



HAL
open science

Méthode de discrétisation adaptée à une logique événementielle pour l'ultra-faible consommation : application à la reconnaissance de signaux physiologiques

Tugdual Le Pelleter

► To cite this version:

Tugdual Le Pelleter. Méthode de discrétisation adaptée à une logique événementielle pour l'ultra-faible consommation : application à la reconnaissance de signaux physiologiques. Micro et nanotechnologies/Microélectronique. Université Grenoble Alpes, 2015. Français. NNT : 2015GREAT043 . tel-01217996

HAL Id: tel-01217996

<https://theses.hal.science/tel-01217996>

Submitted on 20 Oct 2015

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Nano Électronique Nano Technologies**

Arrêté ministériel : 7 août 2006

Présentée par

Tugdual LE PELLETER

Thèse dirigée par **Laurent FESQUET** et
codirigée par **Agnès BONVILAIN**

préparée au sein du **Laboratoire TIMA**
dans l'**École Doctorale Électronique Électrotechnique,**
Automatique & Traitement du Signal

**Méthode de discrétisation adaptée à une logique
événementielle pour l'ultra-faible consommation :
application à la reconnaissance de signaux
physiologiques**

Thèse soutenue publiquement le **13 mai 2015**,
devant le jury composé de :

Monsieur, Yves, LEDUC

Professeur, Université de Nice Sophia Antipolis (Rapporteur)

Monsieur, Norbert, NOURY

Professeur, INSA de Lyon (Rapporteur)

Monsieur, Pascal, NOUET

Professeur, Université de Montpellier 2 (Président)

Monsieur, Laurent, FESQUET

Maître de conférence, Université de Grenoble (Directeur de thèse)

Madame, Agnès, BONVILAIN

Maître de conférence, Université de Grenoble (Co-encadrante de thèse)

Monsieur, Hamid, LAMRAOUI

PDG de UroMems (Invité)



Remerciements

Je tiens tout d'abord à témoigner de ma profonde gratitude envers le membre

– Laurent Fe , mon directeur de thèse
de travailler sur un sujet passionnant et riche en perspective
La confiance dont il m'a fait preuve m'a permis d'ex
idée
et innovante
la recherche et mon goût pour la réflexion.

– Agnès , ma co-encadrante, qui a été pré
pro
dans le biomédical, un domaine de recherche que j'affectionne particu-
lièrement. Durant ce
pertinent

Cous deux m'ont apporté de
m'ont permis de porter à maturité et d'aboutir ce travail de recherche. Je
tiens à nouveau à le
lité et de leur soutien inflexible dont il
n'aurait été po

Je souhaite remercier le
 tenir le
 de

- Alexandre Moreau-Gaudry qui m'a consacré une grande partie de son temps à l'élaboration et la mise en forme du protocole clinique
- Pierre Mozzer qui a mis en œuvre le cadre ex
 tocole et qui a récupéré avec grand soin le
 auprès
 m'avoir permis d'être au « cœur de l'action » en assistant à une ex-
 plantation d'un dispo
 précier toute la gratification de s'imve
 ty

Je tiens à remercier chaleureusement Alexandre Chagoya et Robin Rolland pour leur pédagogie, leur patience et leur sens de l'écoute quand il m'a fallu d'« dompter » le matériel lors de la phase de prototy
 reconnaissance va également à mon stagiaire, Giuse , qui a été
 d'un précieux soutien.

Je tiens également à remercier toute
 soutenance de cette thèse

- Yve Norbert Pourry et Pascal Pouet, qui m'ont fait l'honneur d'être membre de mon jury. Leur regard éclairé sur mon travail
 de thèse
 perspective
- Hamid Lamraoui, qui a généreusement acce

mon jury de thèse
veillance, sa générosité
long de mon doctorat.

Je remercie également mes
tuteurs durant ma thèse
Otto Roloff et Leonel Guimarães qui sont arrivés
à une période charnière de ma thèse
Il est un membre du Club de composition
de personnalité
voir, Léandro Rocha, Alexandre Ayre, Karla « Karlinha » Romão,
Chiago Leite, Raphael Viera mais également Jean Simatic qui, chaque
jour, devient un peu plus bricoleur
avant tout un état d'esprit
Alice De Bignicourt, Chadi Al-Khatib, Abdelkarim Cherkaoui, Amani
Darwish et Rodrigo Poisson ainsi que Chibaut, Cairn, Alvie, Ma-
non, Mégane, Camille, Stéphanie et Jade.

Une mention particulière à mes
amis de l'année Lyubomir Kerachev,
Micael Correia, Morgan Saint-Jore et Om Kulkarni. Un grand merci
pour tous les conseils
n'ayant aucune emprise.

Je remercie chaleureusement Eric Gascard qui m'a soutenu « paternel-
lement » durant toute ma thèse
conseil

Enfin, je tiens à remercier affectueusement me
qui m'ont toujours é

Cugdual Le Pelleter

Table des matières

Remerciements	2
Introduction	16
Problématique de recherche	19
I Échantillonnage et reconnaissance de formes	27
1 Méthodes d'échantillonnage	29
1.1 Principe de l'échantillonnage	29
1.2 Échantillonnage uniforme	30
1.3 Échantillonnage non-uniforme	31
1.3.1 JITTERED RANDOM SAMPLING (JRS)	34
1.3.2 ADDITIVE RANDOM SAMPLING (ARS)	34
1.3.3 ADDITIVE PSEUDO-RANDOM SAMPLING (APRS)	35
1.3.4 LEVEL CROSSING SAMPLING (LCS)	36
1.3.5 Comparaison entre APRS et LCS	38
2 Reconnaissance de formes de signaux unidimensionnels	41
2.1. Définitions et approches	42
2.1.1 Formes	42

2.1.2	Reconnaissance de formes	43
2.1.3	Evaluation des approches	48
2.2.	Méthode de discrétisation	51
2.2.1	Objectifs	51
2.2.2	Architecture	53
2.2.3	Méthode de sélection et d'extraction des primitives	53
2.2.4	Reconnaissance de primitives	54
2.2.5	Classification	59
2.3.	Représentation et reconnaissance de formes	65
2.3.1	Représentation symbolique	65
2.3.2	Définitions	66
2.3.3	Éléments de la théorie des graphes	66
2.3.4	Automate fini déterministe	67
2.4.	Résultats de simulation	75
2.4.1	Données expérimentales	75
2.4.2	Partition	76
2.4.3	Représentation et reconnaissance de formes	83
II	Prototypage et architecture matérielle	89
3	Logique asynchrone	91
3.1	Concepts de base	92
3.1.1	Fonctionnement synchrone	92
3.1.2	Fonctionnement asynchrone	94
3.1.3	Caractéristiques des opérateurs	95
3.1.4	Principe de base	96

3.2	Motivations	102
3.2.1	Calcul en temps minimum	102
3.2.2	Support fiable pour les traitements non-déterministes	103
3.2.3	Modularité	104
3.2.4	Faible consommation	104
3.2.5	Faible bruit et faible émission	105
3.3	Classification des circuits asynchrones	107
3.3.1	Aléas	107
3.3.2	Classes de circuits asynchrones	108
4	Validation matérielle	115
4.1	Architecture du système de reconnaissance de formes	115
4.2	Conditions expérimentales	117
4.3	Convertisseur par traversée de niveaux	119
4.3.1	Détecteur de traversée de niveau	120
4.3.2	Convertisseur analogique-numérique	121
4.3.3	Générateur de niveaux	122
4.3.4	Compteur	124
4.4	Convertisseur par traversée de niveaux modifié	125
4.5	Détecteur de formes	126
4.6	Réalisation	128
4.7	Résultats	129
5	Consommation énergétique	133
5.1	Architecture du système de reconnaissance de formes	133
5.1.1	Architecture asynchrone	134
5.1.2	Architecture synchrone	134
5.2	Conditions expérimentales	136

5.3	Flot de conception	136
5.4	Résultats	137
5.4.1	Approche asynchrone	138
5.4.2	Approche synchrone	139
5.4.3	Comparaison de l'approche synchrone et asynchrone	140
	Conclusion	142
	Perspectives	144

Table des figures

1	Architecture générique d'un système embarqué.	22
1.1	(a) Numérisation par échantillonnage uniforme pour $f_e = 7$ Hz (b) Instants d'acquisition des échantillons (Peigne de Dirac).	32
1.2	(a) Numérisation par échantillonnage JITTERED RANDOM SAMPLING. (b) Instants d'acquisition des échantillons avec un intervalle moyen de 0.40s.	33
1.3	(a) Numérisation par échantillonnage ADDITIVE RANDOM SAMPLING. (b) Instants d'acquisition des échantillons.	35
1.4	(a) Numérisation par échantillonnage ADDITIVE PSEUDO RANDOM SAMPLING. (b) Instants d'acquisition des échantillons.	36
1.5	(a) Numérisation par échantillonnage par traversée de niveaux répartis uniformément sur la dynamique du signal (16 niveaux de quantification donnant un quantum de 0.1) (b) Instants d'acquisition des échantillons.	38
2.1	Capteur excité par des forces Φ_i induites par son environnement. La réponse à ces sollicitations se traduit sous forme de signal temporel \mathbf{x}_j	43
2.2	Superposition de formes issues d'un même capteur générées par des processus de nature similaire.	44

2.3	Représentation des images médicales dans un espace paramétrique composé des paramètres p_1 et p_2 . La ligne droite est la frontière de décision qui permet de séparer les deux classes \mathcal{C}_1 et \mathcal{C}_2	45
2.4	Représentation du signal électrocardiographique (normal) sur un cycle.	46
2.5	Représentation hiérarchique du signal ECG qui se compose de trois niveaux. Les primitives se décomposent au moyen d'une courbe croissante (c) ou décroissante (d).	47
2.6	Association (a) linéaire et (b) non-linéaire d'échantillons entre une forme de référence (prototype) et une forme à comparer (candidate).	48
2.7	(a) Forme continue caractérisée par des alternances positives et négatives. (b) Forme numérisée au moyen de l'échantillonnage par traversée de niveaux.	52
2.8	Architecture pour la détermination de partitions basée sur une structure de reconnaissance syntaxique de formes.	53
2.9	(a) Forme échantillonnée par traversée de niveaux au moyen d'une partition initiale uniforme. (b) Représentation binaire obtenue au moyen de la fonction signe.	55
2.10	(a) Forme échantillonnée par traversée de niveaux au moyen d'une partition initiale uniforme. (b) Poids des primitives.	55
2.11	(a) Poids des deux premières primitives contiguës. (b) Poids cumulé des deux premières primitives au moyen d'une fenêtre glissante Δ	56
2.12	(a) Première primitive extraite. (b) Seconde primitive extraite.	56
2.13	Module dédié à la reconnaissance de primitives de l'architecture FIGURE 2.8 qui est basé sur une architecture générique de reconnaissance de formes statistique.	57
2.14	Module d'extraction et de sélection de paramètres.	58

2.15	Représentation des probabilités jointes $p(x, \mathcal{C}_k)$ et de la frontière de décision \mathcal{D}	60
2.16	Matrice de confusion	62
2.17	(a) Position de plusieurs seuils de décision pour la séparation des distributions. Chaque seuil de décision correspond à un point dans l'espace ROC.	64
2.18	Représentation de la forme numérisée en séquence de symboles : <i>bbaabbaa</i>	66
2.19	Graphe \mathcal{G}	67
2.20	Matrice d'incidence du graphe \mathcal{G}	67
2.21	Reconnaissance d'une séquence de symboles.	68
2.22	Représentation des interactions entre le TEACHER qui est la source d'exemples et le LEARNER qui propose un état de sa compréhension de son apprentissage (conjecture).	69
2.23	Table d'observation composée des ensembles <i>prefix-closed</i> E , <i>suffix-closed</i> S , de l'ensemble issu de la concaténation de S et A , noté $S \cdot A$ et de la fonction T	70
2.24	Automate stochastique.	73
2.25	Matrice de transition d'états de référence Π^0 , celle après analyse d'une séquence de symboles d'une forme Π^k et la mesure de divergence \mathbf{M}^k	73
2.26	Capteur accélérométrique MEMS triaxial.	76
2.27	(a) Les classes de formes pour deux primitives (b) Formes non-uniformément échantillonnées modélisées par deux primitives.	77
2.28	Pourcentage moyen du taux de rejet des formes d'intérêt qui n'appartiennent pas aux classes \mathcal{C}_1 ou \mathcal{C}_2	78
2.29	Pourcentage moyen du taux de compression.	79
2.30	Pourcentage moyen de détection.	80

2.31	Estimation de la densité de probabilité de l'amplitude des primitives positives issues des formes d'intérêt (foncé) et celles des primitives à rejeter (clair).	81
2.32	Aire sous la courbe ROC pour deux primitives.	81
2.33	Ensemble des seuils qui constitue les partitions pour des formes d'intérêt constitués de deux primitives.	82
2.34	Architecture du système de reconnaissance de séquences de formes.	83
2.35	Automate Fini Déterministe pour la détection de séquences de formes.	84
2.36	Mesures des performances de détection des séquences de formes pour l'axe z	84
2.37	Architecture du système de reconnaissance de formes.	85
2.38	Méthode d'extraction de fenêtres dans les signaux physiologiques.	85
2.39	Taux moyen de vrais positifs.	86
2.40	Taux moyen de faux positifs.	87
3.1	Mécanisme de fonctionnement d'un circuit synchrone par l'intermédiaire d'un automate à états finis	92
3.2	Caractéristiques d'un module asynchrone	94
3.3	Opérateurs asynchrones	96
3.4	Protocole 2 phases (adapté de [38]).	97
3.5	Protocole 4 phases (adapté de [38]).	98
3.6	Chaîne de communication de données regroupées (adaptée de [45]).	99
3.7	Chaîne de communication avec le protocole 4-phases double rail (adaptée de [45]).	100
3.8	Codage (a) 3 et (b) 4 états des données (adapté de [38]).	101
3.9	Acquittement en codage double rail 4 phases.	101

3.10	Spectre fréquentiel de 2 microcontrôleur 80C51. Ils exécutent le même programme aux mêmes performances. (a) Le spectre de la version synchrone montre une suite de pics aux harmoniques 3-6 MHz dû à l'horloge (b) Le spectre de la version asynchrone ne montre pas ces pics et montre clairement que le spectre possède moins d'énergie.	106
3.11	Architecture simplifiée du récepteur radio-message avant et après l'utilisation d'un microcontrôleur faible bruit asynchrone.	107
3.12	Classes de circuits asynchrones.	108
3.13	Terminologie des classes de circuits asynchrones.	109
3.14	Circuit Micropipeline	112
4.1	Système implémenté matériellement.	116
4.2	Dispositif expérimental pour la validation matérielle du système de reconnaissance de formes.	118
4.3	Architecture <i>générique</i> du convertisseur analogique-numérique par traversée de niveaux qui génère en sortie un couple amplitude-temps (v_s, dt).	119
4.4	Détecteur de traversé de niveaux composé d'un réseau d'amplificateur opérationnel.	120
4.5	(a) Traversée d'un niveau de référence haut (b) Actualisation des niveaux de référence pour poursuivre la conversion.	123
4.6	Échantillonnage d'un signal bruité avec la présence d'un intervalle entre les niveaux de références $v_{REF}^{d^{(i)}}$ et $v_{REF}^{u^{(i-1)}}$	124
4.7	Matrice d'incidence de l'automate \mathcal{A}	126
4.8	Signaux obtenus au moyen du convertisseur analogique-numérique asynchrone.	129
5.1	Système de reconnaissance de formes asynchrone.	135
5.2	Système de reconnaissance de formes synchrone.	135

5.3	Flot de conception pour la simulation électrique du circuit asynchrone et synchrone au moyen de l'outil ULTRASIM.	137
5.4	Consommation du circuit asynchrone composé du module contrôleur couplé au module de reconnaissance de formes pour une tension d'alimentation de 3.3.V.	138
5.5	Consommation du circuit synchrone composé du module contrôleur couplé au module de reconnaissance de formes pour une tension d'alimentation de 3.3.V.	139
5.6	(a) Comparaison du nombre d'échantillons moyen et de la (b) consommation moyenne entre le circuit asynchrone et synchrone.	140
5.7	Ratio d'échantillons entre l'approche synchrone et asynchrone.	141
5.8	Ratio de consommation entre l'approche synchrone et asynchrone.	141
5.9	Récapitulatif de la consommation du circuit asynchrone composé du module contrôleur couplé au module de reconnaissance de formes pour plusieurs tensions d'alimentation.	142

Introduction

Le développement des ordinateurs portables et des réseaux sans-fil (e.g. WLAN) au début des années 1990 a mené vers de nouvelles perspectives de recherche concernant la disponibilité de ressources informatiques à des clients mobiles quel que soit le temps et le lieu où ils se trouvent. Ceci a donné lieu à la naissance d'un nouveau domaine, l'informatique embarquée. Les contraintes qui émergent naturellement de la mobilité concernent : les variabilités non-prévisibles de la qualité de réseau pour la communication à distance, la fiabilité et la robustesse réduites, ainsi que les limitations des ressources imposées par les contraintes de poids et de taille, et la préoccupation de la limitation énergétique de la batterie qui alimente ces systèmes mobiles [39] [41].

Les innovations dans l'informatique embarquée ont donné lieu au développement d'un nouveau type de traitement, le traitement pervasif, dont le précurseur est M. Weiser [55]. Sa vision consistait à rendre la technologie *invisible*, c'est-à-dire à la fondre dans l'environnement. Ces systèmes regroupent tous les appareils possédant une intelligence active et passive. Contrairement aux systèmes mobiles classiques qui réagissent aux problèmes de localité et de mobilité (e.g. qualité de réseau), les systèmes pervasifs sont plus complexes car ils sont sensibles au contexte dans lequel ils évoluent au moyen de capteurs [39] [41]. De plus, ils s'adaptent en conséquence pour satisfaire les besoins de l'utilisateur par l'intermédiaire d'une intelligence artificielle. Les systèmes pervasifs sont conçus pour s'intégrer dans les infrastructures tout en minimisant l'attention de son utilisateur, c'est-à-dire satisfaire ses besoins avec un minimum d'interactions et donc de

façon quasi-autonome. La mobilité faisant partie intégrante de la vie de tous les jours, les recherches dans le domaine des systèmes pervasifs se sont naturellement dirigées vers les systèmes ubiquitaires. En plus des caractéristiques des systèmes pervasifs, les systèmes ubiquitaires comportent une contrainte supplémentaire de mobilité [39] [41].

Les appareils de téléphonie mobile de dernière génération peuvent être considérés comme un des éléments majeurs des systèmes ubiquitaires, étant donné leur popularité. Ils possèdent une grande gamme de capteurs (e.g. accéléromètres, GPS, gyroscopes, microphones et caméras) [5] [7] [27]. Les informations qu'ils délivrent permettent aux systèmes ubiquitaires ainsi qu'aux systèmes pervasifs de s'adapter en fonction de leur environnement sous contrainte des besoins de leurs utilisateurs. La reconnaissance d'une situation ou d'un contexte particulier peut se décrire au moyen de la reconnaissance d'une forme particulière contenue dans un signal généré par des capteurs. Les appareils mobiles possèdent des ressources limitées en terme de puissance de calcul, de mémoire et d'autonomie qui limitent leurs performances. Ainsi le traitement de reconnaissance de formes doit être conçu de manière intelligente pour optimiser les performances en fonction de l'autonomie [7] [5].

L'évolution technologique des batteries étant lente par rapport à la progression technologique des systèmes, décrite par la loi de Moore, ceci justifie naturellement l'intérêt des recherches pour les techniques de faible consommation afin de réduire leur taille et augmenter leur autonomie [46] [8].

Une approche intéressante pour prolonger l'autonomie des systèmes pervasifs et ubiquitaires est de réduire le nombre de données à traiter. Une stratégie couramment employée pour la numérisation d'un signal est basée sur un échantillonnage uniforme où des échantillons sont prélevés périodiquement, cependant elle ne permet pas une évaluation de la pertinence des données. Il faut dans ce cas proposer un type d'échantillonnage qui soit adapté aux formes à reconnaître, de telle sorte qu'il ne capture des échantillons qu'en présence d'information pertinente, ce qui permettrait d'alléger considérablement

la charge de calcul [1] [27] [42] [48] [54].

Ce manuscrit a pour objectif de présenter de nouvelles stratégies de réduction de consommation dédiées aux systèmes limités en ressources. Il est structuré en deux parties : la première abordera les aspects théoriques de l'échantillonnage et de la reconnaissance de formes. La seconde partie traite de l'implémentation matérielle sur FPGA et des estimations de consommation sur ASIC.

Dans la première partie, le chapitre 1 décrit les différentes stratégies d'échantillonnage qui se divisent en deux approches : uniforme et non-uniforme. Les différentes stratégies de l'échantillonnage non-uniforme sont présentées afin de déterminer la plus adaptée. L'approche retenue, nommée échantillonnage par traversée de niveaux, sera appliquée à la reconnaissance de formes dans le chapitre 2. Une méthode pour déterminer les niveaux adaptés sera élaborée. Des résultats de simulation issus de signaux expérimentaux viendront justifier la pertinence de l'approche retenue.

Dans la deuxième partie, le chapitre 3 explique les mécanismes et l'intérêt de la logique asynchrone. Une implémentation matérielle sur FPGA de la chaîne de traitement pour la reconnaissance de formes fait l'objet du chapitre 4. L'acquisition des données est réalisée par l'échantillonnage non-uniforme par traversée de niveaux. Le chapitre 5 compare la consommation énergétique de la partie détection de la chaîne de traitement, dans les cas synchrone et asynchrone (liés respectivement à l'échantillonnage uniforme et non-uniforme). Les conclusions et perspectives viennent clore ce manuscrit.

Problématique de recherche

Les systèmes pervasifs sont de plus en plus présents et nous accompagnent au quotidien. Cette omniprésence entraîne une course à la miniaturisation. Ils sont conçus de manière à répondre aux besoins de leur utilisateur en limitant les interactions, ils sont ainsi quasi-autonome. Ceci est réalisable en récoltant les informations pertinentes présentes dans leur environnement [8] [29]. Ces systèmes sont dotés d'un sous-système capteur qui est composée d'organes de perception et permettent ainsi de percevoir leur environnement. Dans certains cas, les données issues des capteurs sont traitées par un module de détection (reconnaissance) afin d'en extraire l'information pertinente. Les systèmes pervasifs sont également étudiés de manière à minimiser leur consommation et ainsi augmenter leur durée de vie. C'est dans ce contexte que s'inscrivent ces travaux. Ils ont pour objectif de montrer l'intérêt de proposer de nouvelles stratégies de traitement du signal, afin de réduire la consommation énergétique dans les systèmes médicaux tels que les implants par exemple.

Les implants médicaux

Un implant médical, qui est un système embarqué, est un appareil de petite taille qui se compose de plusieurs sous-systèmes (c.f. **FIGURE 1**) : un capteur (et acquisition) qui permet de mesurer une information dans son environnement ; le sous-système de traitement (et mémoire) qui extrait les informations pertinentes issues du capteur ; un

actionneur qui agit sur son environnement et un sous-système de commande qui effectue l'interface entre le traitement et l'actionneur [8].

Le système est alimenté en énergie par l'intermédiaire d'une batterie en général, et qui présente donc un budget énergétique limité. Il peut être impossible ou peu pratique de recharger la batterie [20]. À plus long terme, des éléments de récupération d'énergie dans leur environnement seraient une solution plus définitive. Cependant, l'implant doit posséder une durée de vie suffisamment longue pour remplir les exigences de son application. Il est donc primordial de s'intéresser aux méthodes qui puissent permettre de réduire la consommation énergétique de ce type de système [8].

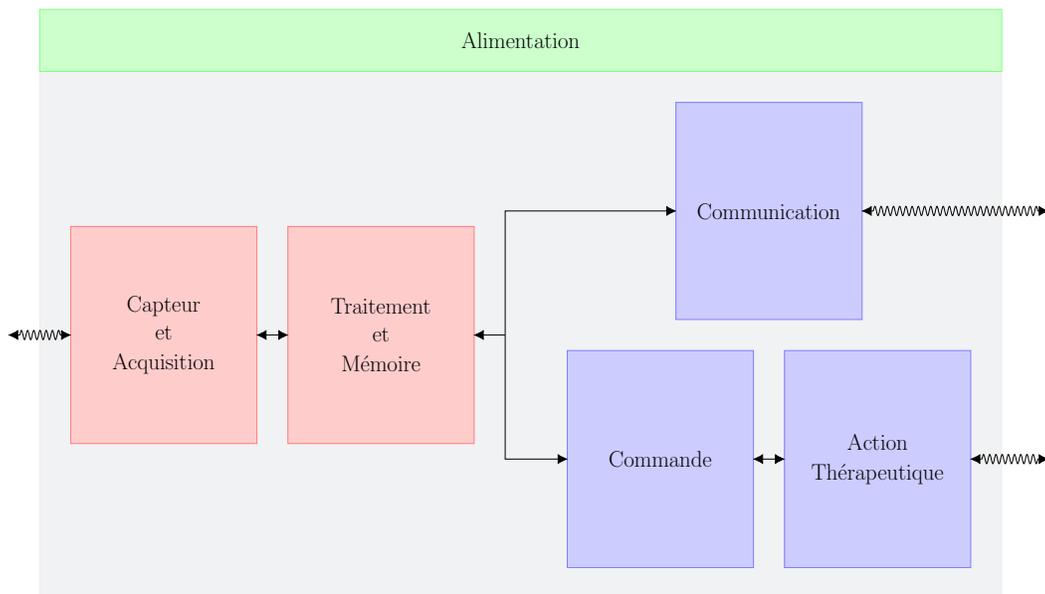


FIGURE 1 - Architecture générale d'un système embarqué.

Des mesures expérimentales ont démontré que l'actionnement est une opération très coûteuse en terme d'énergie alors que le coût de traitement des données est plus faible. Par ailleurs, la consommation énergétique du sous-système de traitement dépend du type de capteur. Enfin l'énergie consommée par les modules de traitement et commande peut être évaluée à environ 20 % à 50 % de la consommation énergétique de l'implant, selon son fonctionnement. Une économie importante ne sera donc pas négligeable pour

le système.

Capteur unique

Le propre d'un implant médical est d'être inséré dans le corps humain. La durée et la complexité de l'opération chirurgicale pour l'implantation sont induits principalement par le nombre de capteurs à l'extérieur du boîtier de contrôle. Chacun des capteurs doit être positionné de façon précise dans le corps du patient pour effectuer des mesures précises ce qui demande du temps et de la dextérité de la part du chirurgien. Par conséquent, plus le nombre de capteurs est faible, plus la durée de l'opération est courte. L'utilisation d'un seul capteur est ainsi fortement apprécié pour alléger l'opération mais également l'anesthésie. Ces conditions permettent en plus un rétablissement plus rapide de l'état de santé du patient concerné. L'unicité du capteur est donc un enjeu important (médical, sociétal et par conséquent économique) [20].

Bien que l'utilisation d'un unique capteur permette un meilleur confort pour le patient et le chirurgien, les exigences envers le système doivent être plus rigoureuses. En effet, l'utilisation d'un capteur ne permet pas d'obtenir une redondance temporelle et spatiale de l'information pour augmenter la robustesse au bruit de l'implant. Le capteur pourra aussi être amené à mesurer plusieurs signaux qui devront être discriminés par le sous-système traitement. Les données du capteur doivent être ainsi traitées de manière efficace pour en extraire au mieux l'information pertinente. L'implant doit, en plus des contraintes énergétiques, posséder de bonnes performances et assurer la robustesse.

Réduction d'énergie

Sachant que le sous-système capteur est le premier élément de la chaîne de traitement de l'implant, il possède un impact majeur dans la consommation énergétique du système. Ce module comprend notamment un convertisseur analogique-numérique qui

est responsable de l'échantillonnage du signal continu issu du capteur. Un échantillonnage inadapté des données contribue à une activité inutile de la partie capteur et qui induit naturellement une perte d'énergie. L'énergie peut être conservée au moyen d'une stratégie d'échantillonnage adaptée [1] [26] [42] [48] [54].

Une solution intuitive qui pourrait considérablement réduire l'activité serait de capturer les échantillons seulement s'ils sont jugés pertinents. La réduction du nombre d'échantillons va permettre d'induire une diminution de l'activité d'échantillonnage du sous-système capteur et par conséquent de son énergie consommée. Cette réduction dans le nombre de données à traiter va se répercuter positivement sur l'ensemble de la chaîne de traitement si l'échantillonnage est conçu de manière judicieuse. En effet, ce bénéfice est effectif si l'information pertinente contenue dans le signal issu du capteur est préservée. Dans le cas contraire, même si le nombre d'échantillons est réduit, le coût de traitement sera important étant donné qu'il nécessitera un plus grand nombre d'opérations pour *compenser* cette perte d'information. En d'autres termes, le sous-système capteur doit condenser l'information pour être en adéquation avec le sous-système traitement.

L'activité de la chaîne de traitement et plus particulièrement celle du sous-système capteur dépend de la nature du signal d'intérêt à échantillonner. Naturellement, plus ce signal est sporadique (peu fréquent) plus faible sera l'activité et donc l'énergie consommée [26] [27].

Conclusion

L'implant médical est un système qui doit de préférence comporter un seul capteur pour faciliter son implantation chez le patient. Cette contrainte induit des critères pointus en termes de performances et de robustesse au bruit. Étant donné que l'implant est un système autonome, il possède un budget énergétique restreint qui limite sa durée

de vie.

Le système n'étant pas facilement rechargeable, il faut utiliser cette énergie de manière efficace et effectuer un traitement seulement quand il est nécessaire, voire primordial [20]. Il est aussi possible d'envisager la mise en veille du système lors d'une longue inactivité.

Une solution intéressante pour la conservation d'énergie consiste à adapter l'échantillonnage en fonction de la pertinence de l'information des données issues des capteurs. La détermination du schéma d'échantillonnage fera donc l'objet du chapitre suivant. La problématique de ces travaux s'inscrit dans la recherche sur l'optimisation énergie-performance.

Première partie

Échantillonnage et reconnaissance de formes

Chapitre 1

Méthodes d'échantillonnage

Introduction

Dans un objectif de minimisation d'énergie, une des premières étapes dans la reconnaissance de formes est l'échantillonnage. Dans ce chapitre nous allons procéder à un état de l'art de ce domaine afin d'effectuer un choix pertinent et justifié pour notre application. Dans une première partie nous décrirons le principe de l'échantillonnage. Les deux parties suivantes présenteront les cas uniforme et non-uniforme. Nous terminerons ce chapitre par une conclusion dans laquelle nous justifierons le choix de la méthode retenue.

1.1 Principe de l'échantillonnage

L'échantillonnage consiste à représenter un signal en fonction du temps $x(t)$ par ses valeurs discrètes $x(t_k)$ aux instants t_k . La séquence d'échantillonnage $s(t)$, appelée peigne de Dirac est une distribution de masse unitaire aux instants t_k et nulle ailleurs. Elle est définie par :

$$s(t) = \sum_{k=-\infty}^{+\infty} \delta(t - t_k) \quad (1.1)$$

Dans le domaine temporel, la discrétisation du signal continu $x(t)$ s'effectue en le multipliant par $s(t)$. L'expression du signal échantillonné est définie par :

$$x_e(t) = x(t) \cdot s(t) = \sum_{k=-\infty}^{+\infty} x(t) \cdot \delta(t - t_k) \quad (1.2)$$

Dans le domaine fréquentiel, l'échantillonnage est un produit de convolution entre le spectre $x(t)$ et $s(t)$:

$$X_e(f) = X(f) \otimes S(f) \quad (1.3)$$

L'échantillonnage est directement influencé par les instants d'échantillonnage t_k car selon sa distribution, l'échantillonnage se décline en 2 approches : uniforme et non-uniforme.

1.2 Échantillonnage uniforme

La théorie de la communication associée au mode d'échantillonnage uniforme a été introduite par Shannon. L'échantillonnage s'applique à un signal à bande limitée de période d'échantillonnage T_e respectant le théorème d'échantillonnage WKS [18]. La séquence d'échantillonnage $s(t)$ est, dans le cas uniforme, une distribution de masse unitaire aux instants multiples de la période d'échantillonnage T_e . Elle représente une abstraction théorique de l'échantillonnage. En réalité, les instants d'échantillonnages ne sont pas déterministes à cause de fluctuations et provoquent de la non-uniformité. En effet, ces variations aléatoires induisent un *jitter* autour du front idéal de l'horloge.

$$s(t) = \sum_{k=-\infty}^{+\infty} \delta(t - k \cdot T_e) \quad \text{avec} \quad t_k = k \cdot T_e \quad (1.4)$$

Pour un signal analogique $x(t)$ à bande limitée entre $[-f_{\text{MAX}}, +f_{\text{MAX}}]$, la fréquence d'échantillonnage uniforme f_e , qui est l'inverse de la période d'échantillonnage T_e doit

être supérieure à la période de Nyquist f_{NYQUIST} définie par l'équation suivante :

$$f_{\text{NYQUIST}} = 2 \cdot f_{\text{MAX}} \quad (1.5)$$

Le signal obtenu, en échantillonnant $x(t)$ par $s(t)$ est défini par l'équation (1.6) et illustrée sur la **FIGURE 1.1**.

$$x_e(t) = x(t) \cdot s(t) = \sum_{k=-\infty}^{+\infty} x(k \cdot T_e) \cdot \delta(t - k \cdot T_e) \quad (1.6)$$

Le spectre du signal échantillonné $X_e(f)$ a pour expression le produit de convolution de $X(f)$ par $S(f)$, respectivement les transformées de Fourier de $x(t)$ et $s(t)$.

$$S(f) = \frac{1}{T_e} \sum_{k=-\infty}^{+\infty} \delta(f - k \cdot f_e) \quad (1.7)$$

$$X_e(f) = X(f) \otimes S(f) = \frac{1}{T_e} X(f - k \cdot T_e) \quad (1.8)$$

L'échantillonnage induit la périodisation du contenu spectral dans le domaine fréquentiel $X_e(f)$ autour de k/T_e selon l'équation 1.8. Le signal $x(t)$ peut être récupéré grâce à un filtrage passe-bas idéal si la fréquence d'échantillonnage respecte le critère de Nyquist. Si ce n'est pas le cas, le signal récupéré est distordu à cause du repliement spectral.

1.3 Échantillonnage non-uniforme

L'échantillonnage non-uniforme, intrinsèque aux systèmes réels, est soit non désiré ou utilisé de façon délibéré. Par exemple, en astronomie, la luminosité de différentes étoiles est enregistrée pour décrire leur contenu fréquentiel. Les mesures sont faites sur plusieurs années, et celles-ci sont corrompues par les conditions météorologiques et

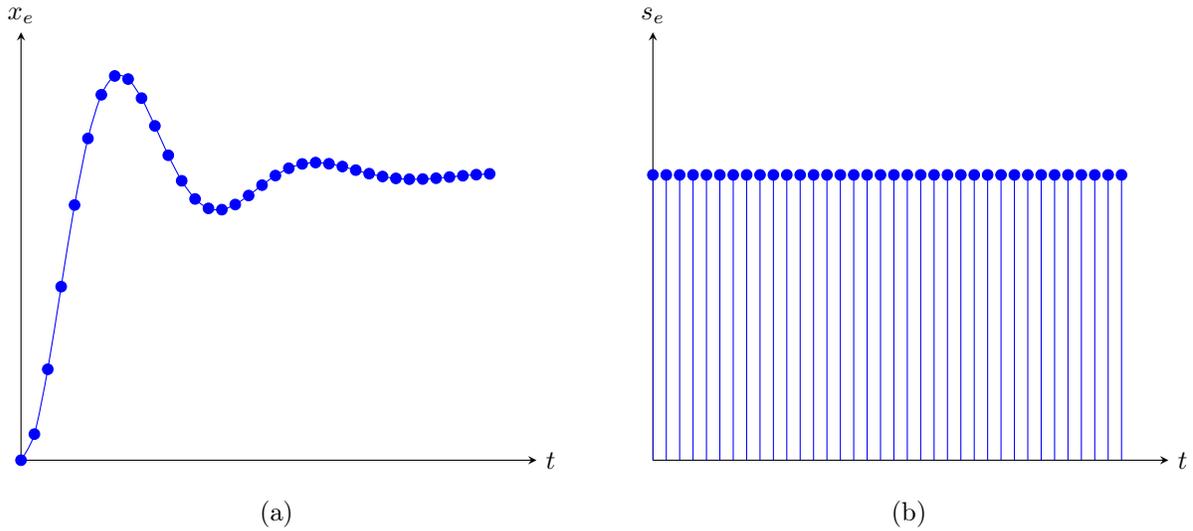


FIGURE 1.1 - (a) Numérisation par échantillonnage uniforme pour $f_e = 7$ Hz (b) Instants d'acquisition des échantillons (Peigne de Dirac).

les pannes de télescope. Ceci mène à une longue période de données manquantes et donc induit une non-uniformité dans l'échantillonnage. La séquence d'échantillonnage non-uniforme pour les instants irréguliers t_k est également un peigne de Dirac mais contrairement au cas uniforme, les instants d'échantillonnages peuvent avoir n'importe quelle distribution possible.

$$s(t) = \sum_{k=-\infty}^{+\infty} \delta(t - t_k) \quad (1.9)$$

Le modèle d'échantillonnage reste linéaire dans le domaine temporel. Le signal numérique obtenu à partir des échantillons irréguliers $x(t_k)$ s'écrit selon l'équation suivante :

$$x_e(t) = x(t) \cdot s(t) = \sum_{-\infty}^{+\infty} x(t_k) \cdot \delta(t - t_k) \quad (1.10)$$

L'analyse spectrale n'utilise pas les mêmes outils que dans le cas de données uniformément réparties car elle est dépendante de plusieurs conditions et paramètres.

L'implémentation matérielle de l'échantillonnage mène à des fluctuations dans les

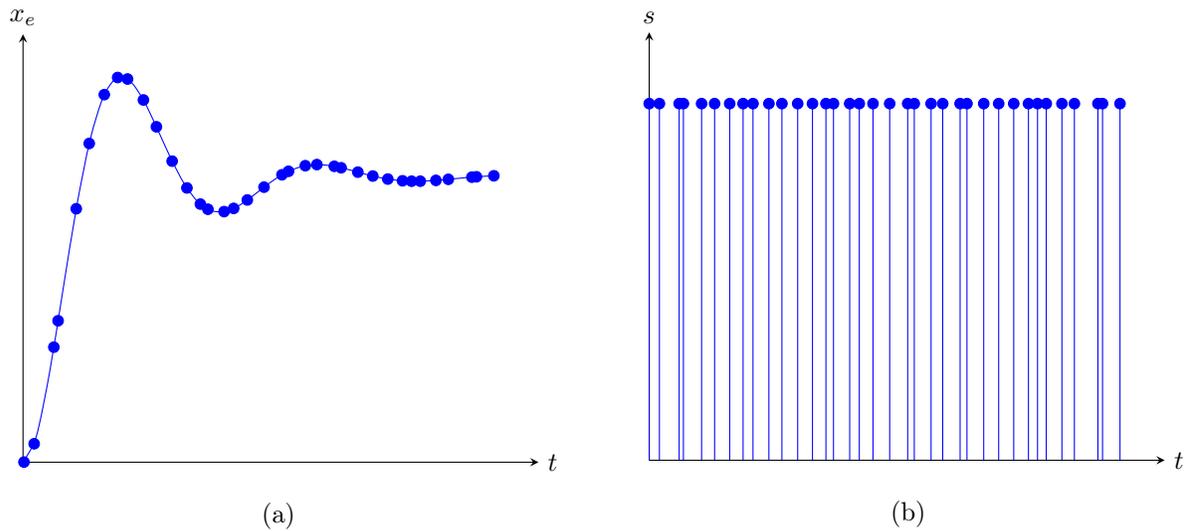


FIGURE 1.2 - (a) Numérisation par échantillonnage JITTERED RANDOM SAMPLING. (b) Instants d'acquisition des échantillons avec un intervalle moyen de 0.40s.

instants d'échantillonnage. Ils ne sont plus déterministes comme décrit par l'équation 1.1. Ceci provient du *jitter* des appareils générants l'échantillonnage. Ce phénomène est toujours présent dans les systèmes de conversion basés sur un échantillonnage uniforme à cause du bruit de phase de l'horloge d'échantillonnage. Dans la majeure partie des cas, ce phénomène peut être négligé pour conserver les techniques habituelles de traitement du signal. Cependant, une fluctuation de la fréquence d'échantillonnage peut présenter un avantage.

H.S. Shapiro et R.A. Silverman (1960) [44] ont proposé d'utiliser le phénomène pour palier le problème de recouvrement spectral (en anglais *aliasing*) présent quand la fréquence d'échantillonnage ne respecte pas le critère de Nyquist. En effet, si la fréquence d'échantillonnage peut prendre n'importe quelle valeur dans une bande donnée, le spectre du signal échantillonné n'est plus périodisé et donc il n'y a plus de repliement spectral. Cette observation a abouti à 2 types d'échantillonnages : le JITTERED RANDOM SAMPLING (JRS), le ADDITIVE RANDOM SAMPLING (ARS) et sa variante, le ADDITIVE PSEUDO RANDOM SAMPLING (APRS).

1.3.1 JITTERED RANDOM SAMPLING (JRS)

Pour cet échantillonnage défini par [44], illustré en **FIGURE 1.2**, les marqueurs temporels t_k sont définis par une composante déterministe et aléatoire.

$$t_k = k \cdot T_e + \xi_k \quad \text{avec} \quad k \in \mathbb{Z} \quad (1.11)$$

Avec ξ_k est une famille de variables aléatoires indépendantes et indentiquement distribuées, de valeur moyenne nulle et généralement gaussienne et avec T_e l'intervalle moyen entre 2 instants d'échantillonnage.

$$p(\tau) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma}} \cdot e^{-\frac{\tau^2}{2\sigma^2}} \quad (1.12)$$

La condition d'un faible jitter s'exprime par $\sigma \ll T_e$, en effet quand $\sigma = 0$, $p(\tau)$ se réduit à une fonction de Dirac se traduisant par un échantillonnage idéal et périodique.

1.3.2 ADDITIVE RANDOM SAMPLING (ARS)

Cet échantillonnage, illustré en **FIGURE 1.3**, a également été proposé par H.S. Shapiro et R.A. Silverman (1960) [44] pour supprimer entièrement le recouvrement spectral. Chaque instant d'échantillonnage, définis par l'équation 1.13, dépend du précédent (cumulatif) en additionnant une variable aléatoire indépendante.

$$t_k = t_{k-1} + \xi_k = t_0 + \sum_{i=1}^k \xi_i \quad \text{avec} \quad k \in \mathbb{Z} \quad (1.13)$$

Avec ξ_k , une famille de variables aléatoires indépendantes et indentiquement distribuées, de valeur moyenne nulle et généralement gaussienne et avec T_e l'intervalle moyen entre deux instants d'échantillonnage. La densité de probabilité du k^e instant d'échantillonnage est défini par k opérations de convolutions sur $p_1(t)$ qui est la densité de probabilité du premier instant d'échantillonnage.

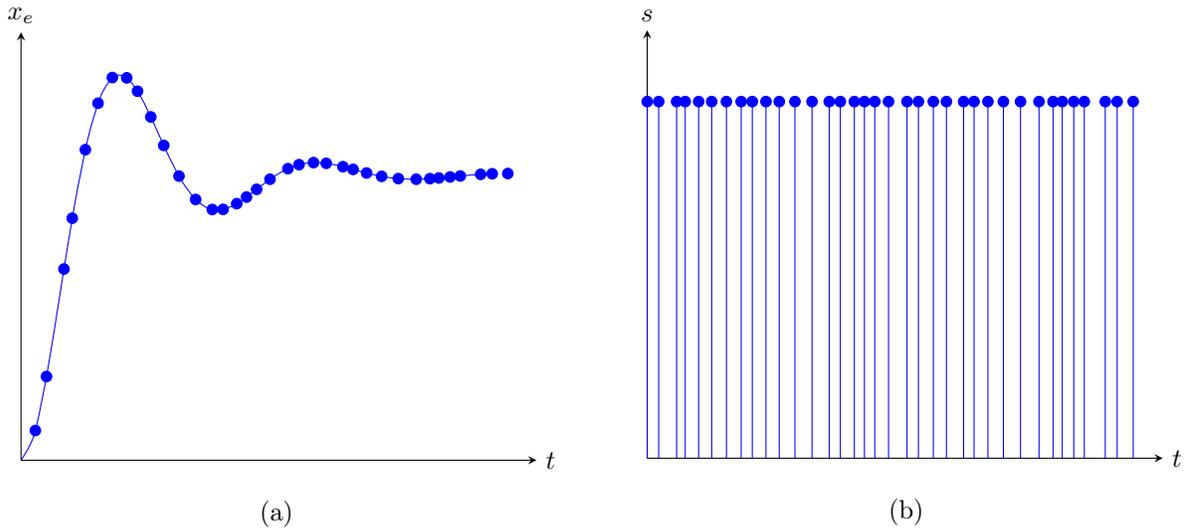


FIGURE 1.3 - (a) Numérisation par échantillonnage ADDITIVE RANDOM SAMPLING. (b) Instants d'acquisition des échantillons.

1.3.3 ADDITIVE PSEUDO-RANDOM SAMPLING (APRS)

Cet échantillonnage, illustré en **FIGURE 1.4**, introduit par P.C. Bagshaw et M. Sarhadi [4] a été mis en évidence avec des signaux large bande et échantillonnés en dessous de la fréquence de Nyquist, dans le cadre de la suppression du recouvrement spectral. Les instants d'échantillonnages sont définis par l'équation suivante :

$$t_k = t_{k-1} + \xi \quad \text{avec} \quad k = 1, 2, \dots, N - 1 \quad (1.14)$$

Avec ξ qui prend une valeur ξ_i parmi un ensemble fini de P valeurs.

$$\xi \in \{\xi_i \mid i = 1, 2, \dots, P\} \quad (1.15)$$

Afin de supprimer le recouvrement spectral, il est nécessaire de déterminer les limites de la bande de fréquence du signal d'entrée $x(t)$. Soit $X_e(f)$, le spectre du signal échantillonné et f_p la fréquence du spectre du signal $x(t)$.

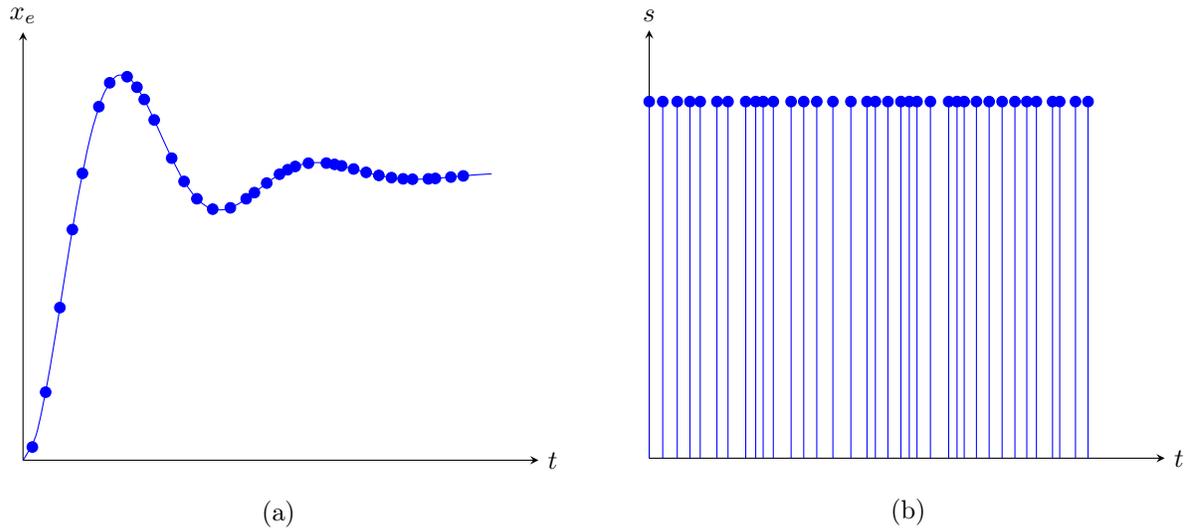


FIGURE 1.4 - (a) Numérisation par échantillonnage ADDITIVE PSEUDO RANDOM SAMPLING. (b) Instants d'acquisition des échantillons.

$$X_e(f + f_p) = \sum_{k=0}^{N-1} e^{-2j\pi(f+f_p)t_k} = \sum_{k=0}^{N-1} e^{-2j\pi f t_k} \cdot e^{-2j\pi f_p t_k} \quad (1.16)$$

Ainsi, $X_e(f + f_p) = X_e(f)$ si et seulement si $e^{-2j\pi f_p t_k} = 1$. Il faut déterminer f_p de telle manière que pour tout t_k , le produit $f_p \cdot t_k \in \mathbb{N}$ et ainsi éviter le recouvrement spectral. Il n'y a pas de recouvrement spectral si $f_{\text{MAX}} \leq f_p/2$. Ceci est valide tant que $F_{\text{MAX}} < f_{\text{MAX}} \leq f_p/2$ avec F_{MAX} la fréquence maximale obtenue avec l'ensemble des périodes t_k .

1.3.4 LEVEL CROSSING SAMPLING (LCS)

L'échantillonnage dit par traversée de niveaux a été employé par J.W. Mark et T.D. Todd [27] dans le cadre de la compression de données (c.f. **FIGURE 1.5**). Cette méthode consiste à définir un ensemble de niveaux sur l'axe d'amplitude. Contrairement aux types d'échantillonnages précédents, il n'est plus nécessaire d'utiliser une horloge pour effectuer l'échantillonnage. Un échantillon est capturé quand le signal d'entrée $x(t)$

croise un niveau pré-défini [42] [54]. J.P. Crutchfield et N.H. Packard [31] ont démontré que pour certains signaux déterministes il peut exister un placement de seuil ou le partitionnement est optimal sauf dans le cas de signaux stochastiques.

L'échantillonnage par croisement de niveaux permet de prendre en compte les caractéristiques du signal $x(t)$. Un partitionnement efficace permet d'éliminer les échantillons contenant peu d'information, correspondant aux zones de faible activité. Et à l'opposé capture les échantillons contenant de l'information pertinente caractérisé par une forte activité. Le nombre d'échantillons ainsi réduit, permet de diminuer l'activité de l'ensemble de la chaîne de traitement du signal et par conséquent sa consommation énergétique.

Chaque échantillon est un couple (x_k, t_k) représentant respectivement l'amplitude et le temps. Les instants d'échantillonnages sont définis par l'équation suivante :

$$t_k = t_{k-1} + dt_k \quad k = 1, 2, \dots, N - 1 \quad (1.17)$$

Avec t_k , l'instant courant d'échantillonnage t_{k-1} son prédécesseur et dt_k le temps écoulé entre ces deux instants. Pour $k = 0$, sont définis par une valeur nulle. Contrairement aux autres méthodes non-uniformes présentées, il n'y a pas de connaissance *a priori* des instants d'échantillonnage.

Ce schéma d'échantillonnage s'avère intéressant dans le cas où, dans un signal, il existe des régions (amplitude) distinctes qui contiennent l'information pertinente. De plus, l'échantillonnage par traversée de niveaux offre la possibilité d'adapter l'acquisition spécifiquement aux signaux d'intérêts.

S. Garcia et al. [17] présente une taxonomie des différentes méthodes pour la détermination du positionnement des niveaux de discrétisation.

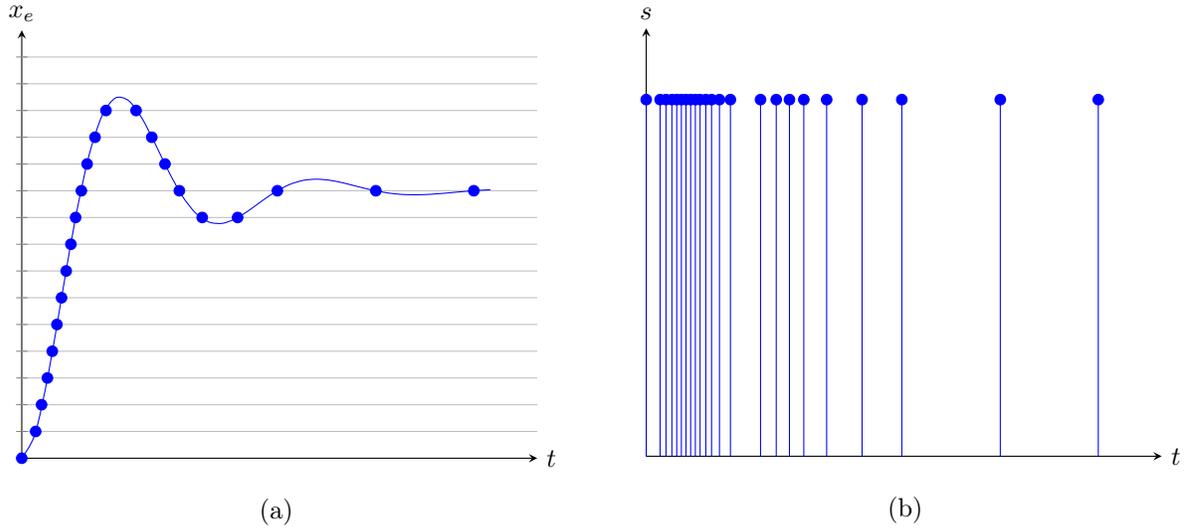


FIGURE 1.5 - (a) Numérisation par échantillonnage par traversée de niveaux répartis uniformément sur la dynamique du signal (16 niveaux de quantification donnant un quantum de 0.1) (b) Instants d'acquisition des échantillons.

1.3.5 Comparaison entre APRS et LCS

Pour l'échantillonnage LCS, un timer de résolution finie est utilisé pour mesurer dt_k . Soit T_{TIMER} le pas du timer, l'équation suivante définit les valeurs de dt_k :

$$t_k = \alpha \cdot T_{\text{TIMER}} = \{T_{\text{TIMER}_k} \mid k = 1, 2, \dots, P\} \quad \text{avec} \quad \alpha \in \mathbb{N}^* \quad (1.18)$$

En pratique, T_{TIMER_k} est un ensemble fini dont sa longueur dépend de la résolution du timer. T_{TIMER} définit la résolution du timer et si $P \cdot T_{\text{TIMER}}$ est la valeur maximale que peut atteindre le timer (e.g. saturation), alors α s'étend de $\alpha = 1, 2, \dots, P$. L'établissement du lien entre ce schéma d'échantillonnage et l'APRS a été proposé par S.M. Qaisar [33]. Dans l'équation 1.14, ξ est le pas d'instant d'échantillonnage pour le APRS et il peut prendre n'importe quelle valeur à partir d'un ensemble fini de pas d'échantillonnage ξ_k . De façon similaire, dans l'équation 1.17, lors de la conversion, dt_k peut prendre une valeur égale parmi les valeurs pré-calculées T_{TIMER_k} .

Malgré les ressemblances, il existe une différence majeure entre les 2 techniques, c'est la façon de choisir le ξ et le dt_k à partir des ensembles ξ_k et T_{TIMER_k} . Pour le APRS, le choix de ξ_k dépend principalement de la fonction génératrice R. Pour le LCS, le choix de dt_k de T_{TIMER_k} dépend du placement des seuils et des caractéristiques du signal d'entrée $x(t)$.

Conclusion

L'échantillonnage est la première étape dans la discrétisation des signaux analogiques utilisés dans un système numérique. Par conséquent, le type d'échantillonnage doit être choisi judicieusement car il impose les performances d'une chaîne de traitement du signal.

L'échantillonnage uniforme est cadencé par une horloge réglée de manière à respecter le critère de Nyquist pour éviter tout recouvrement spectral. Ce type d'échantillonnage ne prend pas en compte les caractéristiques du signal et par conséquent capture périodiquement des échantillons même s'ils ne possèdent pas forcément de pertinence. Cela mène par conséquent à une activité inutile du système de traitement et de ses périphériques [48] [54].

Une amélioration pertinente des ressources consiste à choisir un type d'échantillonnage qui prend en compte les caractéristiques du signal. Avec l'échantillonnage LCS, le taux d'échantillonnage s'adapte en fonction du signal : il est faible quand le signal varie lentement et augmente quand le signal varie plus rapidement [42]. Ainsi, cela pourrait permettre d'éviter toute surcharge inutile de la chaîne de traitement et de ses ressources. Pour cela, le nombre et le positionnement des seuils doivent être choisis judicieusement en fonction des caractéristiques du signal d'intérêt.

Chapitre 2

Reconnaissance de formes de signaux unidimensionnels

Définitions et approches

Introduction

Ce chapitre permet dans un premier temps de définir de façon générale une forme dans un signal issu d'un capteur et le principe de la reconnaissance de formes.

La problématique de reconnaissance peut être résolue au moyen de plusieurs stratégies, dont les concepts sont fondamentalement différents et seront décrits dans ce chapitre. Ces stratégies sont évaluées en fonction de leurs caractéristiques, pour permettre de déterminer celles qui sont les plus adaptées à nos travaux et ainsi choisir l'architecture de reconnaissance de formes optimale.

2.1.1 Formes

On considère le sous-système capteur (et acquisition) de l'architecture de l'implant (c.f. **FIGURE 2.1**). Les signaux issus du capteur considéré dans ces travaux évoluent de façon continue au cours du temps.

Le capteur subit des sollicitations (forces) qui sont produites par différents processus physiques dans l'environnement dans lequel il se situe. Ces forces produisent des variations dans la réponse temporelle du capteur [32]. En considérant un processus particulier de nature non-aléatoire, la réponse du capteur va posséder une certaine structure identifiable. Cette structure qui évolue dans le temps est nommée forme [43]. C'est une sous-séquence temporelle contenue dans le signal temporel du capteur.

Soit un ensemble de processus physiques de même nature qui exercent des sollicitations sur le capteur.

Pour chacun des processus Φ_i , une forme particulière \mathbf{x}_i va être générée. Malgré l'exis-

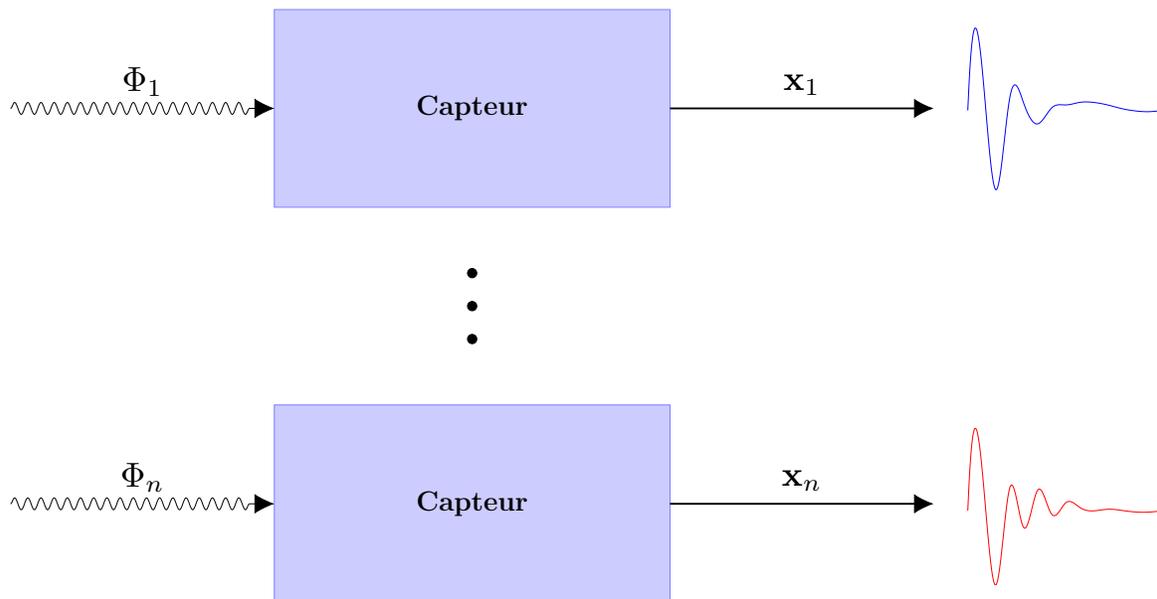


FIGURE 2.1 - Capteur excité par des forces Φ_i induites par son environnement. La réponse à ces sollicitations se traduit sous forme de signal temporel x_j .

tence de distorsions temporelles et de variations d'amplitude, ces formes possèdent une structure quasi-similaire et donc une même signature sous-jacente étant donné que les processus sont de même nature (c.f. **FIGURE 2.2**).

2.1.2 Reconnaissance de formes

La reconnaissance de formes a pour objectif de concevoir des outils automatiques et informatiques capables d'associer des formes dans des catégories, communément appelées classes. Il n'existe pas une théorie unifiée de la reconnaissance de formes [24]. Il y a plusieurs approches qui présentent des concepts fondamentalement différents. Les principales sont : l'approche syntaxique (structurelle), l'approche statistique et l'approche par filtrage de formes, que nous allons décrire ci-après.

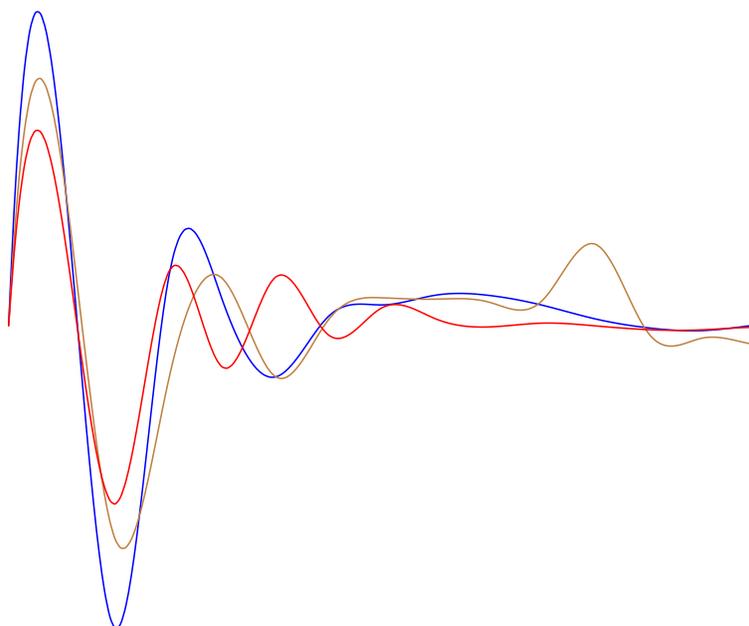


FIGURE 2.2 - Superposition de formes issues d'un même capteur générées par des processus de nature similaire.

2.1.2.1 Approche statistique

Dans cette approche, un ensemble de paramètres, appelé vecteur paramétrique, est extrait des formes. Ce vecteur, noté p , permet de définir une forme [21] [14]. Soit un espace construit à partir des paramètres du vecteur, une forme est représentée par un point dont ses coordonnées sont définies par chacune des mesures des paramètres. Cet espace est appelé espace paramétrique (c.f. **FIGURE 2.3**).

En considérant un ensemble de formes, l'approche statistique consiste dans un premier temps à délimiter des régions dans l'espace paramétrique. Les lignes de séparation sont appelées frontières de décision. Chaque région est associée à une classe particulière, qui est au nombre de 2 dans le cas de la **FIGURE 2.3**). Par la suite, chaque point (forme) est associée à sa classe correspondante.

L'espace de représentation doit être construit à partir d'un vecteur de paramètres qui puisse représenter les formes de classe identique dans des régions compactes et

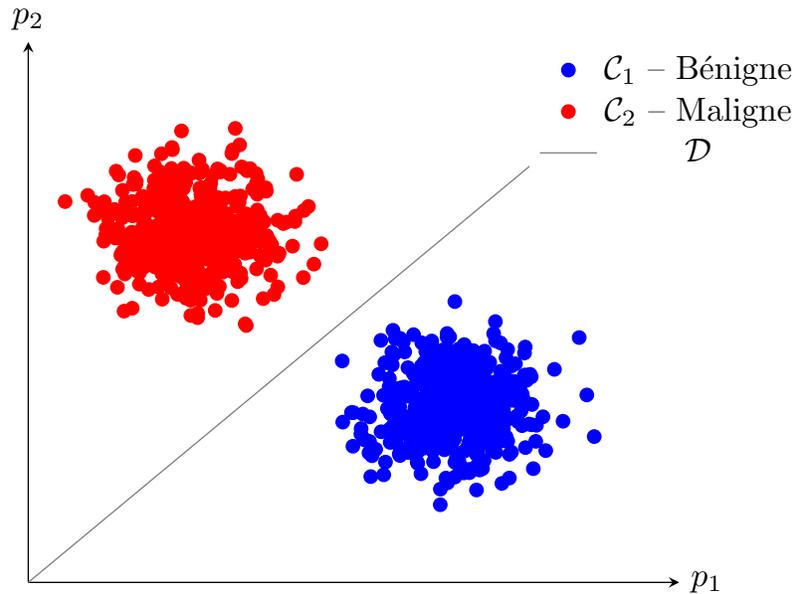


FIGURE 2.3 - Représentation des images médicales dans un espace paramétrique composé des paramètres p_1 et p_2 . La ligne droite est la frontière de décision qui permet de séparer les deux classes \mathcal{C}_1 et \mathcal{C}_2 .

disjointes. L'efficacité de la représentation est déterminée au moyen de la qualité de séparation des différentes classes.

Pour illustrer cette approche, on considère un exemple de classification d'images médicales de deux catégories d'images (c.f. **FIGURE 2.3**). Le premier type représente des lésions bénignes qui sont associées à une classe et le second type représente des lésions malignes (tumeurs) qui sont associées à une classe différente. Dans un premier temps, on identifie des mesures quantifiables pour pouvoir représenter les images dans l'espace paramétrique. Par la suite, les frontières de décision sont déterminées pour permettre de classifier les nouvelles images.

2.1.2.2 Approche syntaxique

L'approche syntaxique est basée sur une représentation hiérarchique des formes (c.f. **FIGURE 2.4**). Une forme est ainsi considérée comme une entité qui se compose de sous-

formes simples qui sont elles-mêmes construites à partir de sous-formes plus simples et ainsi de suite. La sous-forme élémentaire est appelée primitive. Une forme se représente en termes d'interrelations entre ses primitives [14].

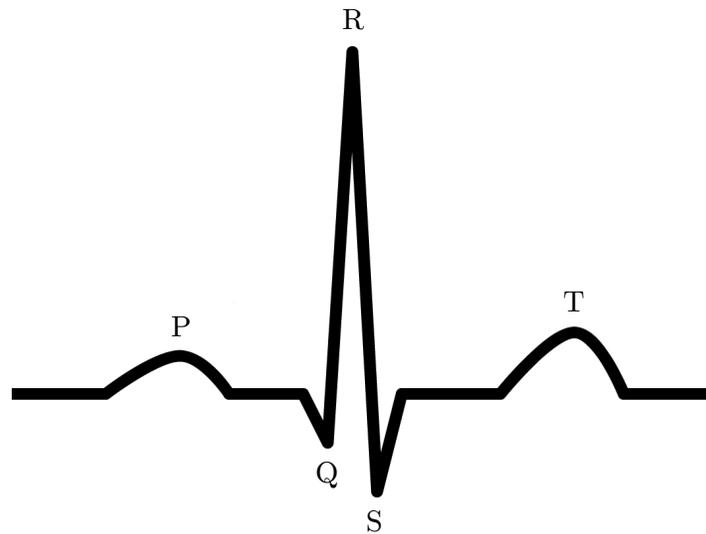


FIGURE 2.4 - Représentation du signal électrocardiographique (normal) sur un cycle.

Une analogie peut être établie entre la structure des formes et la syntaxe d'un langage. Une grammaire permet de fixer les règles de construction et l'analyse des phrases. Pour une phrase donnée, l'analyse syntaxique au moyen de la grammaire permet de déterminer si la phrase est grammaticalement correcte. Ces notions peuvent être généralisées à la reconnaissance syntaxique des formes, où les formes sont considérées comme des phrases composées de plusieurs primitives qui sont les mots. Les règles de combinaison des primitives entre elles sont fixées par la grammaire. L'ensemble des formes d'une même classe correspond à un même langage qui est associée à une grammaire. La reconnaissance syntaxique des formes consiste à identifier les primitives et à effectuer une analyse syntaxique des phrases (formes) pour vérifier l'exactitude vis-à-vis de la grammaire correspondante [14].

Pour illustrer cette approche, on considère la reconnaissance syntaxique d'un signal

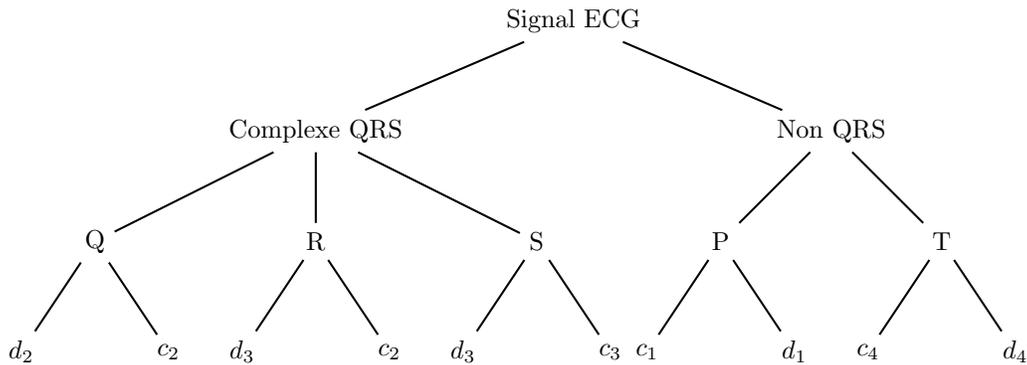


FIGURE 2.5 - Représentation hiérarchique du signal ECG qui se compose de trois niveaux. Les primitives se décomposent au moyen d'une courbe croissante (c) ou décroissante (d).

électrocardiographique (ECG) représentée en **FIGURE 2.4** proposée par M. Kunt [24]. Ce signal quasi-périodique traduit l'activité électromécanique du cœur. Dans son état normal et sur un cycle, il est constitué de cinq ondes appelées P, Q, R, S et T. Chacune de ces ondes correspond à un phénomène précis. De plus, elles doivent apparaître dans cet ordre. L'approche syntaxique consiste à donner une description de cette forme. La phénoménologie du cœur donne naturellement cinq primitives : P, Q, R, S et T. Les ondes Q, R et S forment un complexe, dit complexe QRS, morphologiquement très différent des ondes P et T. Chaque onde peut être décrite par une courbe croissante (c) suivit d'une courbe décroissante (d) et vice versa. Le signal ECG sur un cycle peut alors être décomposé en une structure hiérarchique (c.f. **FIGURE 2.5**). Dans cet exemple, la phrase signal ECG est reconnue comme correcte par l'ordre des indices des primitives et du respect des alternances. Si la séquence obtenue après l'analyse syntaxique n'est pas la séquence $c_1, d_1, d_2, c_2, d_3, c_3, c_4, d_4$ alors la forme observée possède une anomalie.

2.1.2.3 Approche par filtrage de formes

La première approche employée historiquement pour la reconnaissance est basée sur le filtrage de formes [21]. Son fonctionnement repose sur l'analyse de la similarité entre

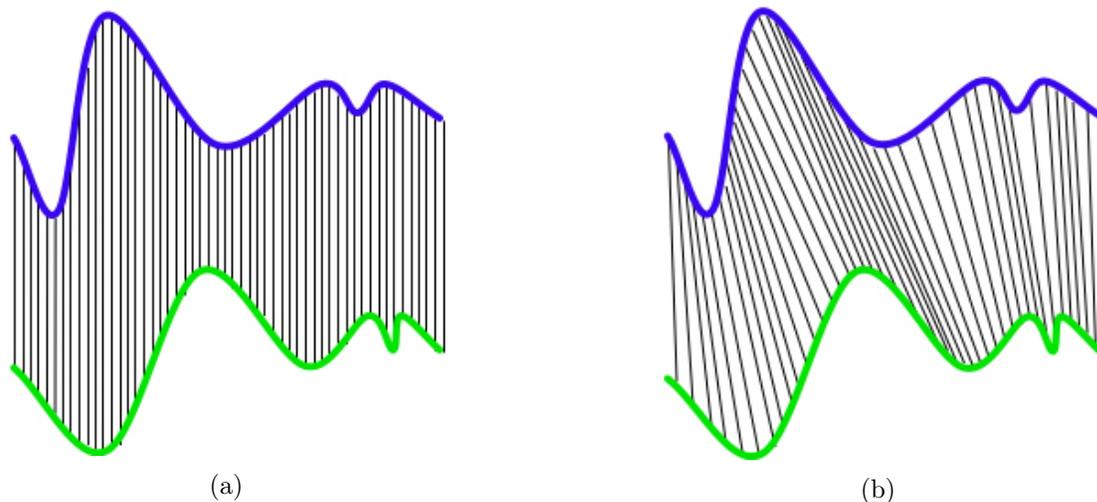


FIGURE 2.6 - Association (a) linéaire et (b) non-linéaire d'échantillons entre une forme de référence (prototype) et une forme à comparer (candidate).

deux formes de même nature.

On considère un ensemble de formes à reconnaître et une forme particulière utilisée comme modèle de base qui sert de support à l'analyse de similarité (correspondance). Cette forme de référence peut être considérée comme un modèle (prototype). Plusieurs transformations (par exemple translation, rotation et changement d'échelle) peuvent être réalisées sur les formes à reconnaître pour procéder à l'opération de similarité. La correspondance entre le modèle de base et la forme à analyser est quantifiée au moyen d'une mesure de similarité comme par exemple la distance euclidienne qui est couramment utilisée.

2.1.3 Evaluation des approches

Le filtrage de formes implique l'utilisation d'une forme de référence [35]. Pour simplifier le traitement, chaque échantillon de la forme de référence est associée de façon linéaire à l'échantillon de la forme à comparer (c.f. **FIGURE 2.6** (a)). Cependant, dans le cas de formes qui appartiennent à une même classe et qui possèdent d'importantes variations structurelles les unes par rapport aux autres (distorsion temporelle), cette

approche ne s'avère pas adéquate et il est préférable d'envisager une association non-linéaire (c.f. **FIGURE 2.6** (b)). Cependant, cette flexibilité alourdit considérablement le traitement ce qui peut présenter un sérieux handicap pour notre problématique de faible consommation.

Dans l'approche statistique, la caractérisation des formes au moyen d'un vecteur ne permet pas de préserver l'information contenue dans leur structure. Dans certains cas, cette information peut être pertinente pour permettre un traitement efficace. La complexité de la mise en œuvre de l'approche statistique de reconnaissance de formes dépend de la dimension de l'espace paramétrique.

Dans le cas de l'approche syntaxique, la représentation hiérarchique permet de reconnaître des formes dont la structure est complexe. Pour effectuer l'analyse grammaticale, les primitives sont extraites des formes. Cependant, une segmentation imprécise peut réduire les performances de reconnaissance.

On peut déduire que les approches statistiques et syntaxiques sont complémentaires. L'approche syntaxique conserve la structure des formes mais est limitée par le bruit du fait de la segmentation. Les outils issus de l'approche statistique doivent être utilisés pour permettre une segmentation appropriée des primitives [23].

L'architecture de reconnaissance des formes adaptée à la problématique de nos travaux doit se baser sur une architecture syntaxique qui doit incorporer un traitement statistique [23]. C'est donc une architecture hybride.

Conclusion

La compréhension des différentes stratégies de reconnaissance de formes a permis de mettre en évidence la pertinence de chacune. L'approche de reconnaissance la plus adaptée dans notre cas s'avère être de nature hybride composée d'un système syntaxique qui incorpore un traitement statistique.

Le traitement de reconnaissance sera effectué, par la suite, sur des formes non uni-

formément échantillonnées. Pour assurer l'efficacité du traitement, il est primordial de déterminer un ensemble de niveaux de quantification qui puissent proposer une représentation adaptée des formes qui préserve l'information structurelle qui fera l'objet du chapitre suivant.

Méthode de discrétisation

Introduction

L'échantillonnage par traversée de niveaux nécessite de définir un ensemble de niveaux de discrétisation (partition) pour son fonctionnement. Les niveaux doivent être adaptés aux formes à échantillonner pour ainsi préserver les éléments structurels les plus importants et assurer par la suite une reconnaissance de formes efficace.

La méthode proposée pour la détermination de la partition recherchée s'inscrit dans le cadre de la reconnaissance de formes. En effet, elle se base sur une architecture de reconnaissance syntaxique modifiée qui intègre une architecture de reconnaissance statistique.

2.2.1 Objectifs

La stratégie employée pour la détermination d'une partition repose sur l'attribution d'un niveau particulier à chaque primitive. Dans le cadre de nos travaux, on définit une primitive par les alternances de nature opposée autour d'une valeur de référence. Dans le cas d'une référence nulle, ces alternances sont positives et négatives. Une alternance est constituée de tous les échantillons situés entre deux valeurs consécutives d'amplitude nulle. Avec cette transformation, une forme se représente sous une séquence composée de valeurs binaires (0 ou 1) dont les valeurs s'alternent [35].

On suppose que cette approche permet de préserver la signature des formes avec l'utilisation d'un minimum de niveaux. Le bien fondé de cette hypothèse est illustré sur la **FIGURE 2.7**. La **FIGURE 2.7** (a) représente une forme continue qui possède des alternances positives et négatives. En supposant, par exemple, qu'il y ait quatre

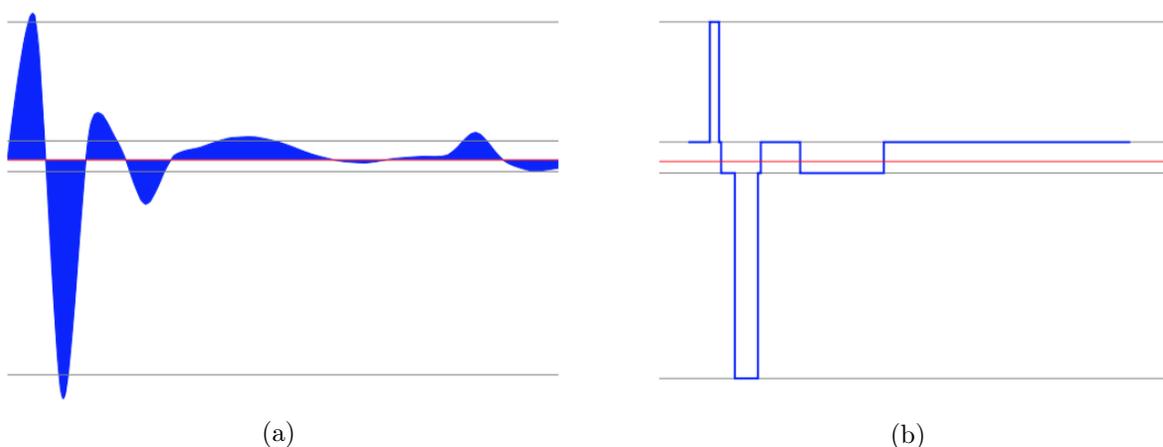


FIGURE 2.7 - (a) Forme continue caractérisée par des alternances positives et négatives. (b) Forme numérisée au moyen de l'échantillonnage par traversée de niveaux.

alternances (primitives) qui définissent la signature de ladite forme, on peut remarquer que sa version numérisée par traversée de niveaux en **FIGURE 2.7** (b) a sa signature sous-jacente qui est préservée.

Pour sélectionner ces niveaux, on définit dans un premier temps une partition initiale. Chacun de ses éléments est un seuil potentiel qui peut constituer la partition recherchée. Ainsi, la méthode proposée est une approche dite TOP-DOWN. De plus, la dimension de la partition finale est directement déterminée par le nombre de primitives sélectionnées.

Le positionnement d'un niveau de discrétisation va dépendre de son indice de performance. L'indice retenu est le couple constitué du taux de bonnes détections (taux de vrais positifs) et de mauvaises détections (taux de faux positifs) des primitives.

Le fonctionnement de la méthode proposée pour la détermination de la partition dépend des paramètres suivants : du nombre de primitives qui constitue une forme, des performances de détection (taux de vrais et faux positifs) ainsi que du nombre de seuils de la partition initiale.

2.2.2 Architecture

La méthode de détermination de partition repose sur l'architecture de la reconnaissance syntaxique de formes (c.f. **FIGURE 2.8**) [14]. Elle se compose d'un module de sélection de primitive, d'un module de prétraitement, d'un module de segmentation et d'un module de reconnaissance de primitive. Il faut noter que le module de reconnaissance de primitives est basé sur l'architecture d'un système de reconnaissance statistique de formes.

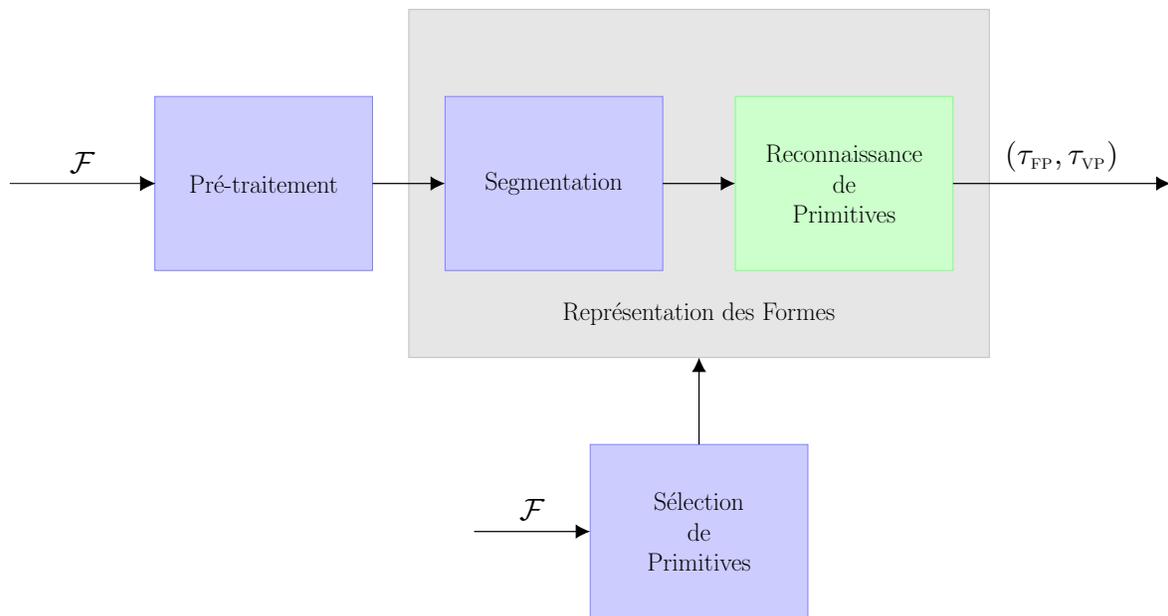


FIGURE 2.8 - Architecture pour la détermination de partitions basée sur une structure de reconnaissance syntaxique de formes.

2.2.3 Méthode de sélection et d'extraction des primitives

Cette méthode a pour objectif d'extraire les primitives prépondérantes qui ont un fort impact dans la structuralité de cette forme. On considère une forme à l'entrée de l'architecture **FIGURE 2.8**. Il faut dans un premier temps échantillonner par traversée de niveaux cette forme avec une partition initiale. Ceci est effectué avec le module de

Prétraitement. Dans l'exemple de la **FIGURE 2.9** (a), cette partition est constituée de niveaux uniformément espacés. Le signe de cette forme est déterminée pour lui donner une représentation binaire et permettre de segmenter ses primitives [3].

Afin de sélectionner les primitives les plus prépondérantes dans la définition de la structure de la forme, il faut attribuer un poids à chacune d'entre elles. Une fonction de pondération simple est la valeur absolue de l'extremum des primitives. De cette façon, l'importance des primitives est indépendante de leur signe. Le résultat de la pondération w est illustré sur la **FIGURE 2.10** (b).

Étant donné que la structure d'une forme ne dépend seulement que de quelques primitives, une fenêtre glissante va être utilisée. Si l'on souhaite seulement récupérer les deux primitives les plus prépondérantes, on définit la fenêtre de largeur deux. Initialement, la fenêtre est positionnée au poids de la première primitive et s'étend jusqu'au poids de la seconde primitive. Un second poids est alors déterminé en faisant la somme du poids des primitives contenues dans la fenêtre. Cette fenêtre glissante est décalée d'une position dans w et ainsi cette opération est renouvelée pour obtenir la représentation w_{Δ} de la **FIGURE 2.11** (b).

Finalement, les primitives prépondérantes sont celles qui sont définies dans la fenêtre glissante dans w et qui débute à la position i définie par l'équation suivante :

$$i = \underset{j}{\operatorname{argmax}} (w_{\Delta})$$

Les deux primitives les plus importantes dans la structure de la forme sont la première p_i et la deuxième p_{i+1} , représentée sur la **FIGURE 2.12**.

2.2.4 Reconnaissance de primitives

La reconnaissance de primitive repose sur l'architecture d'un système de reconnaissance statistique de formes. Les primitives constituent ainsi l'ensemble des formes à

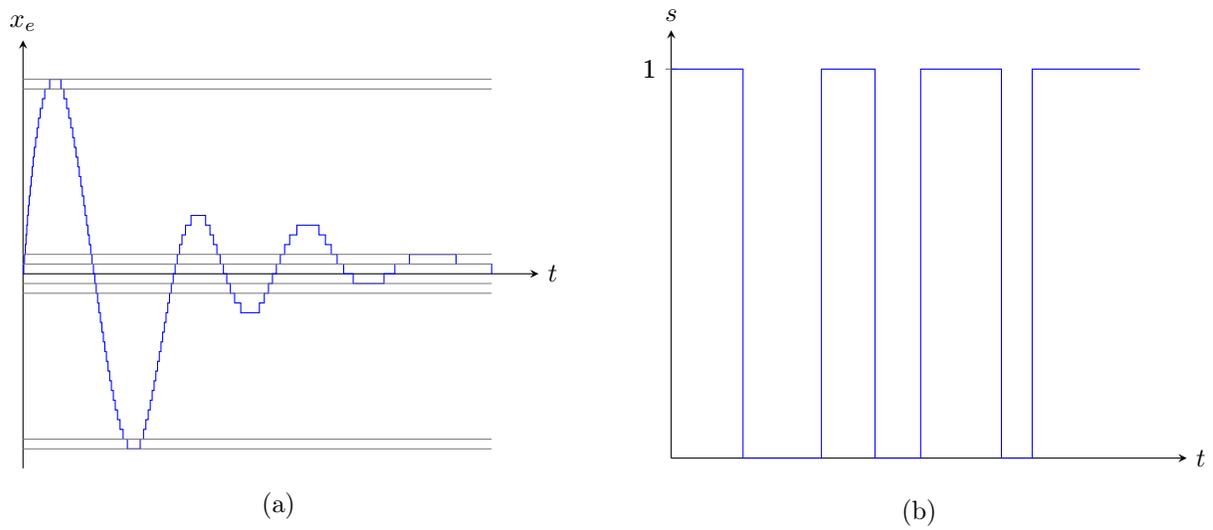


FIGURE 2.9 - (a) Forme échantillonnée par traversée de niveaux au moyen d'une partition initiale uniforme. (b) Représentation binaire obtenue au moyen de la fonction signe.

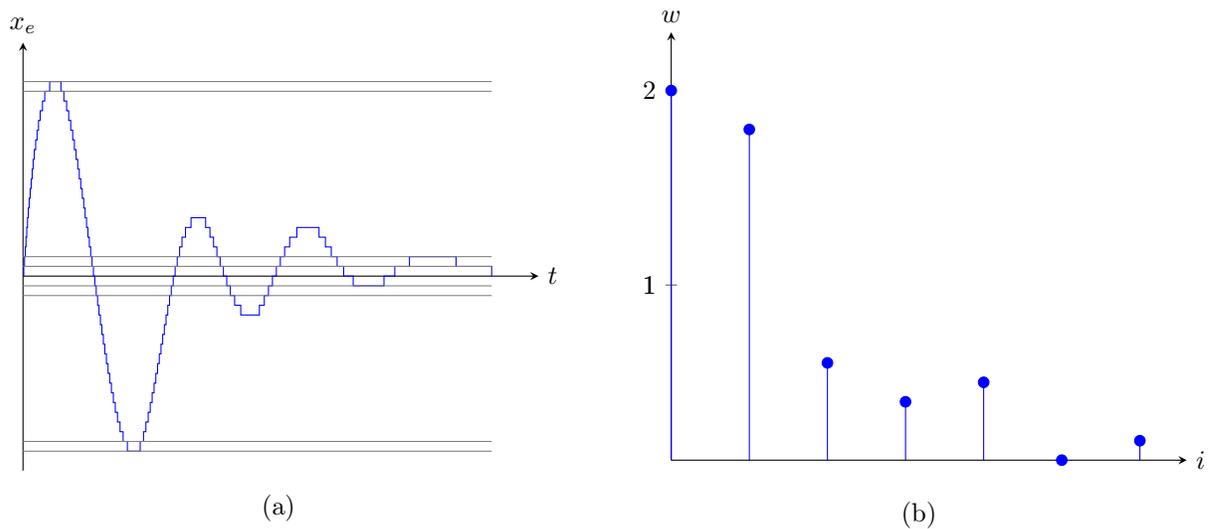


FIGURE 2.10 - (a) Forme échantillonnée par traversée de niveaux au moyen d'une partition initiale uniforme. (b) Poids des primitives.

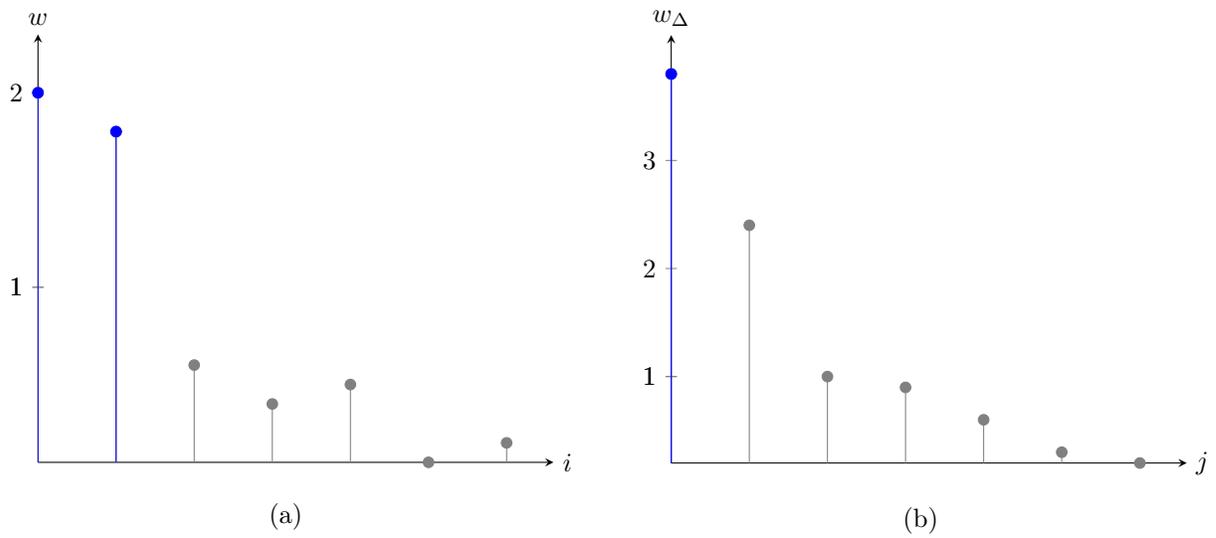


FIGURE 2.11 - (a) Poids des deux premières primitives contiguës. (b) Poids cumulé des deux premières primitives au moyen d'une fenêtre glissante Δ .

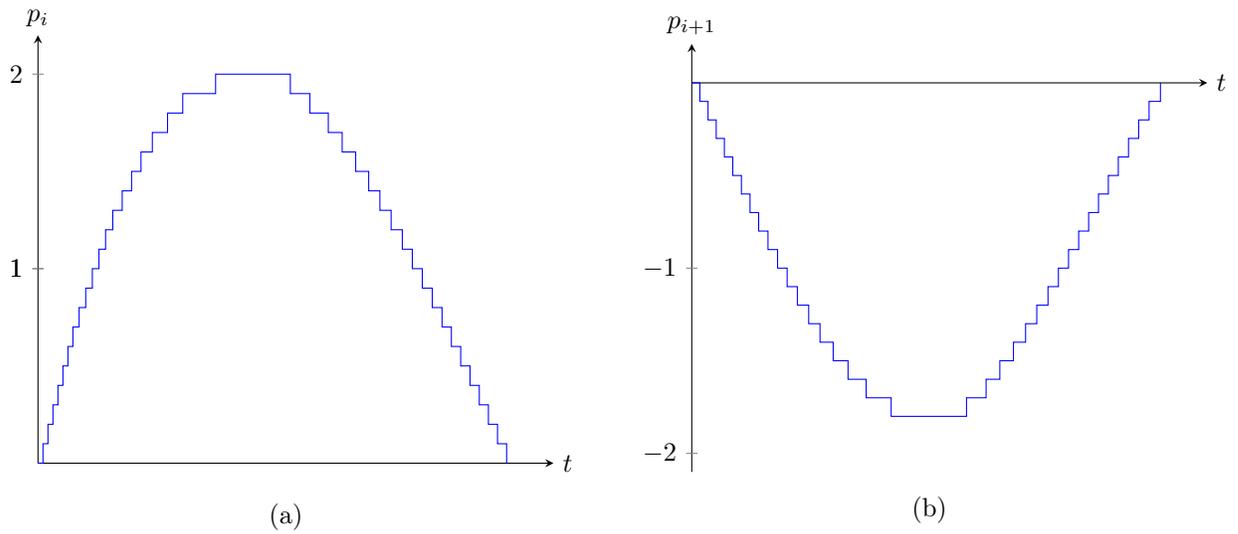


FIGURE 2.12 - (a) Première primitive extraite. (b) Seconde primitive extraite.

détecter.

2.2.4.1 Architecture

L'architecture du module de reconnaissance de primitives est un sous-ensemble de l'architecture de reconnaissance syntaxique de formes (c.f. **FIGURE 2.13**) [21].

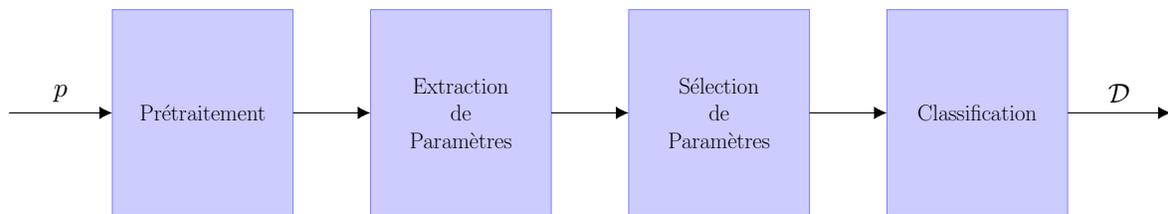


FIGURE 2.13 - Module dédié à la reconnaissance de primitives de l'architecture **FIGURE 2.8** qui est basé sur une architecture générique de reconnaissance de formes statistique.

Cette architecture générique comporte un module de prétraitement, un module d'extraction et de sélection de paramètre et un module de classification. Pour des raisons de simplicité, seuls les modules Extraction de Paramètres et Classification seront utilisés par la suite. Chaque module qui constitue cette architecture est détaillée ci-après.

2.2.4.2 Prétraitement

L'un des objectifs de ces travaux est d'effectuer l'inspection de différentes architectures pour la reconnaissance de formes, à commencer par la plus simple. La reconnaissance statistique de formes la plus simple consiste à manipuler un unique paramètre. De plus, ce paramètre doit pouvoir s'extraire directement à partir des formes sans traitement préalable. Par conséquent, le module de prétraitement ne sera pas utilisé dans l'architecture proposée.

2.2.4.3 Extraction et sélection des paramètres

Le module d'extraction et de sélection de paramètre a pour rôle de déterminer un sous-ensemble parmi un ensemble de paramètres. Cette sélection dépend de plusieurs critères, à savoir , qu'un paramètre doit avoir des valeurs significativement différentes pour des formes appartenant à des classes différentes et avoir des valeurs très similaires pour des formes de même classe. Dans un ensemble de paramètres choisis, un paramètre doit être indépendant d'un autre [21] [24]. De plus, le coût et la complexité en temps de calcul du système de traitement croissent en fonction du nombre de paramètres utilisés. Il faut donc choisir les meilleurs paramètres pour diminuer ces critères tout en satisfaisant les performances de détection.

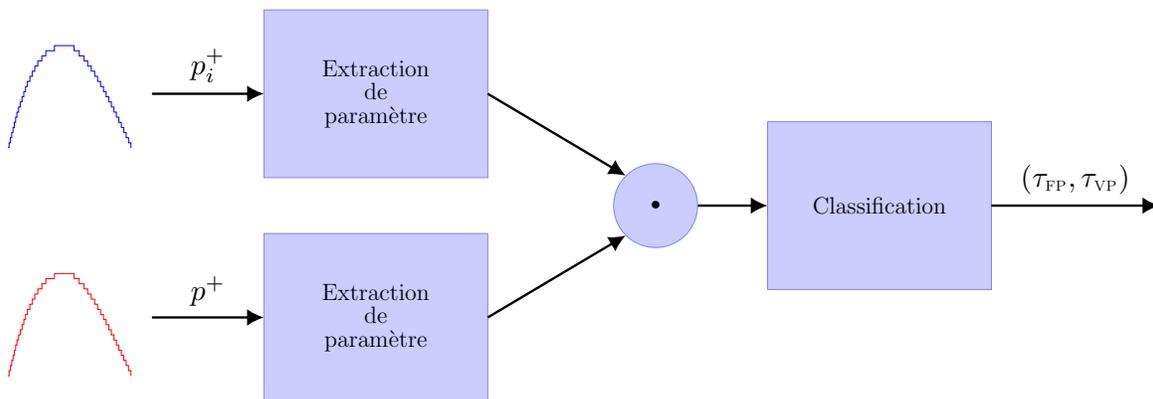


FIGURE 2.14 - Module d'extraction et de sélection de paramètres.

Un des objectifs de nos travaux étant de proposer l'architecture de reconnaissance de formes la plus simple possible, seulement un unique paramètre sera évalué. De plus, pour s'affranchir de traitements supplémentaires comme par exemple du module de prétraitement, le paramètre sera extrait directement de la représentation d'origine des primitives (amplitude). Ce paramètre correspond à la valeur extrême des primitives (positive ou négative).

La FIGURE 2.14 illustre le procédé de construction du vecteur de paramètre pour

une primitive. Un premier module d'extraction de paramètre est dédié à la détermination de la valeur extrême de la primitive pour chaque forme d'intérêt. Si le signe de la primitive est positif alors le deuxième module d'extraction de paramètre s'effectue sur les primitives positives du reste du signal dans lequel les formes d'intérêt ont été extraites. Le vecteur résultant fourni est la concaténation de ces deux vecteurs de paramètres. Ce procédé de construction du vecteur de paramètres est réitéré pour chacune des primitives sélectionnées dans les formes d'intérêt.

2.2.5 Classification

2.2.5.1 Théorie de la décision

Pour illustrer la théorie de la décision, on suppose un vecteur de paramètres x auquel est associé une variable \mathcal{C} . La théorie de la décision est basée sur la prédiction de \mathcal{C} en fonction de x . Dans le contexte de la classification, une variable représente un paramètre extrait d'une forme et l'autre représente la classe dans laquelle on peut associer la forme. La loi jointe $p(x, \mathcal{C})$, appelée également loi de probabilité à plusieurs variables, fournit toutes les informations concernant l'incertitude associée à ces variables.

$$p(\mathcal{C}_k|x) = \frac{p(x|\mathcal{C}_k) \cdot p(\mathcal{C}_k)}{p(x)} \quad (2.1)$$

On reprend le problème de diagnostic médical dans lequel nous avons l'image à rayons X d'un patient et que l'on souhaite déterminer si une tumeur est cancéreuse ou non (c.f. **FIGURE 2.3**). Le vecteur paramétrique peut être construit au moyen de mesures obtenues sur l'intensité des pixels de l'image. Le problème associé à cet exemple implique la détermination de la loi jointe $p(x, \mathcal{C})$ qui donne une description complète de la situation. Cette quantité va permettre de décider de donner un traitement ou non, cela implique d'effectuer un choix optimal. Quand l'image à rayon X d'un patient est obtenue, l'objectif est de décider laquelle des deux classes assigner à l'image. On

s'intéresse ainsi à la probabilité d'appartenance des deux classes de cette image. Cette quantité peut être obtenue par le théorème de Bayes qui s'exprime avec l'équation (2.1).

Notre souhait est de minimiser la probabilité d'assigner une forme à la mauvaise classe. En d'autres termes, il faut minimiser le taux de mauvaises classifications.

2.2.5.2 Minimisation du taux de mauvaises classifications

Pour minimiser le taux de mauvaises classifications il est nécessaire de définir un critère (règle) qui assigne chaque forme à une des classes possibles. Afin de trouver une règle de décision optimale, pour raison de clarté, on considère par exemple deux classes, comme pour le problème de la détection de présence de cancer. Une erreur de décision apparaît quand une forme qui appartient à la classe 1 est associée à la classe 2 et vice versa. Cette probabilité d'occurrence est définie par l'équation (2.2).

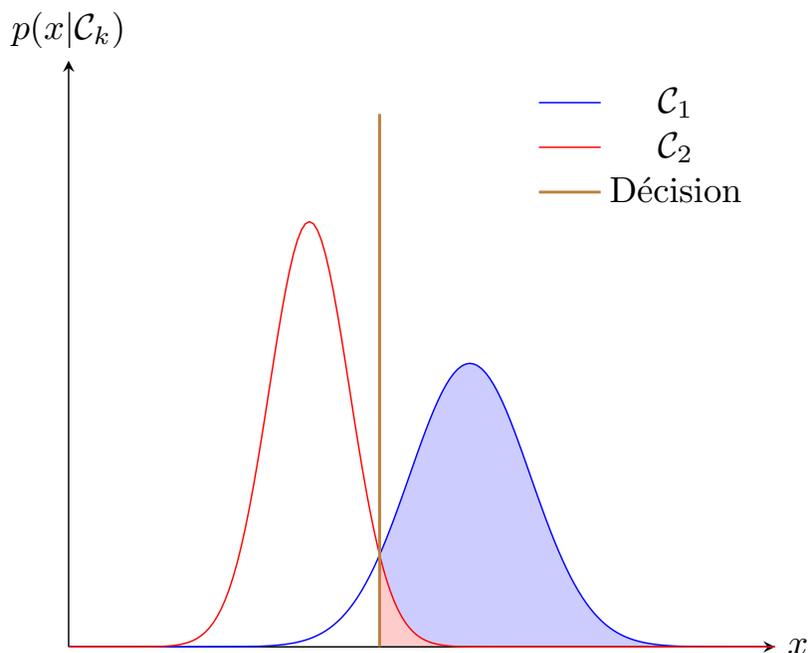


FIGURE 2.15 - Représentation des probabilités jointes $p(x, \mathcal{C}_k)$ et de la frontière de décision \mathcal{D} .

Nous sommes libres de choisir la règle de décision qui assigne chaque forme à l'une

des deux classes. Cependant, pour minimiser la probabilité d'erreur, on doit faire en sorte que chaque forme soit assignée à la classe qui présente la plus petite valeur de l'intégrande dans l'équation (c.f. l'équation (2.2)).

$$p_e = \int_{\mathcal{R}_1} p(x, \mathcal{C}_2) \cdot dx + \int_{\mathcal{R}_2} p(x, \mathcal{C}_1) \cdot dx \quad (2.2)$$

La théorie de la décision est illustrée avec la figure **FIGURE 2.15**. Elle représente la modélisation de distributions jointes qui sont modélisées par deux distributions de loi Gaussienne de caractéristiques différentes. Un choix optimal de décision peut être défini pour la valeur d'abscisse pour laquelle les courbes se croisent. Ceci équivaut au taux minimum de mauvaise classification.

Un seul paramètre est extrait des primitives (signe positif et négatif) : c'est la valeur extrême de leur amplitude. Le vecteur de paramètre des primitives est de dimension 1, de même que pour l'espace paramétrique.

Pour effectuer l'opération de classification il est nécessaire de définir différentes régions dont chacune d'entre elle est associée à une des classes possibles. Dans nos travaux, la problématique de reconnaissance de formes est restreinte à deux classes. Par conséquent, un unique seuil (de décision) doit être défini pour séparer l'espace paramétrique. Dans l'exemple de la **FIGURE 2.15**, la classification s'effectue de la manière suivante : les échantillons, respectivement au dessus et en dessous, du seuil de décision sont classés dans \mathcal{C}_1 et \mathcal{C}_2 .

Comme il a été décrit précédemment, on souhaite placer le seuil de décision de manière à minimiser le taux de mauvaises classifications (c.f. équation (2.2)). Cependant, pour nos travaux, il n'est pas adéquat de caractériser les performances d'un classificateur au moyen d'une seule valeur (probabilité d'erreur). En effet, nous voulons ajouter plus de flexibilité à l'utilisateur pour définir les performances du système de reconnaissance de formes. Ainsi il peut être plus approprié de caractériser ses performances par l'intermédiaire du couple composé du taux de Vrais Positifs (VP) et du taux de Faux

Positifs (FP) décrit par l'équation (2.3).

$$\tau_{VP} = \frac{VP}{V} \quad \text{et} \quad \tau_{FP} = \frac{FP}{F} \quad (2.3)$$

Une classification binaire induit quatre types de décisions possibles. Si une forme appartient à la classe 1 ou 2 et quelle est associée à la vraie classe, alors elle sera comptabilisée, respectivement, en tant que vrai positif et vrai négatif. Dans le cas où il y a une erreur dans la classe prédite, alors la décision est comptabilisée en tant que faux négatif et faux positif. Cette information est affichée dans une matrice dite de confusion (c.f. **FIGURE 2.16**).

Classe vraie	Classe prédite	
	\mathcal{C}_1	\mathcal{C}_2
\mathcal{C}_1	VP	FN
\mathcal{C}_2	FP	VN

FIGURE 2.16 - Matrice de confusion

Ceci peut être représenté au moyen de la courbe ROC (Receiver Operating Characteristic) qui est un graphe qui se construit avec le taux de vrais positifs sur l'axe des ordonnées et le taux de faux positifs sur l'axe des abscisses [6] [13]. La courbe ROC se construit en faisant varier le seuil de décision où le taux de vrais positifs et le taux de faux positifs est déterminé (c.f. **FIGURE 2.17**).

Dans l'espace ROC, le seuil de décision qui effectue le compromis entre le taux de vrais positifs et de faux positifs est le point le plus éloigné de la diagonale définie par $y = x$. Un seuil de décision qui engendre un point situé sur cette diagonale représente une décision aléatoire de classification.

Conclusion

L'architecture de reconnaissance syntaxique permet d'extraire les éléments structuraux importants des formes d'intérêt. La reconnaissance statistique permet de positionner les niveaux afin d'assurer un compromis entre le taux de vrais positifs et le taux de faux positifs de détection des primitives. Ainsi, cette architecture hybride permet de conserver à la fois la dynamique (signature) des formes d'intérêt en prenant en compte des indices de performances.

Pour effectuer de la reconnaissance de formes, il faut à présent définir une méthode qui puisse représenter les formes d'une même classe en prenant en compte l'information structurelle. La stratégie envisagée s'appuie sur la théorie des automates et fera l'objet du chapitre suivant.

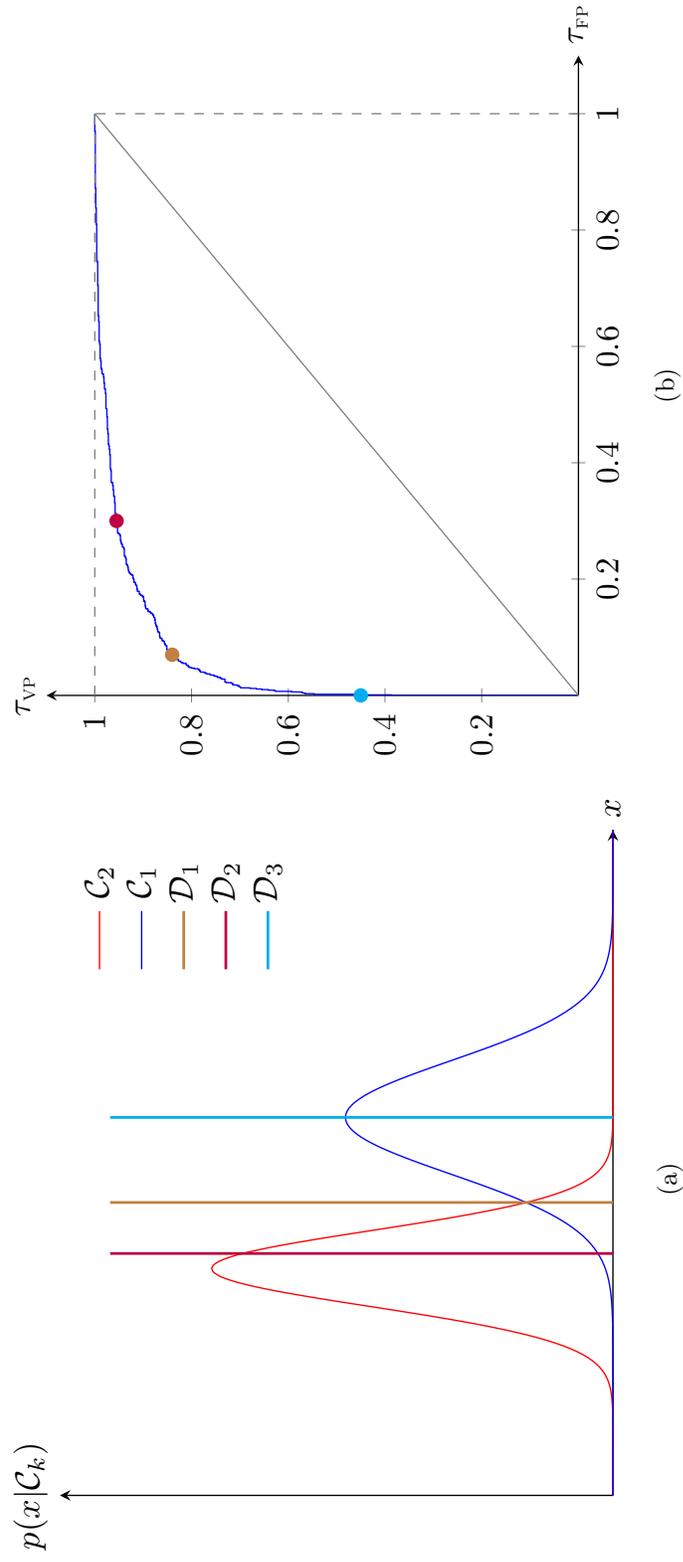


FIGURE 2.17 - (a) Position de plusieurs seuils de décision pour la séparation des distributions. Chaque seuil de décision correspond à un point dans l'espace ROC.

Représentation et reconnaissance de formes

Introduction

La structure des formes comporte de l'information pertinente. De ce fait, il faut la préserver pour obtenir une chaîne de traitement efficace. L'information structurelle d'une forme est présente car il existe un degré de corrélation entre les échantillons. Cette information peut être conservée en donnant une représentation adéquate des formes. Ce qui est effectué par l'intermédiaire de graphe qui prend en compte la relation existante entre les échantillons.

L'ensemble des réponses du capteur qui induit les formes à reconnaître peut être interprété comme un seul et unique système dynamique. Ce système peut être modélisé sous forme d'automate qui se représente sous forme de graphe.

Dans ce chapitre plusieurs types de représentation des formes vont être explorés, à savoir les automates finis déterministes et les automates stochastiques.

2.3.1 Représentation symbolique

Un signal qui est numérisé au moyen de l'échantillonnage par traversée de niveaux peut être représenté sous forme d'une séquence de symboles. En effet, la position des niveaux utilisés est connue de manière exacte. On considère la forme continue représentée sur la **FIGURE 2.18** qui est échantillonnée au moyen de deux niveaux de discrétisation : l_1 et l_2 . Étant donné que l'amplitude de ces niveaux est exacte, on peut respectivement les associer, par exemple, aux lettres a et b . De ce fait, la forme numérisée peut être représentée par la séquence de symboles : $bbaabbaa$ et permet ainsi l'utilisation des outils de la théorie des automates pour effectuer la reconnaissance de formes.

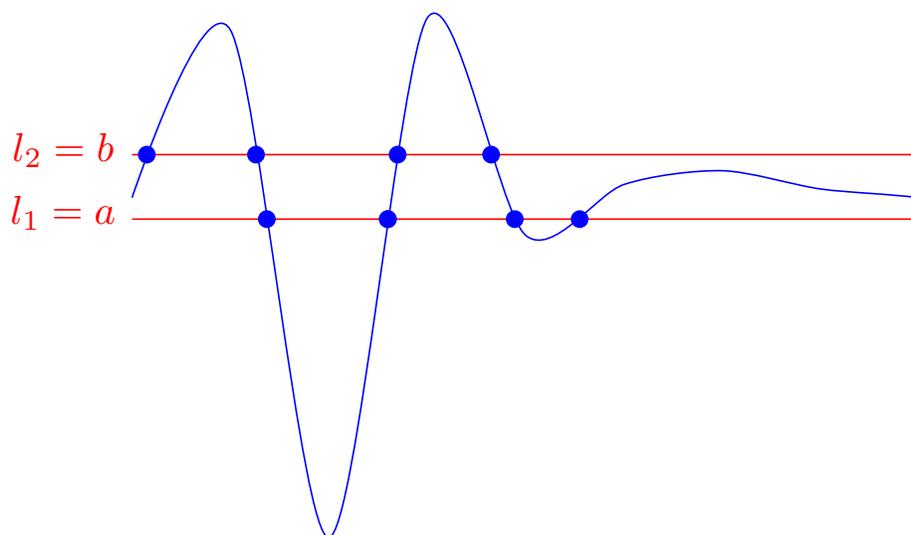


FIGURE 2.18 - Représentation de la forme numérisée en séquence de symboles : *bbaabbaa*.

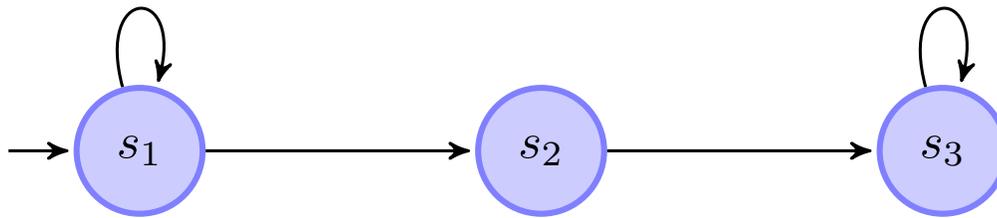
2.3.2 Définitions

La terminologie, empruntée à la linguistique, vient de l'intérêt que les linguistes ont portés, et portent encore, à la théorie des automates [24].

Un alphabet est un ensemble, non vide, *a priori* fini, qui sera l'ensemble des symboles que l'on notera A . Logiquement, les éléments de A , les symboles, sont appelés lettres, et les suites de lettres sont appelés mots. On peut prendre comme alphabet un ensemble de chiffres : par exemple $\{0, 1\}$ ou $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Les mots sont des entiers qui peuvent être écrits dans différentes bases définies par l'alphabet choisie : 1101 ou 013 sont les deux écritures dans les bases 2 et 10 respectivement du nombre 13 [40].

2.3.3 Éléments de la théorie des graphes

Les automates sont considérés comme des graphes. Un graphe \mathcal{G} est défini par deux ensembles. Le premier est l'ensemble des sommets et le second est l'ensemble des arcs.

FIGURE 2.19 - Graphe \mathcal{G}

Chaque arc possède une origine et une extrémité qui sont deux sommets du graphe. On représente les sommets par des cercles et les arcs par des flèches qui joignent l'origine à l'extrémité. Un graphe \mathcal{G} est illustré en **FIGURE 2.19**.

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

FIGURE 2.20 - Matrice d'incidence du graphe \mathcal{G} .

Un graphe se représente naturellement par une matrice booléenne, appelée matrice d'incidence. Un coefficient de la matrice vaut 1 s'il existe (ou moins) un arc ayant une origine et une extrémité. La matrice d'incidence du graphe \mathcal{G} est présentée en **FIGURE 2.20**.

2.3.4 Automate fini déterministe

Le terme déterministe provient du fait que pour chacune des entrées, l'automate transite vers un seul et unique état à partir de son état courant. Un automate fini déterministe se compose d'un ensemble d'états, de symboles d'entrées, d'une fonction de transition, d'un état de départ et d'un état final (accepteur). Ce type d'automate sera nommé par l'acronyme AFD [19].

Il est nécessaire de comprendre comment un AFD décide ou non d'accepter une séquence de symboles présentée à son entrée. Le langage d'un AFD est l'ensemble de tous

les mots qui sont acceptés par cet automate. L'AFD commence dans son état de départ. La fonction de transition est consultée pour déterminer dans quel état l'automate doit transiter après le traitement du premier symbole de la séquence d'entrée (mot). Le second symbole est traité et de la même manière la table de transition est consultée pour trouver l'état suivant et ainsi de suite. Si l'état dans lequel se trouve l'AFD après le traitement du dernier symbole de sa séquence d'entrée est un état qui appartient à l'ensemble des états finaux alors la séquence est acceptée sinon elle est rejetée. Un exemple est illustré en **FIGURE 2.21**.

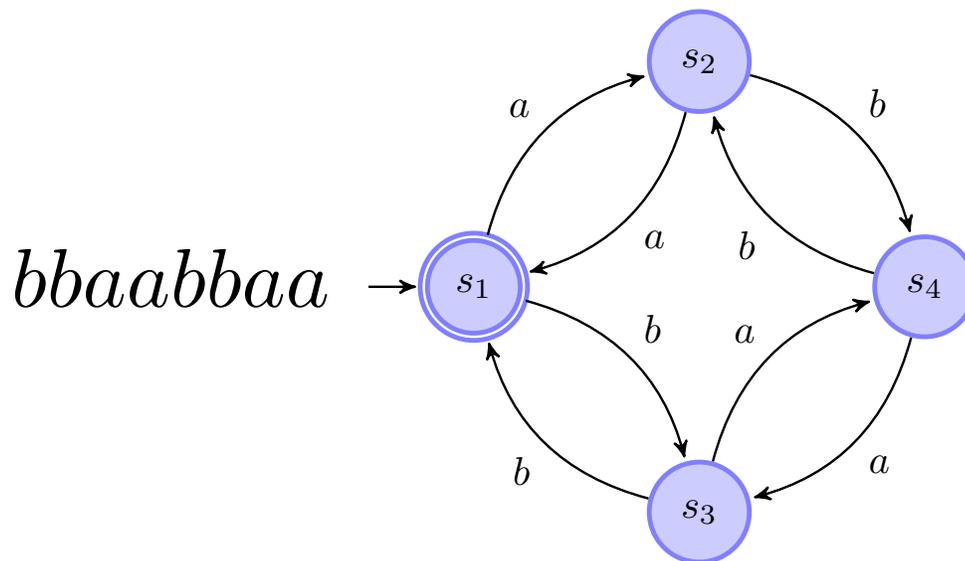


FIGURE 2.21 - Reconnaissance d'une séquence de symboles.

À chaque état est associé un nœud. Pour chaque état et chaque symbole en entrée, il existe un état dans lequel l'AFD transite. Le diagramme de transition possède un arc qui a pour origine un nœud et dont son extrémité est un autre nœud. Il y a une flèche dont l'extrémité est reliée à l'état de départ. Cette flèche n'a aucun nœud d'origine. Les nœuds qui correspondent aux états accepteurs sont marqués par un double cercle. Les états qui n'appartiennent pas à l'ensemble des états accepteurs sont notés par un cercle unique.

2.3.4.1 Inférence grammaticale

On imagine un professeur qui souhaite transmettre les connaissances d'une méthode particulière à des étudiants. Les exemples positifs et négatifs doivent occuper une place centrale dans sa communication. Dans son exercice de communication, les exemples doivent être choisis de manière pertinente et non de façon arbitraire ou aléatoire afin d'assurer la convergence de l'apprentissage de la méthode par les étudiants (hypothèse correcte). La plupart des méthodes d'apprentissage se basent sur des exemples arbitraires ou aléatoires. Cependant, l'observation précédente suggère qu'il serait raisonnable d'examiner des méthodes qui se basent sur une source qui fournisse des exemples informatifs (utiles). Pour accentuer cet aspect, on nomme la source des exemples le TEACHER et l'algorithme d'apprentissage est dénommé le LEARNER (c.f. **FIGURE 2.22**).

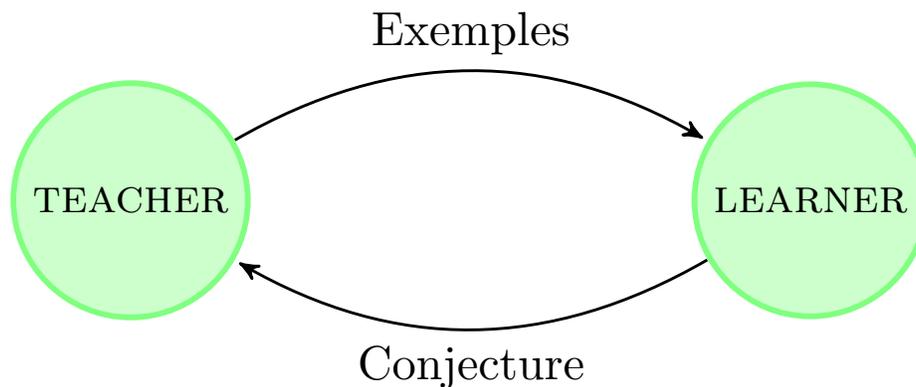


FIGURE 2.22 - Représentation des interactions entre le TEACHER qui est la source d'exemples et le LEARNER qui propose un état de sa compréhension de son apprentissage (conjecture).

Un TEACHER est supposé répondre correctement à deux types de questions posées par le LEARNER concernant un ensemble régulier inconnu. La première question concerne l'appartenance d'un mot t à l'ensemble régulier. La réponse est positive si t appartient à l'ensemble ou négative dans le cas contraire. Le second type de question est posé sous la forme d'une conjecture qui décrit l'ensemble régulier S . La réponse est

positive si elle décrit le langage inconnu de l'ensemble S . Dans le cas contraire, il existe une chaîne t , appelée contre-exemple, pour lequel la conjecture est incorrecte. Dans la méthode proposée, la description de l'ensemble régulier par la conjecture s'effectue sous forme d'un AFD. Elle exploite de manière efficace les informations contenues dans les questions d'appartenance pour trouver efficacement (convergence rapide) un AFD compatible avec les exemples et les réponses aux questions.

On décrit l'algorithme d'apprentissage L^* qui apprend efficacement un ensemble régulier inconnu au moyen d'un TEACHER [2]. L'ensemble régulier est noté U qui est défini sur un alphabet A .

À n'importe quel moment, l'algorithme L^* a accès à des informations sur un ensemble de mots définis sur A en les classant en tant que membre ou non-membre de l'ensemble régulier inconnu U . Cette information est classée dans une table d'observation, représentée en **FIGURE 2.23**, qui se compose de trois éléments : un ensemble fini et non vide de mots *suffix-closed* E , un ensemble fini et non vide de mots *prefix-closed* S et une fonction T (association). À noter qu'un ensemble est *prefix-closed* si et seulement si chaque préfixe de chaque membre de l'ensemble est aussi un membre de l'ensemble. Un ensemble *suffix-closed* est défini de manière analogue.

	E
S	T(S · E)
S · A	T(S · A · E)

FIGURE 2.23 - Table d'observation composée des ensembles *prefix-closed* E , *suffix-closed* S , de l'ensemble issu de la concaténation de S et A , noté $S \cdot A$ et de la fonction T

L'interprétation de T est telle que $T(u)$ est égal à 1 si et seulement si u est un membre de l'ensemble régulier inconnu U . La table d'observation peut se visualiser comme un tableau de deux dimensions où les lignes sont labélisées par les éléments de $(S \cup S \cdot A)$

et les colonnes labélisées par les éléments de E . Les entrées pour la ligne s et la colonne e est égale à $T(s \cdot e)$ avec s qui appartient à $(S \cup S \cdot A)$. L'algorithme L^* utilise la table d'observation pour construire un AFD. Les lignes labélisées par les éléments de S sont des candidats pour les états de accepteurs et les colonnes définissent les différents états possibles. Les lignes labélisées par les éléments de $S \cdot A$ sont utilisées pour construire la fonction de transition. Une table d'observation est fermée si pour chaque t de $S \cdot A$ il existe un s dans S de telle manière que $LIGNE(t) = LIGNE(s)$. Une table d'observation est consistante si quelque soit s_1 et s_2 , des éléments de S , on obtient $LIGNE(s_1) = LIGNE(s_2)$ pour tout a dans A , on a $LIGNE(s_1 \cdot a) = LIGNE(s_2 \cdot a)$. Si la table est fermée et consistante on peut définir un accepteur correspondant $M(S, E, T)$ sur l'alphabet A avec un ensemble d'états Q , un état initial q_0 , des états accepteurs F et une fonction de transition δ .

La table d'observation est initialisée avec $S = E = \{\lambda\}$. Pour déterminer T , l'apprenti L^* pose la question d'appartenance de λ et de chaque a dans A au TEACHER. Cette table d'observation peut être fermée et consistante ou non. La boucle principale de L^* teste la table d'observation pour vérifier si elle est fermée et consistante. Si la table n'est pas consistante, alors L^* trouve s_1 et s_2 dans S , e dans E et a dans A de telle manière que $LIGNE(s_1) = LIGNE(s_2)$ mais que $T(s_1 \cdot a \cdot e)$ n'est pas égal à $T(s_2 \cdot a \cdot e)$. L^* ajoute le mot $a \cdot e$ à E et étend T avec $(S \cup S \cdot A) \cdot (a \cdot e)$ en demandant l'appartenance pour les éléments manquants. Si la table n'est pas fermée, alors L^* trouve s_1 dans S et a dans A de telle manière que $LIGNE(s_1 \cdot a)$ est différent de $LIGNE(s)$ pour tout s dans S . L^* ajoute le mot $s_1 \cdot a$ à S et étend T à $(S \cup S \cdot A) \cdot E$ en demandant l'appartenance pour les éléments manquants. Ces deux opérations sont répétées tant que la table (S, E, T) n'est pas fermée ni consistante. Si la table est fermée et consistante alors L^* établit la conjecture de $M(S, E, T)$. Le TEACHER répond de manière positive si la conjecture est correcte ou répond avec un contre-exemple t dans le cas contraire. Si le TEACHER répond affirmativement, l'algorithme L^* se termine. Par contre, dans le cas contraire, le

TEACHER répond par un contre-exemple t et tout ses préfixes sont ajoutés à l'ensemble S et les questions d'appartenance sont posées aux nouvelles entrées. C. de la Higuera [11] présente un état de l'art sur les méthodes d'inférence grammaticale.

2.3.4.2 Automate stochastique

Une classe particulière d'automates probabilistes déterministes, appelée D-Markov, peut être utilisée pour la représentation des formes [34] [36]. Le cœur de cette approche repose sur l'approximation d'un processus symbolique avec un certain niveau de précision. Un processus symbolique stochastique est appelé processus Markovien d'ordre D (c.f. équation (2.4)), si la probabilité de l'observation courant dépend seulement des D observations précédentes. En d'autres termes, le processus possède une mémoire de longueur D . Un tel processus peut se représenter avec un automate probabiliste déterministe (c.f. **FIGURE 2.24**). Les états de l'automate sont représentés par une chaîne de symboles de longueur D définie sur un alphabet. La machine D-Markov peut se représenter par sa matrice de transition d'état.

$$P[s_k | s_{k-1} s_{k-2} \cdots s_1 s_0] = P[s_k | s_{k-1} s_{k-2} \cdots s_{k-D+1} s_{k-D}] \quad \forall s_i \in A \quad (2.4)$$

La profondeur de la machine D-Markov est un paramètre crucial puisque le nombre d'états varie exponentiellement avec D . Une mémoire faible ne peut pas permettre une représentation correcte de la dynamique du processus symbolique. Par contre, une grande mémoire induirait un nombre d'états important qui mènerait à de très faibles probabilités d'états et à une matrice de transition imprécise.

Pour pouvoir effectuer de la reconnaissance de formes, il est nécessaire de quantifier les similarités ou les divergences dans les formes au moyen d'une mesure. Cette mesure est appelée mesure d'anomalie (c.f. **FIGURE 2.25**) et quantifie la déviation des formes

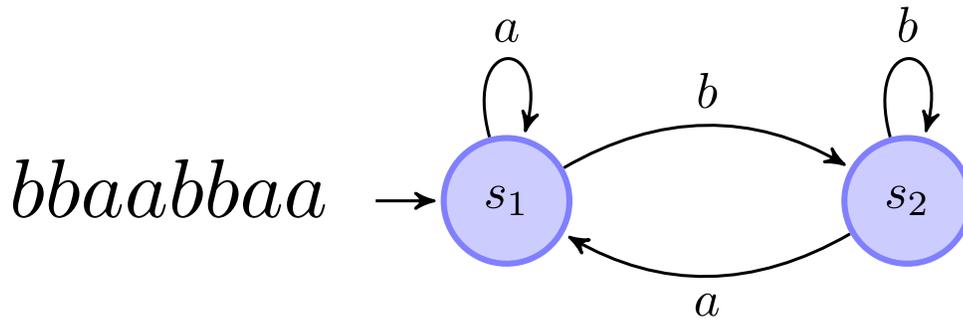


FIGURE 2.24 - Automate stochastique.

par rapport à une forme de référence. Par la suite un seuil de décision est défini pour associer les formes aux différentes classes possibles.

$$\Pi^0 = \begin{pmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{pmatrix} \quad \Pi^k = \begin{pmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{pmatrix}$$

$$\mathbf{M}^k = \left\| \Pi^0 - \Pi^k \right\|$$

FIGURE 2.25 - Matrice de transition d'états de référence Π^0 , celle après analyse d'une séquence de symboles d'une forme Π^k et la mesure de divergence \mathbf{M}^k .

Conclusion

L'utilisation d'un automate déterministe à états finis dans le module de détection offre deux avantages majeurs. Le premier réside dans la simplicité dans la reconnaissance d'une forme particulière. Le seul fait de transiter vers un des états finaux (possibles) de l'automate indique que la forme évaluée respecte la grammaire des formes d'intérêts et ne nécessite pas de post-traitements contrairement aux automates stochastiques. Le deuxième provient du fait que les outils d'implémentation matérielle, notamment sur circuit logique programmable, sont parfaitement adaptés et optimisés pour la réalisation

d'un automate.

Le sous-chapitre suivant fera l'objet des résultats de simulation de la reconnaissance de formes non-uniformément échantillonnées avec une partition.

Résultats de simulation

Introduction

Ce chapitre décrit les résultats de simulation obtenus avec le logiciel MATLAB. Les données manipulées, décrites ci-après, proviennent d'un protocole expérimental.

Dans un premier temps, les niveaux qui constituent la partition pour l'échantillonnage par traversée de niveaux ont été déterminés. Ces niveaux ont été sélectionnés de manière à obtenir le compromis entre le taux de vrais positifs et de faux positifs. Une fois les partitions déterminées une comparaison est effectuée entre le nombre d'échantillons capturés pour le cas uniforme et non-uniforme.

Ensuite, les formes, après échantillonnage non-uniforme, sont utilisées pour donner une représentation unique sous forme d'automate : déterministe et stochastique. Les performances de reconnaissance de formes obtenues avec ces deux types de représentation sont comparées.

2.4.1 Données expérimentales

Les données proviennent d'un capteur accélérométrique composé de 3 axes (c.f. **FIGURE 2.26**). Ce capteur a été intégré dans un dispositif qui échantillonne uniformément son signal. Ce signal de nature continu a été échantillonné à la fréquence la plus élevée que l'on peut obtenir avec ce dispositif et qui est de 100 Hz. Cette fréquence d'échantillonnage maximum est supérieure à la fréquence de Nyquist requise pour échantillonner correctement des signaux physiologiques.

Le dispositif de mesure, porté par un patient, subit plusieurs sollicitations qui sont également subies par le capteur. En raison de la confidentialité du contexte expéri-

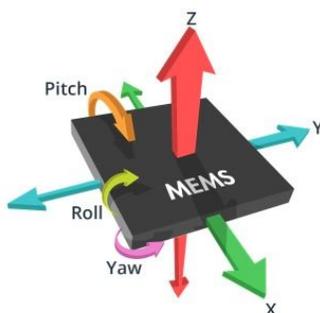


FIGURE 2.26 - Capteur accélérométrique MEMS triaxial.

mental, les sollicitations peuvent être considérées comme des exercices de toux pour illustrer le protocole. Un exercice est constitué d'une séquence de toux consécutives. Ces sollicitations sont effectuées à intervalle régulier durant la totalité de l'expérience. Une toux, durant un exercice, est suffisamment espacée d'une autre pour considérer que chaque forme induite par ces sollicitations soit indépendante les unes des autres. Il n'existe ainsi aucun recouvrement entre deux toux consécutives. De la même manière, les exercices sont suffisamment espacés les uns des autres.

Les patients notent à quel moment au cours de la journée ils effectuent un exercice de toux. De ce fait, chaque exercice, et par conséquent, chaque forme peut être localisée précisément dans le signal temporel du capteur. Cette caractéristique permet d'effectuer de la reconnaissance de formes supervisée.

2.4.2 Partition

Le choix du nombre de primitive permet de représenter une forme avec un certain degré d'approximation. Une forme se décompose ainsi en un nombre fini de primitives. Un seuil est associé à chacune de ces primitives. On considère une forme détectée si l'extremum de chacune de ses primitives est supérieure au seuil qui sert à leur détection. En considérant une classe de forme à reconnaître, la transformation binaire des formes est une approximation qui fait apparaître deux sous-classes. En effet, cette représenta-

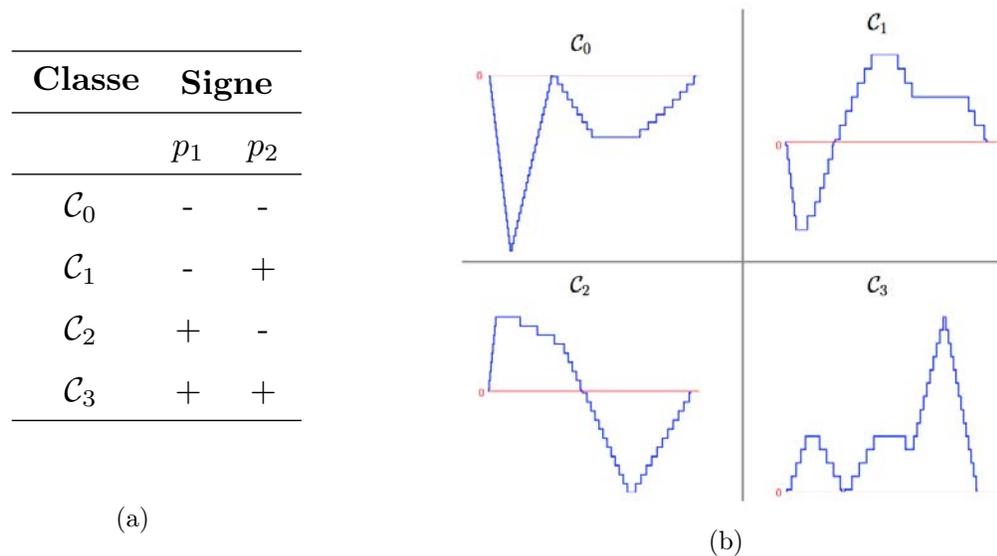


FIGURE 2.27 - (a) Les classes de formes pour deux primitives (b) Formes non-uniformément échantillonnées modélisées par deux primitives.

tion permet de modéliser n'importe quel signal par une séquence binaire alternant les symboles 0 et 1 et qui correspondent aux classes \mathcal{C}_1 et \mathcal{C}_2 de la **FIGURE 2.27**.

Une forme commence ainsi par 0 (signe négatif) ou par 1 (signe positif) le reste de leur séquence est entièrement définie par une alternance de 0 et 1 (c.f. **FIGURE 2.27** (a)). Cette représentation ne va pas convenir à la totalité des formes qui constituent une classe. Il faut ainsi rejeter les formes qui ne possèdent pas les séquences précédemment définies.

La **FIGURE 2.28** représente le pourcentage de rejet des formes pour un nombre de primitives égal à deux. Les faibles valeurs prouvent que les classes \mathcal{C}_1 et \mathcal{C}_2 permettent une bonne modélisation des formes d'intérêt. Pour un nombre de primitive unitaire, aucune forme n'est rejetée car il y a seulement les classes \mathcal{C}_1 et \mathcal{C}_2 , donc aucune n'est rejetée.

Outre le fait de correctement modéliser la classe des formes à reconnaître, la qualité d'une partition est évaluée par le taux de compression de donnée qu'elle permet

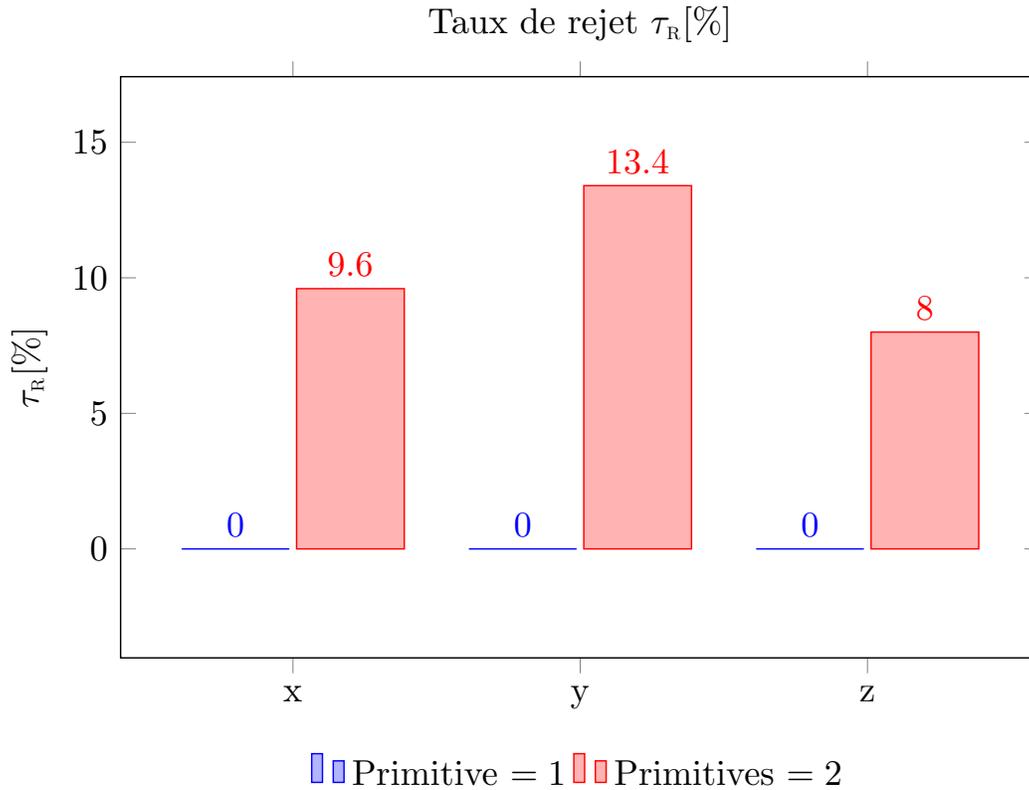


FIGURE 2.28 - Pourcentage moyen du taux de rejet des formes d'intérêt qui n'appartiennent pas aux classes \mathcal{C}_1 ou \mathcal{C}_2 .

d'obtenir. Les données expérimentales étant obtenues au moyen d'un échantillonnage uniforme, on évalue le taux de compression \mathcal{R} en effectuant le rapport du nombre de d'échantillons obtenu par l'échantillonnage par traversée de niveaux et celui obtenu par l'échantillonnage uniforme. Les ratios d'échantillons exprimés en pourcentages sont affichés sur la **FIGURE 2.29**. Le ratio est, excepté un cas, inférieur 0,2 % et, dans les meilleurs cas, peut être inférieur à 0,1 % (axe y).

Le faible taux de rejet des formes d'intérêt et le très faible taux de compression permettent d'affirmer que la partition déterminée permet de condenser l'information. En effet, le taux de rejet indique que la partition a permis de conserver les éléments structuraux qui définissent la classe des formes (faible variation intra-classe) mais également

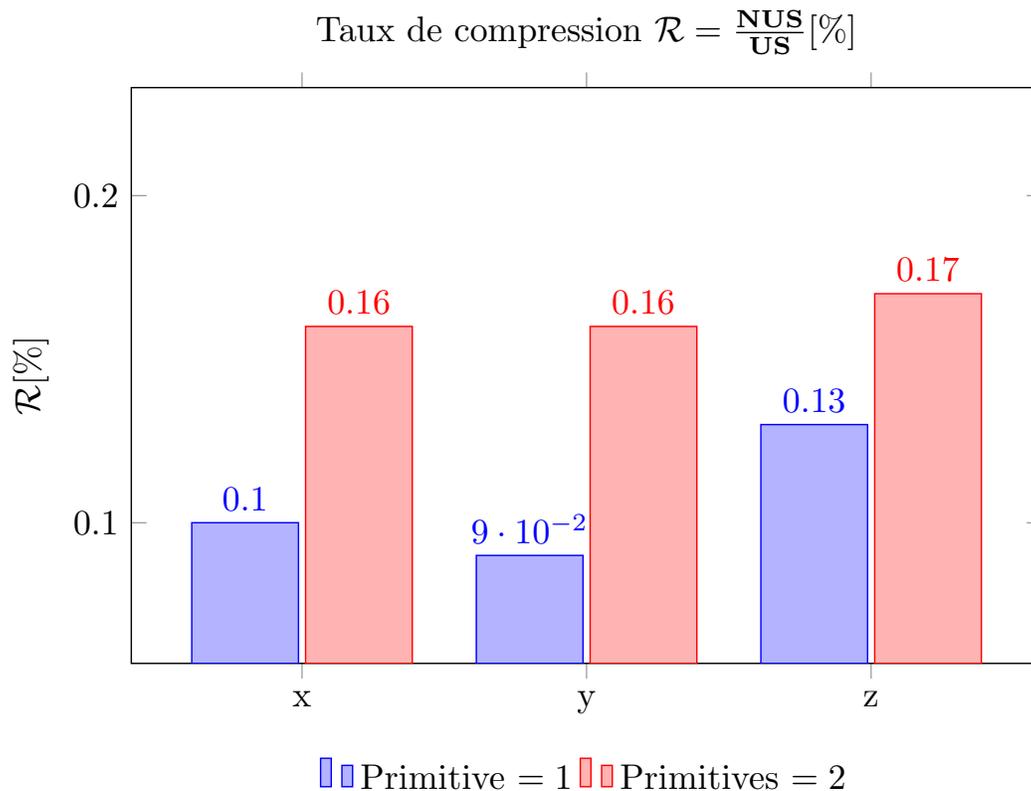


FIGURE 2.29 - Pourcentage moyen du taux de compression.

a permis de capturer seulement les échantillons contenant de l'information pertinente.

La **FIGURE 2.30**, obtenu au moyen de l'algorithme L^* [2] représente le pourcentage de taux de détection moyen des formes pour les trois axes. De façon complémentaire au taux de rejet des formes, ces résultats prouvent que les seuils qui constituent les partitions sont adaptés à l'ensemble des primitives des formes.

La **FIGURE 2.31** représente un exemple de représentation de l'espace paramétrique pour un type de primitive. La courbe claire et foncée représentent respectivement la densité de probabilité des primitives qui ne proviennent pas des formes d'intérêt et celles qui proviennent des formes d'intérêt. Le faible recouvrement entre les deux distributions indique une bonne séparabilité. La qualité de séparation peut se mesurer au moyen de l'aire sous la courbe ROC. Plus le recouvrement entre les distributions est faible, plus

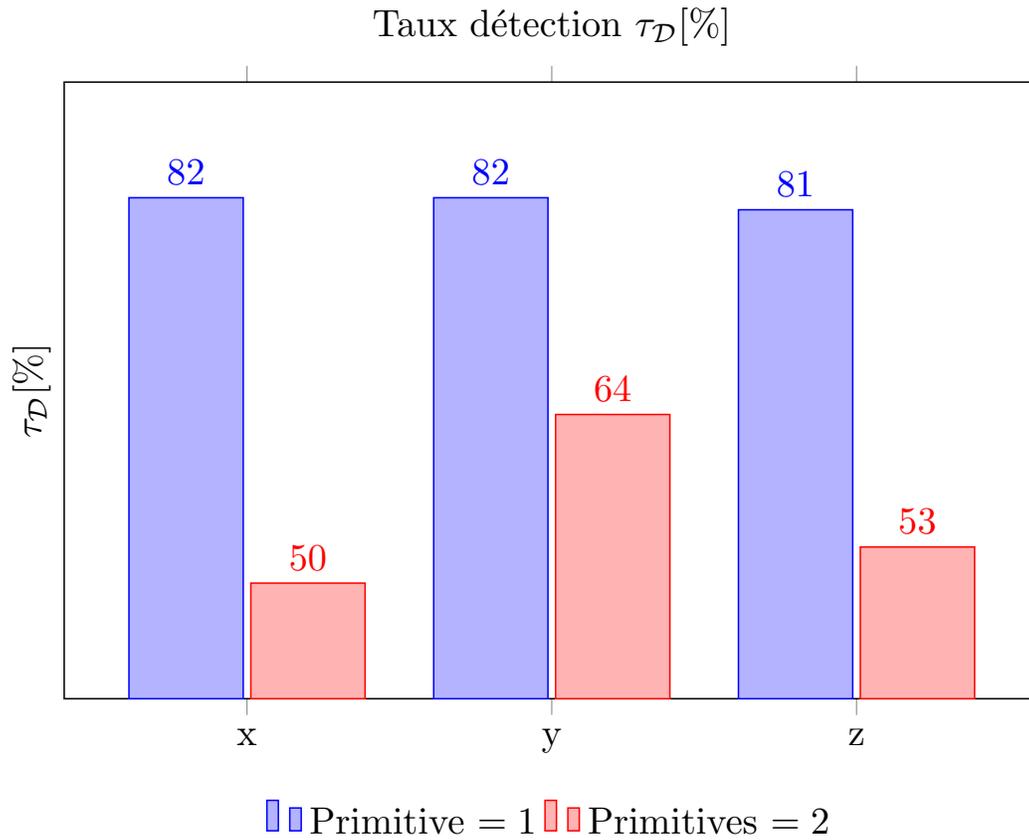


FIGURE 2.30 - Pourcentage moyen de détection.

l'aire sous la courbe est proche de 1. Quand le recouvrement est maximale, l'aire est égale à 0,5.

La **FIGURE 2.32** récapitule les aires sous la courbe ROC selon les trois axes de l'accéléromètre (x , y et z) et les enregistrements. Les classes \mathcal{C}_1 et \mathcal{C}_2 correspondant respectivement aux formes d'intérêt qui débutent avec une primitive positive et négative. Les résultats montrent des aires supérieures à 0.90. Ceci affirme que le paramètre utilisé (amplitude) permet une très bonne distinction entre les primitives provenant des formes d'intérêt et celles à rejeter.

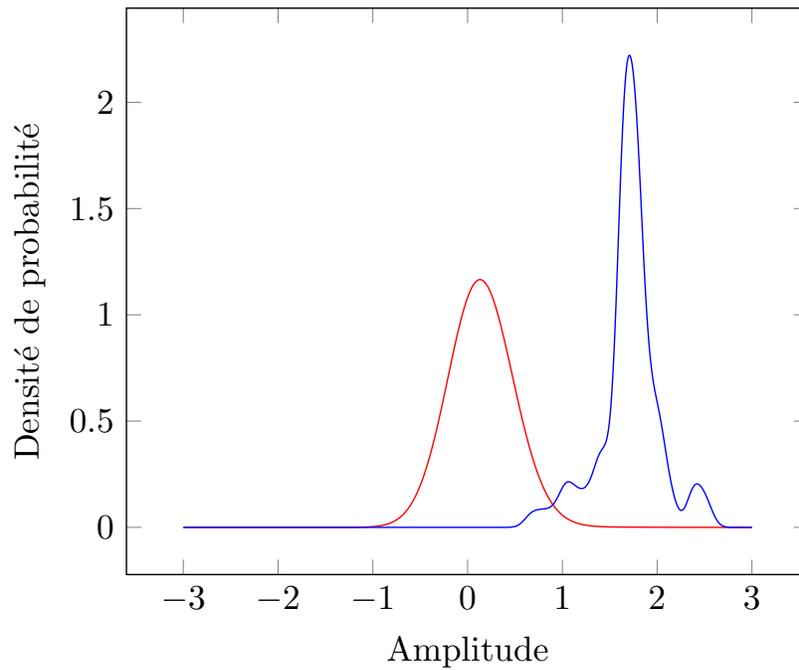


FIGURE 2.31 - Estimation de la densité de probabilité de l'amplitude des primitives positives issues des formes d'intérêt (foncé) et celles des primitives à rejeter (clair).

	Classe	Axe		
		x	y	z
1	\mathcal{C}_1	[0.99, 0.99]	[0.94, 0.99]	[0.99, 0.99]
	\mathcal{C}_2	[0.99, 0.91]	[0.98, 0.89]	[0.99, 0.96]
2	\mathcal{C}_1	[0.98, 0.99]	[0.95, 0.99]	[0.98, 0.99]
	\mathcal{C}_2	[0.99, 0.98]	[0.99, 0.94]	[0.99, 0.99]
3	\mathcal{C}_1	[0.92, 0.99]	[0.99, 0.99]	[0.96, 0.99]
	\mathcal{C}_2	[0.99, 0.99]	[0.99, 0.99]	[0.99, 0.99]
4	\mathcal{C}_1	[0.98, 0.98]	[0.92, 0.99]	[0.98, 0.98]
	\mathcal{C}_2	[0.99, 0.81]	[0.97, 0.86]	[0.97, 0.82]
5	\mathcal{C}_1	[0.99, 0.99]	[0.96, 0.99]	[0.99, 0.99]
	\mathcal{C}_2	[0.99, 0.97]	[0.98, 0.94]	[0.99, 0.97]

FIGURE 2.32 - Aire sous la courbe ROC pour deux primitives.

Sur la **FIGURE 2.31**, par exemple, le seuil qui est sélectionné correspond à la valeur de l'abscisse où il y a le croisement entre les deux courbes. Ce seuil respecte le critère de compromis entre le taux de vrais positifs et de faux positifs.

La **FIGURE 2.33** récapitule les seuils utilisés pour la discrétisation des formes d'intérêt appartenant aux classes \mathcal{C}_1 et \mathcal{C}_2 pour chaque axe de l'accéléromètre (x , y et z) et chaque enregistrement.

	Classe	Axe		
		x	y	z
1	\mathcal{C}_1	$[-0.55, 0.67]$	$[-0.18, 0.43]$	$[-0.55, 0.67]$
	\mathcal{C}_2	$[-0.43, 0.31]$	$[-0.31, 0.18]$	$[-0.55, 0.31]$
2	\mathcal{C}_1	$[-1.29, 0.92]$	$[-0.55, 0.36]$	$[-1.16, 0.92]$
	\mathcal{C}_2	$[-1.29, 0.80]$	$[-0.65, 0.26]$	$[-1.40, 0.92]$
3	\mathcal{C}_1	$[-0.67, 0.55]$	$[-0.55, 0.67]$	$[-0.67, 0.55]$
	\mathcal{C}_2	$[-1.04, 0.67]$	$[-0.67, 0.31]$	$[-1.04, 0.67]$
4	\mathcal{C}_1	$[-0.06, 0.55]$	$[-0.18, 0.43]$	$[-0.31, 0.55]$
	\mathcal{C}_2	$[-0.18, 0.55]$	$[-0.43, 0.31]$	$[-0.31, 0.31]$
5	\mathcal{C}_1	$[-1.04, 1.53]$	$[-0.43, 0.67]$	$[-0.06, 0.55]$
	\mathcal{C}_2	$[-0.18, 0.55]$	$[-0.55, 0.31]$	$[-0.55, 0.55]$

FIGURE 2.33 - Ensemble des seuils qui constitue les partitions pour des formes d'intérêt constitués de deux primitives.

Conclusion

Les résultats ont mis en évidence que la méthode de détermination de partition permet d'une part de conserver les éléments structurels des formes (très faible variabilité) et d'autre part de réduire de façon drastique le nombre d'échantillons par rapport au cas uniforme. La partition permet, a proprement dit, de condenser l'information.

2.4.3 Représentation et reconnaissance de formes

Les différentes stratégies de détection sont détaillées ci-après. La première concerne la reconnaissance de séquence de formes dont les paramètres pris en compte sont le temps et l'amplitude. La deuxième, plus générale, est un algorithme de reconnaissance de formes qui n'utilise que l'amplitude.

2.4.3.1 Détection de séquences de formes

Cet algorithme, dont l'architecture est représentée en **FIGURE 2.34** les échantillons sont traités en flux de données, il n'y a pas de stockage d'échantillons.

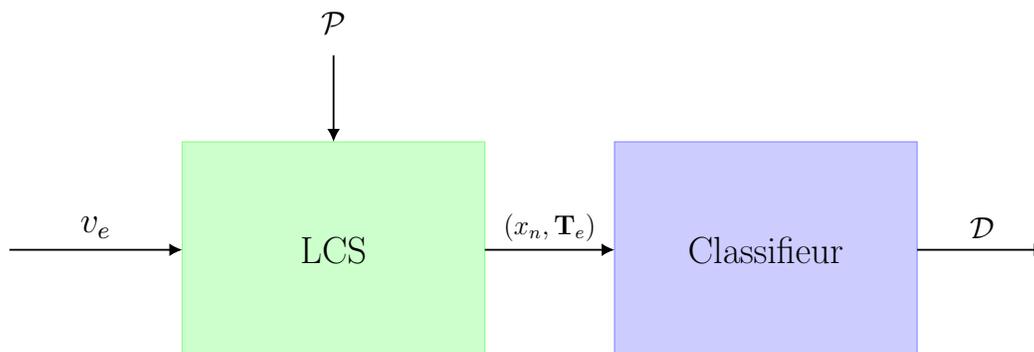


FIGURE 2.34 - Architecture du système de reconnaissance de séquences de formes.

L'avantage de détecter des séquences permet d'exploiter la redondance d'information pour ainsi diminuer le taux de faux positifs. Cet algorithme est basé sur un automate fini déterministe conçu manuellement qui exploite l'amplitude et le temps.

La **FIGURE 2.36** récapitule l'ensemble des performances obtenues avec cet algorithme. Ces mesures sont obtenues uniquement pour l'axe z . On atteint en moyenne 76 % de vrais positifs pour environ 0 % de faux positif, ce qui prouve la bonne efficacité de l'algorithme.

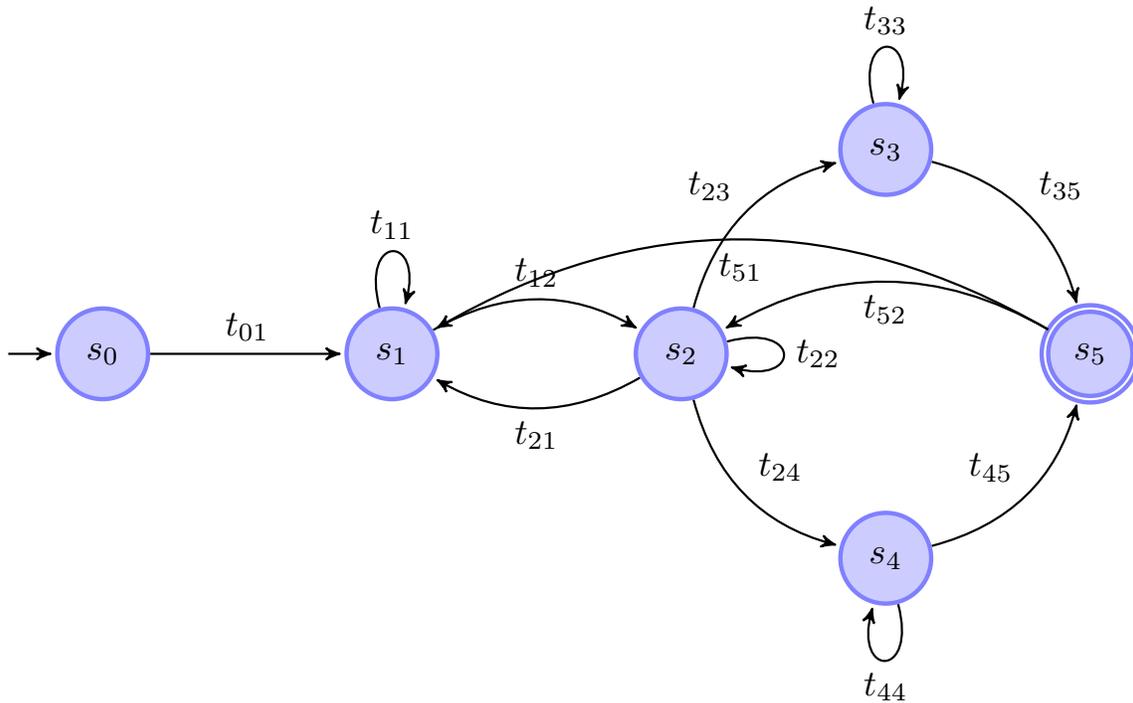


FIGURE 2.35 - Automate Fini Déterministe pour la détection de séquences de formes.

Enregistrement	VP	FP	τ_{VP}
1	31/37	0	84 %
2	26/36	0	72 %
3	12/30	0	40 %
4	25/25	0	100 %
5	33/39	1	85 %

FIGURE 2.36 - Mesures des performances de détection des séquences de formes pour l'axe z .

2.4.3.2 Détection de formes

Contrairement au cas précédent, cette approche, illustrée en FIGURE 2.37 est basée sur la reconnaissance individuelle de formes. Il n'y a plus d'information *a priori*, en d'autres termes les contraintes sont réduites pour acquérir plus de généralité. La construction de l'automate s'effectue de façon entièrement automatique par inférence

grammaticale.

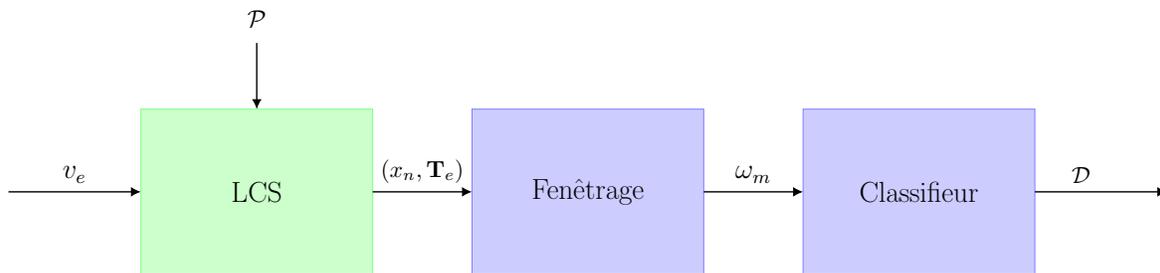


FIGURE 2.37 - Architecture du système de reconnaissance de formes.

La détermination de l'automate sur lequel repose le fonctionnement de cet algorithme s'effectue au moyen d'une méthode d'inférence grammaticale [2]. Cette méthode exploite l'information fournie par les exemples positifs et négatifs pour déterminer la structure de l'AFD qui permet de reconnaître l'ensemble des formes à reconnaître. Dans le cadre de nos travaux, les exemples positifs sont les formes qui sont définies en tant que séquence de symboles. L'information fournie par les exemples négatifs n'est pas utilisée.

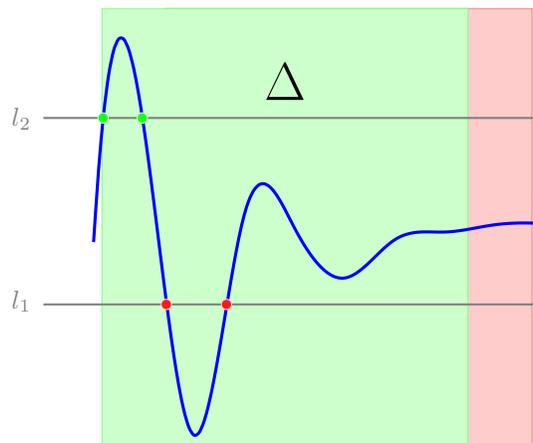


FIGURE 2.38 - Méthode d'extraction de fenêtres dans les signaux physiologiques.

L'extraction des formes est effectuée par une opération de fenêtrage. Une fenêtre se déclenche quand un niveau de détection de la première primitive de la classe \mathcal{C}_1 et \mathcal{C}_2

est franchi. Dans l'exemple de la **FIGURE 2.38**, il y a deux franchissements du niveau l_1 et du niveau l_2 , ainsi quatre fenêtres sont donc extraites.

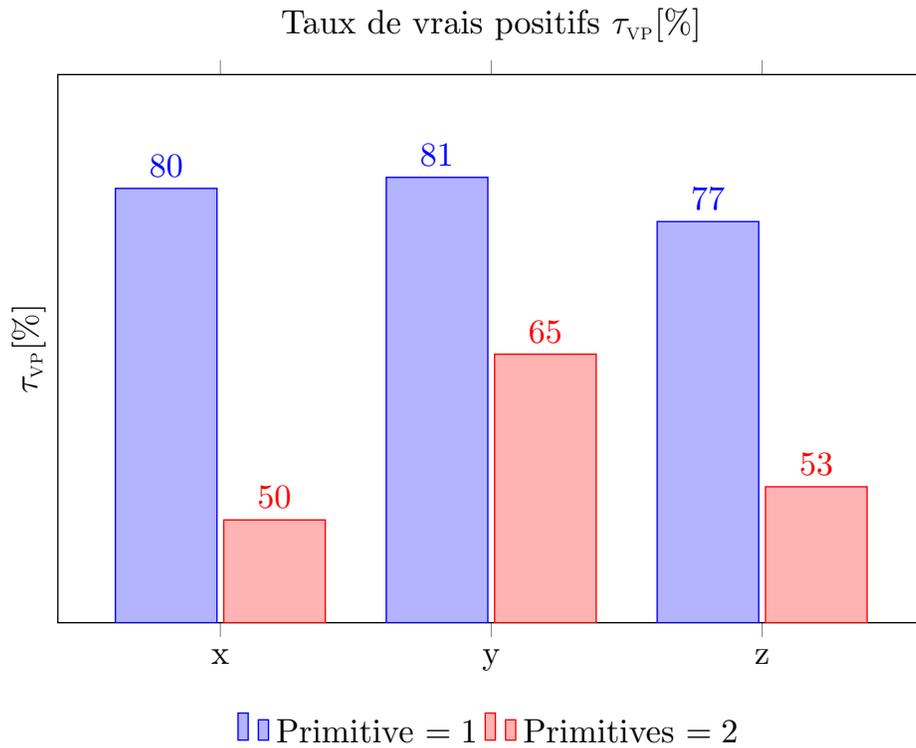


FIGURE 2.39 - Taux moyen de vrais positifs.

Les résultats obtenus avec cet algorithme sont répertoriés dans la **FIGURE 2.39** pour les taux de vrais positifs et la **FIGURE 2.40** pour les taux de faux positifs.

Conclusion

Une méthode pour la détermination des niveaux de quantification pour l'échantillonnage par traversée de niveaux est proposée. Les niveaux sont positionnés de manière à préserver l'information structurelle des formes d'intérêt. Les formes d'intérêt sont approximées en deux classes. La première se compose de formes dont leurs primitives alternent le signe positif puis négatif et pour la seconde, il y a également une alternance

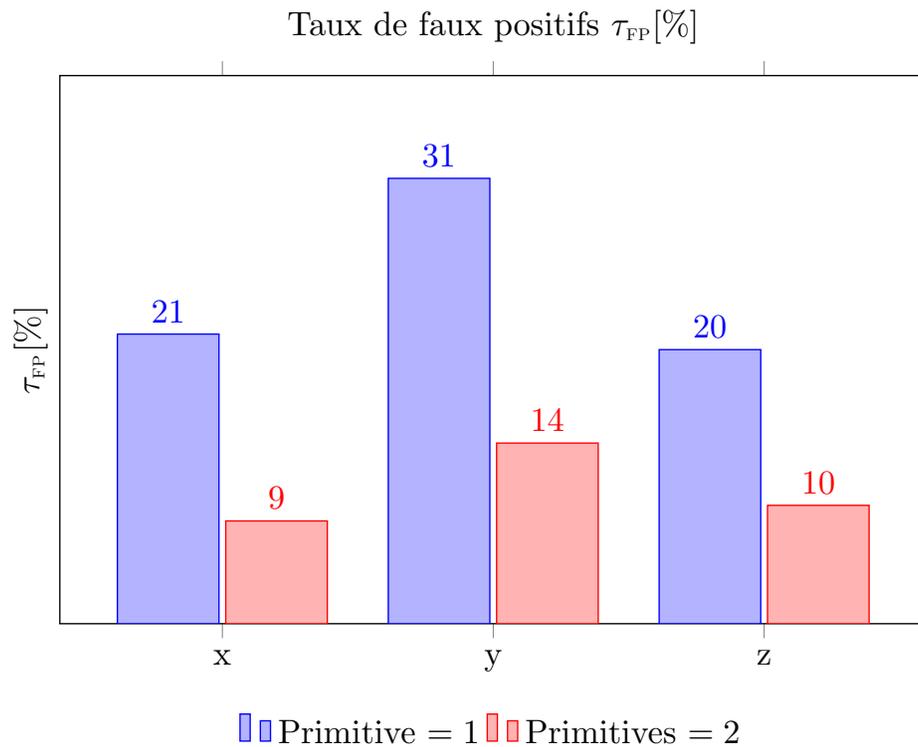


FIGURE 2.40 - Taux moyen de faux positifs.

de signe mais négatif puis positif. Cette approximation permet d'utiliser un minimum de niveaux pour l'échantillonnage.

L'algorithme prend en compte un unique paramètre à savoir le nombre de primitives qui compose les formes. Pour chaque primitive, une courbe ROC est tracée et le point qui correspond au compromis entre vrais et faux positifs est désigné comme niveau pour l'échantillonnage. Les très bons résultats de l'aire sous la courbe ROC en **FIGURE 2.32** indiquent que le paramètre utilisé pour la représentation des formes d'intérêt, à savoir l'amplitude, est pertinent.

L'algorithme d'inférence grammaticale proposé par D. Angluin [2] permet de générer un automate déterministe fini adapté aux formes d'intérêts. Les résultats du tableau **FIGURE 2.39** démontrent que l'échantillonnage non-uniforme effectué au moyen des niveaux déterminés par notre méthode de discrétisation permet de conserver l'infor-

mation structurelle des formes d'intérêt. De plus, étant donné les résultats du ratio de compression du tableau **FIGURE 2.29**, on peut conclure que la partition ROC permet de condenser l'information, c'est-à-dire de compresser l'information (utile) dans très peu de données. Le couple composé du taux de vrais positifs et du taux de compression peut définir un critère d'évaluation et de comparaison avec d'autres méthodes de partitionnement.

L'algorithme de D. Angluin [2] a été appliqué seulement pour les formes isolées. Les résultats du graphe en **FIGURE 2.40** montrent qu'il y a un pourcentage non négligeable de faux positifs. Pour réduire le taux de faux positifs tout en préservant le taux de vrais positifs une solution consiste à déclencher le fenêtrage non seulement en scrutant l'amplitude, comme c'est le cas dans le cas précédent, mais en scrutant à la fois l'amplitude et le temps. De plus, en considérant non plus une forme isolée mais une séquence de formes, cela rajoute de la redondance d'information. Comme le montre les résultats en **FIGURE 2.36**, le taux de faux positifs est quasiment nul mais le taux de vrais positifs est élevé. Un avantage majeur de cet algorithme est que l'on s'affranchit de l'opération de fenêtrage : la reconnaissance de formes s'effectue en flux de données ce qui permet d'assurer un débit maximal.

Ces résultats permettent de confirmer toute la pertinence de la stratégie employée pour effectuer une chaîne de traitement efficace. En effet, le schéma d'échantillonnage capture seulement les échantillons utiles et ne suscite le module de reconnaissance de formes seulement quand cela est nécessaire. La chaîne de traitement complète sera par la suite implémentée sur un circuit logique programmable, à savoir un FPGA, afin de valider le fonctionnement de cette approche théorique. Le chapitre 3 expliquera les mécanismes et les vertues de la logique asynchrone. Le caractère événementiel de l'échantillonnage par traversée de niveaux justifie naturellement le choix d'une implémentation matérielle asynchrone.

Deuxième partie

Prototypage et architecture matérielle

Chapitre 3

Logique asynchrone

Introduction

Il existe principalement deux classes de circuits numériques : synchrone et asynchrone. Ces deux méthodes n'étant pas indépendantes, certaines applications les combinent. Cependant, de nos jours, la plupart des circuits fabriqués sont synchrones. Ils sont basés sur deux hypothèses fondamentales : tous les signaux sont binaires et tous les composants partagent la même notion de temps, qui est définie par une horloge distribuée dans le circuit. La binarisation des signaux permet une implémentation électrique simple et offre un cadre de conception maîtrisé grâce à l'algèbre de Boole. L'utilisation d'une horloge permet de façon fonctionnelle de simplifier la synchronisation des composants.

Le fonctionnement des circuits asynchrones est fondamentalement différent. Bien que les signaux soient binaires, les composants ne partagent pas la même notion de temps, ainsi ils ne sont donc plus commandés par un signal périodique unique. C'est un protocole de communication qui est utilisé pour faire la synchronisation, la communication, et le séquençement des opérations.

3.1 Concepts de base

3.1.1 Fonctionnement synchrone

Les circuits synchrones sont composés d'un réseau de registres cadencés et de circuits combinatoires. La **FIGURE 3.1** illustre un circuit synchrone par un automate à états finis. L'état courant de l'automate est stocké dans un ensemble de registres. L'état suivant et la sortie sont fonction de l'état courant et de l'entrée. À chaque nouveau front d'horloge, les registres sont activés et l'état courant est remplacé par l'état suivant. Le fonctionnement du circuit synchrone s'apparente au modèle RTL (en anglais *Register Transfer Level*) [37] [50].

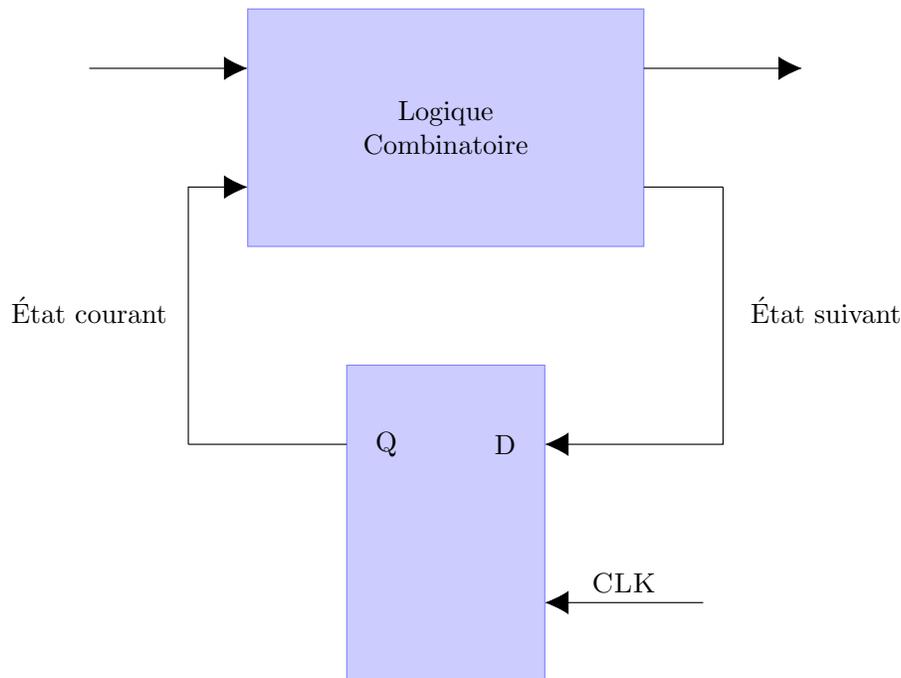


FIGURE 3.1 - Mécanisme de fonctionnement d'un circuit synchrone par l'intermédiaire d'un automate à états finis

Le signal d'horloge joue le rôle d'un actionneur global. Tous les éléments du système évoluent ensemble lors de l'occurrence d'un événement d'horloge (front montant ou des-

endant), l'exécution de tous les éléments est donc synchronisé. Ce mécanisme global d'activation introduit une contrainte temporelle. Pour assurer le bon fonctionnement du circuit, tous les éléments doivent respecter un temps d'exécution maximum fixé qui est défini comme le plus long chemin combinatoire (chemin critique). Tous les éléments du système sont donc synchronisés dans le temps. À l'opposée, les systèmes asynchrones évoluent de façon localement synchronisés. Pour dimensionner la période d'horloge, il faut prendre en compte les variations de vitesse des éléments du circuit qui sont principalement induites par les variations dans le processus de fabrication, de température et de tension d'alimentation mais également le biais d'horloge. Ainsi, on ajoute une marge de sécurité d'un facteur 2.

La mise en place des réseaux de distribution d'horloge est devenue un enjeu-important pour la conception des circuits synchrones car ils peuvent affecter le bon fonctionnement des circuits et détériorer leur fiabilité. Les signaux d'horloge sont particulièrement affectés par la génération technologique, par le fait que les lignes d'interconnexions deviennent de plus en plus résistives au fur et à mesure qu'elles se réduisent (K. van Berkel et al., 1999 [50]). L'augmentation de la résistance est l'une des raisons principales de l'importance de la distribution d'horloge. De plus, les différences dans les délais des signaux d'horloge peuvent dégrader les performances du système et même créer un dysfonctionnement par le déclenchement du traitement de données invalides. L'un des enjeux de la distribution d'horloge est de supprimer le biais d'horloge. Le biais d'horloge entre 2 registres est défini comme la différence de temps d'arrivée. Quand les signaux d'horloge sont complètement synchronisés, le biais est nul. La distribution d'horloge et les méthodes de débiaisage sont nombreuses mais elles sont coûteuses en terme de consommation et de surface. Ces coûts augmentent au fur et à mesure que la fréquence d'horloge augmente. Une méthode couramment utilisée est de distribuer une horloge pour une configuration en H avec des *buffers* amplificateurs aux points de distributions (en anglais *fanout*). Le problème de cette approche est que plus de buffers

sont utilisés, plus le biais d'horloge augmente. Une technique commune pour contrôler le biais d'horloge est d'utiliser des boucles à verrouillage de phase (PLL). Son but est de réduire dynamiquement la phase d'un signal pour le faire concorder avec la phase d'un autre.

3.1.2 Fonctionnement asynchrone

Pour s'affranchir du signal d'horloge, les circuits asynchrones utilisent des modules de communication représentés en **FIGURE 3.2**. La détection de la présence de données sur le port d'entrée déclenche le traitement de ces données, dont le résultat est envoyé sur le port de sortie. Ce fonctionnement est très similaire à des systèmes de flot de données (en anglais *data-flow*), dont l'arrivée de données déclenche une opération [37].

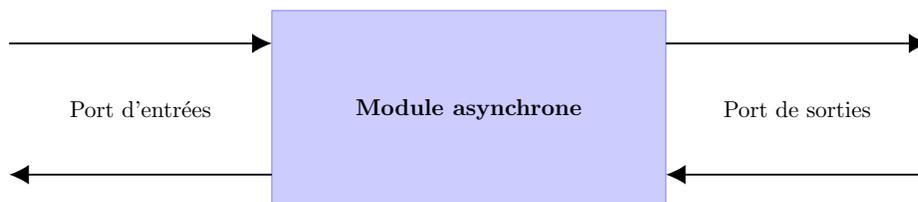


FIGURE 3.2 - Caractéristiques d'un module asynchrone

Les données de communication à travers les ports ne sont pas contrôlées par un signal externe, comme une horloge, mais par un protocole de communication implémenté dans le module asynchrone. Ce type de protocole requiert un échange bi-directionnel d'informations entre les émetteurs et les récepteurs appelé *protocole à poignée de main* (en anglais *handshake protocol*). Ce protocole utilise des signaux de requête qui génèrent une action et des signaux d'acquittements pour achever cette action. La plupart des protocoles de communication requierent une alternance de requêtes et d'acquittements.

3.1.3 Caractéristiques des opérateurs

Un opérateur asynchrone peut être considéré comme une cellule réalisant une certaine fonction et communiquant avec son environnement à travers des canaux de communication. Ces canaux de communication servent à échanger des données avec l'environnement et des informations de synchronisation (parfois les deux simultanément). Un opérateur asynchrone peut être caractérisé par au moins 4 paramètres fondamentaux :

- **Temps de latence** : Correspond à la chaîne combinatoire la plus longue (critique) nécessaire à une sortie. Cette caractéristique ne dépend pas de la mémoire de l'opérateur. En asynchrone, un opérateur de mémorisation inoccupée ou vide se comporte comme une cellule combinatoire qui laisse simplement la donnée le traverser. Le temps de latence est variable en fonction des données en entrée, dépendant de l'algorithme et de l'implémentation choisit.
- **Temps de cycle** : Caractérise le temps minimum qui sépare l'acceptation de 2 informations en entrée. C'est dans le cas général, le temps nécessaire pour échanger une donnée entre 2 ressources de mémorisation connectées.
- **Profondeur du pipeline** : Définit le nombre maximum de données ou d'informations que l'opérateur peut mémoriser. Cela correspond à la différence maximale entre l'indice de la donnée présente à la sortie et l'indice de la donnée présente à l'entrée, alors que toutes les ressources de mémorisation sont occupées. Dans le cas asynchrone, le nombre de données présentes dans le pipeline n'est pas imposé.
- **Protocole de communication** : Permet d'échanger des informations avec l'environnement. Les cellules doivent comporter le même protocole de communication pour pouvoir les connecter entre elles. Les protocoles d'échange asynchrone assurent la détection de la présence des données en entrées et assurent la génération d'une signalisation qui indique qu'une information est en cours de traitement

et d'autre part qu'elle est disponible en sortie.

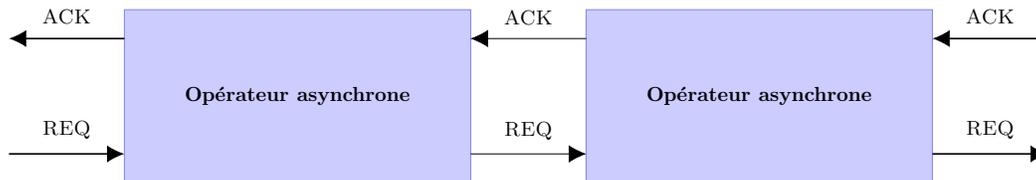


FIGURE 3.3 - Opérateurs asynchrones

Les opérateurs asynchrones sont caractérisés par un nombre de paramètres plus importants que les opérateurs synchrones. Ainsi, l'asynchrone, avec ces paramètres, va permettre de définir une classe plus importante de circuits.

3.1.4 Principe de base

Le mode de fonctionnement asynchrone est caractérisé par un transfert d'informations qui est généré localement par une signalisation adéquate. Le contrôle local doit être à l'écoute des communications entrantes, déclencher le traitement localement si toutes les informations sont disponibles et produire des valeurs sur les sorties. Afin d'émettre de nouvelles données, les émetteurs doivent être informés qu'elles ont bien été traitées. Ainsi, le contrôle local doit prendre en charge une signalisation bi-directionnelle. Toute action de communication doit être acquittée par le récepteur afin que l'émetteur puisse émettre à nouveau. Les communications sont dites à *poignée de mains* ou de type *requête-acquittement* (c.f. **FIGURE 3.3**).

En prenant par exemple deux cellules asynchrones : un émetteur A et un récepteur B. Une requête est émise de A vers B pour indiquer que A demande une action de la part de B. Quand B fini l'action, il acquitte la requête en émettant un signal d'acquittement vers A.

3.1.4.1 Protocoles de communication

Il existe principalement 2 classes de protocoles, le protocole 2 phases et 4 phases. Il n'y a pas d'hypothèse sur le délai entre 2 événements successifs car le changement de signal par l'émetteur est acquitté par un changement d'un signal du récepteur et vice-versa.

Le protocole 2-phases, aussi appelé *non retour à zéro* (en anglais *non return to zero* (NRZ)) est représenté en **FIGURE 3.4**. Le récepteur détecte la présence de nouvelles données, effectue le traitement et génère le signal d'acquiescement (Phase 1). L'émetteur détecte le signal d'acquiescement et émet les nouvelles données si elles sont disponibles (Phase 2). La signalisation est basée sur événement puisque la détection des données et des acquiescements sont générés au moyen de front montants et descendants.

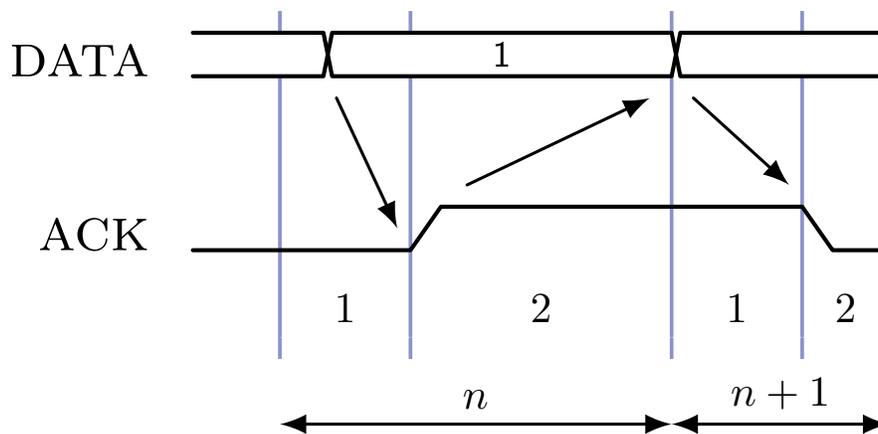


FIGURE 3.4 - Protocole 2 phases (adapté de [38]).

Le protocole 4-phases, aussi appelé *retour à zéro* (en anglais *return to zero* (RZ)) est représenté en **FIGURE 3.5**. La présence des nouvelles données est détectée par le récepteur qui effectue le traitement et génère l'acquiescement (Phase 1). L'émetteur

détecte le signal d'acquittement et émet des données invalides (retour à zéro) (Phase 2). Le récepteur détecte le passage des données dans l'état invalide et place le signal d'acquittement dans l'état initial (retour à zéro) (Phase 3). L'émetteur détecte le retour à zéro de l'acquittement. Il est prêt à émettre de nouvelles données. Contrairement au protocole NRZ, le protocole RZ est basé sur la détection de niveaux.

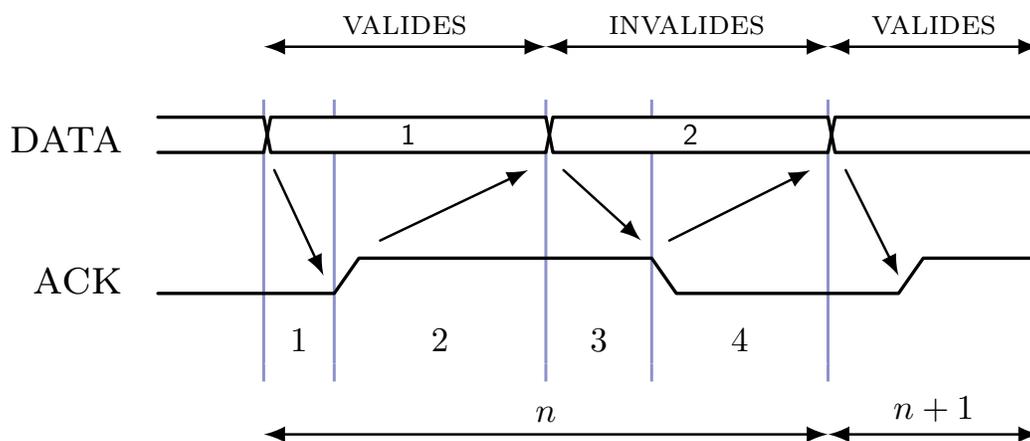


FIGURE 3.5 - Protocole 4 phases (adapté de [38]).

Le protocole 4-phases demande 2 fois plus de transitions que le protocole 2-phases. Également, il semble plus lent et consomme plus d'énergie. Cependant, les techniques d'optimisations du pipeline permettent de s'affranchir de la pénalité apparente des phases de retour à zéro.

Le protocole 2-phases requiert un matériel plus important que le protocole 4-phases car il nécessite de détecter des transitions et non des niveaux. Le faible nombre de transitions du protocole 2-phases se répercute sur la complexité du matériel et la consommation est comparable à celle des réalisations 4-phases.

Le protocole 4-phases est majoritairement utilisé pour implémenter les parties internes d'un circuit intégré. Le développement des processeurs asynchrones sous le nom

de AMULET est doté d'une architecture en micropipeline. Le but était de démontrer la faisabilité dans le développement d'un circuit complexe asynchrone et évaluer le potentiel de la logique asynchrone pour les applications à faible consommation pour la conception d'un processeur entièrement asynchrone. La première version AMULET1 [15] utilise un protocole 2-phases. En terme de consommation, il ne démontre pas un avantage significatif par rapport à son homologue synchrone ARM6 qui a été le support de sa conception.

L'amélioration de ce processeur a mené vers une deuxième version [16] qui utilise un protocole 4-phases et qui montre des performances en terme de consommation supérieures à son prédécesseur. Il n'est pas possible de déterminer l'avantage d'un protocole par rapport à l'autre car les efforts d'optimisations ont été apportées sur l'architecture.

3.1.4.2 Codage des données

Le protocole de données groupées (en anglais *bundled-data*) se réfère à la situation où les signaux de requête et d'acquiescement sont regroupés avec les fils de données (c.f. **FIGURE 3.6**). On sépare l'information de contrôle de l'information des données.

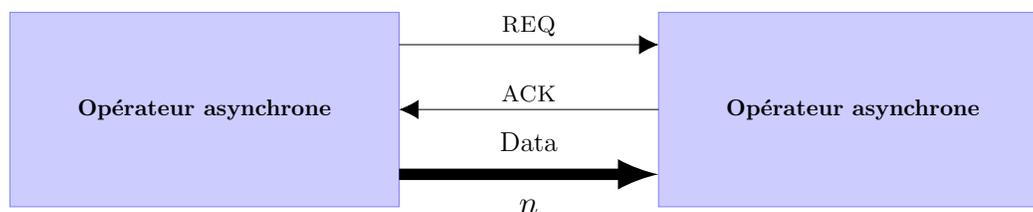


FIGURE 3.6 - Chaîne de communication de données regroupées (adaptée de [45]).

Une approche plus sophistiquée pour détecter la présence des données est d'indiquer la fin d'un traitement est d'utiliser un encodage. Dans ce cas, le signal de requête est encodé dans les données en utilisant 2 fils par bit d'information (c.f. **FIGURE 3.7**). Un seul fil de données ne permet pas de détecter que la nouvelle donnée prend un état identique à la précédente. Les protocoles utilisent en général un codage bifilaire ou

double rail (en anglais *dual-rail*) pour chaque bit de donnée. Cela double le nombre de fils par rapport aux réalisations synchrones. Avec deux fils par bit de données, 4 états sont utilisables pour exprimer deux valeurs logiques (0 et 1). Il y a 2 types de codages qui sont couramment utilisés : le codage 3-états et le codage 4-états.

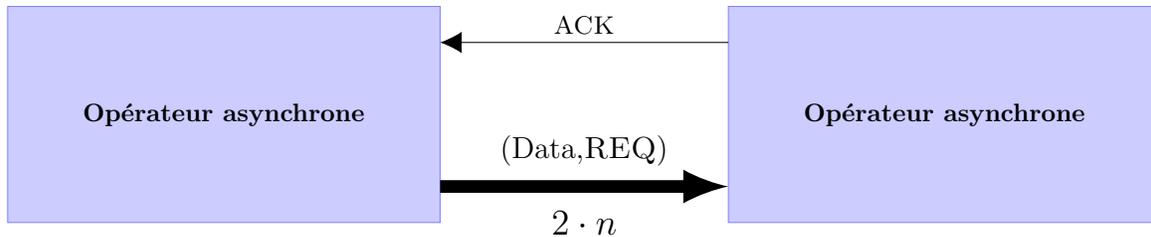


FIGURE 3.7 - Chaîne de communication avec le protocole 4-phases double rail (adaptée de [45]).

Le codage 3-états (c.f. **FIGURE 3.8** (a)) à 3 états détectables : 1 logique, 0 logique et l'état nul N . Après la transmission de chaque bit (1 ou 0), un nul (N) représente l'invalidité des données. Les données sont définies par la valeur se situant entre 2 nuls successifs. Ce codage garantit que le passage d'un état à un autre se fait toujours par changement de l'état d'un seul bit sur les 2.

Le codage 4-états (c.f. **FIGURE 3.8** (b)) permet de coder les valeurs 0 et 1 d'un seul bit avec 2 combinaisons. L'une des combinaisons est considérée comme étant de parité impaire et l'autre de parité paire. Deux états représentent le 1 logique et 2 autres représentant le 0 logique. Chaque fois qu'une donnée est émise, on change sa parité permettant de passer d'une valeur logique à une autre, sans passer par l'état d'invalidité, ainsi on alterne entre les phases P et Q.

3.1.4.3 Signal d'acquittement

Les protocoles décrits utilisent un signal d'acquittement que chaque opérateur asynchrone doit générer. Une méthode proposée par A.L. Davis [10] consiste à utiliser des horloges locales. L'arrivée d'une requête déclenche une horloge interne du module et

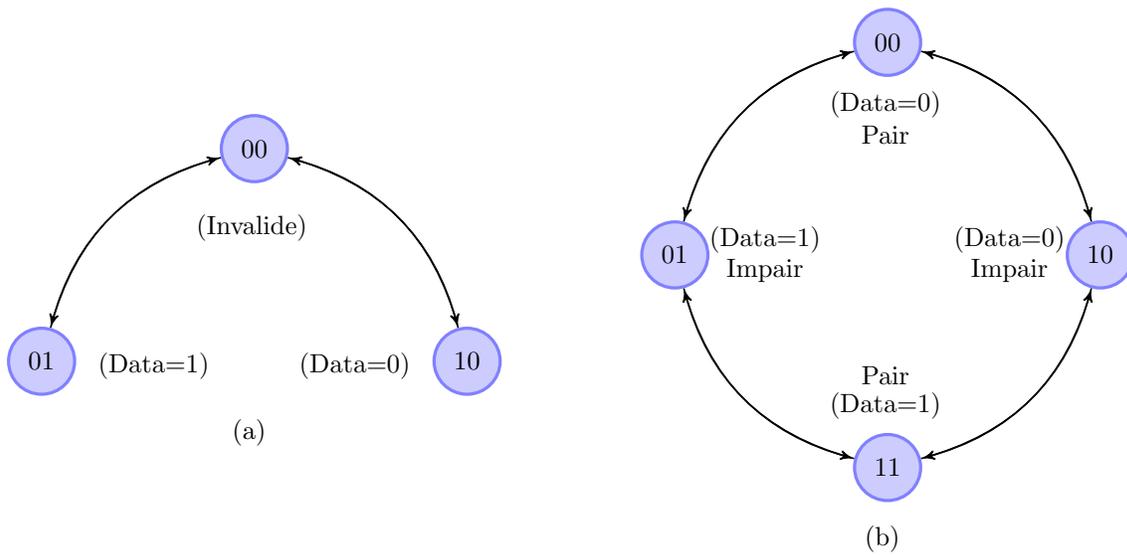


FIGURE 3.8 - Codage (a) 3 et (b) 4 états des données (adapté de [38]).

après un nombre de cycles déterminé, le calcul est terminé. L'horloge interne est stoppée et un signal d'acquittement est généré. Cette technique s'avère très coûteuse et lente pour des modules de petite taille.

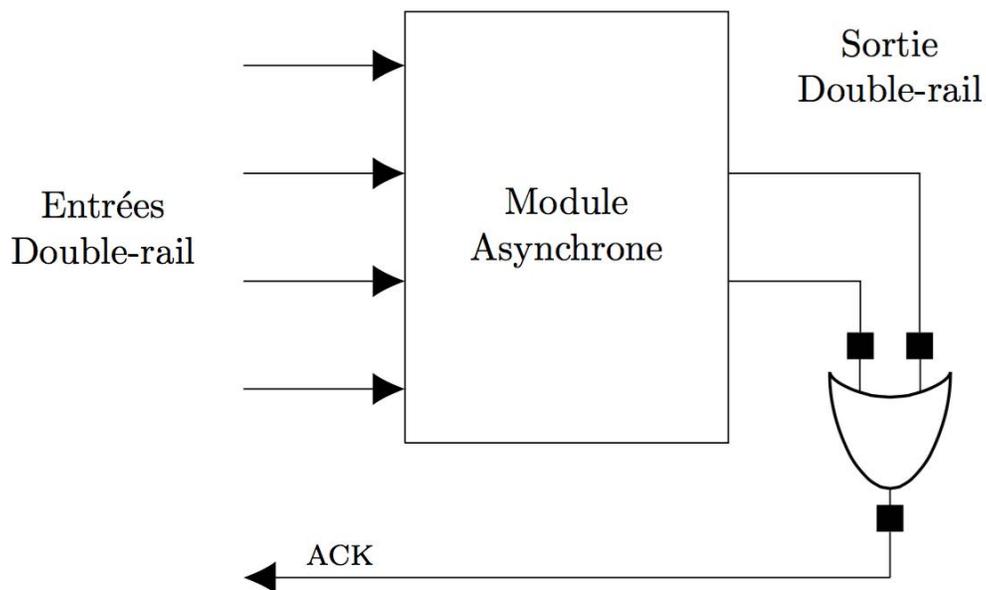


FIGURE 3.9 - Acquittement en codage double rail 4 phases.

Une technique simple consiste à avoir recours à des hypothèses temporelles ou des modèle de délai. On retarde le signal de requête d'une certaine durée en concordance avec le temps de calcul du module asynchrone réalisant le traitement. Ce signal retardé peut directement être utilisé comme signal d'acquiescement. Cette méthode est très proche des circuits synchrones car elle est basée sur un modèle de porte à temps de traversé borné.

Une autre méthode porte sur l'analyse de courant. Elle est basée sur le fait qu'en technologie CMOS, le courant consommé est très différent lorsqu'il existe des transitions, que lorsque les signaux sont stables. L'idée est de placer des capteurs de courant pour détecter la fin de calcul en détectant une consommation nulle [12].

L'utilisation du codage des données avec les protocoles double-rail permet une meilleure robustesse. En considérant un module conçu avec un codage 3-états et un protocole 4-phases, on utilise une porte OU sur la sortie double-rail (c.f. **FIGURE 3.9**). Avec l'utilisation d'un protocole 2-phases, une porte OU exclusif doit être utilisé sur la sortie double-rail.

3.2 Motivations

3.2.1 Calcul en temps minimum

Dans les circuits, le délai de traitement dépend de l'état courant et de la valeur des entrées. Pour les circuits synchrones, la période d'horloge est déterminée par la durée du chemin combinatoire critique, ainsi qu'une marge de sécurité prenant en compte le délai des bascules et du biais d'horloge, ce qui limite la vitesse du circuit. Le délai de traitement est ainsi toujours plus faible que la période d'horloge.

À l'inverse, les circuits asynchrones exploitent de façon intrinsèque les variabilités locales des délais de la logique combinatoire et donc un traitement débute dès la fin du précédent. Il n'est pas nécessaire d'attendre le prochain front d'horloge comme dans le

cas synchrone. Ceci offre potentiellement un avantage mais cependant il faut prendre en compte les frais d'implémentations (en anglais *overhead*) de détection de fin de traitement. On peut illustrer la dépendance de traitement aux délais avec un additionneur avec propagation de retenue [50].

Les caractéristiques de vitesse d'un circuit varient en fonction de paramètres qui influencent le fonctionnement des dispositifs élémentaires, tels que les variations des paramètres technologiques, la température ou la tension d'alimentation. Le fonctionnement en flot de données des circuits asynchrones les rends très robustes vis-à-vis de ces variations. Ils ralentissent quand la tension d'alimentation chute ou quand la température augmente. Les circuits asynchrones fonctionnent toujours à vitesse maximale permise par les dispositifs élémentaires et les conditions de fonctionnement.

3.2.2 Support fiable pour les traitements non-déterministes

Le traitement d'un signal asynchrone avec une horloge est une opération périlleuse pour un circuit synchrone. En effet, l'échantillonnage asynchrone peut faire apparaître un état métastable dont la durée est indéterminée, non bornée. T.J. Chaney et C.E. Molnar [9] mettent en évidence l'existence de comportements oscillatoires et métastables de bascules en réponse à une entrée présentant un état logique indéfini qui peut apparaître avec les synchroniseurs. Les signaux synchrones imposent à tous les signaux un temps d'établissement borné compatible avec la période d'horloge. Ils ne permettent pas d'assurer un comportement correct dans tous les cas. Des mécanismes de division d'horloge sont mis en œuvre pour limiter la probabilité d'apparition d'états métastables. Par contre, en asynchrone, il est possible d'attendre le temps nécessaire pour la fin de l'état métastable.

3.2.3 Modularité

L'intégration VLSI des circuits synchrones a fait apparaître des problèmes de réduction d'échelle (C. Seitz, 1978 [30]). En effet, si le circuit est réduit d'un facteur d'échelle, le temps que met un signal électrique pour traverser 2 points du circuit peut parfois dépasser la période d'horloge. Ainsi, le délai des interconnexions est devenu non négligeable avec la technologie actuelle. Le délai dans les portes décroît rapidement en fonction de l'évolution technologique, à l'inverse, la latence induite par les interconnexions augmente (C.V. Berkel et al., 1999 [50]).

Le fonctionnement des circuits asynchrones est invariant en fonction du facteur d'échelle car ceci est dû à la localité du contrôle et à l'utilisation par tous les opérateurs d'un protocole de communication. Ceci permet d'avoir une modularité des circuits asynchrones car il n'y a pas d'hypothèses temporelles qui garantissent le fonctionnement. Les blocs asynchrones peuvent être construits indépendamment.

3.2.4 Faible consommation

Un circuit en veille ne consomme seulement qu'un courant de fuite. Avec la technologie CMOS, ce courant est négligeable par rapport au courant dynamique quand le circuit est actif. Un circuit synchrone est soit en veille (e.g. l'horloge est éteinte) soit actif. Les circuits asynchrones ne consomment de l'énergie que quand ils sont actifs. Chaque sous-circuit est en veille jusqu'au moment où il est activé. Après avoir effectué son traitement, il retourne à l'état non-actif.

La comparaison du diviseur de fréquence synchrone et asynchrone permet d'illustrer les caractéristiques inhérentes de faible consommation de la logique asynchrone par rapport au synchrone (K. van Berkel et al., 1999 [50]). Une mise en cascade de N éléments diviseurs par 2 divise la fréquence d'entrée par 2^N . Le second élément fonctionne à une vitesse 2 fois plus faible que le premier et dissipe la moitié de la puissance, le troisième

consomme le quart et ainsi de suite. Ainsi, le circuit asynchrone consomme sur une période d'horloge moins de la moitié que le premier élément, indépendamment de N . À l'inverse, son homologue synchrone dissipe en fonction de N .

Des avantages similaires se sont retrouvés dans les compteurs modulo N [51]. K. van Berkel et M. rem [53] ont réalisés 2 conceptions asynchrones d'un décodeur (simple-rail et double-rail) et établis la comparaison avec la version synchrone. Une autre application est un récepteur de radio-message (en anglais *pager*) [22].

3.2.5 Faible bruit et faible émission

Pour les circuits asynchrones, les pics de consommation qui est un artefact créé par la synchronisation des circuits synchrones n'existe pas car il y a une distribution du contrôle dans toute la structure du circuit. Par conséquent, l'activité électrique d'un circuit asynchrone est mieux répartie dans le temps que celle d'un circuit synchrone, ce qui réduit le bruit dans les lignes d'alimentation.

Les sous-modules d'un système peuvent interagir de façon inattendue. Par exemple, un sous-module numérique génère un bruit sur la tension d'alimentation ou induit un courant dans le substrat de silicium. Ce bruit peut affecter les performances d'un convertisseur analogique numérique qui utilise la même source d'alimentation ou se situe sur le même substrat. Un autre exemple est un sous-module qui émet des radiations électromagnétiques à sa fréquence d'horloge qui perturbe un récepteur radio. Avec l'absence d'horloge, les circuits ont de meilleures performances électromagnétiques [52]. L'activité du circuit est en général maximale peu de temps après le front d'horloge. Elle décline jusqu'à ce que le circuit soit en veille avant l'arrivée du prochain front d'horloge. Le signal d'horloge module le courant d'alimentation. Les chutes locales de tension d'alimentation peuvent avoir un impact sur les performances et un bruit excessif peut avoir un impact sur la fiabilité du circuit. Il faut analyser en fréquence le courant d'alimentation d'un circuit synchrone pour se rendre compte de l'impact qu'il peut avoir. Les

harmoniques générées peuvent interférer avec les antennes et les circuits analogiques sensibles, incluant les circuits radio du FM (100 MHz) aux téléphones portables (1 à 2 GHz).

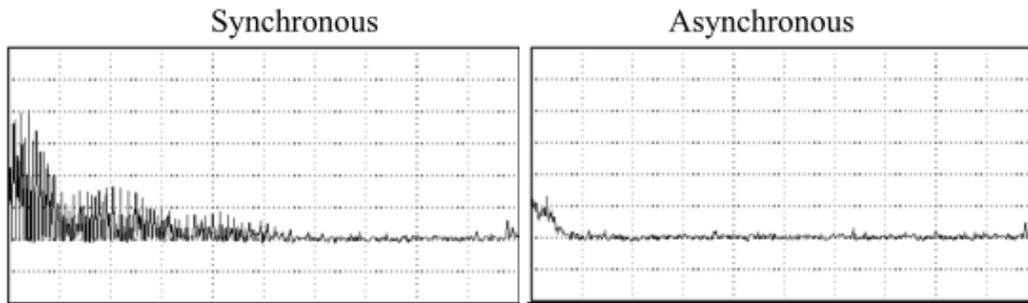


FIGURE 3.10 - Spectre fréquentiel de 2 microcontrôleurs 80C51. Ils exécutent le même programme aux mêmes performances. (a) Le spectre de la version synchrone montre une suite de pics aux harmoniques 3-6 MHz dû à l'horloge (b) Le spectre de la version asynchrone ne montre pas ces pics et montre clairement que le spectre possède moins d'énergie.

La conséquence du faible rayonnement électromagnétique des circuits asynchrones est à l'origine de la commercialisation du premier circuit asynchrone, un processeur 8 bit compatible 80C51. La Figure **FIGURE 3.10** montre le spectre du courant d'alimentation de la version synchrone (**FIGURE 3.10** (a)) et de la version asynchrone (**FIGURE 3.10** (b)) d'un microcontrôleur 80C51. La version synchrone montre une série d'harmoniques de la fréquence d'échantillonnage (autour de 3 à 6 MHz), jusqu'à 300 MHz. Le récepteur radio-message peut interférer avec les circuits analogiques très sensibles. À l'inverse, la faible émission de la version asynchrone permet d'avoir un microcontrôleur actif durant la phase de récepteur du récepteur de radio-message. La **FIGURE 3.11** représente l'architecture synchrone et asynchrone de ce circuit.

La réduction de la consommation généralement réduit l'énergie contenue dans le spectre. Un exemple d'émission électromagnétique mesuré d'un microprocesseur asynchrone est présenté par S.B. Furber et al. [16].

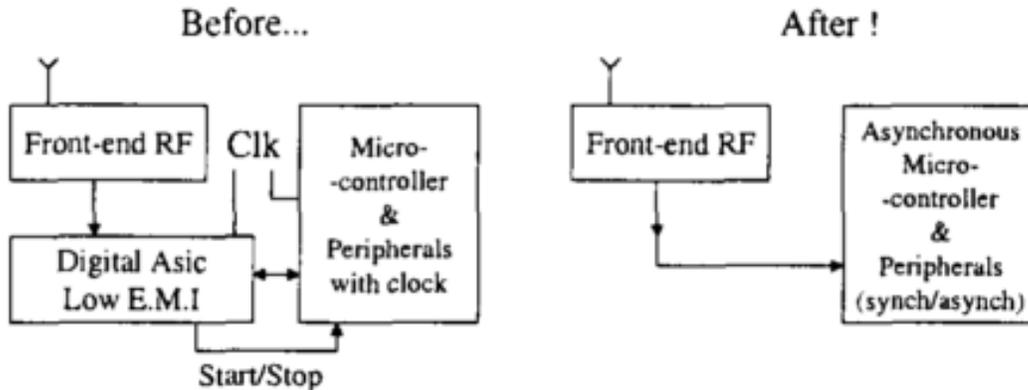


FIGURE 3.11 - Architecture simplifiée du récepteur radio-message avant et après l'utilisation d'un microcontrôleur faible bruit asynchrone.

3.3 Classification des circuits asynchrones

3.3.1 Aléas

Pour les systèmes synchrones, le traitement s'effectue entre 2 fronts d'horloge. Les aléas (en anglais *glitch*) sur les fils pendant une période d'horloge ne sont en général pas un problème. Le système fonctionne correctement aussi longtemps qu'un résultat stable et valide est produit avant le prochain front d'horloge, quand le résultat est échantillonné. À l'inverse, les systèmes asynchrones n'ont pas d'horloge, le résultat n'est plus échantillonné à des intervalles discrets. Par conséquent, n'importe quel aléa peut être traité par le système comme un changement réel dans la valeur et peut causer un dysfonctionnement du système.

Plusieurs approches ont été utilisées pour éliminer les aléas combinatoires. La première est l'utilisation de délais inertiels pour atténuer les pics non désirés. La plupart des travaux de recherche des aléas dans les circuits asynchrones est basé sur les délais inertiels [49]. La seconde, si on suppose un délai borné, les aléas peuvent être ignoré en ajoutant un délai approprié pour ralentir certains chemins dans un circuit. La troisième :

certain aléas sont tolérés si ils ne sont pas préjudiciables.

3.3.2 Classes de circuits asynchrones

Au niveau porte, les circuits asynchrones peuvent être classifiés comme auto-temporisés (en anglais *self-timed*), indépendants de la vitesse (en anglais *speed-independent*) ou insensibles aux délais (en anglais *delay-insensitive*) dépendants des hypothèses temporelles. La **FIGURE 3.12** permet d'illustrer cette classification.

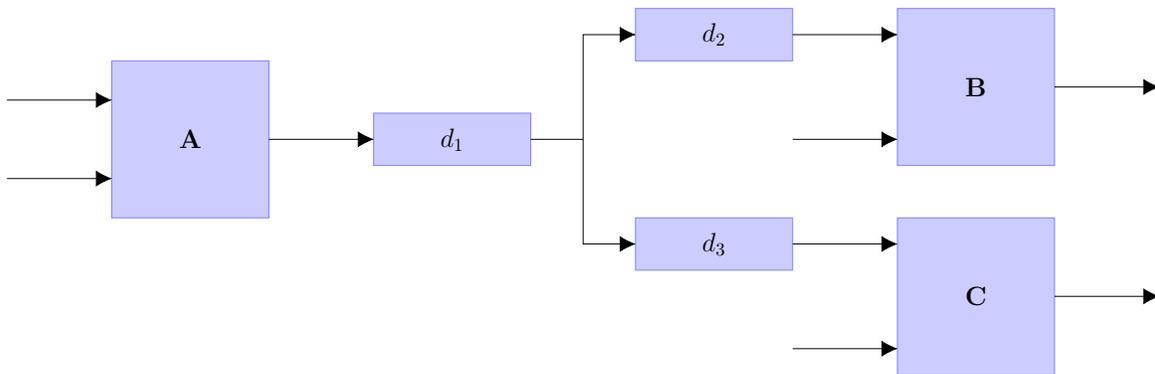


FIGURE 3.12 - Classes de circuits asynchrones.

Un circuit indépendant de la vitesse est un circuit qui fonctionne correctement en supposant des délais positifs, bornés mais inconnus dans les portes mais des fils parfaits (délais nuls). En référence avec la **FIGURE 3.12**, cela signifie des délais arbitraires d_A , d_B et d_C mais $d_1 = d_2 = d_3 = 0$. Supposer des fils sans délai n'est pas très réaliste. En supposant des valeurs arbitraires pour d_1 et d_2 et en supposant $d_2 = d_3$, le circuit est toujours *speed-independent*.

Un circuit qui fonctionne correctement avec un délai dans les fils positif, borné mais inconnu (également dans les portes) est insensible aux délais. Sur la **FIGURE 3.12**, cela veut dire des valeurs arbitraires de d_A , d_B , d_C , d_1 , d_2 et d_3 . De tels circuits sont extrêmement robustes. Cette classe est plutôt petite. Seulement les circuits composés de porte de Muller et d'inverseurs peuvent être *delay-insensitive* (e.g. circuit pipeline

de Muller).

Les circuits qui sont insensibles aux délais avec l'exception de $d_2 = d_3$ sont appelés quasi-insensible aux délais. Un fil où les transitions du signal s'effectuent en même temps sont appelées isochroniques. Ces fourches peuvent se trouver dans les implémentations au niveau porte où les concepteurs peuvent contrôler le délai dans les fils.

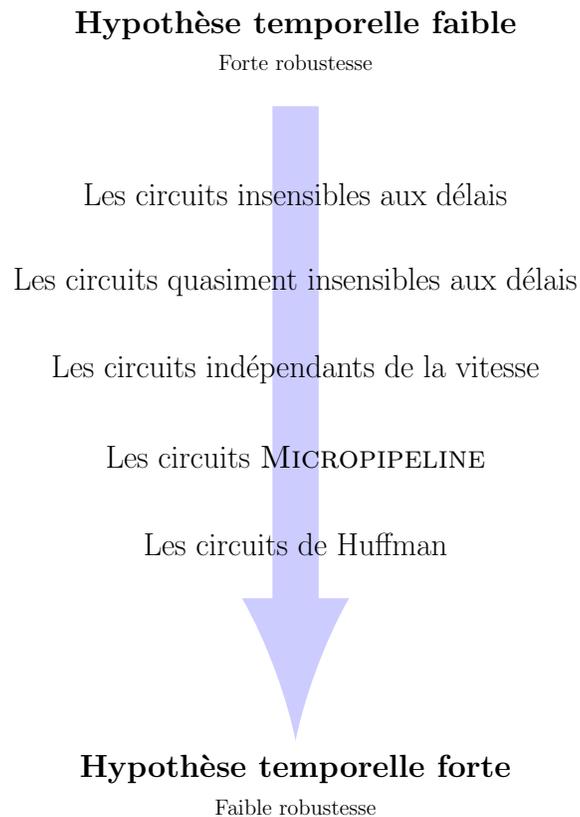


FIGURE 3.13 - Terminologie des classes de circuits asynchrones.

Le terme de circuits temporisés est souvent utilisé pour désigner des circuits asynchrones. Il désigne les circuits dont le séquençement est déterminé par des signaux internes aux circuits plutôt que par un signal d'horloge. La terminologie utilisée pour qualifier les circuits asynchrones est illustrée en **FIGURE 3.13**.

3.3.2.1 Circuits insensibles aux délais

Cette classe de circuit ne comporte aucune hypothèse temporelle. En d'autres termes, le fonctionnement des circuits est correct indépendamment des délais introduits par les fils ou les éléments logiques. Cela signifie qu'ils sont basés sur un modèle de délais pour les fils et les éléments qui sont non-bornés. Ceci impose au récepteur d'un signal de toujours informer l'expéditeur que l'information a été reçue. Les circuits récepteurs doivent être capables de détecter la réception d'une donnée et/ou la fin de son traitement. Les circuits émetteurs quand à eux doivent attendre un compte rendu avant d'émettre une nouvelle donnée.

3.3.2.2 Circuits quasi-insensibles aux délais

Cette classe de circuit adopte le même modèle de délai non-borné mais y ajoute la notion de fourche isochrone (en anglais *isochronic fork*). On appelle fourche, un fil qui connecte un expéditeur unique à deux récepteurs. On la qualifie d'isochrone lorsqu'on suppose que les délais entre l'expéditeur et les deux récepteurs sont identiques. Elle résout le problème d'utilisation de portes logiques à une seule sortie. La fourche isochrone rend les circuits insensibles aux délais réalisables pour obtenir des circuits quasi insensibles aux délais avec des portes logiques classiques [28]. Les circuits quasi insensibles aux délais se caractérisent par l'adoption d'un modèle de délais pour les connexions qui est de type non borné avec en plus l'hypothèse de fourche isochrone et un modèle de type porte simple pour les composants élémentaires des circuits. Les circuits quasi insensibles aux délais sont donc implémentables avec des cellules standard telles qu'on les utilise pour la conception de circuits synchrones.

3.3.2.3 Circuits indépendants de la vitesse

Les circuits indépendants de la vitesse font l'hypothèse que les délais dans les fils sont négligeables tout en conservant le modèle non borné pour les délais dans les portes.

La fourche isochrone est une façon équivalente de formuler le modèle indépendant de la vitesse.

3.3.2.4 Circuits de Huffman

Les circuits de cette classe utilisent un modèle de délais identique aux circuits synchrones. Ils supposent que les délais dans tous les éléments du circuit et les connexions sont bornés ou même de valeurs connues. Les hypothèses temporelles sont donc du même ordre que la conception de circuits synchrones. Leur conception est basée sur l'analyse des délais dans tous les chemins et boucles de façon à dimensionner les signaux de contrôle locaux qui s'apparente à des horloges locales.

3.3.2.5 Micropipeline

La technique de Micropipeline a été introduite par I. Sutherland [47]. Les circuits sont composés de parties contrôles insensibles aux délais qui commandent des chemins de données conçus en utilisant le modèle de délai borné. La structure de base est le contrôle d'une queue de type FIFO. Elle se compose d'éléments identiques connectés tête-bêche. Le circuit réagit à des transitions de signaux et non pas à des états (protocole deux phase). En supposant que toutes les portes de Muller ont été initialisées à zéro, l'environnement de gauche **FIGURE 3.14** peut commencer le protocole de communication. Considérons l'élément i , $C[i]$. Il propage un 1 à partir de son prédécesseur $C[i - 1]$, seulement si $C[i + 1]$ est à 0 (c.f. **FIGURE 3.14**). Peu importe si le protocole 2-phases ou 4-phases est utilisé, cela reste le même circuit.

La motivation première pour le développement de cette classe de circuits était de permettre un pipeline élastique. En effet, le nombre de données présentes dans le circuit peut être variable, les données progressant dans le circuit aussi loin que possible en fonction du nombre d'étages disponibles ou vides.

Le problème d'aléas a été écarté en ajoutant des retards sur les signaux de contrôle.

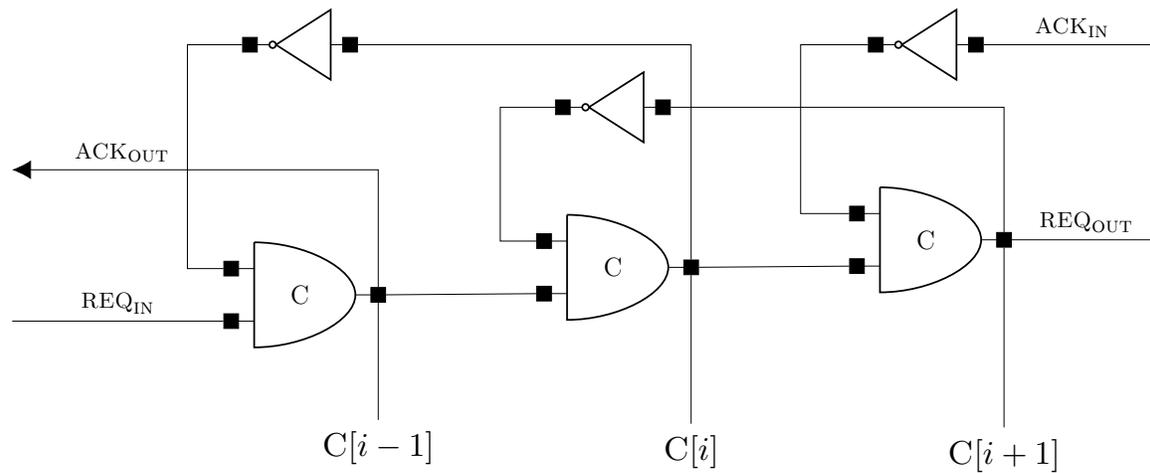


FIGURE 3.14 - Circuit Micropipeline

Cela permet en fait de se ramener à un fonctionnement en temps discret, avec l'autorisation de la mémorisation de données seulement lorsqu'elles sont stables (à la fin des retards). Les délais étant de durée fixe, c'est une approche pire cas. On ne peut pas tirer de la variation dynamique de la chaîne critique des opérateurs de traitement. Il est possible de s'affranchir de cette contrainte en utilisant des registres et des opérateurs combinatoires capables de générer leur propre signal de fin de calcul sans avoir recours à des délais fixes. Les délais sont remplacés par des délais variables implémentés dans les registres et la logique combinatoire. Ainsi, le temps de calcul peut varier en fonction des données. Le circuit n'utilise le modèle de délai borné et le circuit peut être rendu indépendant de la vitesse, car la seule hypothèse est que le signal de contrôle succède aux données.

Conclusion

Ce chapitre a permis d'expliquer les mécanismes de la logique asynchrone et de souligner les différences avec la logique synchrone.

L'asynchrone étant une logique événementielle, le coupler à l'échantillonnage par

traversée de niveaux peut permettre d'obtenir une architecture faible consommation.
Cette hypothèse sera vérifiée dans le chapitre suivant.

Chapitre 4

Validation matérielle

Introduction

Les résultats de simulation ont permis de démontrer que l'échantillonnage par traversée de niveaux permet de réduire de façon drastique le nombre d'échantillons à traiter tout en préservant l'information pertinente.

Une architecture intéressante a été proposée pour cette stratégie d'échantillonnage mais fonctionne uniquement pour des niveaux uniformément espacés. Ce chapitre aura pour principal objectif de valider la nouvelle architecture proposée mais également de confirmer la véracité des résultats de simulation.

4.1 Architecture du système de reconnaissance de formes

Le système de reconnaissance de formes à mettre en place est destiné à devenir un système implantable. Il se compose de plusieurs sous-systèmes qui a été précédemment illustré sur la **FIGURE 1**. Cet implant se compose d'une partie capteur (et acquisition), d'une partie de traitement (et mémoire) et d'une partie commande qui réalise l'interface entre la partie traitement et l'actionneur.

L'implémentation de la totalité du système ne sera pas envisagée. Ce type de système, représenté en **FIGURE 4.1**, qui sera par la suite réalisé matériellement sur circuit logique programmable, se compose d'une partie acquisition et traitement (et mémoire). D'une part l'acquisition est réalisée au moyen d'un échantillonnage par traversée de niveaux et d'autre part la partie traitement est dédiée à la reconnaissance de formes. Les formes à reconnaître, appelées également les formes d'intérêt sont *enfouies* dans le signal (temporel) issu du capteur accélérométrique [29]. Leur détection doit être la plus efficace possible en termes de coût énergétique et de performance. La partie capteur ne sera pas implémentée dans le cadre de la validation matérielle car les signaux issus des capteurs v_g , enregistrés sur des patients lors du protocole expérimental, seront directement injectés à la partie acquisition.

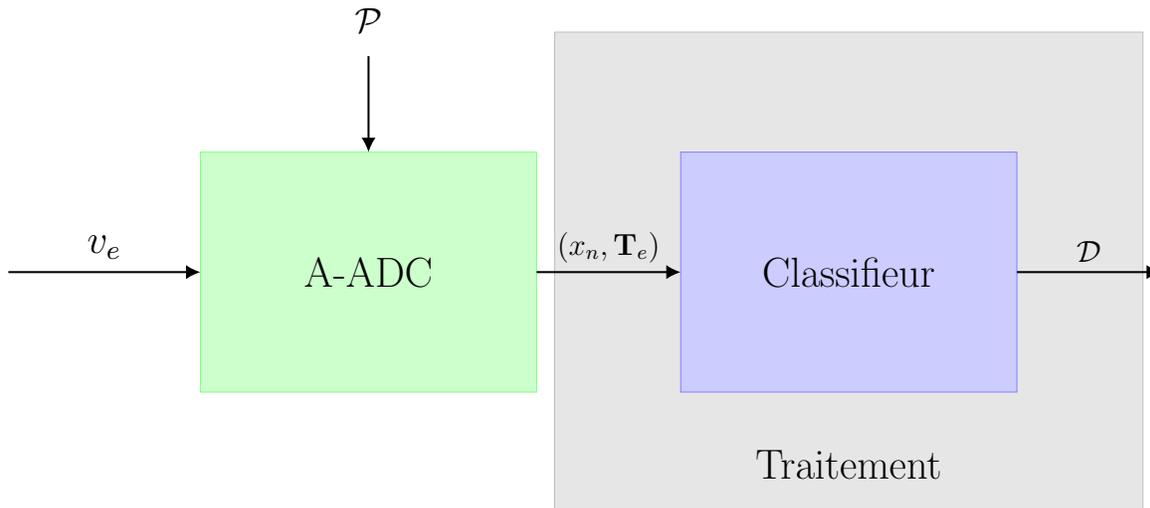


FIGURE 4.1 - Système implémenté matériellement.

Le module traitement du système en **FIGURE 4.1** sera dédié à la reconnaissance des formes issus du capteur et émis en flux de données. Le résultat de cette détection est un signal logique v_d .

4.2 Conditions expérimentales

L'acquisition des données est effectuée au moyen de l'implémentation matérielle d'une méthode d'échantillonnage par traversée de niveaux. Les niveaux de quantification ont été précédemment déterminés lors des résultats de simulation sur les signaux du protocole expérimental.

Les signaux issues du protocole expérimental ont été discrétisés au moyen d'un échantillonnage de Nyquist. Ces données étant discrètes, elles ne peuvent évidemment pas être directement transmises au module d'acquisition car elles doivent être de nature analogique. Ces données sont injectées à la partie acquisition par l'intermédiaire d'un générateur de signaux qui a la possibilité d'émettre des données *arbitraires* stockées dans sa mémoire.

Le dispositif expérimental pour la validation matérielle est représentée en **FIGURE 4.2**. Il comporte le générateur de signaux qui émet la version analogique v_e des données issues du protocole expérimental, l'échantillonneur par traversée de niveaux, située dans la partie CNA & Détecteur de niveaux ainsi que dans la partie FPGA dans lequel est également implémenté le détecteur de formes.

Les formes d'intérêt discrétisées, issues du protocole expérimental v_g , sont sauvegardées en mémoire du générateur pour être transmises à la sortie analogique de l'appareil v_e . La partie acquisition est basée sur le convertisseur analogique-numérique de E. Allier [1] et qui fera l'objet d'une description. La partie numérique du convertisseur est implémentée sur un circuit logique programmable de type FPGA (Field-Programmable Gate Array) [6] [23] qui fait partie d'une carte de développement. La partie analogique est implémentée sur un circuit spécifique et connectée à cette carte. Les signaux v_u et dt sont les échantillons d'amplitude et de temps issus de l'échantillonnage par traversée de niveaux et qui sont ensuite utilisés par la module de reconnaissance de formes, dont v_d est le résultat de cette détection.

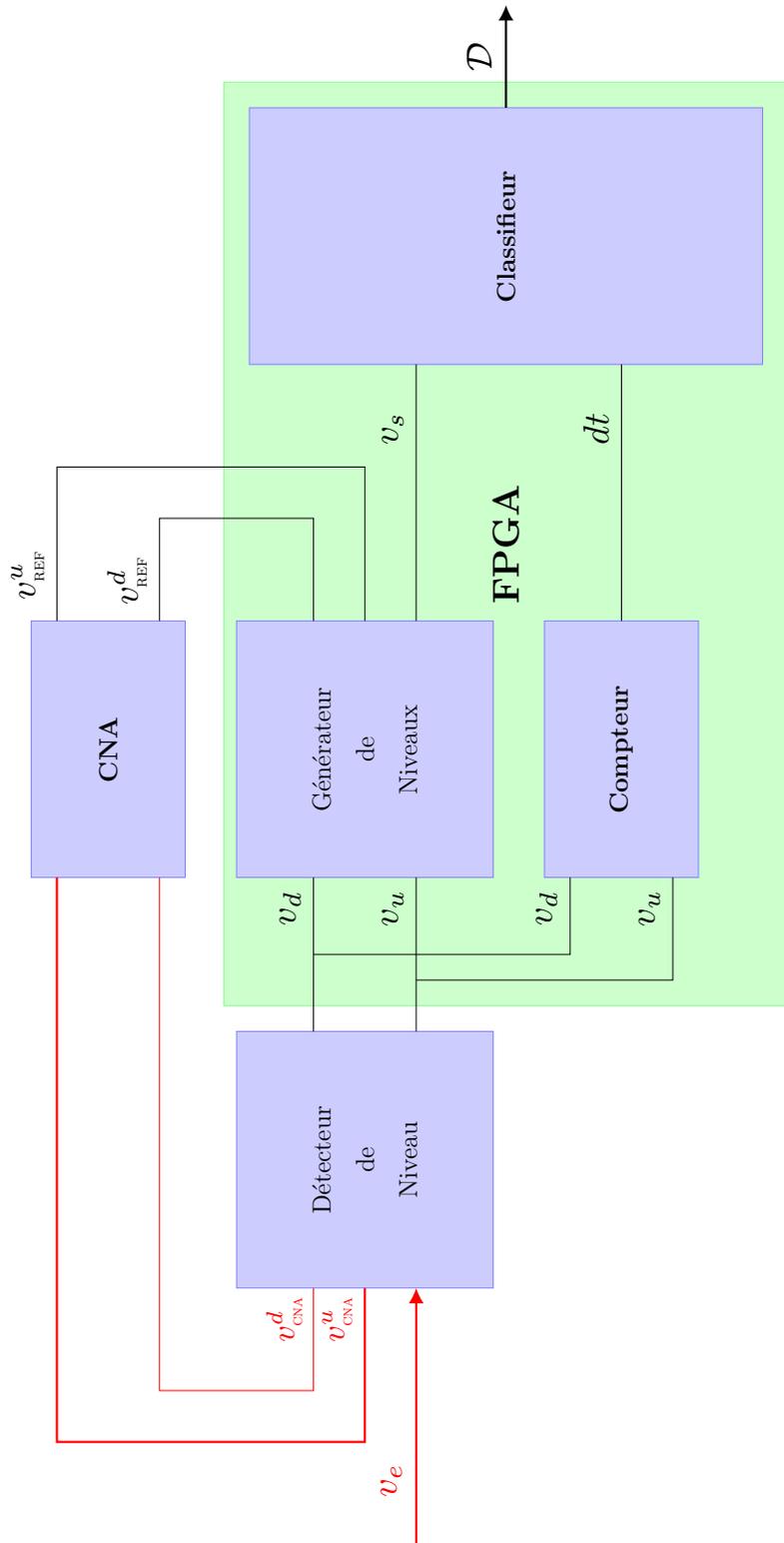


FIGURE 4.2 - Dispositif expérimental pour la validation matérielle du système de reconnaissance de formes.

4.3 Convertisseur par traversée de niveaux

Le convertisseur analogique-numérique, basée sur la traversée de niveaux, utilisé pour l'acquisition des données est basée sur la réalisation de E. Allier [1]. Ce convertisseur est nommé par l'acronyme A-ADC (Asynchronous ADC) et qui correspond à un convertisseur analogique-numérique par traversée de niveaux qui fonctionne de manière asynchrone [48]. L'architecture qui est proposée, illustrée sur la **FIGURE 4.3**, fonctionne uniquement avec des niveaux de quantification uniformément disposés le long de la dynamique du signal à numériser v_e .

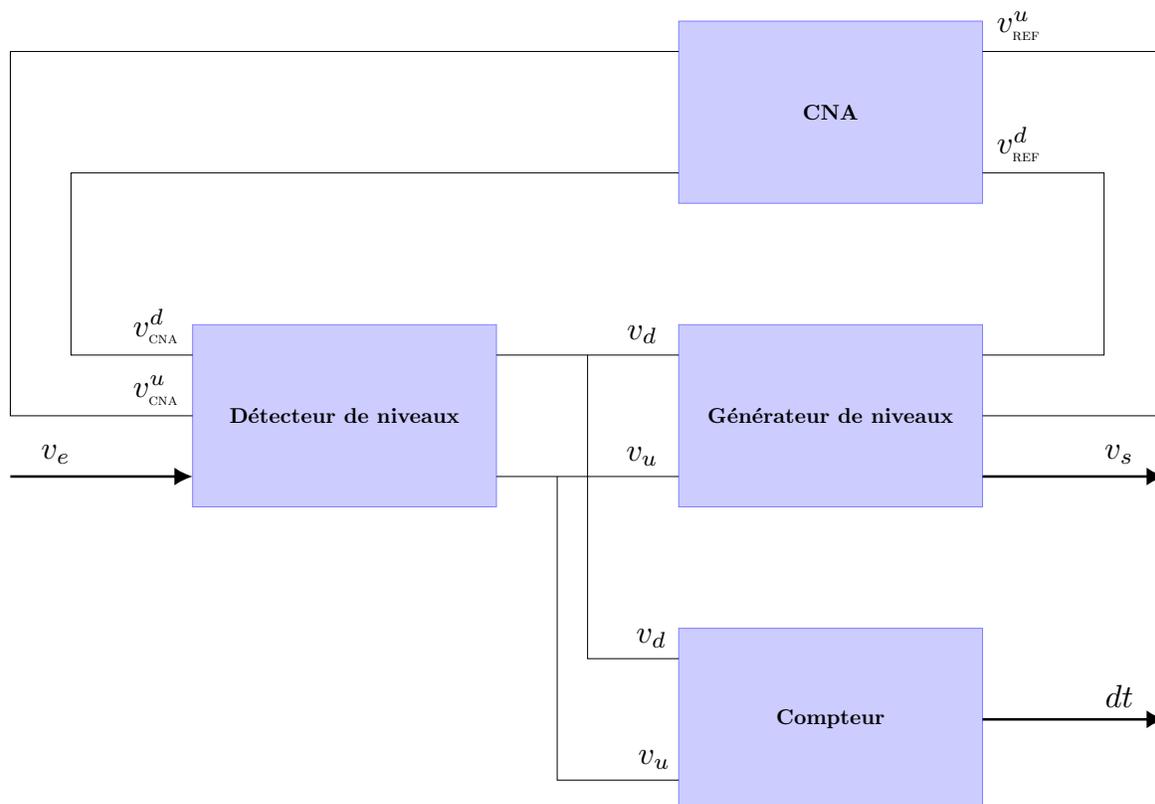


FIGURE 4.3 - Architecture *générique* du convertisseur analogique-numérique par traversée de niveaux qui génère en sortie un couple amplitude-temps (v_s, dt) .

L'architecture de l'A-ADC se compose de quatre modules : un module de détection

de la traversée d'un niveau (de quantification), un module de génération de niveau, d'un compteur et d'un convertisseur numérique analogique qui effectue la conversion des niveaux proposés par le générateur de niveaux. Chaque module de l'architecture de l'A-ADC sera détaillé par la suite.

4.3.1 Détecteur de traversée de niveau

Ce module a pour fonction de déterminer quand le signal d'entrée à numériser, émis par le générateur, traverse un des niveaux de quantification. Ce détecteur détermine la direction de la traversée d'un niveau : ascendante ou descendante. En considérant un seul niveau v_{CNA} , la détection de la traversée est possible mais l'information de direction n'est pas accessible.

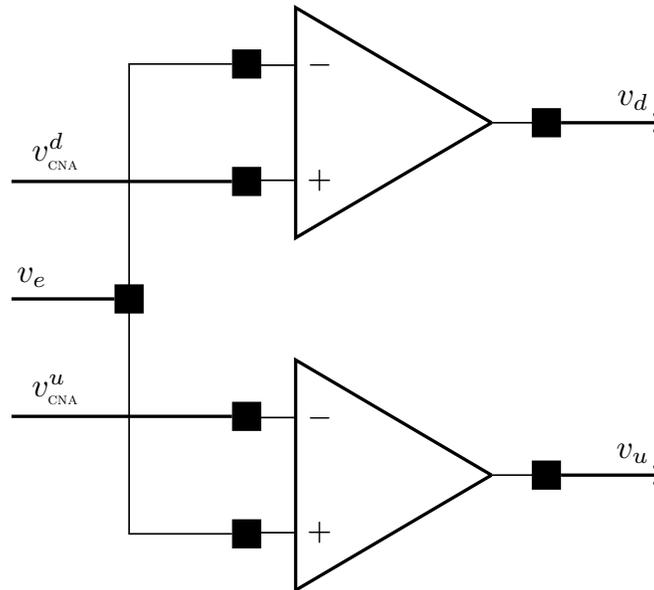


FIGURE 4.4 - Détecteur de traversée de niveaux composé d'un réseau d'amplificateur opérationnel.

Pour cela, il est nécessaire d'utiliser deux niveaux de quantification au lieu d'un seul. Dans la même configuration, on dispose ces deux niveaux de part et d'autre (dessus et dessous) du niveau initial qui sont notés v_{CNA}^u et v_{CNA}^d et qui sont respectivement le niveau

supérieur et inférieur à v_{CNA} . Ainsi, pour un ensemble constitué de plusieurs niveaux de quantification, il faut associer un couple de niveaux et qui seront utilisés par le module de détection de traversée de niveau : v_u pour ascendante et v_d pour descendante. Le détecteur de traversée de niveau a pour entrée le signal analogique à numériser qui est émit par le générateur de signal ainsi que les couples de niveaux composé de v_{CNA}^u et v_{CNA}^d . Sa sortie fournit les informations de la direction de la traversée d'un niveau qui sera utilisée par le module de génération de niveau : v_d et v_u .

L'information de direction de traversée est obtenue au moyen d'un réseau de deux amplificateurs différentiels qui fonctionnent en mode comparateur. Un comparateur permet de générer l'information de traversée ascendante et l'autre génère l'information de traversée descendante. Le comparateur de traversée ascendante a son entrée non-inverseuse connectée au signal analogique provenant du générateur et son entrée inverseuse reliée à v_{CNA}^u . De cette façon, quand le signal d'entrée v_e est supérieur à v_{CNA}^u alors la sortie du comparateur, nommée v_u , est au niveau logique haut. À l'inverse, le comparateur de traversée descendante a son entrée inverseuse connectée au signal analogique provenant du générateur et son entrée non-inverseuse reliée à v_{CNA}^d . De cette façon, quand le signal d'entrée v_e est inférieur à v_{CNA}^d alors la sortie du comparateur, nommée v_d , est au niveau logique haut.

Les couples de niveaux de références qui sont reliés aux entrées des comparateurs, v_{CNA}^u et v_{CNA}^d , sont de nature analogique qui sont fournies par un convertisseur numérique-analogique.

4.3.2 Convertisseur analogique-numérique

Le convertisseur analogique-numérique réalise l'interface entre le module de détection de la traversée d'un niveau et le module de génération de niveau. Ce dernier fournit une version numérique des couples de niveaux de référence nécessaires pour obtenir l'information de direction de traversée. Le couple de niveaux de référence est converti dans

le domaine analogique pour permettre au module de détection de traversée de niveau de pouvoir comparer ces niveaux au signal d'entrée analogique.

Le convertisseur doit pouvoir fournir les niveaux de référence de manière simultanée. Il doit ainsi comporter une sortie pour la détection d'une traversée de niveau ascendante et une autre pour permettre de détecter une traversée de niveau descendante.

Le couple de niveaux de référence est généré par le module de génération de niveaux. Ces niveaux de référence sont actualisés par le module de génération de niveaux après chaque activation (niveau logique haut) d'un des signaux de direction de traversée v_u ou v_d .

4.3.3 Générateur de niveaux

Les signaux de direction de traversée de niveaux permettent d'actualiser de façon dynamique le couple de niveaux de référence utilisés pour la comparaison avec le signal d'entrée. Dans le cas où v_u est au niveau logique haut, cela signifie que la traversée du niveau de référence est ascendante et qu'il est nécessaire de réactualiser les niveaux de références à des valeurs supérieures au signal d'entrée pour poursuivre la conversion analogique-numérique.

Par contre, si c'est v_d qui est au niveau logique haut, cela signifie que la traversée du niveau de référence est descendante et qu'il est nécessaire de réactualiser les niveaux de référence à des valeurs supérieures au signal d'entrée v_e (condition de poursuite).

Dans le cas de niveaux uniformément répartis le long de la dynamique du signal d'entrée v_e , chaque niveau est espacé par un pas de quantification constant (quantum). Chaque niveau est défini par un multiple du quantum. Dans l'architecture de l'A-ADC, E. Allier [1] propose de générer les niveaux de référence au moyen d'un simple compteur. Un niveau logique haut sur v_u ou v_d va respectivement incrémenter ou décrémenter le compteur. Le nouveau niveau de référence sera défini par le produit entre la valeur (courante) du compteur et le quantum. Dans le cas où le compteur a atteint ses limites,

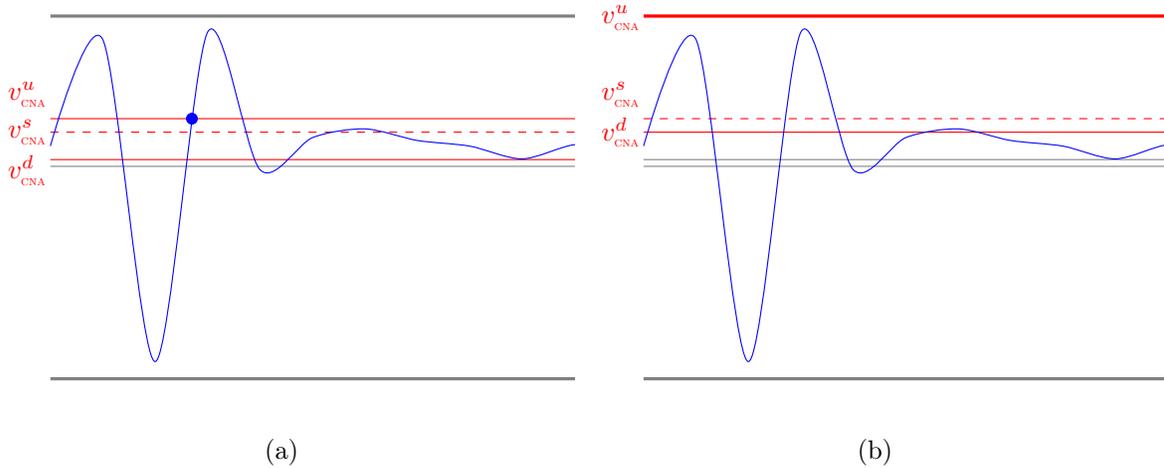


FIGURE 4.5 - (a) Traversée d'un niveau de référence haut (b) Actualisation des niveaux de référence pour poursuivre la conversion.

alors il n'y a pas de réactualisation des niveaux de référence.

L'utilisation d'un compteur pour la génération des niveaux de référence est simple et efficace en termes de ressources matérielles et de vitesse. Cependant, cette architecture ne fonctionne que pour des niveaux uniformément espacés et n'est donc par conséquent pas *générique*. Une modification du générateur de niveaux sera effectuée pour permettre de faire fonctionner le convertisseur analogique-numérique avec des niveaux pas nécessairement espacés uniformément (quantum variable).

La **FIGURE 4.5** illustre le procédé de génération de niveaux uniformément répartis. Les niveaux v_{REF}^u et v_{REF}^d sont choisis d'être positionnés à équidistance du niveau de référence v_{REF} . Sur la **FIGURE 4.5** (a), il y a un franchissement du niveau v_{REF}^u qui provoque un niveau logique haut sur v_u et qui mène à un rafraîchissement des niveaux de référence. La **FIGURE 4.5** (b) illustre les niveaux de référence mis-à-jour nécessaire pour *poursuivre* le signal v_e . Il faut noter que sur la **FIGURE 4.5** (b), le niveau de référence v_{REF}^d est le niveau v_{REF}^u de la capture précédente. Le respect de cette condition de position des niveaux est nécessaire pour assurer le bon fonctionnement de l'A-ADC. En effet, si ce critère n'est pas validé, en présence d'un fort niveau de bruit le circuit

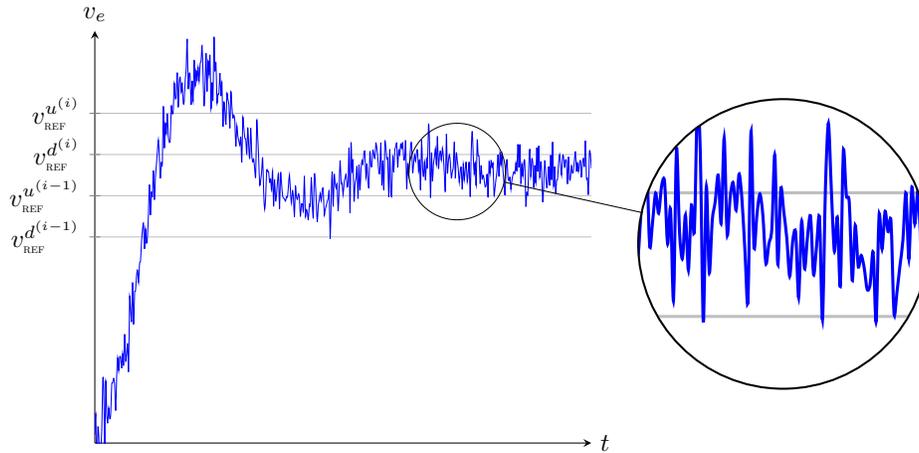


FIGURE 4.6 - Échantillonnage d'un signal bruité avec la présence d'un intervalle entre les niveaux de références $v_{\text{REF}}^{d(i)}$ et $v_{\text{REF}}^{u(i-1)}$.

serait instable. Pour illustrer cette affirmation, on considère le signal corrompu par un bruit gaussien de la **FIGURE 4.6**. Il y a un intervalle entre le niveau de référence v_{REF}^u de l'échantillon $i - 1$ et v_{REF}^d de l'échantillon i . Le fort bruit va générer un grand nombre de transitions (instabilité) et de rafraîchissements de niveaux dans un faible interval de temps. Le but dans la conception du système étant de minimiser l'activité du circuit, il n'est pas recommandable de laisser un interval entre $v_{\text{REF}}^{d(i)}$ et $v_{\text{REF}}^{u(i-1)}$.

4.3.4 Compteur

Dans l'échantillonnage par traversée de niveaux, un échantillon est capturé seulement quand le signal d'entrée à convertir traverse un des niveaux de quantification. Contrairement à l'échantillonnage de Nyquist, les échantillons ne sont pas régulièrement espacés dans le temps car cela dépend des variations du signal d'entrée. Par conséquent, il est nécessaire d'enregistrer (quantifier) le temps écoulé depuis le dernier échantillon capturé. Un compteur de période T_c est utilisé pour réaliser cette quantification temporelle.

Le compteur s'incrémente au rythme des fronts montants des signaux d'informa-

tion de traversée v_u ou v_d . Lors d'un débordement, le compteur emet sa valeur avant d'être réinitialisé pour permettre un nouveau cycle de comptage et ne pas perdre cette information temporelle.

4.4 Convertisseur par traversée de niveaux modifié

Les niveaux de quantification ont été précédemment déterminés lors des résultats de simulation sur les signaux du protocole expérimental. Selon les résultats, la répartition des seuils suivent une loi de distribution fortement non-uniforme. Le quantum n'étant pas fixe, il n'est pas possible d'utiliser un compteur pour générer les niveaux de quantification. Par conséquent, pour faire fonctionner l'A-ADC proposé par E. Allier [1] il faut modifier l'architecture du générateur de niveaux.

Une possibilité consiste à simplement utiliser une table de correspondance ou LUT (Look-Up Table) qui doit être préalablement initialisée avec les niveaux de référence. Chaque emplacement mémoire est accessible au moyen d'une adresse où les niveaux sont stockés par valeur croissante. Pour faciliter l'implémentation matérielle, les niveaux de références sont stockés dans deux tables de correspondance distinctes ce qui permet d'utiliser la même variable d'adresse.

En considérant une variable (pointeur) qui stocke l'adresse de l'emplacement mémoire courant, un niveau logique haut sur v_u va incrémenter cette variable et va permettre d'accéder à l'adresse du niveau de référence de valeur supérieure au niveau de référence courant. À l'inverse, un niveau logique haut sur v_d va décrémenter cette variable et va permettre d'accéder à l'emplacement du niveau de référence de valeur inférieure au niveau de référence courant. Aux conditions aux limites, si le niveau de référence courant est le plus grand ou le plus faible, en d'autres termes, si le pointeur se situe aux limites de l'espace mémoire réservé pour la table de correspondance, un niveau logique haut sur v_u ou v_d n'aura aucun effet.

La modification de l'architecture du générateur de niveau par une table de correspondance permet de faire fonctionner l'A-ADC quelque soit la répartition des niveaux de référence. Cette nouvelle architecture permet un fonctionnement *générique*.

4.5 Détecteur de formes

L'algorithme de reconnaissance de formes consiste à détecter des séquences composées de plusieurs formes. Une séquence est détectée si les formes qu'elle contient sont reconnues, c'est-à-dire que leurs primitives sont détectées. Il est supposé que les formes d'intérêt qui constituent une séquence sont suffisamment espacées pour être indépendantes les unes des autres. De plus, l'utilisation de cet algorithme nécessite que les niveaux utilisés pour la détection de la première primitive soient d'amplitude supérieure aux niveaux utilisés pour la détection de la seconde primitive.

$$T_{\mathcal{A}} = \begin{pmatrix} t_{11} & t_{12} & 0 & 0 & 0 \\ t_{21} & t_{22} & t_{23} & t_{24} & 0 \\ 0 & 0 & t_{33} & 0 & t_{35} \\ 0 & 0 & 0 & t_{44} & t_{45} \\ t_{51} & t_{52} & 0 & 0 & 0 \end{pmatrix}$$

FIGURE 4.7 - Matrice d'incidence de l'automate \mathcal{A} .

L'algorithme de reconnaissance de formes est implémenté au moyen d'un automate déterministe à états finis \mathcal{A} à six états représenté en **FIGURE 2.35**.

On considère que chaque forme d'intérêt se compose de deux primitives. Sachant que les formes d'intérêt ont été approximées en deux classes et qu'un niveau est dédié à la détection de ces primitives, la partition ROC se compose de quatre niveaux. Le fonctionnement de l'algorithme nécessite la spécification de deux niveaux de silences dont leur amplitude est inférieure aux niveaux de détection. Le déclenchement de l'algorithme est basée sur la détection de la transition d'un niveau de silence vers un niveau

de détection.

1. État S_1

- t_{11} – L'échantillon courant est sur un niveau de silence et le temps de silence n'a pas été validé ou l'échantillon courant est sur un niveau de détection.
- t_{12} – L'échantillon courant est sur un niveau de silence et le temps de silence a été validé.

2. État S_2

- t_{21} – L'échantillon courant est sur un niveau de détection et la fenêtre a été dépassée ou l'échantillon courant est sur un niveau de silence et le temps de silence n'a pas été validé.
- t_{22} – L'échantillon courant est sur un niveau de silence et la fenêtre n'a pas été dépassée ou l'échantillon courant est sur un niveau de silence et la fenêtre a été dépassée mais le silence pré-séquence a été validé.
- t_{23} – L'échantillon courant est sur un niveau de détection positif et la fenêtre n'a pas été dépassée.
- t_{24} – L'échantillon courant est sur un niveau de détection négatif et la fenêtre n'a pas été dépassée.

3. État S_3

- t_{33} – La fenêtre n'a pas été dépassée.
- t_{35} – La fenêtre a été dépassée.

4. État S_4

- t_{44} – La fenêtre n'a pas été dépassée.
- t_{45} – La fenêtre a été dépassée.

5. État S_5

- t_{51} – Le temps de silence précédent la forme suivante n'est pas validé, on attend la validation d'un silence pré-séquence.

- t_{52} – Le temps de silence précédent la forme suivante est validé, la détection de séquence se poursuit. Ou le temps de silence précédent la forme suivante n'est pas validée, la détection de séquence recommence au début.

La détection des séquences de formes s'effectue en flux de données. Le déclenchement de l'algorithme repose sur des conditions en temps et en amplitude. Le nombre de formes qui compose les séquences doit être préalablement fixé. Une forme est considérée détectée si au moins une primitive a été reconnue. Cependant, pour augmenter la sélectivité et rejeter le taux de faux positifs, un seuil peut être spécifié sur le nombre de primitive que doit comporter une forme dans une séquence.

4.6 Réalisation

La partie numérique du convertisseur analogique-numérique par traversée de niveaux qui comporte le générateur de niveaux et le compteur sont implémentés sur une carte de développement ALTERA DE1 qui comporte un FPGA de type CYCLONE II EP2C20F484. La partie analogique du convertisseur est implémentée sur une carte électronique de réalisation *ad hoc* qui a été réalisée à cet effet. Celle-ci est connectée à la carte DE1 pour permettre le fonctionnement du A-ADC. Les amplificateurs opérationnels utilisés pour la réalisation du module de détection de traversée de niveaux sont de référence LM339N. Le convertisseur numérique-analogique qui effectue l'interface entre le générateur de niveaux et le détecteur de traversée de niveaux est un AD7302. Ce convertisseur comporte deux entrées afin de permettre la conversion de deux signaux numériques distincts définis sur 8 bits. La mise en activité du convertisseur s'effectue au moyen d'une requête (asynchrone). Cette propriété intéressante va permettre une réalisation asynchrone du convertisseur analogique-numérique qui est la principale vertu du convertisseur de E. Allier [1]. L'opération de conversion nécessite une signalisation particulière qui est réalisée au moyen d'un automate à états finis et qui est implémentée

sur le FPGA.

Comme il l'a été spécifié précédemment, un générateur de signal est utilisé pour fournir le signal d'entrée au module d'acquisition qui est le convertisseur analogique-numérique par traversée de niveaux. Étant donné que la mémoire du générateur utilisée est insuffisante pour stocker la totalité des signaux issus du protocole expérimental, seulement des portions du signal seront utilisées (fenêtres). Ces fenêtres comprendront les formes d'intérêt et les formes à rejeter. Dans notre cas, les formes d'intérêt sont des séquences composées de plusieurs formes qui contiennent l'information utile.

4.7 Résultats

L'objectif de ce dispositif expérimental est de déterminer l'activité du circuit du module d'acquisition et des performances du module de reconnaissance de formes pour permettre de valider les résultats de simulation.

On considère un ensemble de fenêtres constituées des formes d'intérêt et des formes à rejeter. Pour déterminer l'activité du circuit on détermine, pour chaque fenêtre, le nombre de front montant de la commande d'écriture REQ du CNA. Le nombre de fronts dans une fenêtre est normalisée par le nombre de commandes d'écritures pour un échantillonnage de Nyquist.



FIGURE 4.8 - Signaux obtenus au moyen du convertisseur analogique-numérique asynchrone.

La **FIGURE 4.8** correspond à une fenêtre d'une forme d'intérêt. Le signal analogique v_e est émis par le générateur de signaux de la **FIGURE 4.2**. Le signal v_{CNA} correspond de référence v_{CNA}^u ou v_{CNA}^d émis par la convertisseur numérique-analogique et REQ est la commande d'écriture émise par le module générateur de niveaux de la **FIGURE 4.3** vers le convertisseur numérique-analogique pour la réactualisation des niveaux de références.

Le signal d'écriture REQ peut être considéré comme l'image de l'activité électrique du circuit. Chaque front montant indique que le signal d'entrée v_e a traversé un niveau de référence v_{CNA}^u ou v_{CNA}^d . Entre chaque commande, le convertisseur numérique-analogique n'est pas sollicité et est donc à l'état de veille. Sur la **FIGURE 4.8** le signal v_e est quasi-statique avant la première et dernière commande de REQ et donc ne présente par d'information pertinente. Si une stratégie d'échantillonnage uniforme était utilisée, le convertisseur analogique-numérique correspondant capturerait des échantillons dans ces régions peu pertinentes ce qui aurait mené à une activité inutile et donc une perte d'énergie.

On peut observer que sur la **FIGURE 4.8**, le signal v_{CNA} possèdent de fortes similarités structurelles (forme d'onde) que le signal d'entrée v_e . Ceci permet d'illustrer le fait que la méthode de détermination des niveaux de discrétisation permet de préserver l'information pertinente avec un minimum d'échantillons et donc de condenser l'information.

Conclusion

La méthode de détermination des niveaux de quantification fournit des niveaux qui sont (fortement) non-uniformément espacés. Cette distribution non-uniforme ne permet pas d'utiliser le convertisseur analogique-numérique asynchrone (A-ADC). En effet, il ne fonctionne qu'avec des niveaux uniformément répartis. Pour permettre l'utilisation de niveaux non-uniformes, une table de correspondance remplace le module de générateur

de niveaux de l'A-ADC représenté en **FIGURE 4.3**. Ainsi, à chaque traversée de niveaux les niveaux de références v_{CNA}^u et v_{CNA}^d sont actualisés et permettent de continuer la numérisation du signal d'entrée v_e .

Les observations sur la similarité du signal d'entrée v_e et sa version analogique préalablement numérisée v_{CNA} montrent que les niveaux de référence permettent de capturer principalement les échantillons qui comportent de l'information pertinente en d'autres termes de condenser l'information.

Chapitre 5

Consommation énergétique

Introduction

La réalisation matérielle a permis de valider le fonctionnement de l'échantillonnage par traversée de niveaux répartis non-uniformément. De plus, des niveaux adaptés permettent de condenser l'information, qui se traduit par un fort taux de compression de données et de détection.

Ce chapitre a pour but d'évaluer les performances du système de reconnaissance asynchrone pour connaître les bénéfices en terme de consommation énergétique. Ces résultats seront comparés à son homologue synchrone afin de situer les bénéfices apportés par l'échantillonnage par traversée de niveaux couplé implémenté en logique asynchrone.

5.1 Architecture du système de reconnaissance de formes

Le système de reconnaissance de formes à développer doit intégrer un système implantable qui se compose d'une partie capteur (et acquisition), d'une partie de traitement (et mémoire) et d'une partie commande qui réalise l'interface entre la partie traitement et l'actionneur. Ce système a fait, dans le chapitre précédent, l'objet d'une

implémentation matérielle sur circuit logique programmable.

Le module de reconnaissance de formes, qui correspond à la partie traitement du système, fera l'objet d'une évaluation de consommation.

Cette simulation a pour objectif d'évaluer les bénéfices du système apporté par une implémentation asynchrone en termes de consommation énergétique par rapport à son équivalent synchrone. Chaque système, asynchrone et synchrone, fera l'objet d'une évaluation de consommation afin de connaître la meilleure stratégie d'implémentation.

5.1.1 Architecture asynchrone

Le système de reconnaissance de formes asynchrone, illustré sur la **FIGURE 5.1**, se compose d'un module de détection qui est cadencé au rythme d'un signal de requête. Ce signal de requête est généré par un module de contrôle asynchrone. Le module de reconnaissance de formes utilise le couple amplitude-temps qui est doit être émis par le module d'échantillonnage par traversée de niveaux. Dans cette simulation, ces données sont émises par deux sources distinctes dédiées respectivement à l'émission des échantillons d'amplitude et de temps.

5.1.2 Architecture synchrone

Le module de reconnaissance de formes qui est utilisé pour cette architecture, illustré sur la **FIGURE 5.2**, est identique à celui utilisé dans son équivalent asynchrone. Étant une architecture synchrone ce module n'est évidemment pas cadencé par un signal de requête mais un signal d'horloge. La source émet des échantillons d'amplitude au rythme de l'horloge et qui sont uniformément espacés dans le temps. Par conséquent n'étant pas possible de connecter la source d'échantillons au module de reconnaissance il faut utiliser un étage intermédiaire pour réaliser un module d'interface.

L'interface est effectuée au moyen d'un module d'interpolation linéaire. À partir des échantillons uniformément espacés émis par la source, il doit générer le couple temps-

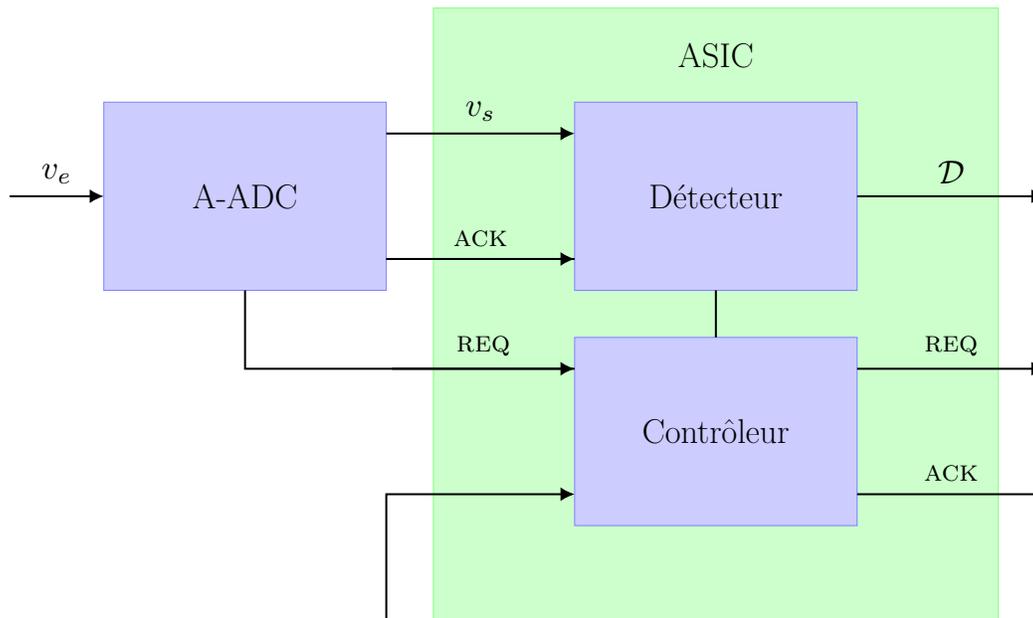


FIGURE 5.1 - Système de reconnaissance de formes asynchrone.

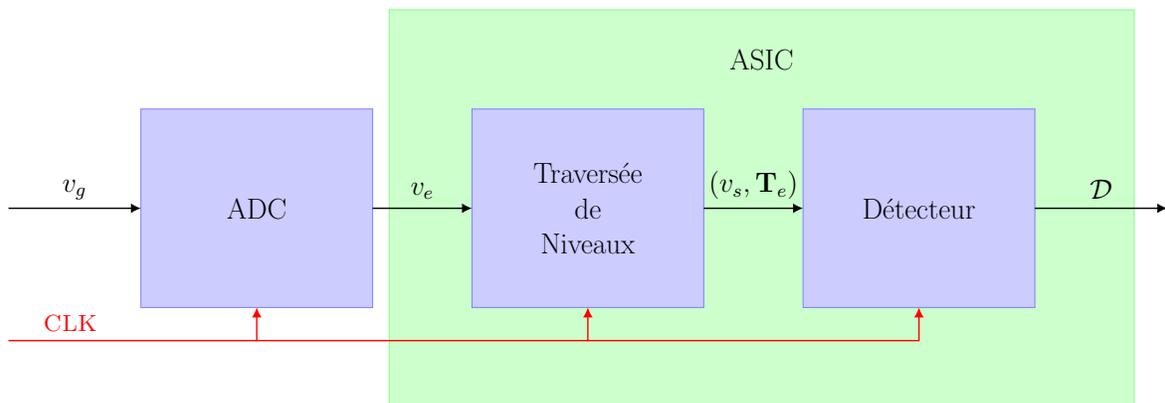


FIGURE 5.2 - Système de reconnaissance de formes synchrone.

amplitude émis par la source utilisée dans l'architecture asynchrone. En reprenant le système implémenté matériellement dans le chapitre précédent, ce module doit réaliser à la fois la fonction du générateur de signaux et du module d'acquisition.

5.2 Conditions expérimentales

Les systèmes de reconnaissance synchrone et asynchrone sont implémentés matériellement sur un ASIC (Application-Specific Integrated Circuit) [25] dans la technologie AMS de 0.35 μm . Les simulations électriques pour déterminer la consommation énergétique sont réalisées au moyen du logiciel CADENCE. Pour des raisons de rapidité de résultats, le moteur de simulation ULTRASIM est utilisé à cet effet.

Comme dans le cadre de la réalisation sur FPGA, les résultats sont obtenus en effectuant l'analyse de fenêtres contenant des formes d'intérêt (séquences) et des formes à rejeter. Pour chaque fenêtre sont obtenues les formes d'ondes du courant électrique. La tension d'alimentation du système est fixe. L'énergie est obtenue en effectuant simplement l'intégrale en fonction du temps définies aux intervalles de la fenêtre du produit du courant instantané et de la tension.

5.3 Flot de conception

Le flot de conception se constitue de deux parties : le *data path* (DP) et le *control path* (CP). Dans la première partie, on réalise la transcription en RTL, puis la synthèse *design compiler* (DC). S'ensuit la synthèse structurelle du DP qui génère la Netlist qui est combinée à celle du *control path*. Suite à la synthèse DP structurelle, l'analyse statique temporelle est réalisée au moyen de l'outil PRIMETIME (SYNOPSIS). De plus, une évaluation des chemins critiques locaux est réalisée (entre registres RTL) puis un report des valeurs de délais dans le CP est effectué. Les simulations électriques sont réalisées au moyen de ULTRASIM (CADENCE) et plus particulièrement de l'outil FAST SPICE car les circuits à évaluer comportent un grand nombre de transistors.

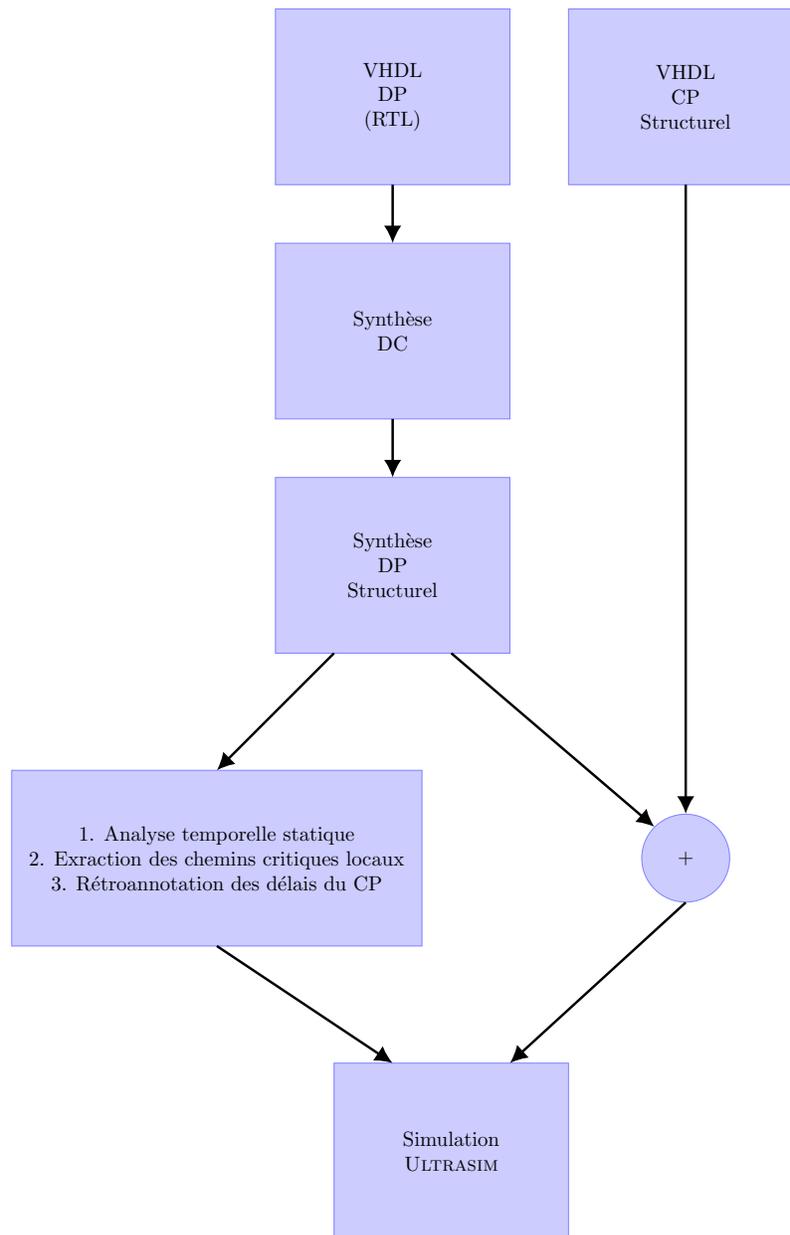


FIGURE 5.3 - Flot de conception pour la simulation électrique du circuit asynchrone et synchrone au moyen de l'outil ULTRASIM.

5.4 Résultats

Les résultats de consommation électrique obtenus par simulation sous CADENCE sont présentés pour l'approche synchrone et asynchrone.

5.4.1 Approche asynchrone

Le signal se divise en régions actives et inactives dont ces dernières sont majoritaires. Le circuit d'échantillonnage par traversée de niveaux ne produit pas d'événements durant les parties inactives du signal. Durant ces longues périodes d'inactivités, la consommation est produite par le courant de fuite qui très faible dans le cas de la technologie CMOS de 0.35 μm .

Séquence	Durée (ns)	Nombre d'échantillons NUS	Énergie (nJ)	Énergie (pAh)
1	383 793	52	6.16	0.52
2	1 146 585	50	6.72	0.57
3	545 788	52	5.10	0.43
4	453 396	48	6.33	0.53
5	626 039	56	6.31	0.53
6	635 617	56	6.72	0.57
7	600 373	50	6.15	0.52
8	671 319	46	5.47	0.46
9	799 828	50	6.18	0.52
10	572 959	48	4.99	0.42
11	778 271	62	7.50	0.63
12	778 489	50	6.05	0.51
13	361 702	54	5.90	0.50
14	543 232	60	5.23	0.44
15	744 268	54	6.07	0.51

FIGURE 5.4 - Consommation du circuit asynchrone composé du module contrôleur couplé au module de reconnaissance de formes pour une tension d'alimentation de 3.3.V.

La consommation énergétique est principalement produite par le traitement de reconnaissance de formes durant les régions actives du signal. Plusieurs stimuli ont été utilisés de différentes durées. Les mesures effectuées pour une tension de 3.3 V sont répertoriées sur la **FIGURE 5.4**. On peut remarquer que la consommation par forme

est quasiment constante quelque soit les stimuli (environ 6 nJ par forme).

5.4.2 Approche synchrone

Afin d'effectuer une comparaison cohérente, un circuit d'échantillonnage par traversée de niveaux a été implémenté pour imiter le comportement de l'A-ADC. Ceci permet d'utiliser le même algorithme de reconnaissance de formes mais dans le contexte synchrone.

Pour permettre une comparaison homogène, l'algorithme existant implémenté sur microcontrôleur de faible consommation n'a pas été utilisé. Cet algorithme utilisé pour cette application est basé sur un algorithme de corrélation qui est bien plus exigeant en terme d'énergie contrairement à la méthode présentée. Par conséquent, on s'est focalisé sur une architecture similaire au A-ADC en utilisant un ADC classique connecté à la fonction de détection de traversée de niveaux capable de fournir des échantillons non-uniforme.

Séquence	Durée (ns)	Nombre d'échantillons US	Énergie (nJ)	Énergie (pAh)
1	383 793	3837	3087	259
2	1 146 585	11465	9236	777
3	545 788	5457	3182	535

FIGURE 5.5 - Consommation du circuit synchrone composé du module contrôleur couplé au module de reconnaissance de formes pour une tension d'alimentation de 3.3.V.

L'architecture synchrone réutilise le module de détection utilisé dans l'approche asynchrone. Le module de détection de traversée de niveaux est ajouté afin de générer la séquence de données appropriée à l'entrée du module de reconnaissance de formes.

Les trois premiers stimuli ont été utilisés pour la simulation. Les mesures sont répertoriées sur la **FIGURE 5.5**. Le nombre d'échantillons traités par le circuit synchrone est d'environ cent fois plus important que dans le cas asynchrone. Avec ce cas particulier

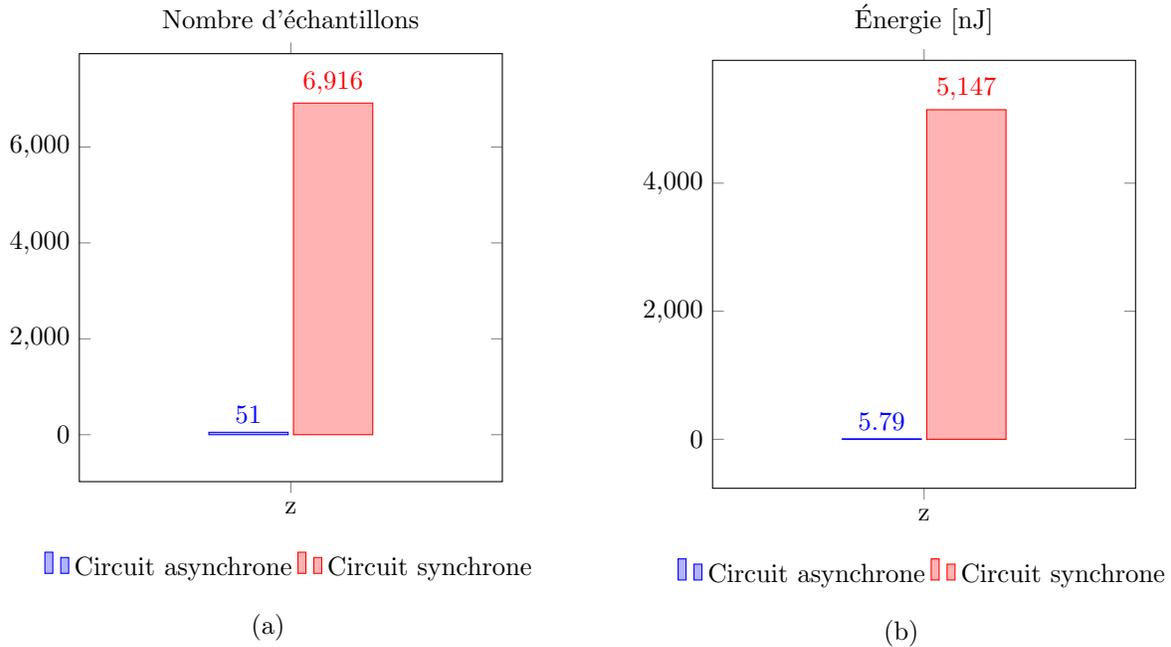


FIGURE 5.6 - (a) Comparaison du nombre d'échantillons moyen et de la (b) consommation moyenne entre le circuit asynchrone et synchrone.

d'application, il est possible de gagner de trois ordres de grandeur sur la consommation (c.f. **FIGURE 5.6**).

5.4.3 Comparaison de l'approche synchrone et asynchrone

La **FIGURE 5.6** illustre l'important écart en termes d'échantillons (a) et d'énergie (b) pour la reconnaissance de séquences de formes uniforme et non-uniforme. L'écart énergétique s'explique par le fait que le circuit synchrone effectue un traitement à chaque front d'horloge contrairement à son équivalent asynchrone qui effectue un traitement seulement en présence de nouvelles données à ses entrées.

Il peut être intéressant d'établir la relation entre le ratio d'échantillons et le ratio de consommation (respectivement **FIGURE 5.7** et **FIGURE 5.8**). On peut définir approximativement qu'un ratio d'échantillons de 100 induit un ratio de consommation de 1000.

La logique asynchrone est capable de fonctionner à faible tension, on a souhaité évaluer la réduction de consommation à différents niveaux de tension. Ceci affecte fortement la vitesse du circuit, cependant la détection de formes était fonctionnelle jusqu'à la tension de 1,5 V. Les résultats sont répertoriés dans la **FIGURE 5.9**.

Séquence	US	NUS	$\mathcal{R} = \frac{US}{NUS}$
1	3830	52	74
2	11 465	50	229
3	5453	52	105
Moyenne	6916	51	136

FIGURE 5.7 - Ratio d'échantillons entre l'approche synchrone et asynchrone.

Séquence	E_{US} [nJ]	E_{NUS} [nJ]	$\mathcal{R} = \frac{E_{US}}{E_{NUS}}$
1	3087	6.16	501
2	9236	6.38	1447
3	3120	4.84	644
Moyenne	5147	5.79	864

FIGURE 5.8 - Ratio de consommation entre l'approche synchrone et asynchrone.

Ces résultats indiquent que l'approche asynchrone reste fonctionnelle pour une grande gamme de tensions. Ceci est extrêmement important dans la perspective de conception des implants médicaux. En effet, la réduction de la consommation est un bon moyen de réduire la consommation, augmenter l'autonomie de l'implant et retarder l'opération de remplacement de la batterie de l'implant.

Tension d'alimentation (V)	3.3	3	2.5	1.5	1
Énergie (nJ)	6.20	4.72	3.00	0.18	0.0050

FIGURE 5.9 - Récapitulatif de la consommation du circuit asynchrone composé du module contrôleur couplé au module de reconnaissance de formes pour plusieurs tensions d'alimentation.

Conclusion

La combinaison de l'échantillonnage non-uniforme et de la logique asynchrone permettent de réduire drastiquement le nombre d'échantillons et d'activer le traitement seulement en présence de nouvelles données à traiter. Dans notre application, le gain est spectaculaire car il est possible d'avoir deux ordres de grandeur sur la consommation électrique seulement en échantillonnant différemment. De plus, le traitement asynchrone est, dans ce cas, plus simple et un autre ordre de grandeur peut être gagné.

Cette méthode est dédiée à l'application et de tels gains ne pourrait pas être obtenus dans d'autres cas. Cependant, on peut prétendre que cette approche pourrait être considérée cas par cas. Cette application particulière souligne le potentiel dans la réduction de consommation quand on considère une chaîne de traitement entière. Ceci montre également que déterminer le schéma d'échantillonnage approprié est la clé pour réduire drastiquement la consommation.

Conclusion

Nous avons présenté dans ce travail de thèse une approche originale pour exploiter efficacement l'échantillonnage par traversée de niveaux. Cet échantillonnage a été associé à une logique événementielle afin de réduire drastiquement la consommation énergétique. Nous nous sommes concentrés sur l'acquisition des données car cette fonction constitue le premier élément de la chaîne de traitement numérique, ainsi les performances globales dépendent majoritairement de l'efficacité de la représentation des données. Une réalisation matérielle sur FPGA a permis de valider le fonctionnement de l'approche proposée et les résultats de consommation électrique ont permis, d'une part de justifier la pertinence de la méthode proposée et d'autre part de mettre en évidence l'adéquation entre l'échantillonnage par traversée de niveaux et la logique événementielle.

Dans la première partie du mémoire, nous avons présenté différentes méthodes d'échantillonnage. L'approche qui nous a semblé la plus pertinente est l'échantillonnage par traversée de niveaux. Une méthode de génération des niveaux de discrétisation a été proposée. Elle s'inscrit dans le cadre d'une application de reconnaissance de formes. Elle permet de prendre en compte les caractéristiques structurelles des formes d'intérêt pour permettre de conserver l'information jugée pertinente. La position des niveaux est également déterminée par des indices de performances spécifiés par l'utilisateur, à savoir le taux de vrais et de faux positifs. Ces paramètres permettent à l'utilisateur d'adapter l'ensemble des niveaux de discrétisation à son application. La représenta-

tion des données non-uniformément échantillonnées permet de condenser (compresser) l'information contrairement au cas uniforme, c'est-à-dire de conserver la majorité de l'information pertinente et de rejeter celle qui ne l'est pas. La méthode de reconnaissance de formes exploite l'information d'amplitude et de temps et permet ainsi d'être très sélectif dans les segments du signal capteur à analyser et d'avoir un nombre de fenêtres à analyser très restreint. De plus, ce ne sont pas des formes isolées du sujet de reconnaissance mais des séquences composées de formes. Ces caractéristiques de sélectivité et de redondance d'information permettent de faire de la reconnaissance de formes en flux de données et de s'affranchir de l'utilisation d'un module de fenêtrage dans la chaîne de traitement. Ceci permet d'alléger l'architecture du circuit de reconnaissance de formes et ainsi de réduire la consommation électrique.

Dans la seconde partie du mémoire, la chaîne de traitement dédiée à la reconnaissance de formes a été implémentée sur FPGA. Cette réalisation matérielle a permis de valider d'une part le fonctionnement de la stratégie de reconnaissance de formes non-uniformément échantillonnées et, d'autre part la capacité de la méthode de discrétisation à condenser les données. Cela nous a permis également de souligner le bien-fondé de l'association de l'échantillonnage par traversée de niveaux avec la logique événementielle asynchrone. Étant donné que les niveaux de quantification sont adaptés à l'application visée, c'est tout naturellement qu'une réalisation ASIC a été envisagée pour estimer la consommation électrique. Ceci a permis de mettre en avant le potentiel de faible consommation apportée par la chaîne de traitement de signaux non-uniformément échantillonnés par rapport au cas classique (uniforme) puisqu'un gain de 3 ordres de grandeurs a été obtenus sur cet exemple particulier.

Perspectives

Ce travail possède de nombreuses possibilités d'améliorations et d'optimisation logicielles et matérielles. Les solutions, ci-après notées 1 à 3, envisagent de réduire le taux de faux positifs qui est basé sur la réduction de la probabilité de déclenchement intempestif de l'algorithme de reconnaissance de formes. L'approche 4 permet de prendre en compte la dérive temporelle ou l'évolution temporelle des formes d'intérêt. La perspective 5 envisage de simplifier et de s'approcher de l'architecture matérielle du système de reconnaissance final en utilisant un capteur accélérométrique qui génère un signal continu sans traitement préalable. Le cas 6 permet de prendre en compte l'amplitude et le temps des échantillons pour guider le choix des niveaux de discrétisation. L'approche 7 permet d'adapter l'architecture de reconnaissance de formes en fonction des besoins de l'utilisateur. La dernière approche envisage de minimiser la surface du circuit de reconnaissance.

(1) La méthode générique proposée dans ce mémoire pour comparer les différentes méthodes de discrétisation consiste à construire un automate fini déterministe (inférence grammaticale) à partir d'un ensemble de formes d'intérêt. Il a été démontré que le problème de reconnaissance de formes peut s'approximer avec deux classes de formes. La méthode est la suivante : deux ensembles représentatifs de chaque classe ont dans un premier temps déterminé puis un automate est construit pour chacun des ensembles par inférence grammaticale. Une stratégie intéressante pour réduire le taux de faux positifs (déclenchement intempestif) est d'attribuer un automate par primitive. Ceci permet de

ne pas poursuivre le déclenchement si la première primitive n'est pas reconnue ce qui permet d'être sélectif en terme de détection mais également de diminuer l'activité du circuit de déclenchement et naturellement la consommation électrique.

(2) À part le début (déclenchement) et la fin de la détection, la classe d'automates mise en œuvre dans la problématique de reconnaissance de formes ne prend seulement en compte que l'amplitude des échantillons. Il serait intéressant de prendre en compte la durée de séjour dans un état de l'automate pour augmenter la sélectivité de reconnaissance de formes et ainsi réduire le taux de faux positifs. L'idée est simple : si la durée dans un état est supérieure à une certaine limite, alors l'automate est réinitialisé pour permettre un autre cycle de détection. L'architecture matérielle d'échantillonnage par traversée de niveaux A-ADC permet de rendre réalisable cette stratégie. Pour optimiser cette approche, il faudrait adapter les niveaux de discrétisation à la fois en fonction de l'amplitude et du temps, ce qui nécessite d'optimiser en deux dimensions au lieu d'une seule.

(3) La méthode de génération des niveaux de discrétisation étant basée sur une architecture supervisée, la pertinence de leur placement dépend directement de la pertinence de l'ensemble des formes d'intérêt choisies. En effet, si des formes trop singulières et pas assez représentatives de la classe de formes à modéliser sont utilisées, celles-ci vont pénaliser la bonne recherche de placement des niveaux (scrutation dans une mauvaise région). Les niveaux obtenus peuvent donner lieu à une représentation des formes d'intérêt qui sont proches des formes à rejeter (marge faible) et donc pénaliser les performances de détection. Une stratégie intéressante serait d'évaluer le degré de structuralité (ou complexité) des formes. Les formes dont l'indice de structure est jugé trop faible seront rejetées. De façon similaire, les formes jugées trop complexes sont également rejetées car pas assez représentatives de la classe de formes à modéliser.

(4) Le sujet de cette thèse s'est focalisé sur les systèmes implantables dont leur caractéristique principale est d'assurer un fonctionnement sous-cutané pendant une

longue durée. L'hypothèse d'avoir des formes quasiment-fixes (variant dans une certaine gamme d'amplitude) n'est plus valide. En effet, le patient peut avoir eu une augmentation importante de poids, une blessure ou autre, ce qui peut altérer considérablement la structure des formes à reconnaître, donnant lieu à une dérive temporelle des formes. Ainsi, il faudrait définir un paramètre qui soit caractéristique (image) de la structure de la forme. Une approche simple serait de définir plusieurs partitions pour chaque classe de formes d'intérêts et de sélectionner la plus adéquate en fonction d'un certain nombre de critères sur le paramètre. On suppose dans ce cas, que chaque partition a été préalablement déterminée.

(5) Les signaux qui ont été traités proviennent d'un capteur accélérométrique. Ce capteur effectue un filtrage qui supprime une partie du bruit. Pour réduire le coût énergétique de la chaîne de traitement, il serait intéressant de s'affranchir de ce traitement en utilisant un capteur qui génère un signal physique continu brut sans cette opération de filtrage. Pour simuler un tel cas, il faut ré-évaluer l'algorithme de génération des niveaux de quantification et de reconnaissance de formes en ajoutant un bruit équivalent à celui filtré par le capteur. Cette approche permettrait également de déterminer les limites de fonctionnement (robustesse au bruit) de l'architecture de reconnaissance de formes mise en œuvre.

(6) L'efficacité de représentation des formes d'intérêt dépend de la pertinence de la position des niveaux de quantification. Les performances de la méthode de génération de niveaux est fortement dépendante du choix des primitives. Elles doivent être choisies en fonction de leur importance dans la structure de la forme auxquelles elles appartiennent. Dans le cadre de la thèse, seule l'amplitude a été utilisée en tant que critère d'importance. Cependant, la structure est définie à la fois en amplitude et en temps ce qui doit être pris en compte dans le choix des primitives. L'énergie d'une primitive peut être une grandeur pertinente qui permettrait également une plus grande robustesse de l'algorithme de détermination de niveaux avec l'ajout de bruit.

(7) Les niveaux de discrétisation ont été choisis afin de respecter un compromis entre le taux de vrais positifs et de faux positifs. Cependant, si l'on souhaite rendre la solution générique, la détermination des niveaux doit pouvoir s'effectuer avec n'importe quelle valeur de taux de vrais positifs (taux de faux positifs fixé) ou du taux de faux positifs (taux de vrais positifs fixé). Une première approche serait d'effectuer un ré-échantillonnage des courbes ROC par une méthode d'interpolation et choisir le niveau adapté aux performances désirées. Une seconde approche consisterait à modéliser les courbes ROC par une certaine distribution. L'utilisation des distributions également pourrait permettre d'évaluer et de quantifier la dérive temporelle des formes d'intérêts au cours du temps.

La plupart des approches énumérées sont principalement des stratégies logicielles plutôt que matérielle. Elles visent dans la majeure partie des cas à réduire le taux de faux positifs pour augmenter la robustesse de détection tout en minimisant l'activité du circuit de reconnaissance de formes et donc la consommation électrique. Après la mise en œuvre des diverses stratégies d'optimisation, il serait intéressant d'effectuer un bilan énergétique du système global ou si ce n'est pas possible, borner la consommation électrique de chaque module qui constitue l'architecture du système afin d'évaluer le potentiel d'utilisation du dispositif avec des systèmes de récupération d'énergie, comme la pile au glucose et donc pour avoir un système entièrement autonome.

Bibliographie

- [1] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin. Asynchronous level crossing analog to digital converters. *Measurement*, 37(4) :296–309, 2005.
- [2] D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2) :87–106, 1987.
- [3] A.J. Bagnall and J. Janacek. Clustering time series from arma models with clipped data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'04)*, 2004.
- [4] P.C. Bagshaw and M. Sarhadi. Anaysis of samples of wideband signals taken at irregular, sub-nyquist, intervals. *Electronics Letters*, 1991.
- [5] K. Bazaka and M. V. Jacob. Implantable devices : issues and challenges. *Electronics*, 2(1) :1–34, 2013.
- [6] A.P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7) :1145–1159, 1997.
- [7] B.C. Brodie, R.K. Cytron, and D.E. Taylor. A scalable architecture for high-throughput regular-expression pattern matching. In *Proceedings of the 33rd international symposium on computer architecture (ISCA 2006)*, 2006.
- [8] A.P. Chandrakasan, N. Verma, and D.C. Daly. Ultralow-power electronics for biomedical applications. *Annual Review of Biomedical Engineering*, 10 :247–274, 2008.

-
- [9] T.J. Chaney and C.E. Molnar. Anomalous behavior of synchronizer and arbiter circuits. *IEEE Transactions on computers*, (421-422), 1973.
- [10] A.L. Davis. The architecture and system method of ddm1 : a recursively sstructure data driven machine. In *Proceedings of the 5th annual symposium on computer architecture (ISCA '78)*, pages 210–215, 1978.
- [11] C. de la Higuera. A bibliographical study of grammatical inference. *Pattern Recognition*, 38(9) :1332–1348, 2005.
- [12] M. Dean, D. Dill, and M. Horowitz. Self timed logic using current sensing completion detection. In *Computer design :VLSI in computers and processors (ICCD'91)*, 1991.
- [13] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8) :861–874, 2006.
- [14] K.S. Fu. *Syntactic pattern recognition and applications*. Prentice-Hall Advances In Computing Science and Technology Series. Prentice Hall, 1982.
- [15] S.B. Furber, P. Day, J.D. Garside, and N.C. Paver. Amulet1 : a micropipelined arm. In *Compcon Spring '94*, pages 476–485, 1994.
- [16] S.B. Furber, D. Garside, P. Riocreux, and S. Temple. Amulet2e : an asynchronous embedded controller. *Proceedings of the IEEE*, 1999.
- [17] S. García, J. Luengo, J.A. Sáez, V. López, and F. Herrera. A survey of dicretization techniques : taxonomy and empirical analysis in supevised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4) :734–750, 2013.
- [18] F. Gensun. Whittaker–kotelnikov–shannon sampling theorem and aliasing error. *Journal of approximation theory*, 85(33) :115–131, 1996.
- [19] J.E. Hopcroft and R. Motwani. *Introduction to automata theory, languages, and Computation*. Prentice Hall, 2006.

-
- [20] C.C. Huang, S.F. Yen, and C.C. Wang. A li-ion battery chargin design for biomedical implants. In *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2008)*, pages 400–403, 2008.
- [21] A.K. Jain, P.W. Duin, and J. Mao. Statistical pattern recognition : A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1) :4–37, 2000.
- [22] J. Kessels and P. Marston. Desdesign asynchronous standby circuits for a low-power pager. *Proceedings of the IEEE*, 1999.
- [23] A. Koski, M. Juhola, and M. Meriste. Synactic recognition of ecg signals by attributed finite automata. *Pattern Recognition*, 28(12) :1927–1940, 1995.
- [24] M. Kunt. *Reconnaissance des formes et analyse de scènes*. Presses Polytechniques et Universitaires Romandes, 2000.
- [25] I. Kuon and J. Rose. Measuring the gap between fpgas and asics. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 26(2) :203–215, 2007.
- [26] Y. Li, D. Zhao, and W.A. Serdijn. A sub-microwatt asynchronous level-crossing adc for biomedical applications. *IEEE Transactions on biomedical circuits and systems*, 7(2) :149–157, 2013.
- [27] J.W. Mark and T.D. Todd. A nonuniform sampling approach to data compression. *IEEE Transactions on Communications*, 29(1) :24–32, 1981.
- [28] A.J. Martin. *Synthesis of asynchronous VLSI circuits*. PN, 2000.
- [29] M.J. Mathie, A.C.F. Coster, N.H. Lovell, and B.G. Celler. Accelerometry : providing an integrated, practical for long-term ambulatory monitoring of human movement. *Physiological Measurement*, 25(2) :1–20, 2004.
- [30] C. Mead and L. Conway. *Introduction to VLSI systems*. Addison-Wesley, 1979.
- [31] J.P. Crutchfield N.H. Packard. Symbolic dynamics of noisy chaos. *Physica D : Nonlinear Phenomena*, 2002.

-
- [32] R.F. Port and T. van Gelder. *Mind as motion : explorations in the dynamics of cognition*. A bradford book, 1998.
- [33] S.M. Qaisar. *Échantillonnage et traitement conditionnés par le signal : une approche prometteuse pour des traitements efficace à pas adaptatifs*. PhD thesis, Institut polytechnique de Grenoble, 2009.
- [34] V. Rajagopalan and A. Ray. Symbolic time series analysis via wavelet-based partitioning. *Signal processing*, 2006.
- [35] C. Ratanamahatana, E. Keogh, A.J. Bagnall, and S. Lonardi. A novel bit level time series representation with implication of similarity search and clustering. *Advances in Knowledge Discovery and Data Mining*, 3518 :771–777, 2005.
- [36] A. Ray. Symbolic dynamic analysis of complex systems for anomaly detection. *Signal processing*, 84(7) :1115–1130, 2004.
- [37] M. Renaudin. Asynchronous circuits and systems : a promising design alternative. *Microelectronic Engineering*, 54(1-2) :133–149, 2000.
- [38] M. Renaudin and J.B. Rigaud. état de l’art sur la conception des circuits asynchrones : perspective pour l’intégration des systèmes complexes. Technical report, TIMA Laboratory, 2000.
- [39] D. Saha and A. Mukherjee. Pervasive computing : a paradigm for the 21st century. *IEEE Computer Society*, 36(3) :25–31, 2003.
- [40] J. Sakarovitch. *Elements of automata theory*. Cambridge University Press, 2009.
- [41] M. Satyanaryanan. Pervasive computing : vision and challenges. *IEEE Personal communications*, 8(4) :10–17, 2001.
- [42] N. Sayiner, H.V. Sorensen, and T.R. Viswanathan. A level-crossing sampling scheme or a/d conversion. *IEEE Transactions on Circuis and Systems–II : Analog and digital signal processing*, 43(4) :335–339, 1996.

-
- [43] C.R. Shalizi and J.P. Crutchfield. Computational mechanics : pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104(314) :817–879, 2001.
- [44] H.S. Shapiro and R.A. Silverman. Alias-free sampling of random noise. *Journal of the Society for Industrial and Applied Mathematics*, 8(2) :225–248, 1960.
- [45] J. Sparsø. *Principles of asynchronous circuit design : a systems perspective*. Springer, 2002.
- [46] T.E. Starner. Powerful change part 1 : batteries and possible alternatives for the mobile market. *IEEE Pervasive Computing*, 2(4) :86–88, 2003.
- [47] I.E. Sutherland. Micropipelines. *Communication of the ACM*, 32(6) :720–738, 1989.
- [48] Y. Tsvividis. Event-driven data acquisition and digital signal processing—a tutorial. *IEEE Transactions on circuits and systems-II : express briefs*, 57(8) :557–581, 2010.
- [49] S.H. Unger. *Asynchronous sequential switching circuits*. Wiley, 1969.
- [50] C.H. van Berkel, M.B. Josephs, and S.M. Nowick. Applications of asynchronous circuits. *Proceedings of the IEEE*, 87(2) :223–233, 1999.
- [51] K. van Berkel and M. Rem. *Handshake circuits : an asynchronous architecture for VLSI programming*. Cambridge international series on parallel computation, 1994.
- [52] K. van Berkel et al. Asynchronous circuits for low power. *IEEE design & test of computers*, 11(2) :22–32, 1994.
- [53] K. van Berkel et al. A single-rail re-implementation of a dcc error detector using a generic standard-cell library. *Asynchronous design methodologies*, 1995.
- [54] C. Vezyrtzis and Y. Tsvividis. Processing of signals using level-crossing sampling. *IEEE International Symposium on Circuits and Systems*, pages 2293–2296, 2009.

- [55] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3) :94–104, 1991.

Résumé

Les systèmes embarqués mobiles font partis intégrante de notre quotidien. Afin de les rendre plus adaptés aux usages, ils ont été miniaturisés et leur autonomie a été augmentée, parfois de façon très considérable. Toutefois, les propositions d'amélioration butent désormais sur les possibilités de la technologie des circuits intégrés. Pour aller plus loin, il faut donc envisager de repenser la chaîne de traitement du signal afin de réduire la consommation de ces dispositifs. Cette thèse développe une approche originale pour exploiter efficacement l'échantillonnage par traversée de niveaux d'une part et, d'autre part, associe cet échantillonnage à une logique événementielle afin de réduire drastiquement la consommation d'énergie des systèmes intégrés autonomes. Une méthode de discrétisation adaptée à une application de reconnaissance de signaux physiologiques, utilisée comme exemple dans cette thèse, y est présentée. Un premier prototype en logique événementielle (asynchrone) sur circuit FPGA a permis de valider cette stratégie et de démontrer les bénéfices de cet échantillonnage dédié en termes de réduction de l'activité par rapport à un échantillonnage uniforme. Un second prototype en logique asynchrone et conçu en technologie CMOS AMS 0.35 μm a permis de valider par simulation électrique un gain extrêmement important sur la consommation électrique du dispositif.

Mots clés : système embarqué, implant médical, faible consommation, reconnaissance de formes, échantillonnage par traversée de niveaux, automate fini déterministe, logique événementielle, logique asynchrone, consommation électrique, réseau de portes programmables, circuit intégré propre à une application.

Abstract

Our everyday life is highly dependent on mobile embedded systems. In order to make them suitable to different applications, they have underwent size reduction and lifetime extension. However, these improvements are currently limited by the possibilities of the integrated circuits technologies. In order to push back the boundaries, it is necessary to reconsider the whole digital signal processing chain from scratch to sustain the power consumption reduction in this kind of system. This work develops on the first hand a strategy that smartly uses the level-crossing sampling scheme and on the other combines this sampling method with event-logic to highly reduce the power consumption in mobile embedded systems. A discretisation method adapted to the recognition of physiological patterns application is described. A first event-logic (asynchronous) prototype implemented on FPGA proved the potential benefits that an adapted sampling scheme could offers to reduce activity compared to a uniform sampling scheme. Electrical simulations performed on a second prototype, also designed in asynchronous logic, with CMOS AMS 0.35 μm technology, validated a high gain in power consumption.

Keywords : mobile embedded system, medical implant, low-power consumption, pattern recognition, level-crossing sampling scheme, deterministic finite automata, event-logic, asynchronous logic, power consumption, field-programmable gate array, application specific integrated circuit.