)$.

Cette puissance dépendra donc de la séquence des impulsions d'entrée (SPSR). Considérons deux cas extrêmes :

Pour une séquence infinie d'impulsions ($K = \text{infini}$) la tension V_{LI} tendra vers la valeur asymptotique $V_{LI}(\text{infini}) = K_V \cdot LI(\text{infini})$ où $LI(\text{infini}) = 2,831$ et K_V est le coefficient de dénormalisation. Si on impose que $V_{LI}(\text{infini}) < V$, alors le coefficient devra respecter $K_V < V/LI(\text{infini})$. Nous choisirons $K_V = V/3$. La puissance consommée par le signal est alors $P_{CRRC(\text{max})} = V^2/R_{Comp}$ où R_{Comp} est la résistance d'entrée du comparateur à seuil.

Pour une séquence unitaire ($K=1$), la tension de sortie s'écrit :

$$V_{LI}(1) = \frac{1}{4} \cdot \frac{\delta}{T} \cdot \frac{e^2}{9} \cdot V^2 \quad (VII.4.5)$$

La puissance de ce signal élémentaire sera donné par :

Chapitre VII

$$P_{CRRC(K=1)} = \frac{1}{4} \cdot \frac{\delta}{T} \cdot \frac{e^2}{9} \cdot \frac{V^2}{R_{Comp}} \text{ avec } \delta = 5ms, T = 250ms \quad (\text{VII.4.6})$$

Il vient alors que $P_{CRRC(K=1)} = 0,0041 * P_{CRRC(max)}$.

Cependant pour une séquence K quelconque, on sait que la puissance totale du signal sera supérieure à la somme des puissances élémentaires. Pour une estimation typique nous considérons une séquence de K=5 ce qui donne $P_{CRRC(K=5)} \approx P_{CRRC(max)}/50$.

Avec $R_{Comp} = 1,0e5$ Ohms et $V = 1,2V$, on aura :

$$P_{CRRC(max)} = 14,4uW \quad \text{et} \quad P_{CRRC(K=5)} = 0,30uW .$$

Selon les publications trouvées dans les dernières années, nous pourrions envisager d'employer des amplificateurs opérationnels en régime de faible inversion [1]-[9]. Leur consommation énergétique est estimée à 280nW. Cette consommation statique devra être ajoutée à la puissance signal calculée ci-dessus.

iv. *Comparateur et la sortie neurone (I & F)*

La seconde partie du troisième module comporte le comparateur à seuil suivi d'un monostable qui délivre une impulsion en sortie. Cette impulsion sera le signal de déclenchement (firing) émis par le neurone. Pour le comparateur, la contrainte sera le temps de commutation que nous fixons à 1V/10us. Pour une capacité de charge (l'entrée du monostable) de 1pF un courant de 100nA est nécessaire. La puissance consommée sera dominée par la puissance de commutation $P_{comp} = C_{comp} V^2 f_{SR}$ où f_{SR} sera de 4Hz au maximum. Le monostable aura une charge plus importante que l'on peut estimer à 5pF et sa puissance à la commutation est $P_{mono} = C_{mono} V^2 f_{SR}$.

On peut donc capitaliser la puissance consommée par un neurone en distinguant les puissances de commutation des portes digitales (SR, Monostable) de celle des blocs analogiques :

$$\begin{aligned} P_{num} &= P_{SR} + P_{Comp} + P_{Mono} \\ &= C_{SR} \cdot V^2 \cdot f_{SR} + C_{com} \cdot V^2 \cdot f_{SR} + C_{mono} \cdot V^2 \cdot f_{SR} \\ &= (400 + 1 + 5) * (1,2)^2 * 4pW \\ &= 2,339nW \end{aligned} \quad (\text{VII.4.7})$$

$$\begin{aligned}
 P_{analog} &= P_{multip} + P_{memoire} + P_{CRRC} \\
 &= 3V \cdot DR_s + 3V \cdot DR_s + P_{CRRC(k=5)} \\
 &= 3,6\mu W + 3,6\mu W + 0,30\mu W \\
 &= 7500nW \quad \text{avec } DR_s = 1\mu A
 \end{aligned}
 \tag{VII.4.8}$$

La puissance totale consommée est alors :

$$P_{total} = P_{num} + P_{analog} = 7502nW \tag{VII.4.9}$$

On constate que la principale consommation vient de la partie analogique et plus particulièrement de la partie synaptique. Sachant qu'un système neuronale est adaptatif en cours d'apprentissage, la question de la linéarité nécessaire à l'apprentissage (donc au fonctionnement en temps réel car l'apprentissage est « on-line ») peut être remise en cause. Si nous réduisons le « besoin » en linéarité, on pourra réduire la puissance du multiplieur et de la mémoire par 3. D'autre part, si l'on considère des structures fonctionnant en régime de faible inversion, la plage dynamique de courant pourra être réduite d'un ordre de grandeur ($DR_s = 100nA$). Ainsi la puissance de la partie synaptique et donc la puissance totale pourra être réduite d'un facteur 30, ramenant la consommation (estimée) à $(2,339+240+300)nW = 542,339nW$.

Ce bilan de puissance présente un fort impact sur les choix architecturaux des structures des « composants » systèmes. Il doit donc être fait dès la fin du partitionnement et avant la sélection topologique. En ce sens, il doit alimenter l'ensemble des critères de sélection des topologies candidates qui resteront ensuite en liste pour l'étape 4 de la méthodologie présentée au chapitre II.

Le résultat tend à valider la faisabilité d'une implémentation analogique dont les performances fonctionnelle et énergétique sont compatibles avec les objectifs fixés. Une telle implémentation peut donc être une alternative à une implémentation purement digitale.

Cependant les questions qui sont soulevées par le bilan de puissance peuvent aussi induire des remises en question de la fonctionnalité car en réduisant les grandeurs physiques, la question de la précision et du bruit suggère d'approfondir l'étude globale du système. Ce questionnement adresse autant la solution numérique que la solution analogique. C'est ici que l'on commence réellement à ressentir les bénéfices de l'approche PVF, car peu de temps a été mis dans des réalisations suivies de mesures et d'interprétations. Si des changements doivent intervenir ils ne pénaliseront que peu le temps de développement. De plus la méthodologie

Chapitre VII

permet clairement à un concepteur métier débutant de pointer le maximum de points de blocage avant réalisation.

Conclusion

Dans ce chapitre VII, nous avons détaillé les divers aspects de notre méthodologie de conception. Celle-ci s'adresse à des projets industriels sous fortes contraintes économiques (Time-To-Market et First-Time-Success). Elle doit aussi guider les concepteurs tant systèmes que domaines, mêmes débutants, afin d'adresser des projets innovants. La démarche s'appuie sur les étapes décrites au chapitres II.

L'utilisation du prototypage virtuel permet de conduire un partitionnement fonctionnel vers le point charnière entre système et domaine. Notre démarche permet d'extraire les structures fonctionnelles et les spécifications de ces « composants ». Elle permet de valider chaque partitionnement et ceci sur plusieurs domaines de compétences. Cela permet à un projet multidisciplinaire de statuer sur la décision de « Go/ No Go ».

Dans ce chapitre nous avons mis l'accent sur les préparations nécessaires au basculement dans le domaine métier. Il est entendu que les outils métiers sont ici supposés suffisant pour ensuite conduire les développements vers un produit satisfaisant au premier coup. Les questions abordées portent sur l'extraction des spécifications de chaque « composant ». Mais pour répondre à cette demande, nous avons pointé l'élément charnière qui est la sélection topologique. Celle-ci crée le passage d'une approche abstraite (théorique) vers une approche technique (physique). Nous avons donc proposé de construire deux familles de modèles qui prennent leurs sources dans chaque approche. La construction de ces modèles a été introduite et commentée.

Grâce à ces modèles, nous affinons en parallèle, les spécifications techniques pour chaque « composant » et l'ensemble des topologies potentiellement intéressantes pour l'implémentation visée dans l'étape 4. La présélection des topologies est un passage délicat que nous voulons circonscrire pour limiter les dérapages et accroître la garantie de réussite. Notre approche permet tout au moins de conduire les concepteurs à révéler des points d'achoppements futurs et d'introduire les corrections nécessaires avant toute pénalisation.

Notre approche n'utilise pas explicitement les principes du « ReUse », car nous avons ciblé des applications innovantes. Néanmoins nous avons souligné les liens forts que l'on peut développer avec cette approche industrielle qui a par ailleurs faits ses preuves. On notera que

Chapitre VII

tous les modèles théoriques développés sont réutilisables parce qu'ils sont dédiés aux cas généraux.

Basé sur ces topologies proposées, nous pouvons ensuite adjoindre une autre méthode proposée par Dr. Jacques MICHEL pour estimer la robustesse de chaque topologie face aux dérives de fabrications. Cette méthode introduite dans la thèse de mon coéquipier François SCHWARTZ [36] est basée sur le calcul ensembliste. De ce fait, la conception analogique et mixte peut être accélérée en assurant la performance du système complet et accroître les chances de réussites au premier coup.

Références

- [1] E., Lopez-Morillo; R.G., Carvajal; F., Munoz; H., El Gmili; Lopez-Martin, A.; Ramirez-Angulo, J.; Rodriguez-Villegas, E.; , "A 1.2-V 140-nW 10-bit Sigma-Delta Modulator for Electroencephalogram Applications," *Biomedical Circuits and Systems, IEEE Transactions on* , vol.2, no.3, pp.223-230, Sept. 2008
- [2] Koushaeian, L.; Amirkhanzadeh, R.; Zayegh, A.; , "Dynamic biasing current source for highly linear OTA with constant transconductance and wide input common—mode range," *TENCON 2008 - 2008 IEEE Region 10 Conference* , vol., no., pp.1-4, 19-21 Nov. 2008
- [3] Shuenn-Yuh Lee; Chih-Jen Cheng; , "Systematic Design and Modeling of a OTA-C Filter for Portable ECG Detection," *Biomedical Circuits and Systems, IEEE Transactions on* , vol.3, no.1, pp.53-64, Feb. 2009
- [4] Lopez-Morillo, E.; Carvajal, R.G.; ElGimili, H.; Ramirez-Angulo, J.; Lopez-Martin, A.; Rodriguez-Villegas, E.; , "A Very Low-Power Class AB/AB Op-amp based Sigma-Delta Modulator for Biomedical Applications," *Circuits and Systems, 2006. MWSCAS '06. 49th IEEE International Midwest Symposium on* , vol.2, no., pp.458-462, 6-9 Aug. 2006
- [5] Zadeh, A.E.; , "Nano-power switched-capacitor bandpass filters for medical implantable pacemakers and defibrillators," *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*, vol., no., pp.29-32, 10-13 Aug. 2008
- [6] Ferreira, L.H.C.; Pimenta, T.C.; Moreno, R.L.; , "An Ultra-Low-Voltage Ultra-Low-Power CMOS Miller OTA With Rail-to-Rail Input/Output Swing," *Circuits and Systems II: Express Briefs, IEEE Transactions on* , vol.54, no.10, pp.843-847, Oct. 2007
- [7] Dresher, R.P.; Irazoqui, P.P.; , "A Compact Nanopower Low Output Impedance CMOS Operational Amplifier for Wireless Intraocular Pressure Recordings," *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE* , vol., no., pp.6055-6058, 22-26 Aug. 2007
- [8] Arnaud, A.; Fiorelli, R.; Galup-Montoro, C.; , "Nanowatt, Sub-nS OTAs, With Sub-10-mV Input Offset, Using Series-Parallel Current Mirrors," *Solid-State Circuits, IEEE Journal of* , vol.41, no.9, pp.2009-2018, Sept. 2006
- [9] Chanapromma, C.; Daoden, K.; , "A CMOS fully differential operational transconductance amplifier operating in sub-threshold region and its application," *Signal Processing Systems (ICSPS), 2010 2nd International Conference on* , vol.2, no., pp.V2-73-V2-77, 5-7 July 2010
- [10] Babanezhad, J.N.; Temes, G.C.; , "A 20-V four-quadrant CMOS analog multiplier," *Solid-State Circuits, IEEE Journal of* , vol.20, no.6, pp. 1158- 1168, Dec 1985
- [11] C.-C., Chang ; S.-I., Liu ; , "Weak inversion four quadrant multiplier and two quadrant divider," *Electronics Letters*, vol.34; n°:22; pp.2079-2080; oct. 1998

Chapitre VII

- [12] C.-Y., Wu; C.-H., Cheng; , "A new analog multiplier-divider with compact structure for CMOS neural network applications," in *Proc. 1st Asia Pacific Conf. on ASICs*, april, 1999, pp. 315–317.
- [13] Valle, M.; Diotalevi, F.; , "An analog CMOS four quadrant current-mode multiplier for low power artificial neural networks implementation," in *Circuit Theory and Design, 2001. ECCTD 2001. European Conference on*, Aug. 28-31, 2001, Espoo, Finland.
- [14] Zarabadi, S.R.; Ismail, M.; Chung-Chih Hung; , "High performance analog VLSI computational circuits," *Solid-State Circuits, IEEE Journal of*, vol.33, no.4, pp.644-649, Apr 1998
- [15] Abdelfattah, Khaled M.; Soliman, Ahmed M.; , "A New Approach to Realize Variable Gain Amplifiers," *Analog Integrated Circuits and Signal Processing (AICSP)*, vol.30; pp.257-263; march 2002
- [16] H. Molina; O. Arellano; A. Reyes; L. Flores; J. Moreno & F. Gomez; , "CMOS ANFIS neurofuzzy system prototype," *Instrumentation & developpement*; vol.5; n°2; pp.122-127; july 01
- [17] E. Ibagari; A. Hyogo; K. Sekine;, "A CMOS Analog Multiplier Freee from Mobility Reduction and Body Effet", *AICSP*; vol.25; pp. 281-290; 2000
- [18] I. Baturone; S. Sanchez-Solano & J. Huertas "Cmos design of a current-mode multiplier/divider circuit with applications to fuzzy controllers" *AICSP*; vol.23; pp.199-210; june 2000
- [19] A. Graupner & R. Schuffny "An ultra low power switched-current 2-quadrant multiplier" Conf. ECS'99; sept. 99 Slovakia
- [20] S. Liu & C. Chang "Low-voltage CMOS four-quadrant multiplier" *Electronics letters*; vol.33; n°3; pp.207-208; jan.97
- [21] S. Hsiao& C. Wu "A parallel structure for Cmos Four-quadrant analog multiplier and its application to 2GHz RF downconverter mixer" *IEEE Jour. Solid-State Circuits*; vol.33; n°6; pp.859-869; june 98
- [22] H. Mehrvarz & C. Kwok A novel multi-input floating gate Mos four-quadrant analog multiplier, *IEEE Jour. Solid-State Circuits*; vol.31; n°8; pp.1123-1131; august 96
- [23] S. Liu & C. Chang Cmos analog divider and four-quadrant multiplier using pool circuits, *IEEE Jour. Solid-State Circuits*; vol.30; n°9; pp.1025-1029; sept. 95
- [24] S. Liu & Y. Hwang "Cmos four-quadrant multiplier using bias feedback techniques," *IEEE Jour. Solid-State Circuits*; vol.29; n°6; pp.750-752; june 94
- [25] N. Saxena & J. Clark "A four-quadrant Cmos analog multiplier for analog neural networks," *IEEE Jour. Solid-State Circuits*; vol.29; n°6; pp.746-749; june 94
- [26] B. DeCock; D. Maurissens & J. Cornelis; , "A Cmos pulse-width modulator/pulse-amplitude modulator for four-quadrant analog multiplier" *IEEE Jour. Solid-State Circuits*; vol.27; n°9; pp.1289-1293; sept. 92
- [27] N. Khachab & M. Ismail. "A nonlinear CMOS analog cell for VLSI signal and information processing," *IEEE Jour. Solid-State Circuits*; vol.26; n°11; pp.1689-1699; Nov. 91

- [28] Z.Wang A, "A Cmos four-quadrant analog multiplier with single ended voltage output and improved temperature performance," *IEEE Jour. Solid-State Circuits*; vol.26; n°9; pp.1293-1301; sept.91
- [29] F. Kub ; K. Moon ; I. Mack & F.; , "Programmable analog vector-matrix multiplier," *IEEE Jour. Solid-State Circuits*; vol.25; n°1; pp.207-214; feb.90
- [30] J. Pena-Finol & J. Connelly, "A Cmos four-quadrant analog multiplier using the quarter-square technique" *IEEE Jour. SSC*; vol.22; n°6; pp.1064-1073; dec. 87
- [31] Sawigun, C.; Serdijn, W.A.; , "A nano-power class-AB current multiplier for energy-based action potential detector," *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on* , vol., no., pp.417-420, 23-27 Aug. 2009
- [32] Sawigun, C.; Serdijn, W.A.; , "Ultra-low-power, class-AB, CMOS four-quadrant current multiplier," *Electronics Letters* , vol.45, no.10, pp.483-484, May 7 2009
- [33] Schwartz, F.; Sun, Q.; Michel, J.; Hervé, Y.; , "A robustness-oriented design tool for the topology selection in analog synthesis," in *Proc. International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD)*, Tunis-Gammarth, Tunisia, Oct. 5-6, 2010
- [34] Schwartz, F.; Sun, Q.; Michel, J.; Hervé, Y.; , "An OTA sizing method using interval analysis," *presented at the. XXIIIth Conference on Design of Circuits and Integrated Systems (DCIS'08)*, Grenoble, France, Nov. 12–14, 2008.
- [35] Levi, T. ; "Méthodologie de développement d'une bibliothèque d'IP-AMS en vue de la conception automatisée de systèmes sur puces analogiques et mixtes" Ph.D. dissertation, Université Bordeaux I, 2007
- [36] Schwartz, F.; "Méthodologie de conception d'un système analogique massivement parallèle. Application à la conception de réseaux de neurones travaillant en mode courant." Ph.D. dissertation, Université de Strasbourg, 2010
- [37] McConaghy, T.; Palmers, P.; Steyaert, M.; Gielen, G.G.E.; , "Variation-Aware Structural Synthesis of Analog Circuits via Hierarchical Building Blocks and Structural Homotopy," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.28, no.9, pp.1281-1294, Sept. 2009
- [38] Palmers, P.; McConnaghy, T.; Steyaert, M.; Gielen, G.G.E.; , "Massively multi-topology sizing of analog integrated circuits," *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE '09.* , vol., no., pp.706-711, 20-24 April 2009
- [39] "Energy density" http://en.wikipedia.org/wiki/Energy_density#cite_note-21
- [40] "Properties of non-rechargeable lithium batteries". <http://www.corrosion-doctors.org/PrimBatt/table2.htm>
- [41] "Battery energy storage in various battery types". <http://www.allaboutbatteries.com/Battery-Energy.html>
- [42] "Performance Specifications of the Lithium Batteries". http://www.globalspec.com/Specifications/Electrical_Electronic_Components/Batteries/Lithium_Batteries
- [43] "Lithium battery". http://en.wikipedia.org/wiki/Lithium-manganese_battery
- [44] V. S. Mallela, V. Ilankumaran, and N.S. Rao,; "Trends in Cardiac Pacemaker Batteries," *Indian Pacing Electrophysiol. J.* 2004; 4(4): 201-212

Chapitre VII

- [45] S. Liu and L. W. Nagel, "Small-signal MOSFET models for analog circuit design," *IEEE J. Solid-State Circuits*, vol. SC-17, pp.983-998, Dec. 1982.
- [46] S. Wong and C. A. Salama, "Impact of scaling on MOS analog performance," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 106-114, Feb. 1983.
- [47] R. Castro-Lopez et al., *Reuse-Based Methodologies and Tools in the Design of Analog and Mixed-Signal Integrated Circuits*, Springer, 2006.

Conclusions générales et perspectives

Conclusions générales

Avec l'augmentation importante du nombre de patients souffrant d'insuffisance cardiaque en Europe et dans le monde entier ces dernières années, le pacemaker basé sur la thérapie de resynchronisation cardiaque (CRT-P), comme traitement efficace pour ces patients, a été mise au point non seulement pour améliorer les fonctions cardiaques, mais aussi pour réduire considérablement le risque de mort subite. En vue d'une amélioration adaptée à chaque patient sur la performance du CRT-P, le projet ADAPTER, regroupant trois entreprises (ELA Medical, AI Semiconductor, Systems'ViP) et deux laboratoires universitaires (LTSI, InESS), a envisagé d'aborder une solution auto-adaptative basée sur l'implémentation d'un processeur de réseau de neurones impulsionnels dans un CRT-P de nouvelle génération. L'objectif premier de cette thèse fut de conduire une étude de la faisabilité d'une alternative en version analogique du processeur neuronal du CRT-P. Ce travail fut une partie du projet ADAPTER et devait apporter une réponse comparative au pendant numérique développé initialement par AI Semiconductor.

L'objectif sous jacent fut d'analyser les méthodes de conception dans un cadre interdisciplinaire depuis une première description du produit jusqu'au démarrage effectif des étapes de réalisations physiques conduisant à un premier produit industriel. Le cheminement est long et complexe du fait de l'innovation. Les heuristiques sont souvent faibles pour guider un jeune ingénieur vers LA solution « idéale » si on peut la définir. Pour planifier, guider et assister les concepteurs, nous avons effectué ce travail en quatre phases de développement correspondant aux différents buts de cette thèse :

1. Développer des « modèles structurels comportementaux » de haut niveau d'un processeur neuronal analogique en langage VHDL-AMS, permettant de l'intégrer dans la plateforme de simulation sous VHDL-AMS. Ces modèles découlent du partitionnement hiérarchique en blocs, modules, composants, etc. justifiant l'appellation « structurelle ». La confrontation des modèles de sous ensembles analogique et numérique a permis d'affiner le partitionnement du CRT dans sa globalité puis dans ces détails jusqu'à un niveau de « composants » systèmes. Ces « modèles structurels comportementaux » sont au final nos « entités » en VHDL-AMS de nos « composants » systèmes. L'avantage principale de cette phase est de pouvoir consolider le partitionnement du système au sein d'une collaboration

multidisciplinaire et offrant une garantie de cohérence et de compatibilités entres sous ensembles ainsi qu'une vision globale du produit ou du projet.

2. Réaliser des « modèles comportementaux » pour chaque « composants » du processeur neuronal analogique. Il s'agit des « architectures » en VHDL-AMS décrivant les « composants ». L'élaboration de ces modèles s'est appuyée sur la méthodologie de conception analogique et mixte basée sur le « Prototypage Virtuel Fonctionnel ». L'avantage majeur de cette phase est de mettre en place les fonctionnalités techniques qui devront être réalisées sans avoir à réaliser physiquement ces « composants ». Cela permet une meilleure exploration des solutions fonctionnelles, d'identifier des points de blocage (éventuellement futurs si une heuristique du « composant » existe) afin de pouvoir satisfaire aux contraintes de Time-To-Market et First-Time-Success.
3. Extraire un jeu de spécifications, techniques et fonctionnelles, propre à chaque « composant » du processeur neuronal. Cette phase s'est appuyée sur les simulations du système complet au travers de la plateforme commune de simulation. L'homogénéité des simulations découle du langage unique employé dans les modèles de chaque ensemble et de chaque niveau hiérarchique du partitionnement. La cohérence de la simulation permet, entre autre, de définir des spécifications « nécessaires », c'est à dire d'éviter de sur contraindre des « composants » en prévision de problèmes futurs car non détectés. La « justesse » des spécifications extraites est la contribution essentielle de cette phase.
4. Renforcer la méthodologie de conception analogique et mixte lors du basculement de la conception système vers la conception domaine (conception par métier). Dans le domaine de l'électronique, l'entrée dans les étapes de conceptions de circuits passe par la sélection d'une topologie qu'il faudra dimensionner puis transcrire en dessin des masques. Il est clair pour tout concepteur en électronique que le choix d'une topologie est déterminant pour la réussite de l'implémentation fonctionnelle visée. Dans le contexte industriel et pour un concepteur novice (tout au moins pour des « composants » spécifiques), cette étape peut conduire à une procédure de type « essais – erreurs ». Ce type de procédure introduira de nombreuses itérations et pénalisera le critère de « Time-To-Market ». Notre objectif a donc été de rassembler le maximum d'informations afin de guider l'étape de sélection topologique. Cette étape est en fait au croisement entre une vision abstraite et une connaissance

Chapitre VII

physique d'un « composant ». De la vision abstraite nous avons extrait le jeu de spécifications qui est peu ou mal adapté à la réalisation physique. Nous avons ensuite introduit une « exploration architecturale » (ou topologique) afin d'identifier des topologies candidates. De cet ensemble de topologies candidates, nous pouvons affiner les descriptions théoriques des « composants » et réajuster le jeu de spécifications afin de faire converger le jeu de spécifications vers les contraintes physiques ou technologiques liées à une topologie ou tout au moins à un ensemble réduit de topologies candidates. Cette phase de la méthodologie doit aussi être associée au travail de mon équipier François SCHWARTZ qui a proposé de construire une estimation de la robustesse face aux dérives technologiques pour chaque « composant ». En respectant cette estimation, nous pouvons garantir que la réalisation conduira à un rendement de production de 100%. Ce résultat sera associé à chaque topologie candidate et participera à la sélection topologique finale permettant d'identifier LA topologie à dimensionner. Cette phase vise à répondre aux deux contraintes industrielles : Time-To-Market et First-Time-Success.

Déroulement de la thèse

- ◆ Dans un premier temps, le travail a débuté par l'étude du processeur neuronal numérique de notre partenaire israélien (AI Semiconductor) et les principes fondamentaux du contrôle du cœur. Ceci nous a permis d'étudier plus finement les besoins médicaux et techniques du pacemaker auto-adaptatif du projet ADAPTER. Selon les rapports cliniques d'ELA Medical, ces besoins (médicaux et techniques) ont évolué au cours du projet. En considérant d'une part que toute modification fonctionnelle d'un réseau neuronal analogique va induire un accroissement du temps de développement et d'autre part qu'un réseau de neurone a une potentialité d'adaptation de sa fonctionnalité, nous avons proposé de figer la partie centrale du processeur analogique à savoir le réseau SNN1. Après l'acceptation de la proposition, nous avons développé avec succès un « modèle comportemental » de ce réseau de neurones. Le développement fonctionnel a d'abord été établi sous Matlab® et puis transposé et validé en langage VHDL-AMS.
- ◆ Dans la deuxième étape, nous avons développé le « modèle structurel comportemental » du processeur neuronal analogique. En élaborant un seul réseau de

neurones analogiques fonctionnant dans les deux premiers modes de fonctionnement du processeur, nous avons réduit la surface de silicium et aussi la consommation d'énergie du processeur neuronal analogique par rapport au pendant numérique. En adoptant les algorithmes d'apprentissage de Hebb et par renforcement, le processeur neuronal analogique peut aider à optimiser les délais auriculo-ventriculaires et inter-ventriculaires afin de maximiser le volume d'éjection systolique à chaque battement du cœur.

- ◆ Durant la troisième étape, les performances du processeur neuronal analogique ont été testées par simulation au niveau système avec un modèle simplifié du cœur et en fonction du rythme cardiaque. Pour accroître la pertinence des analyses, plusieurs améliorations ont été introduites. La première amélioration fut l'introduction des modèles de bruit permettant d'analyser le comportement du réseau de neurone dans chaque mode de fonctionnement où les règles d'apprentissage sont très différentes. La seconde amélioration résulte d'observations cliniques d'ELA Medical sur la « forme » des surfaces SV de différents patients et pour différentes situations physiologiques. Nous avons ainsi créé trois surfaces SV évoluant avec le rythme cardiaque et possédant un maximum ou plusieurs maxima. Nous avons testé avec succès l'ensemble du système avec ces modèles. Nous pouvons constater que les deux algorithmes utilisés permettent d'optimiser le volume d'éjection sanguin en temps réel et le système converge toujours vers le maximum (mais sans être certain que ce soit le maximum global) même en présence de bruit. Dans certains cas, le bruit peut être utile pour aider le système à quitter le maximum local.
- ◆ Après les étapes de validations fonctionnelles et de validation du partitionnement en domaines puis en « composants » domaines, nous avons démarré la recherche de solutions topologiques analogiques pour la réalisation du processeur. Nous avons illustré cette problématique en considérant le « composant » multiplieur analogique propre à chaque synapse. Ce « composant » concentre plusieurs difficultés et représente donc un cas d'école difficile plus démonstratif qu'un « composant » trivial. Le multiplieur synaptique doit être linéaire car il conditionne la réponse du neurone. Il délivre un signal qui doit ensuite être ajouté aux autres réponses synaptiques. Compte tenu du nombre de synapse, sa consommation doit être drastiquement réduite. La linéarité pose deux questions : sa définition pour un « composant » à deux entrées et une sortie et son impact réel sur la réponse du

Chapitre VII

neurone sachant que l'apprentissage peut compenser tout ou partie des défauts internes. La sommation des réponses synaptiques justifie le recours aux courants pour limiter la surface de silicium et la puissance consommée par un additionneur. La puissance révèle l'importance des dynamiques des signaux sur la polarisation, les régimes de fonctionnement des transistors et la stratégie d'alimentation des « composants » du neurone. Au travers de ces quelques aspects, nous voulions souligner l'impact qu'ils auront sur la sélection topologique. En ce sens nous avons considéré ce composant comme démonstratif de notre approche méthodologique. L'objectif de cette quatrième étape fut donc d'extraire les spécifications fonctionnelles et techniques de chaque « composant » en nous focalisant ici sur le multiplieur analogique. Afin de disposer de critères d'acceptation des spécifications fonctionnelles nous avons établi une liste de facteurs de qualités du système. Pour affiner les spécifications techniques, nous avons proposé plusieurs modèles théoriques permettant d'introduire d'éventuelles imperfections. L'introduction des imperfections dans les simulations du système a permis de converger vers un jeu pertinent de spécifications pour ce « composant ». Selon ces plages des spécifications acceptables au sens des performances attendues du système, nous avons pu lancer une étude bibliographique afin d'étoffer un ensemble de présélection avec des topologies potentiellement intéressantes.

Apports originaux de ce travail:

- ✓ A notre connaissance, nous avons implémenté pour la première fois un réseau de neurones à impulsions analogiques, dans un pacemaker CRT. Ce réseau de neurones analogiques très compact (avec seulement deux neurones) peut réaliser des objectifs différents dans deux modes de fonctionnement. Cette conception nous permet de réduire à la fois la surface de silicium et la consommation énergétique. Cette étude d'implémentation analogique avec ses deux algorithmes associés ont été publiés dans plusieurs conférences internationales (dont ISCAS 2010) et ont été acceptées sous réserve de révisions mineures par la revue « IEEE Transaction on Neural Networks » (impact factor–2009 JCR Science Edition : 2.889 ; 5-years impact factor : 3.902).
- ✓ Illustration positive de l'intérêt de l'approche basée sur le « Prototypage Virtuel Fonctionnel ». Partant d'une description très abstraite d'un hypothétique produit,

notre démarche méthodologique a permis de conduire la conception vers les premières étapes de la conception domaine en offrant une décomposition du système cohérente. Nous avons ensuite travaillé sur l'extraction de spécifications pertinentes en accord avec la présélection de topologies.

- ✓ En associant ce travail à celui exposé dans la thèse de mon équipier François SCHWARTZ, nous avons proposé un renforcement de la méthodologie de conception et plus particulièrement un renforcement de l'articulation système – domaine en améliorant et sécurisant la sélection d'une topologie.

Les questions introduites par ce travail ou qui sont restées ouvertes

La conception du réseau de neurones analogiques (SNN1) et l'étude topologique menée sur le multiplieur analogique ont permis de formaliser les interrogations suivantes:

- la présélection des topologies (ici par bibliographique) et l'analyse de la robustesse (proposée par F. SCHWARTZ) dépendent toutes des spécifications techniques. Cela pose la question : Au vue de la présélection topologique et de l'analyse de robustesse, peut-on envisager de redéfinir les spécifications « composant »? Dans l'affirmative, cela permettrait de relâcher certaines contraintes si cela est possible et en resserrer d'autres plus critique. Les choix topologiques seront donc plus pertinents. Selon cette option, nous pourrions facilement affirmer qu'un tel « composant » doit être créé, ce qui est un métier spécifique.
- le multiplieur analogique présente un comportement non linéaire pour les grandes dynamiques de signal. Mais s'il est à l'intérieur d'une boucle de contre réaction comme c'est le cas dans un réseau de neurones, comment peut-on définir les vrais besoins en précision et en linéarité du multiplieur ? Ceci est corrélé à une seconde question : qu'elle doit être la plage dynamique des poids synaptique pour atteindre les objectifs fonctionnels sachant que le multiplieur est imparfait ? On constatera que l'une dépend de l'autre avec comme élément commun la dynamique des signaux. Ce point devrait être approfondi.
- les critères de « bon fonctionnement » d'un réseau de neurones ne sont pas faciles à définir. Les spécifications fonctionnelles dépendent tantôt des signaux analogiques (précisions, linéarité, bruit, etc.) tantôt des algorithmes de

Chapitre VII

corrections des poids synaptiques. Ces derniers sont assez typiquement des algorithmes numériques. Une question intéressante est : quelle partie (analogique ou numérique) doit être la plus « souple » en termes de conception ? Un algorithme figé peut être implémenté par une machine d'état utilisant très peu de silicium et de puissance, mais reportant sur l'analogique les contraintes d'adaptabilité. Il devient difficile de définir les bons critères de validation pour l'implémentation d'un réseau de neurones. Ainsi l'extraction des spécifications les plus cohérentes est ouverte à discussion.

- Comme nous l'avons souligné à différentes reprises, la sélection topologique est au croisement des approches théoriques (abstraites) et des approches physiques (techniques). Pour créer la jonction de ces deux approches et optimiser la sélection topologique, il nous faut établir le lien entre ces deux approches. La question que cela soulève est : quel sont les liens quantifiables, s'ils existent, entre les modèles théoriques du « composant » et ses modèles physiques pour une technologie donnée? Si nous pouvons formaliser ces liens, les réponses quantitatives seraient utilisables en temps que critère de sélection dans un processus d'optimisation qui retournerait LA topologie optimale. Notre travail a suggéré des pistes qui nous semblent pertinentes. Donner une suite à cette question me semble fondamentale pour l'objectif de Time-To-Market et First-Time-Success.

Perspectives

A la suite du développement du processeur neuronal analogique très basse consommation d'énergie pour assister à la prédiction des délais cardiaques dans un pacemaker de thérapie de resynchronisation cardiaque (CRT-P) en appliquant la méthodologie renforcée de conception analogique et mixte à la base de Prototypage Virtuel Fonctionnel, certains points d'étude devront être effectués pour améliorer les modèles et la méthodologie de conception en cours.

Comme prévu dans le projet ADAPTER, le nouveau CRT-P fonctionne avec trois modes de fonctionnements. Après avoir validé l'implémentation des deux premiers modes via les simulations du système complet avec un modèle simple du cœur, la validation du système complet basé sur des modèles plus détaillés du cœur et du réseau de neurones pour le troisième mode de fonctionnement (Adaptatif-optimal) devra être effectuée prochainement tant par de nouvelles simulations que par l'expérimentation sur des cochons vivants.

En même temps, la dernière étape portera sur le dimensionnement final des structures analogiques. L'intégration d'une nouvelle méthode basée sur le calcul par intervalles devrait permettre d'automatiser la conception d'un réseau analogique tout en ouvrant l'espace des solutions possibles et en maîtrisant le critère de robustesse des topologies choisies.

Publications et communications associées à ce travail

Les résultats obtenus ont été décrits dans les publications et communications suivantes

Revue internationale

Sun, Q.; Schwartz, F.; Michel, J.; Hervé, Y.; Dal molin, R.; , “Implementation study of an analog spiking neural network for assisting cardiac delay prediction in a cardiac resynchronization therapy device,” *IEEE Transaction on Neural Networks*, article acceptée pour publication en Février 2011.

Conférences internationales à comité de lecture

Schwartz, F.; **Sun, Q.**; Michel, J.; Hervé, Y.; , “A robustness-oriented design tool for the topology selection in analog synthesis,” in *Proc. International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD)*, Tunis-Gammarth, Tunisia, Oct. 5-6, 2010,

Sun, Q.; Schwartz, F.; Michel, J.; Hervé, Y.; , “A reinforcement learning algorithm used in analog spiking neural network for an adaptive cardiac resynchronization therapy device,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS 2010)*, Paris, France, May 30–Jun. 2, 2010, pp. 2546–2549,

Schwartz, F.; **Sun, Q.**; Michel, J.; , “An OTA sizing method using interval analysis,” *presented at the. XXIIIth Conference on Design of Circuits and Integrated Systems (DCIS'08)*, Grenoble, France, Nov. 12–14, 2008.

Sun, Q.; Schwartz, F.; Michel, J.; Rom. R.; , “Implantation study of an analog spiking neural network in an auto-adaptive pacemaker,” in *Proc. Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference (NEWCAS-TAISA'08)*, Montréal, Canada, Jun. 22-25, 2008, pp. 41–44.

Conférences francophones à comité de lecture

Sun, Q.; Schwartz, F.; Michel, J.; Hervé, Y.; , “Implémentation d’un algorithme d’apprentissage par renforcement dans un réseau neuronal impulsionnel pour l’aide à la décision dans un pacemaker auto adaptatif,” in *Proc. XIIIèmes Journées Nationales du Réseau Doctoral de Microélectronique (JNRDM'2010)*, Montpellier, France, Jun. 7-9, 2010.

Sun, Q.; Schwartz, F.; Michel, J.; , “Etude sur l’implémentation d’un réseau neuronal impulsionnel analogique pour l’aide à la décision dans un pacemaker auto adaptatif,” in *Proc. XIIèmes Journées Nationales du Réseau Doctoral de Microélectronique (JNRDM'2009)*, Lyon, France, May 18-20, 2009.

Annexes

Annexe I. Les langages de modélisation mixte multi-domaines

Pour réaliser une conception performante en appliquant notre méthodologie de conception analogique et mixte à base de PVF, un langage de modélisation analogique et mixte multi-domaines et un outil CAO doivent être choisis avec soin pour répondre aux caractéristiques suivantes :

- Co-simulations des abstractions
- Vrai support du mixte (temps continu et discret)
- Normalisé (indépendance fournisseur/université)
- Multi-abstractions
- Multi-disciplines
- Compatible avec les outils anciens (SPICE)
- Accès à la physique (ODE : équations différentielles ordinaires / DAE : équations algébriques différentielles) et à la synthèse logique
- Capacité de gérer les discontinuités

Actuellement, nous pouvons trouver plus de 300 outils/langages/packages/bibliothèques pour la simulation et plus de 30 langages pour la simulation en temps continu et multi-physique. Parmi eux, nous trouvons six langages ayant plutôt de bonnes caractéristiques pour la méthodologie basée sur le PVF : MAST, MATLAB/Simulink, MODELICA, Verilog-AMS, VHDL-AMS, SystemC-AMS.

Annexe I.1 VHDL-AMS

À l'origine, VHDL (Very High Speed Integrated Circuit Hardware Description Language) est un langage puissant normalisé en 1987 (IEEE 1076) pour des descriptions et modélisation des circuits numériques. Complété et enrichi en 1993 (IEEE 1076-1993 et IEEE

1164-1993), en 2000 (IEEE 1076-2000) et plus récemment en 2008 (IEEE 1076-2008), avec ce langage, les outils CAO permettant de passer directement d'une description fonctionnelle VHDL à un schéma en porte logique ont effectué la révolution des méthodes de conception numérique.

Afin de gérer des modélisations des systèmes mixtes (numériques et analogiques) et de répondre aux exigences des industries citées dans l'introduction de ce chapitre, une extension de la norme IEEE 1076-1993 pour supporter des descriptions et des simulations des circuits et systèmes analogiques et mixtes est présenté en 1999 (IEEE 1076.1-1999) [1]. L'ensemble de ces deux normes (IEEE 1076-1993 et IEEE 1076.1-1999) est informellement appelé VHDL-AMS [2]. La dernière version de norme VHDL-AMS est la norme IEEE 1076.1-2007.

Le langage VHDL-AMS, un langage non-propriétaire et normalisé par l'IEEE, semble être très approprié pour répondre aux exigences de notre méthodologie de conception analogique et mixte en présentant ses multi-capacités [2] [3]. Premièrement, la normalisation du VHDL-AMS permet au plus grand nombre de personne de réaliser et communiquer les projets indépendamment des fournisseurs de logiciels choisis, évitant largement des problèmes de dépendances de logiciel dans des entreprises. De ce fait, les entreprises auront plus de liberté sur le choix du logiciel. Les partenaires du projet travaillant avec différents simulateurs VHDL-AMS peuvent facilement échanger des modèles et faire des coopérations sans perdre du temps sur les traductions des modèles.

La capacité native du VHDL-AMS pour la simulation multi-domaine facilite la communication et la coopération des partenaires dans des projets multi-domaines comme pour le projet ADAPTER. Des terminaux de connexion sont liés à des grandeurs physiques respectant les lois de *Kirchhoff* généralisées [10]. Tant que c'est un système conservatif, nous pouvons donner n'importe quelle unité physique aux quantités analogiques dans des terminaux. Grace à ce langage multi-domaine, un électronicien ou un mécanicien peut modéliser une partie de son domaine dans un dispositif complexe sans problème de communication avec les autres parties.

VHDL-AMS permet également des simulations mixtes avec des modèles de multi-abstractions. Il permet non seulement d'effectuer des modèles des hauts niveaux d'abstraction (comportemental) afin de valider rapidement la fonctionnalité du système, mais aussi de simuler des modèles structurels de bas niveaux d'abstraction avec des détails techniques. De plus, l'utilisation du langage VHDL-AMS nous permet de simuler le comportement en temps

Chapitre VII

discret (VHDL) et le comportement en temps continu avec des équations algébriques différentielles et/ou ordinaires (ODE, DAE) en même temps. Il gère de la façon très précise la discontinuées mathématiques.

Annexe I.2 MAST

MAST est un langage propriétaire proposé par la société *Analogy* (actuellement *Synopsys*) en 1984 et a été intégré dans leur simulateur *Saber* [11]. Comme le premier langage réussi de description comportementale orienté système pour la simulation analogique et mixte, MAST a dominé le marché des langages de description de matériel à signaux et technologies mixtes. Ce langage et l'outil *SABER* ont été largement utilisés dans les entreprises automobiles et aéronautiques.

Cependant ce langage ne peut pas être choisi dans la méthodologie renforcée à cause des inconvénients suivants :

- Ce langage est exclusif à l'outil *SABER*, il n'est pas normalisé
- Il supporte mal des systèmes numériques et il ne peut pas faire la synthèse logique.

Afin de répondre au développement rapide de son concurrent VHDL-AMS, le langage d'*initiative OpenMAST* a été développé [12]. Malheureusement, il n'apporte pas d'effets effectifs sur l'offre commerciale.

Annexe I.3 MATLAB/Simulink

Le langage MATLAB[®] a été développé par Cleve Moler à la fin des années 1970 à partir des bibliothèques Fortran, LINPACK et EISPACK [13]. La société *Math Works* a été créée en 1984 et a commercialisé la version 1.0 de MATLAB[®] dans la même année, comme un interpréteur de commande pour accéder aux bibliothèques LINPACK et EISPACK. Aujourd'hui, le MATLAB[®] (Matrix laboratory) est devenu un langage technique informatique de haut niveau et un environnement interactif pour le développement d'algorithmes, la visualisation de données, l'analyse des données et le calcul numérique. En utilisant MATLAB[®], nous pouvons résoudre les problèmes techniques de calcul plus

rapidement qu'avec des langages traditionnels de programmation, tels que C, C++ et Fortran [14].

Avec le code facile à interpréter, les mécanismes simples et les facilités d'accès fournis par MATLAB[®], nous pouvons implémenter des algorithmes, créer et manipuler des matrices, générer et tracer des figures en 2D ou 3D pour une meilleure visualisation, créer des interfaces d'utilisateurs personnalisées et également s'interfacer avec des autres programmes tel que C, C++, Fortran, Java, COM, et Microsoft Excel. Avec ces avantages, MATLAB[®] est devenu le standard de fait dans plusieurs domaines, tel que le contrôle et le traitement du signal numérique. Les applications de MATLAB[®] peuvent être largement complétées par l'utilisation des multiples boîtes à outils appelées *add-on toolboxes*. Parmi les plus importantes, nous trouvons :

- *Communications Toolbox*
- *Control System Toolbox*
- *Excel Link*
- *MATLAB Compiler*
- *Neural Network Toolbox*
- *Optimization Toolbox*
- *Parallel Computing Toolbox*
- *Real-Time Workshop*
- *Robust Control Toolbox*
- *SimMechanics*
- *SimPowerSystems*
- ***Simulink***
- *Statistics Toolbox*
- *System Identification Toolbox*
- *Virtual Reality Toolbox*

L'outil *Simulink*[®] est un environnement pour la simulation multi-domaine et la conception à la base des modèles pour les systèmes dynamiques et intégrés sous forme *SIGNAL FLOW* [15]. Il fournit un environnement graphique interactif et un ensemble de

Chapitre VII

bibliothèques de blocs personnalisables qui nous permettent de concevoir, simuler, mettre en œuvre et tester une variété de variables dans le temps des systèmes, y compris des communications, des commandes, des traitements du signal, de vidéo et d'image.

Afin de compléter les domaines d'application, *Simulink*[®] peut être couplé avec les sous boîtes à outils dans sa famille, tel que *Simscape* pour des systèmes physiques, *SimElectronics* pour des systèmes électroniques et électromécaniques, *SimMechanic* pour des systèmes mécaniques, etc. Ces outils nous permettent d'effectuer des diagrammes spécialisés. Avec l'outil *Real-Time Workshop*, il est possible de convertir des modèles en code C. Les inconvénients principaux de MATLAB[®]/Simulink[®] sont décrits dans les phrases qui suivent. Il n'y a pas une norme pour ces langages et ils sont propriétaires. Avec ses capacités performantes en mathématiques, ils sont plutôt des outils orientés mathématique au lieu de système. Comme il n'y a pas de synchronisations explicites dans la simulation, l'outil MATLAB[®]/Simulink[®] n'est pas complet sur des modélisations et des simulations numériques.

Annexe I.4 MODELICA

MODELICA est un langage orienté objet unifié pour la modélisation du système physique. Par conséquent, il est capable de réaliser des simulations multidisciplinaires (par exemple : électronique, thermique, mécanique et hydraulique) [16].

L'association à but non lucratif « Modelica Association » gère le développement de Modelica. La première version de Modelica a été formalisée en 1997 et l'utilisation industrielle de ce langage a débuté en 2000. De plus en plus d'entreprises, notamment des industries d'automobiles, ont adopté ce langage. Le langage, les bibliothèques des modèles multi-domaines et certains outils de simulation de ce langage sont gratuits, prêt à l'emploi. La version la plus récente de Modelica est la version 3.2 sortie en 2010.

L'avantage de Modelica est le mécanisme puissant de type « langage objet ». Les inconvénients sont également évidents. Il n'existe pas une norme.

Nous pouvons trouver plusieurs simulateurs commerciaux ou gratuits de Modelica dans le marché, citons ci-dessous :

- *CATIA Systems* : add-on dans l'environnement 3D, simulateur commercial développé par Dassault Systèmes. CATIA est initialement basé sur le Dymola qui est abordé par Dynasim.
- *LMS Imagine.Lab AMESim* : simulateur commercial développé par LMS international
- *MapleSim* : simulateur commercial développé par MapleSoft, Canada
- *MathModelica* : simulateur commercial développé par MathCore AB, Sweden
- *SimulationX* : simulateur commercial développé par ITI GmbH, Dresden, Allemagne
- *OpenModelica* : simulateur gratuit développé par Linköping university, Sweden
- *SCICOS* : simulateur gratuit développé par INRIA, France

Annexe I.5 Verilog-AMS

Le langage Verilog-AMS est une extension du langage de description matériel Verilog (IEEE 1364) permettant de définir les comportements des systèmes à signaux analogiques et mixtes (à temps continu et à temps discret). Initialement proposé par *Accellera* (Organisation de normalisation EDA : Electronic Design Automation) [17], cette organisation a sorti la dernière version du Verilog-AMS LRM, version 2.3.1, en juin 2009. Cette version remplace le Verilog-A LRM OVI (à partir de juin 1996) et les versions antérieures de Verilog-AMS LRM. La Figure Annex I-1 montre l'architecture signal mixte/ multi-domaine du langage Verilog-AMS, qui comprend la partie numérique (appelé Verilog-D IEEE-1364-1995), la partie analogique (appelé Verilog-A OVI-96) et la partie des extensions de signaux mixtes.

Le langage Verilog-AMS permet aux utilisateurs de décrire et simuler les conceptions à signaux analogiques et mixtes en utilisant une méthodologie de conception descendante (Top-down) ainsi que l'approche traditionnelle (Bottom-up). La norme Verilog-AMS soutient des modèles analogiques et mixtes à trois niveaux: transistor / gate, transistor / porte-rtl / comportementale, et transistor mixte / porte-rtl / niveaux circuit de comportement. En outre, Verilog-AMS offre des capacités puissantes de modélisation comportementales et structurales pour les systèmes dans lesquels les effets des interactions entre les différentes disciplines comme électriques, mécaniques et thermiques sont importantes.

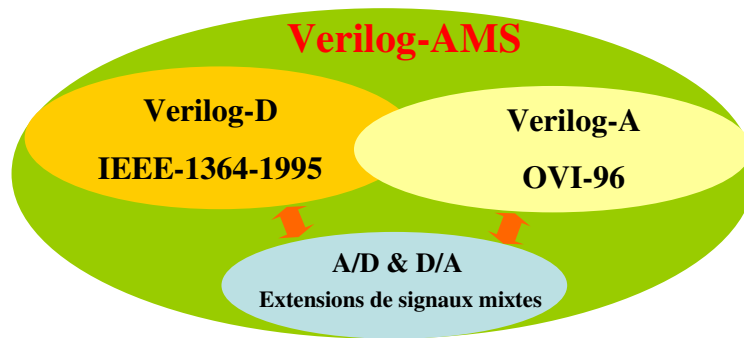


Figure Annex I-1 L'architecture signal mixte/multi-domaine du langage Verilog-AMS [18]

En effet, Verilog-AMS permet d'optimiser les codes à un niveau d'abstraction plus bas, plus proches des circuits. Mais l'inconvénient est qu'il ne permet pas de faire des simulations et des modélisations de haut niveau d'abstraction. Donc, le langage Verilog-AMS est peu utilisé dans la réalité pour les systèmes complexes.

Nous listons ci-dessous quelques simulateurs supportant Verilog-AMS sur le marché :

- *ADVance MS* : développé par Mentor Graphics
- *ModelSim* : développé par Mentor Graphics
- *SaberHDL* : développé par Synopsys

Annexe I.6 Comparaison des langages de modélisation

Le Tableau Annexe I-1 rassemble et compare les principales caractéristiques des cinq langages de modélisation mixte présentés ci-dessus : VHDL-AMS, MAST, Modelica, MATLAB/Simulink et Verilog-AMS.

Selon les études et les comparaisons des caractéristiques de ces langages de modélisation mixte, le langage VHDL-AMS s'adapte mieux aux critères de langage pour notre méthodologie renforcée de conception analogique et mixte listés au début de la section II. 4. Ce langage nous permet d'établir le système du cahier de charges abstraites et simulables jusqu'au niveau le plus bas avec plus de détails technologiques dans un environnement unique. Ses supports sur des modélisations des différentes abstractions (comportementales, structurelles, « signal-flows ») sont des éléments indispensables dans notre méthodologie.

Tableau Annexe I-1 Comparaison de cinq langages de modélisation mixte [7]-[9]

	VHDL-AMS	MAST	Modelica	MATLAB/Simulink	Verilog-AMS
Norme	IEEE 1076.1-2007	Langage Propriétaire lié à l'outil SABER	Modelica Specification 3.2 (Avril 2010)	Langage Propriétaire lié à l'outil MATLAB et ses toolboxes	Accelera standard Version 2.3.2 (Juin 2009)
Présentation	Extension du VHDL (IEEE 1076)	Logiciel propriétaire lié à ??	Langage de modélisation orienté objet	Logiciel de calcul numérique + Simulink + toolbox	Extension de Verilog (IEEE 1364)
Modularité	Entité Active Multi-Architecture	Mono-bloc	Mono-architecture	Mono-architecture	Mono-architecture
Accès à la physique ODE/DAE	OUI/Formes explicites et formes implicites des équations Ordre N	OUI/Formes explicites et support limité des formes implicites Ordre 1	OUI/Formes explicites et formes implicites des équations Ordre 1	OUI/Formes explicites (équation d'état)	OUI/Formes explicites et support limité des formes implicites Ordre 1
Multi-disciplines	OUI	OUI	OUI	OUI	OUI
Conservatif/Flot de signal	OUI/OUI	OUI/OUI	OUI/OUI	OUI/OUI	NON/OUI
Gestion des discontinuités	OUI/Instruction Break	OUI Control Section (complexe)	OUI/ automatique Mécanisme daté	NON	OUI
Dépendance à un outil	NON	OUI	NON	NON	NON
Modélisation de l'environnement	OUI	OUI	OUI	SimPower	
Vrai support mixte	OUI	NON	NON	NON	OUI
Accès à la synthèse logique	OUI	NON	NON	NON	OUI
Compatibles outils anciens	OUI	NON	NON	NON	OUI
Gestion des unités	NON	NON	OUI	NON	NON

Chapitre VII

Du fait de la normalisation VHDL-AMS, permet de réaliser des projets en multi-équipes / multi-sites en prenant un langage commun indépendant des fournisseurs et de la technologie. La grande capacité de ce langage sur des modélisations et des simulations multi-domaines nous donne la possibilité de gérer des projets multi-disciplines en partageant une plateforme commune sans problèmes de dialogue entre des partenaires de différents domaines (électricien, mécanicien, chimiste...).

Le langage VHDL-AMS supporte en même temps des systèmes conservatifs (pour la conception analogique et mixte, c'est la loi de Kirchhoff) et des systèmes « signal-flow » en plus des événements discrets. Les capacités de traitements des équations implicites (ODE et DAE) et l'adoption des lois de *Kirchhoff* généralisées nous permettent de résoudre des relations implicites entre les nœuds différents dans un système complexe.

Un grand atout de VHDL-AMS est la transparence et la réutilisabilité. Chaque concepteur peut modéliser librement son propre système ou modifier les modèles existants pour s'adapter aux besoins spécifiques. Tous les modèles comportementaux ou structurels existants sont archivables afin de construire une bibliothèque de modèles réutilisables comme IP (Intellectual Property). Les modèles des composants fondamentaux analogiques ou mixtes pourraient être réutilisés dans le futur pour implémenter un autre système complexe. De ce fait, nous pouvons augmenter l'efficacité et la productivité de conception. Dans la dernière décennie, beaucoup de laboratoires de recherches et d'industriels ont apporté leurs contributions sur le développement de bibliothèques de modèles en VHDL-AMS.

Un autre atout de VHDL-AMS est d'associer conjointement des modèles comportementaux de haut niveau et des modèles structurels de bas niveau dans le même modèle du système complet. Cette capacité nous permet de nous focaliser plus finement sur certaines parties. Comme les autres parties sont modélisées de façon plus abstraite, nous pouvons étudier l'influence sur les performances de la modification d'un paramètre dans cette partie via la simulation du système complet avec un temps de calcul inférieur [19]. Nous allons montrer une étude sur l'implémentation d'un multiplieur analogique dans le chapitre VII afin de montrer cet atout dans notre méthodologie renforcée de conception analogique et mixte.

En résumé, les multi-capacités du langage VHDL-AMS correspondent bien aux critères du développement d'une méthodologie renforcée de conception analogique et mixte à la base

de PVF pour satisfaire des besoins industriels. De ce fait, nous l'adoptons pour notre méthodologie de conception ainsi que pour le développement du projet ADAPTER.

Annexe II. Outils de simulation VHDL-AMS

Aujourd'hui, nous pouvons trouver plusieurs simulateurs commerciaux dédiés à la simulation du VHDL-AMS. Malheureusement, à notre connaissance, peu d'entre eux supportent totalement la norme IEEE 1076.1. Dans cette section, nous allons citer les simulateurs les plus importants et les plus utilisés sur le marché.

Annexe II.1 Questa ADMS (Mentor Graphics)

Le simulateur Questa ADMSTM (ancien ADVance-MS) [20] est un environnement unifié sous UNIX développé par Mentor Graphics pour la simulation de VHDL-AMS, Verilog-AMS, VHDL, Verilog, SystemVerilog, SPICE et SystemC dans tout niveau de conception. Il possède quatre moteurs de simulation pour des tâches différentes :

- *Questa* : simulateur VHDL/SystemVerilog/SystemC permettant des conceptions et vérifications avec un ou plusieurs langages
- *ELdo* : simulateur analogique de haute performance avec l'avantage de grande vitesse
- *ADiT* : simulateur fast-SPICE pour des applications analogiques et mixtes de niveau transistor
- *Eldo RF* : simulateur pour des systèmes numériques de communication qui comprennent étroitement des RF intégrés avec des signaux analogiques et mixtes et des fonctions DSP

Chapitre VII

Cet outil est très adapté et très utilisé à la simulation de circuits intégrés, mais il ne supporte pas la totalité du langage VHDL-AMS. Par exemple, pour l'initialisation, les instructions « Break » et « Generate » ne sont pas supportées.

Annexe II.2 SystemVision (Mentor Graphics)

SystemVision est aussi un outil de Mentor Graphics. Cet outil permet aux ingénieurs de système et ingénieurs de différentes disciplines de communiquer, de simuler et d'optimiser la conception pour le succès au premier passage [21]. Il fonctionne sous Windows. SystemVision fournit un laboratoire virtuel pour la conception et l'analyse des systèmes analogiques, numériques et signaux mixtes. Il y a aussi une option pour faire l'intégration avec MATLAB/Simulink. Il supporte plusieurs langages normalisés : VHDL-AMS, SPICE, C. Cet outil est utilisé pour des conceptions de systèmes électromécaniques. En possédant une interface graphique, SystemVison supporte des modèles « drag-and-drop » et la création automatique de symboles. L'inconvénient de SystemVison est aussi le support incomplet de la norme du langage VHDL-AMS.

Annexe II.3 SaberHDL (Synopsys)

Cet outil, développé par *Synopsys*, est un environnement de simulation et de modélisation multi-domaine qui permet un prototypage virtuel de l'ensemble du système pour les applications multi-disciplines [11]. SaberHDL est indépendant des langages utilisés et fonctionne sous UNIX, Linux et Windows. Il est capable de simuler et supporter les langages MAST et VHDL-AMS. Cependant, SaberHDL a le même problème que Questa ADMS. Il n'implémente pas l'ensemble de la norme VHDL-AMS : les instructions « Break », « Procédural » et « Quantités Vectorielles » ne sont pas supportées.

Annexe II.4 Simplorer 8.1 (Ansoft)

Simplorer (ancien hAMSter) est un logiciel fonctionnant sous Windows de Ansoft, un leader dans le développement de logiciel pour la conception et la simulation électronique de haute technologie [22]. Les capacités de simulation multi-domaines du Simplorer permettent aux ingénieurs de modéliser, simuler, analyser et optimiser des systèmes complexes. Les applications du Simplorer incluent l'électromécanique, l'électromagnétique, l'électrique et d'autres modèles mécatroniques. Simplorer offre de multiples techniques de modélisation y compris les circuits, schémas, machines d'état, le niveau de l'équation, et langages de modélisation comme VHDL-AMS, SML (Simplorer Modelling Language), et C / C + +. Ceux-ci peuvent être utilisés simultanément permettant aux ingénieurs de facilement faire les conceptions analogiques et signaux mixtes.

Avec l'interface graphique utilisateur du simplorer, les concepteurs peuvent créer une représentation graphique conviviale et intuitive, qui facilite largement l'implémentation de systèmes complexes en associant graphiquement des blocs fonctionnels ou structurels en VHDL-AMS.

Simplorer supporte la plupart de la norme VHDL-AMS (IEEE 1076.1), cependant il ne gère pas l'instruction de quantités vectorielles. Cet outil possède également une bibliothèque VHDL-AMS vaste (open source) mais mal documentées pour des composants variés comme les moteurs, générateurs, circuits (analogiques et numériques), des blocs de contrôle, fluidiques, magnétiques, mécaniques et thermiques. Il est possible d'effectuer la co-simulation avec d'autres langages de simulation spécialisés (ModelSim, QuestaSim, y compris VHDL et Verilog) et des codes propriétaires. Simplorer offre également la possibilité de se relier aux produits Ansoft (Maxwell, Q3D Extractor, RMxprt, PExprt) pour des modélisations des composants électromécaniques et électromagnétiques.

Annexe II.5 Portunus (Groupe CEDRAT)

Portunus est un logiciel unique de la simulation de système de multi-domaine qui rend possible de modéliser un système mécatronique complet dans un même fichier de simulation. Il nous permet de créer facilement des interactions entre plusieurs domaines (électrique, thermique, magnétique, mécanique).

Chapitre VII

Un avantage de ce simulateur est qu'il possède plusieurs bibliothèques standards prêtes à utiliser. Portunus offre différentes approches descriptives (circuit, diagramme de blocs, machine d'état) ainsi que différents types d'analyse (transitoire, harmonique, point de fonctionnement, régime permanent). Ce logiciel peut facilement être utilisé pour communiquer ou échanger des données avec un autre outil métier (par exemple : Simulink®, Motor-CAD® et InCa3D®). En supportant la plupart de la norme VHDL-AMS, ce simulateur est capable de faire des simulations de systèmes multi-disciplinaires. Des concepteurs peuvent aussi effectuer des simulations de multi-niveaux et multi-abstractions pour des conceptions des systèmes complexes. Tous les modèles VHDL-AMS peuvent être gérés par la bibliothèque de Portunus et des échanges de données avec tous les autres modèles sont possibles.

Annexe II.6 SMASH 5.15 (Dolphin Integration)

Actuellement, SMASH, développé par Dolphin Integration, est le simulateur qui supporte le plus complètement la norme VHDL-AMS, assez loin devant les autres simulateurs [23]. Il fonctionne sur des plateformes UNIX et PC (Sun Solais 7 – Linux RHL 3 à RHEL 3 – Windows XP/Vista/Windows 7 32/64 bit).

Un grand atout de cet outil est sa polyvalence très significative sur la compatibilité aux différents langages de modélisation : SPICE, Verilog HDL, Verilog-A, VHDL, SystemC, C, ABCD, et il peut aussi être interfacé avec MATLAB/Simulink. La capacité tout-en-un de SMASH permet d'utiliser un seul noyau pour les simulations logiques, analogiques et mixtes. Il offre également des caractéristiques uniques et brevetées pour suivre les bugs de conception, telles que d'optimiser la conception pour le rendement avec l'analyse de dispersion et d'offset.

Un autre atout est que les utilisateurs peuvent co-simuler des modèles développés par différents langages dans un modèle global. C'est surtout utile pour le développement des projets complexes et multi-domaines. Lorsque les partenaires ne peuvent pas travailler avec le même langage de modélisation comme VHDL-AMS, le simulateur SMASH peut toujours regrouper et co-simuler des modèles écrits par différents langages sans le problème de traduction et de communication. De ce fait, nous pouvons augmenter la productivité et réduire le « time-to-market ». Prenons le projet ADAPTER comme un exemple, notre partenaire rennais (LTSI) a prévu d'intégrer des modèles du cœur développés en langage C. Cette

grande capacité de SMASH a justifié le choix de ce logiciel dans le projet ADAPTER et dans cette thèse.

Annexe III. Synthèse bibliographique sur les neurones de type SNN (Spiking neural network)

Dans les dernières années, la preuve expérimentale a été accumulée à suggérer que les réseaux de neurones biologiques, qui communiquent via les impulsions (spikes) en utilisant les moments de ces impulsions pour encoder et calculer des informations [25]-[27]. Plusieurs hypothèses ont été suggérées que des informations peuvent être codées dans les synchronisations et les oscillations de neurones, ou dans les moments précis de l'impulsion de neurones [28] [29].

Annexe III.1 Physiologie des neurones biologiques

Un neurone typique peut être divisé en trois parties fonctionnelles distinctes : la dendrite, le soma, et l'axone.

Les poids des charges reçus par un neurone sur ses dendrites sont additionnés par un processus non linéaire de sommation spatiale et temporelle [30]. Les signaux d'entrée s'écoulent au soma et souffrent des atténuations de tension, de telle sorte qu'environ la moitié de charges injectées dans la dendrite distale atteignent le soma.

Il y a deux paramètres importants: l'espace constant λ qui donne l'atténuation de la tension avec la distance et le temps constant de membrane τ_m , qui décrit à quelle vitesse la tension décroît.

La fonction première du soma est d'effectuer les maintenances continues nécessaires pour garder les neurones fonctionnels. La partie importante du soma jouant les traitements non-linéaires des signaux d'entrée est de l'axone hillock. Si l'ensemble des signaux d'entrée produit une dépolarisation jusqu'au seuil des neurones, l'axone hillock déclenche un potentiel d'action. Le signal de sortie est transmis à l'axone, qui transmet à d'autres neurones.

Annexe III.1.1 Le potentiel d'action

Le déclenchement d'un neurone, ce qu'on appelle le potentiel d'action, est une réponse de tous ou rien. Les stimuli entrant produisent des potentiels d'action s'ils dépassent la valeur de seuil du neurone, sinon ils ne le font pas.

A noter que chaque neurone a une période réfractaire: même avec les forts signaux d'entrée, il est impossible de produire un second déclenchement pendant ou immédiatement après le premier déclenchement.

La distance minimale entre deux déclenchements définit la période réfractaire absolue de neurones. La période réfractaire absolue est suivie par une phase de réfraction relative où il est difficile mais pas impossible de générer un potentiel d'action.

Annexe III.1.2 Synapse

L'endroit où l'axone d'un neurone pré-synaptique prend en contact avec la dendrite (ou le soma) d'une cellule post-synaptique est la synapse. Lorsqu'un potentiel d'action arrive à une synapse, il déclenche une série complexe de processus biochimiques qui conduit à une libération de transmetteurs du neurone, du terminal pré-synaptique à l'écart synaptique. La réponse du neurone postsynaptique en tension au potentiel d'action du neurone pré-synaptique est considérée comme le potentiel postsynaptique. À la fin de l'action synaptique, le potentiel postsynaptique décroît exponentiellement vers la valeur de repos avec un taux de décroissance donné par la constante de temps τ_m .

Annexe III.2 Les aspects de calcul et de modélisation

L'activité neuronale peut être décrite avec plusieurs niveaux d'abstraction. Plus on prend en compte des détails, plus grandes puissances de calcul le modèle nécessite.

Annexe III.2.1 Modèles Détaillés de neurones

Le modèle proposé par Hodgkin et Huxley, en 1952, offre peut-être l'exemple le plus claire d'un modèle détaillé des processus impliqués dans la génération du potentiel d'action.

Chapitre VII

Elle offre un modèle quantitatif précis des données physiologiques. Un inconvénient majeur de cette famille de modèles est que leur complexité numérique peut empêcher les gens de comprendre les caractéristiques qui sont responsables pour un phénomène particulier et qui ne sont pas pertinents [30].

Pour la construction des modèles détaillés de neurones qui tiennent compte de toutes les complexités des cellules, y compris de la structure du câble ramifié, l'approche standard consiste à diviser le neurone en un nombre fini des compartiments connectés isopotentiels.

Un des arguments principaux en faveur de la modélisation détaillée est que la réplication de clôture des structures du système nerveux comme une base de l'exploration de ses fonctions augmente les possibilités de découvrir ses fonctions inconnues. Les inconvénients viennent de la complexité de ces modèles. C'est à dire qu'ils évoluent mal avec la taille du réseau et leurs activités. Ils représentent une description quantitative, plutôt que d'une description qualitative du comportement de neurones, ce qui peut empêcher les gens de comprendre les caractéristiques essentielles du système.

Annexe III.2.2 Modèles formels de neurones impulsionnels (Spiking neuron)

Une autre recherche en neurosciences computationnelle est orientée vers les modélisations sur la nature d'impulsion des neurones et sur la conservation des éléments essentiels des comportements modélisés. La motivation principale pour la création de modèles simplifiés est qu'ils permettent d'étudier plus facilement les principes de calcul et les comportements fonctionnels de systèmes neuronaux.

La réduction des modèles neuronaux par rapport aux modèles formels requiert des simplifications dans au moins deux aspects. Premièrement, la dynamique non linéaire pour la génération des impulsions doit être réduite à une seule équation différentielle et, deuxièmement, la structure particulière du neurone est négligée et réduite à une seule entrée.

Il est suggéré que l'émission d'un potentiel d'action puisse être décrite par un processus de seuil. Le neurone de type intégré et déclenché (en anglais : integrate & fire) est probablement le meilleur exemple connu d'un modèle de neurones formels. Il simule la dynamique du potentiel de membrane des neurones et répond à un courant synaptique en

mettant à effet un circuit électrique équivalent. La fonction du circuit de type intégré et déclenché est d'accumuler des courants d'entrée et lorsque le potentiel de membrane atteint la valeur du seuil neuronal, il génère un déclenchement. Immédiatement après avoir émis une impulsion, le potentiel est maintenu stable pendant une période réfractaire absolue.

Le modèle de type intégré et déclenché se concentre sur la sommation temporelle de la fonction des neurones. Le modèle de réponse déclenchement proposé par Gerstner [31] simplifie la génération du potentiel d'action à un processus de seuil. Le modèle de type résonné et déclenchement [32] met l'accent sur le fonctionnement du neurone dans un régime de résonance.

En conclusion, il existe d'un problème de compromis qu'un chercheur ait à résoudre, au moment de choisir le niveau de détail approprié pour la tâche à simuler.

Annexe III.3 Communication neuronale avec des déclenchements des impulsions (spikes)

Les séquences des déclenchements sont la langue que le cerveau écoute, la langue que le cerveau utilise pour ses rêveries intérieures, et la langue qu'il parle comme il parle avec le monde extérieur [27].

Il existe deux hypothèses principales à l'égard d'où dans le train de déclenchement l'information neurale est codée: dans le taux de déclenchement neuronal ou dans la répartition précise dans le temps de déclenchement.

Annexe III.3.1 Taux de codage

Adrian introduit le concept de taux de codage [33], par laquelle le nombre des impulsions dans une fenêtre de temps fixée après l'apparition d'un code de stimulus statique pour mesurer l'intensité du stimulus.

Il est important de comprendre que le codage des impulsions représente une extension de la façon dont les neurones codent de l'information, plutôt qu'un remplacement du taux de déclenchements impulsions.

Annexe III.3.2 Codage temporel par des délais relatifs

Dans un code temporel, les informations peuvent être contenues dans les régimes temporels du déclenchement (codages des intervalles entre des déclenchements) ou dans le temps d'arrivée des déclenchements (temps des déclenchements impulsionnels) [34].

Les études neurobiologiques sur les codages sensoriels des stimuli dans les systèmes auditifs et visuels ont révélé que le délai de transmission est un candidat potentiel pour le codage des stimuli. Le délai de la réponse des neurones dans le cortex strié est une fonction de l'impulsion dont le contraste et la synchronisation dépendent du délai, cela peut apporter une contribution importante au contraste contraignant des informations connexes. Le schéma de codage qui représente des informations analogiques codées via des différences sur les moments de déclenchement des différents neurones est considéré comme le codage de délai ou le codage de latence [35]-[40].

Selon [35] et [36], un vecteur de réels (x_1, x_2, \dots, x_n) avec $x_j \in [0,1]$ peut être encodé en temps de déclenchement t_j de n neurones, comme $t_j^{(0)} = T - c \cdot x_j$ où T est le temps de référence et $c \cdot x_j$ représentent les délais de transmission. Le timing peut être défini relativement à un autre déclenchement produit par le même neurone ou au début d'un stimulus. Si l'on ne considère que le délai du premier déclenchement survenu après le stimulus pour chaque neurone, alors on obtient un schéma de codage basé sur le délai au premier déclenchement.

Selon [41], les cellules peuvent fonctionner comme les « convertisseurs de Analog à Délai » C'est-à-dire, le neurone le plus activé aura tendance à déclencher premièrement et signale une forte stimulation, alors que les neurones plus faiblement activés déclencheront plus tard et signaleront des faibles stimulations.

Annexe III.3.3 Propriétés de calcul des neurones impulsionnels

Grâce à l'utilisation de codage temporel de délais, un réseau de neurones impulsionnels (Spiking neural network) peut acquérir plus de puissance informatique que le réseau de neurones traditionnels de taille comparable.

Un modèle simple du réseau de neurones impulsionnels peut effectuer les calculs sur les trains des impulsions d'entrée sous plusieurs modes différents [37]. Les neurones impulsionnels calculent quand les signaux d'entrée sont encodés sous formes temporelles : le taux de déclenchement, les collations temporelles, et les codes du taux d'espace. Une caractéristique essentielle des neurones impulsionnels est qu'elles peuvent agir comme des détecteurs de coïncidence pour les impulsions d'entrée en détectant si elles arrivent presque en même temps [42]-[44].

Lorsqu'il fonctionne dans le mode d'intégration, les taux de sortie changent en fonction des taux moyens d'entrée et elle est indépendante de la structure fine du train d'impulsions d'entrée. En revanche, lorsque le neurone fonctionne comme un détecteur de coïncidence, à la sortie, le taux de déclenchement de sortie est plus élevé si les impulsions d'entrée arrivent simultanément.

Deux contraintes doivent être respectées si le neurone fonctionne comme un détecteur de coïncidence:

- Le potentiel postsynaptique doit évoluer dans le temps selon une fonction de décroissance exponentielle.
- Les délais de transmission doivent avoir les valeurs similaires, de sorte que l'arrivée simultanée du potentiel postsynaptique causant les déclenchements de neurones reflète la coïncidence d'impulsions présynoptiques.

En réseaux de neurones impulsionnels, les neurones peuvent calculer dans deux régimes: mode de détection de coïncidence et mode de seuil-et-déclenchement.

- Agissant comme des détecteurs de coïncidence est plus probable pour les unités cachées, quand ils calculent les impulsions provenant des couches d'entrée.
- Dans le mode de seuil-et-déclenchement, les neurones effectuent une addition linéaire des signaux d'entrée de la même manière pour le modèle intégré et déclenché (integrate & fire).

Annexe IV. Code source VHDL-AMS :

Modèle comportemental structurel

Annexe IV.1 Modèle du pacemaker

```
>>> VHDL
--
--
%%%%
-- Simulation du Pacemaker en version Analogique avec le coeur
-- OU une représentation réuite du coeur qui retourne seulement le Stroke Volume
--
-- Auteur Francois SCHWARTZ, Jacques MICHEL, Qing SUN
-- Laboratoire INESS Strasbourg UMR 7163
--
-- Date de création: 7 avril 2008
-- Date de modification: 13 mai 2008
--
-- Fichier: SNN_PM1_Hrt1_13mai\TestBench_SNN1_V1.nsx
-- Version numéro: 1.2 : Vérification fonctionnelle
--
%%%%
-- %%% Commentaires:
-- %%% Les calculs n'ayant pas possible sur les signaux de type : time,
-- %%% On réalise une conversion Time 2 Real en entrée du module,
-- %%% Puis une transformation Real 2 Time en sortie du module !
-- %%% En interne toutes les grandeurs temporelles seront du type : real !
-- %%% les SNN1 travailleront sur les stimulations:
-- %%% Time_Right_Ventricule et Time_Left_Ventricule (Trv & Tlv) !
-- %%% Les grandeurs AV et VV seront ensuite calculés sur la base des ces timing,
-- %%% afin de guider les algorithmes afin qu'ils optimisent SV.
-- %%% Les grandeurs du systeme sont regroupées dans le module SNNpkg
-- %%% ainsi que les initialisations des variables internes. LIBRARY IEEE;
library ieee;
USE IEEE.electrical_systems.ALL;
```

```

USE IEEE.math_real.ALL;
USE work.SNNpkg.ALL;
USE work.ALL;
ENTITY TestBench_SNN1_V1 IS
END;

--%%%%%%%%%
ARCHITECTURE str OF TestBench_SNN1_V1 IS
-- Contantes temporals
    CONSTANT PulsWidth : time := 500us;      -- width of the pulse exciting the synapses
-- Constantes for the algorith in implant/AlgoBlock/DelMeas
    CONSTANT default_AV : real := 100.0e-3;  -- AV value in Sec defined by Clinic Pratic
    CONSTANT default_VV : real := 25.0e-3;  -- VV value in Sec defined by Clinic Pratic
-- Constante Inconnues
    CONSTANT IE : real := 850.0e-3;          -- Undefined
-- Contantes for the Heart rate
    CONSTANT RA_Pulse : time := 5ms; -- equal 2*HPClk !
    CONSTANT RA_Period : time := 250ms;    -- must be > (NbDumSR_RV+NbSyn+3)*(2*HPClk)
    QUANTITY Q_SV: real := 0.0;
        SIGNAL mitral_open    : boolean := false;
        SIGNAL DET_RA         : real := 0.0;
        SIGNAL DET_RV         : real := 0.0;
        SIGNAL DET_LV         : real := 0.0;
        SIGNAL STIM_RA        : real := 0.0;
        SIGNAL STIM_RV        : real := 0.0;
        SIGNAL STIM_LV        : real := 0.0;
BEGIN
-- Heart model or pseudo model
Heart : ENTITY SimpleHeart_SV(bhv)
    GENERIC MAP(RA_Pulse => RA_Pulse,
               RA_Period => RA_Period)
    PORT MAP(    Q_SV => Q_SV,
               mitral_open => mitral_open,
               DET_RA => DET_RA,
               DET_RV => DET_RV,
               DET_LV => DET_LV,
               STIM_RA => STIM_RA,
               STIM_RV => STIM_RV,
               STIM_LV => STIM_LV);

```



```

USE IEEE.math_real.ALL;
USE work.SNNpkg.ALL;
USE work.ALL;

```

```

ENTITY SimpleHeart_SV IS

```

```

  GENERIC(RA_Pulse, RA_Period : time);

```

```

  PORT( quantity Q_SV : out real := 0.0;
        signal mitral_open : out boolean := false;
        signal DET_RA : out real := 0.0;
        signal DET_RV : out real := 0.0;
        signal DET_LV : out real := 0.0;
        signal STIM_RA : in real := 0.0;
        signal STIM_RV : in real := 0.0;
        signal STIM_LV : in real := 0.0);

```

```

END ENTITY SimpleHeart_SV;

```

```

--
%%%%%%
%%%%%% Simple model of the Heart only for the SV calculs   %%%
-- %%% Based on a fixed Bell Shape curve                    %%%
--
%%%%%%
%%%%%%

```

```

ARCHITECTURE bhv OF SimpleHeart_SV IS

```

```

-- Constante for the Bell Shape curve
  CONSTANT AvOpt : time := 140ms; -- optimal value for the maximum
  CONSTANT VvOpt : time := 20ms;
  CONSTANT SVmax : real := 1000.0; -- arbitrary value for the maximum
  --CONSTANT S_N : real := 20.0;
  SIGNAL S_Delay, TimeRef, CurrentTime : real := 0.0;
  SIGNAL S_RA_D, S_RV_D, S_LV_D : real := 0.0;
  SIGNAL S_AVopt, S_VVopt, Delta_AV, Delta_VV : real :=0.0;
  --SIGNAL S_noise : real := 0.0;

```

```

BEGIN

```

```

-- %%% The Heart Beat %%%

```

Chapitre VII

```
HrtBeat: PROCESS
BEGIN
  DET_RA <= 1.0;
  WAIT FOR RA_Pulse;
  DET_RA <= 0.0;
  WAIT FOR (RA_Period - RA_Pulse);
END PROCESS;

-- Time measurement with a precision of 500us
-- This value of 500us is related to the simulation time!
-- Lower value gives longer simulation time!
TimeMeas : PROCESS
BEGIN
  CurrentTime <= t2r(now);
  S_Delay <= CurrentTime - TimeRef;
  WAIT FOR 500us;
END PROCESS;

DelayRA: PROCESS IS
BEGIN
  WAIT ON STIM_RA;
  IF (STIM_RA = 1.0) THEN
    S_RA_D <= S_Delay;
  END IF;
  WAIT FOR 1us;
END PROCESS DelayRA;

DelayRV: PROCESS IS
BEGIN
  WAIT ON STIM_RV;
  IF (STIM_RV = 1.0) THEN
    S_RV_D <= S_Delay;
  END IF;
  WAIT FOR 1us;
END PROCESS DelayRV;

DelayLV: PROCESS IS
BEGIN
  WAIT ON STIM_LV;
```



```

IF (STIM_LV = 1.0) THEN
    S_LV_D <= S_Delay;
END IF;
WAIT FOR 1us;
END PROCESS DelayLV;

-- %%%%%%%%% The Stroke Volume evaluation

S_Veval: PROCESS IS
BEGIN
    WAIT ON S_RV_D, S_LV_D;
    S_AVopt <= t2r(AvOpt);
    S_VVopt <= t2r(VvOpt);
    WAIT FOR 1us; -- wait that the previous process have refreshed the signals
    Delta_AV <= (S_RV_D - S_RA_D) - S_AVopt;
    Delta_VV <= (S_LV_D - S_RV_D) - S_VVopt;
    WAIT FOR 1us;
END PROCESS;

BREAK ON Delta_AV, Delta_VV;
Q_SV == SVmax - 78125.0*(Delta_AV*Delta_AV) - 50000.0*(Delta_VV*Delta_VV);

END ARCHITECTURE bhv;

```

Annexe IV.3 Modèle du SNN

```

>>> VHDL
--
%%%%%%%%%%
%%%%%%%%%%
-- Sous Bloc d'un neurone SNN1 complet
-- ShiftRegister
--
-- Auteur Francois SCHWARTZ, Jacques MICHEL, Qing SUN
-- Laboratoire INESS Strasbourg UMR 7163
--
-- Date de création: 7 avril 2008

```

Chapitre VII

-- Date de modification: 9 mai 2008

--

-- Fichier: SNN_PM1_Hrt1\synapse.vhd

-- Version numéro: 1.1

--

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

LIBRARY IEEE;

USE IEEE.electrical_systems.ALL;

USE work.ALL;

ENTITY synapse IS

PORT(TERMINAL T_Vdd, T_pulse, T_w, T_out : electrical);

END ENTITY;

ARCHITECTURE str OF synapse IS

TERMINAL T_wmem : electrical;

BEGIN

mem : ENTITY memory(bhv)

PORT MAP(T_in=>T_w, T_out=>T_wmem);

mul : ENTITY AnalogMul(bhv_QS_SPICE_level5) --bhv_QS_level2

PORT MAP(T_in1=>T_wmem, T_in2=>T_pulse, T_out=>T_out);

END ARCHITECTURE;

Annexe IV.4 Modèle des algorithmes HL et RL

--

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-- Bloc modifiant les poids synaptiques suivant les Algorithmes HL et RL

--

-- Auteur Francois SCHWARTZ, Jacques MICHEL, Qing SUN

-- Laboratoire INESS Strasbourg UMR 7163

--

-- Date de création: 7 avril 2008

-- Date de modification: 17 avril 2008 pour bhv_1

-- Date de modification: 13 mai 2008 pour bhv_full

-- Version finalisé qui fonctionne sur tous les modèles dont le full!

```

--
-- Fichier: SNN_PM1_Hrt1_13mai\HRL.vhd
-- Version numéro: 1.1
--
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

LIBRARY IEEE;
USE IEEE.electrical_systems.ALL;
USE IEEE.math_real.ALL;
USE work.SNNpkg.ALL;
USE work.ALL;

ENTITY HRL IS
  GENERIC(HPClk : time);
  PORT(TERMINAL T_weightsRV, T_weightsLV : electrical_vector(1 TO NbSyn);
       SIGNAL S_RA, S_Clk           : IN bit;
       SIGNAL S_AdaptedMode, S_GuidedMode : OUT boolean := false;
       SIGNAL S_defPrv, S_defPlv      : IN real;
       SIGNAL S_Trv, S_Tlv           : IN real;
       QUANTITY Q_SV                  : IN real);
END ENTITY;

--
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
-- %%% Algorithme Réunit à HL poids fixe %%%
--
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ARCHITECTURE bhv_0 OF HRL IS

  QUANTITY V_weightsRV ACROSS I_weightsRV THROUGH T_weightsRV;
  QUANTITY V_weightsLV ACROSS I_weightsLV THROUGH T_weightsLV;
  SIGNAL S_weightsRV, S_weightsLV : vec := (MaxWeight, MaxWeight, MaxWeight, OTHERS=>0.0);
-- INITIALISATION IMPORTANTE
  -- %%% Attention cette initialisation importe pour le .OP systeme ! MAIS
  -- %%% Les premiers poids retournés par l'algorithme démarrent sur les valeurs d'initialisation des
  VARIABLES
  -- %%% weights RV, weights LV ! ILS DOIVENT être IDENTIQUES !

BEGIN

```

Chapitre VII

```
-- Mise à jour des poids synaptiques vers les SNN1
iW : FOR i IN 1 TO NbSyn GENERATE
    I_weightsRV(i) == -1.0*S_weightsRV(i); -- Signe Moins pour rendre le courant sortant !
    I_weightsLV(i) == -1.0*S_weightsLV(i); -- Signe Moins pour rendre le courant sortant !
END GENERATE;
BREAK ON S_weightsRV, S_weightsLV;

-- Process for Synapses Weights changes
PROCESS
    VARIABLE weightsRV, weightsLV : vec := (MaxWeight, MaxWeight, MaxWeight, OTHERS=>0.0);
-- INITIALISATION Sensible!
    VARIABLE P, T : intTab;
    VARIABLE hitRV, hitLV : real := 0.0;
BEGIN

    -- %%% Time position into Synapses rank Translation
    -- %%% for the SNN Firing signals called TRV and TLV
    T(1) :=3; -- poids de RV l'initialisation
    T(2) :=3; -- poids de LV l'initialisation

-- %%% Test of the Learning Mode %%%
-- **** NON ADAPTED MODE ****
    -- %%% Time position into Synapses rank Translation
    -- %%% for the SNN Firing Targets called PRV and PLV
    -- %%% set by the praticer
    P(1) :=11;
    P(2) :=11;

-- SNN RV
-- %%% Test if in HIT mode (default is MISS mode)
    IF ABS(T(1)-P(1)) <= delHIT THEN -- delHIT is defined in SNNpkg := 2
        hitRV := 1.0; -- Hit Mode is Yes
    ELSE
        hitRV := 0.0; -- Hit Mode is NO => Miss Mode is Yes
    END IF;
-- New weights calculation for RV SNN
    weightsRV := weightsCompute(weightsRV,P(1),T(1),etaHL,hitRV); --etaHL Defined in SNNpkg :=
0.5 !
```


Chapitre VII

```
-- %%% Attention cette initialisation importe pour le .OP systeme ! MAIS
-- %%% Les premiers poids retournés par l’algorithme démarrent sur les valeurs d’initialisation des
VARIABLES
-- %%% weightsRV, weightsLV ! ILS DOIVENT être IDENTIQUES !
BEGIN
-- Mise à jour des poids synaptiques vers les SNN1
iW : FOR i IN 1 TO NbSyn GENERATE
    I_weightsRV(i) == -1.0*S_weightsRV(i); -- Signe Moins pour rendre le courant sortant !
    I_weightsLV(i) == -1.0*S_weightsLV(i); -- Signe Moins pour rendre le courant sortant !
END GENERATE;
BREAK ON S_weightsRV, S_weightsLV;

-- Process for Synapses Weights changes
PROCESS
    VARIABLE weightsRV, weightsLV: vec := (OTHERS=>0.0); -- INITIALISATION Sensible !
    VARIABLE TRV, TLV : real := 0.1e-3; -- Synapse rank = 0 -- 0.1 vient de la troncature !
-- Il va créer le pointeur en 3 !!
    VARIABLE PRV, PLV : real := 55.1e-3; -- Synapse rank = 11
    VARIABLE Tstep : real := 5.0e-3;
    VARIABLE P, T : intTab;
    VARIABLE hitRV, hitLV : real := 0.0;
BEGIN
    WAIT UNTIL S_RA = '1';
-- %%% Time position into Synapses rank Translation
-- %%% for the SNN Firing signals called TRV and TLV
    Tstep := 2.0*t2r(HPClk);
    IF ( (S_TrV - t2r(DelayRV)) > 0.0 ) THEN
        TRV := S_TrV - t2r(DelayRV); -- DelayRV is defined in SNNpkg
    END IF;
    T(1) := integer( TRUNC(TRV/Tstep) ); -- Initial value T(1) := 3;
    IF ( (S_TlV - t2r(DelayLV)) > 0.0 ) THEN
        TLV := S_TlV - t2r(DelayLV); -- DelayLV is defined in SNNpkg
    END IF;
    T(2) := integer( TRUNC(TLV/Tstep) );
-- %%% pour test: Comparer T(i) avec S_ClkCount !

-- %%% Test of the Learning Mode %%%
-- **** NON ADAPTED MODE ****?
-- %%% Time position into Synapses rank Translation
```

```

-- %%% for the SNN Firing Targets called PRV and PLV
-- %%% set by the praticer
P(1) := integer( TRUNC(PRV/Tstep) ); -- Initial value P(1) := 11
P(2) := integer( TRUNC(PLV/Tstep) );
-- Control of the internal pointer rank position before weights change
C_TRV <= T(1);
C_PRV <= P(1);
C_TLV <= T(2);
C_PLV <= P(2);
WAIT FOR 1us;

-- SNN RV
-- %%% Test if in HIT mode (default is MISS mode)
IF ABS(T(1)-P(1)) <= delHIT THEN -- delHIT is defined in SNNpkg := 2
    hitRV := 1.0; -- Hit Mode is Yes
ELSE
    hitRV := 0.0; -- Hit Mode is NO => Miss Mode is Yes
END IF;
-- New weights calculation for RV SNN
-- %%% ATTENTION: Les "pointeurs T() et P()" doivent etre recentrés sur l'intervalle
-- %%% [1, NbSyn]!
weightsRV := weightsCompute(weightsRV,P(1),T(1),etaHL,hitRV); --etaHL Defined in SNNpkg :=
0.5 !

-- SNN LV
IF ABS(T(2)-P(2)) <= delHIT THEN
    hitLV := 1.0;
ELSE
    hitLV := 0.0;
END IF;
-- New weights calculation for LV SNN
weightsLV := weightsCompute(weightsLV, P(2), T(2), etaHL,hitLV);

WAIT UNTIL S_RA = '0';
-- Update of the Synapses Weights in SNN1
-- %%% These values are used for the Firing in this Heart Beat
S_weightsRV <= weightsRV;
S_weightsLV <= weightsLV;

```



```

VARIABLE TRV, TLV : real := 0.1e-3; -- Synapse rank = 0 -- 0.1 vient de la troncature !!
VARIABLE PRV, PLV : real := 55.1e-3; -- Synapse rank = 11
VARIABLE Tstep : real := 5.0e-3;
VARIABLE P, T : intTab := (OTHERS=>0);
VARIABLE deltaPT : intTab := (1,1); -- First step in Adapted Mode
VARIABLE SV : realTab := (OTHERS=>0.0);
VARIABLE hitRV, hitLV, realHCRwindow, meanNAHCRRV, meanNAHCRLV, meanAdHCRRV,
meanAdHCRLV, deltaSV : real := 0.0;
VARIABLE NA_HCR_RV, NA_HCR_LV, Ad_HCR_RV, Ad_HCR_LV : HCRTab :=
(OTHERS=>0.0);
VARIABLE AdaptedMode, GuidedMode : boolean := false;

BEGIN

WAIT UNTIL S_RA = '1';

-- Iterations Counter for HCR calculs
Count := Count + 1;

-- In Adapted Mode, we change only the synapse wheigths of one neuron
-- selection of the active SNN RV or SNN LV with a "toggle"
-- sel = 0 => RV and sel = 1 => LV
sel := sel + 1;
sel := intMOD(sel,2);
selbar := intMOD((sel + 1),2); -- sinon ABS(sel - 1);

-- Enregistrement de la nouvelle valeur mesuré du SV
SV(sel+1) := Q_SV; -- acquisition of SV; this new value measure the consquence of the previous
weight change
S_SV(sel+1) <= SV(sel+1);

-- %%% Time position into Synapses rank Translation
-- %%% for the SNN Firing signals called TRV and TLV
Tstep := 2.0*t2r(HPClk);
IF ( (S_TrV - t2r(DelayRV)) > 0.0 ) THEN
TRV := S_TrV - t2r(DelayRV); -- DelayRV is defined in SNNpkg
END IF;
T(1) := integer( TRUNC(TRV/Tstep) ); -- Initial value T(1) := 3
IF ( (S_TlV - t2r(DelayLV)) > 0.0 ) THEN
TLV := S_TlV - t2r(DelayLV); -- DelayLV is defined in SNNpkg

```

Chapitre VII

```
END IF;
T(2) := integer( TRUNC(TLV/Tstep) );
-- %%% pour test: Comparer T(i) avec S_ClkCount !
WAIT FOR 1us;

-- %%% Test of the Learning Mode %%%
IF (AdaptedMode=false) THEN -- **** NON ADAPTED MODE ****?
-- %%% Time position into Synapses rank Translation
-- %%% for the SNN Firing Targets called PRV and PLV
-- %%% set by the praticer
IF ( (S_defPrv - t2r(DelayRV)) > 0.0 ) THEN
PRV := S_defPrv - t2r(DelayRV); -- DelayRV is defined in SNNpkg
END IF;
P(1) := integer( TRUNC(PRV/Tstep) ); -- Initial value P(1) := 11;
IF ( (S_defPlv - t2r(DelayLV)) > 0.0 ) THEN
PLV := S_defPlv - t2r(DelayLV); -- DelayLV is defined in SNNpkg
END IF;
P(2) := integer( TRUNC(PLV/Tstep) );
-- Control of the internal pointer rank position before weights change
C_TRV <= T(1);
C_PRV <= P(1);
C_TLV <= T(2);
C_PLV <= P(2);
WAIT FOR 1us;

-- SNN RV
-- %%% Test if in HIT mode (default is MISS mode)
IF ABS(T(1)-P(1)) <= delHIT THEN -- delHIT is defined in SNNpkg := 2
hitRV := 1.0; -- Hit Mode is Yes
ELSE
hitRV := 0.0; -- Hit Mode is NO => Miss Mode is Yes
END IF;
-- New weights calculation for RV SNN
-- %%% ATTENTION: Les "pointeurs T() et P()" doivent etre recentré sur l'intervalle
-- %%% [1, NbSyn] !
weightsRV := weightsCompute(weightsRV,P(1),T(1),etaHL,hitRV); --etaHL Defined in SNNpkg :=
0.5 !

-- SNN LV
```

```

IF ABS(T(2)-P(2)) <= delHIT THEN
  hitLV := 1.0;
ELSE
  hitLV := 0.0;
END IF;
-- New weights calculation for LV SNN
weightsLV := weightsCompute(weightsLV, P(2), T(2), etaHL, hitLV);

WAIT UNTIL S_RA = '0';
-- Update of the Synapses Weights in SNN1
-- %%% These values are used for the Firing in this Heart Beat
S_weightsRV <= weightsRV;
S_weightsLV <= weightsLV;

-- Calculation of the HCR in NonAdapted Mode
-- The HCR defines the Learning Mode used in this Heart Beat to calculated
-- the Synapses Weights used in the next Heart Beat
-- Progression du HCR en mode NonAdapted
pt := intMOD(Count, HCRwindow);
NA_HCR_RV(pt+1) := hitRV;
meanNAHCRRV := mean(NA_HCR_RV);
NA_HCR_LV(pt+1) := hitLV;
meanNAHCRLV := mean(NA_HCR_LV);
realHCRwindow := REAL(HCRwindow);    -- Integer to real for MODULO function
IF (meanNAHCRRV > ((realHCRwindow -
1.0)/realHCRwindow)) AND ((meanNAHCRLV > (realHCRwindow - 1.0)/realHCRwindow)) THEN
  AdaptedMode := true;
  count := 0; -- Restart the conter for starting in Adapted Mode
END IF;
S_AdaptedMode <= AdaptedMode;
C_HCR_RV <= meanNAHCRRV;
C_HCR_LV <= meanNAHCRLV;

END IF; -- ***** NON ADAPTED MODE *****?

-- %%% Test of the Learning Mode %%%
IF (AdaptedMode=true) THEN -- ***** ADAPTED MODE *****

  IF count > 0 THEN

```

Chapitre VII

```
-- mise à jour de la cible P
-- In Adapted mode the target pacing P shifted to ensure maximum SV
-- At strating in Adapted mode the first step (deltaPT) is fixed to (+1,+1) !
IF (sel=0) THEN -- sel = 0 => Sel+1 = 1 => RV_SNN
    P(1) := P(1) + deltaPT(1);
    P(2) := P(2) + deltaPT(1); -- Change AV only !!
ELSE -- sel = 1 => sel+1 = 2 => LV_SNN
    P(2) := P(2) + deltaPT(2); -- Change VV only !!
END IF;

-- Détermination de l'évolution des cibles Prv ou Plv pour le prochain pas
deltaSV := SV(sel+1) - SV(selbar+1); -- variation du SV du pas précédent
-- Préparation du changement du pacing pour la prochaine période cardiaque!
IF deltaSV >= 0.0 THEN
    deltaPT(selbar+1) := deltaPT(selbar+1);
ELSE
    deltaPT(selbar+1) := -deltaPT(selbar+1);
END IF;
END IF;

-- Control of the internal pointer rank position before weights change
C_TRV <= T(1);
C_PRV <= P(1);
C_TLV <= T(2);
C_PLV <= P(2);
WAIT FOR 1us;

-- Affectation des nouveaux poids synaptiques en fonction du toggle
-- In Adapted mode we are allways in HitMode !
hitRV := 1.0;
hitLV := 1.0;
-- Affectation des nouveaux poids synaptiques RV
weightsRV := weightsCompute(weightsRV,P(1),T(1),etaRL,hitRV); --etaRL Defined in SNNpkg :=
1.0 !
-- Affectation des nouveaux poids synaptiques LV
weightsLV := weightsCompute(weightsLV, P(2), T(2), etaRL,hitLV);

WAIT UNTIL S_RA = '0';
-- Mise à jour des poids synaptiques
S_weightsRV <= weightsRV;
```

```

S_weightsLV <= weightsLV;
C_deltaSV <= deltaSV;
C_deltaPT_RV <= deltaPT(1);
C_deltaPT_LV <= deltaPT(2);

-- Calculation of the HCR in Adapted Mode
-- The HCR defines the Learning Mode used in this Heart Beat to calculated
-- the Synapses Weights
-- Progression du HCR en mode Adapted
pt := intMOD(Count,HCRwindow);
Ad_HCR_RV(pt+1) := hitRV; -- Inutile car hitRV TOUJOURS égale à 1
meanAdHCRRV := mean(Ad_HCR_RV);
Ad_HCR_LV(pt+1) := hitLV;
meanAdHCRLV := mean(Ad_HCR_LV);
realHCRwindow := REAL(HCRwindow); -- Integer to real for MODULO function
IF (meanAdHCRRV>((realHCRwindow-
1.0)/realHCRwindow))AND((meanAdHCRLV>(realHCRwindow-1.0)/realHCRwindow)) THEN
    GuidedMode := true;
END IF;
S_GuidedMode <= GuidedMode;
C_HCR_RV <= meanAdHCRRV;
C_HCR_LV <= meanAdHCRLV;

END IF; -- **** ADAPTED MODE ****

END PROCESS;
END ARCHITECTURE bhv_full;

```

Références

- [1] Y. Hervé, VHDL-AMS : Applications et enjeux industriels. DUNOD, 2002
- [2] J. Weber and M. Meaudre, *Le langage VHDL*. DUNOD, 2001
- [3] Y. Hervé, “VHDL-AMS: Un atout pour la conception de système,” *IEEE*, 2002
- [4] IEEE Computer Society, “IEEE Standard VHDL Language Reference Manual”, Amendment 1: Procedural Language Application Interface, IEEE Std 1076cTM-2007
- [5] IEEE Computer Society, “IEEE Standard VHDL Analog and Mixed-Signal Extensions”, IEEE Std 1076.1TM-2007, pp. 342, New York-USA, 2007
- [6] A. Dobioli and R. Vemuri, “Behavioral modeling for high-level synthesis of analog and mixed-signal systems applications,” *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on, vol. 22, no. 11, pp. 1504-1520, Nov. 2003
- [7] D. Guihal, “Modélisation en langage VHDL-AMS des systèmes pluridisciplinaire,” Ph.D. dissertation, Université Toulouse III, 2007
- [8] L. Zhou, “Modélisation VHDL-AMS multi-domaines de structures intelligentes, autonomes et distribuées à base de MEMS” Ph.D. dissertation, Université Louis Pasteur, 2007
- [9] T. Ibrahim, “Contribution au développement de modèles pour l’électronique de puissance en VHDL-AMS” Ph.D. dissertation, INSA de Lyon, 2007
- [10] Homepage of the IEEE standard VHDL-AMS (IEEE P1076.1 Working Group): [Online, 10/10/2010], <http://www.eda.org/twiki/bin/view.cgi/P10761/WebHome>
- [11] Saber : multi-domain modeling and simulation, SYNOPSYS, [Online, 10/10/2010], <http://www.synopsys.com/Tools/SLD/Mechatronics/Saber/Pages/default.aspx>
- [12] OPENMAST: open source mixed-technology and mixed-signal hardware description language, SYNOPSYS, [Online, 10/10/2010], <http://www.openmast.org/>
- [13] The origins of MATLAB, Mathworks, [Online, 10/10/2010], http://www.mathworks.com/company/newsletters/news_notes/clevescorner/dec04.html
- [14] MATLAB – the language of technical computing, Mathworks, [Online, 10/10/2010], <http://www.mathworks.com/products/matlab/>
- [15] Simulink – simulation and model-based design, Mathworks, [Online, 10/10/2010], <http://www.mathworks.com/products/simulink/>
- [16] Modelica – modeling of complex physicals systems, Modelica, [Online, 20/10/2010], <http://www.modelica.org/>
- [17] Accellera Verilog Analog Mixed-Signal Group, EDA industry working groups, [Online, 20/10/2010], <http://www.eda-stds.org/verilog-ams/>
- [18] Accellera, “Verilog-AMS Langage Reference Manual, Analog & Mixed-Signal Extensions to Verilog-HDL,” Version 2.2., pp. 309, U.S.A., November 2006.

- [19] F. Pecheux, C. Lallement, and A. Vachoux, "VHDL-AMS and Verilog-AMS as alternative hardware description languages for efficient modeling fo multidiscipline systems," *Computer-Aided Design of Intergrated Circuits and Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 204-225, Feb. 2005.
- [20] Questa – advanced verification & debug technologie, Mentor Graphics, [Online, 20/10/2010], <http://www.mentor.com/products/fv/questa/>
- [21] SystemVision – system modeling, Mentor Graphics, [Online, 20/10/2010], http://www.mentor.com/products/sm/system_integration_simulation_analysis/systemvision/
- [22] Simplorer – System Simulation Software for Multi-Domain Design, ANSOFT, [Online, 20/10/2010], <http://www.ansoft.com/products/em/simplorer/>
- [23] SMASH Vision, DOLPHIN Integration, [Online, 20/10/2010], http://www.dolphin.fr/medal/smash/smash_overview.php
- [24] Portunus introduction, CEDRAT, [Online, 20/10/2010], <http://www.cedrat.com/fr/solutions-logicielles/portunus.html>
- [25] Abeles, M., Bergman, H., Margalit, E., & Vaadia, E. "Spatiotemporal firing patterns in the frontal cortex of behaving monkeys," *Journal of Neurophysiology*, vol. 4, pp. 70, 1993.
- [26] Thorpe, S., Fize, D., & Marlot, C., "Speed of processing in the human visual system," *Nature*, vol. 381, pp. 520–522, 1996.
- [27] Rieke, F., Warland, D., van Steveninck, R. R., & Bialek, W., "Spikes: exploring the neural code," Cambridge, Ma, London: MIT Press. 1997.
- [28] Recce, M., Encoding information in neuronal activity. In W. Maass & C. Bishop (Eds.), *Pulsed neural networks*. MIT Press. 1999.
- [29] Gerstner, W., & Kistler, W., Spiking neuron models. Single neurons, populations, plasticity, Cambridge University Press. 2002.
- [30] Koch, C., & Segev, I. (Eds.), *Methods in neuronal modeling: From ions to networks* (Second ed.). MIT Press. 1998.
- [31] Gerstner, W., Spiking neurons. In W. Maas & C.M.Bishop (Eds.), *Pulsed neural networks* (pp.3–53). MIT Press. 1999.
- [32] Izhikevich, E., "Resonate-and-fire neuron," *Neural Networks*, vol.14, page 883–894. 2001.
- [33] Adrian, E. "The impulses produced by sensory nerve endings," *Journal of Physiology (London)*, vol. 61, pp.49–72, 1926.
- [34] Cariani, P., Temporal coding of sensory information. In J. M. Bower (Ed.), *Computational neuroscience: Trends in research* (pp. 591–598). New York: Plenum. 1997.
- [35] Hopfield, J. (1995). Pattern recognition computation using action potential timing for stimulus rep-resentation. *Nature*, 376, 33–36.
- [36] Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10, 1659–1671.

Chapitre VII

- [37] Maass, W. (1999). Computing with spiking neurons. In W. Maass & C. Bishop (Eds.), *Pulsed neural networks*. MIT Press.
- [38] Maass, W., & Bishop, C. (Eds.). (1999). *Pulsed neural networks*. MIT Press.
- [39] Gawne, T., Kjaer, T., & Richmond, B. (1996). Latency: another potential code for feature binding in striate cortex. *Journal of Neurophysiology*, 76(2), 1356–1360.
- [40] Thorpe, S., & Gautrais, J. (1998). Rank order coding. In *Computational neuroscience: Trends in research 1998* (pp. 113–118). New York: Plenum Press.
- [41] Van Rullen, R., & Thorpe, S. (2001). Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Computation*, 13, 1255–1283.
- [42] Abeles, M. (1982). Role of cortical neuron: integrator or coincidence detector? *Isr. Journal Medicine Science*, 18, 83–92.
- [43] Softky, W., & Koch, C. (1993). The highly irregular firing of cortical cells is inconsistent with temporal integration of random ESPs. *Journal of Neuroscience*, 13, 334–350.
- [44] Kempter, R., Gerstner, W., & van Hemmen, J. L. (1998). How the threshold of a neuron determines its capacity for coincidence detection. *BioSystems*, 48, 105–112.

Etude et conception d'un processeur neuronal analogique très faible consommation, application au pilotage d'un pacemaker de nouvelle génération

Résumé : L'objectif de cette recherche est de développer un réseau de neurones impulsionnels analogiques afin d'améliorer la performance d'un pacemaker bi-ventriculaire (aussi appelé le CRT-P) de nouvelle génération. L'implémentation sur silicium utilise l'approche réseau de neurones analogiques qui nécessite le développement d'une solution technique satisfaisant à une contrainte de très basse consommation énergétique. Nous proposons une approche de un réseau de neurones impulsionnels analogiques pour optimiser la prédiction des délais cardiaques avec l'algorithme d'apprentissage Hebb et l'algorithme d'apprentissage par renforcement dans des modes de fonctionnement différents. L'amélioration des prévisions permet au CRT-P de fournir des battements cardiaques optimaux en temps réel. Nous décrivons le comportement et les qualités de notre algorithme au travers de simulations mathématiques et comportementales. Des simulations complètes et cohérentes du système basées sur des modèles simples du cœur (rythme cardiaque constant puis rythme cardiaque variable) avec des bruits uniformes aléatoires sont illustrées avec succès pour la validation de la faisabilité du système.

Nous proposons aussi une méthodologie renforcée de la conception analogique et mixte. Les simulations de tous niveaux (de hauts et bas niveaux) peuvent être faites rapidement afin de vérifier des performances du système dans chaque phase de conception et ainsi fournir une plage des spécifications acceptables facilitant la synthèse analogique et mixte suivante.

Mots clés : dispositif de thérapie de resynchronisation cardiaque (CRT), apprentissage Hebb, apprentissage par renforcement, réseau de neurones impulsionnels analogiques, VHDL-AMS, méthodologie de conception analogique et mixte.

Study and design of an analog neural processor of very low power consumption, applied to the navigation of a new generation pacemaker

Abstract: The objective of this research is to develop an analog spiking neural network so as to improve the performance of a bi-ventricular pacemaker (also called the CRT-P) of new generation. The implementation on silicon using the analog neural network approach requires the development of a satisfactory technical solution to meet the constraint of very low energy consumption. We propose an analog spiking neural network approach to optimize the cardiac delay prediction with the Hebbian learning algorithm and the reinforcement learning algorithm in different functional modes. The delay improvement allows the CRT-P to provide optimal heartbeat in real time. We describe the behavior and the qualities of our algorithm through mathematical and behavioral simulations. The complete and coherent system simulations based on the simple heart models (constant heart rate and variable heart rate) with random uniform noise are shown successfully to validate the system feasibility.

We also propose an enhanced methodology of the analog and mixed signal design. The simulations of all levels (high and low levels) can be carried out quickly in order to verify the system performance in each design phase and also carry out the acceptable specification space for facilitating the following analog and mixed signal synthesis.

Keywords: Cardiac Resynchronization Therapy (CRT) device, Hebbian learning, Reinforcement learning, analog spiking neural network, VHDL-AMS, analog and mixed signal design methodology.