**UCL** Université catholique de Louvain
Laboratoire d'Électrométrie et d'Instrumentation

# Développement et applications de méthodes de mesure différentielle à haute résolution

Pascal DUPUIS
Ingénieur civil électricien

**Jury :**
Président :
Prof. P. SOBIESKI (FSA/ELEC)
Promoteur :
Prof. Ch. EUGÈNE (FSA/ELEC/LEI)
Membres :
Prof. H. BUYSE (FSA/ELEC/LEI)
Prof. Cl.VERAART (MD/FSIO/GREN)
Prof. G. BASTIN (FSA/ELEC/AUTO)
Prof. J. OKSMAN (SUPELEC, Gif sur Yvette, France)

Thèse présentée en vue de l'obtention du grade de Docteur en Sciences Appliquées

Louvain-la-Neuve,
Avril 2001

**UCL** Université catholique de Louvain
Laboratoire d'Électrométrie et d'Instrumentation

# Développement et applications de méthodes de mesure différentielle à haute résolution

Pascal Dupuis
Ingénieur civil électricien

**Jury :**
Président :
Prof. P. Sobieski (FSA/ELEC)
Promoteur :
Prof. Ch. Eugène (FSA/ELEC/LEI)
Membres :
Prof. H. Buyse (FSA/ELEC/LEI)
Prof. Cl.Veraart (MD/FSIO/GREN)
Prof. G. Bastin (FSA/ELEC/AUTO)
Prof. J. Oksman (SUPELEC, Gif sur Yvette, France)

Thèse présentée en vue de l'obtention du grade de Docteur en Sciences Appliquées

Louvain-la-Neuve,
Avril 2001

# Préface

Tout projet a une fin, et ce texte est l'étape finale d'une démarche dont les prémices remontent à l'époque ou j'effectuais mes études d'ingénieur.

Ce travail est la sanction d'un parcours de recherche qui n'aurait pas été possible sans le soutien de nombreuses personnes à qui je vais essayer ici de rendre justice, en les priant de bien vouloir m'excuser si je suis incomplet quant à l'étendue de leurs mérites.

En tout premier lieu je désire citer le professeur Christian Eugène, qui m'a offert les moyens matériels et humains de mener à bien ce travail. Je le remercie pour le temps et l'énergie qu'il m'a consacré, et pour avoir cru en l'achèvement de ce travail.

Je remercie les membres du Comité d'Encadrement et ensuite du Jury pour les conseils qu'ils m'ont prodigués lors de chaque étape intermédiaire et pour m'avoir guidé et orienté tout au long de ce travail.

Une partie des mesures ont été effectuées en collaboration directe avec des médecins. Je remercie en particulier le Prof. J.-M. Lachapelle de l'Unité de Dermatologie Professionnelle et les membres de l'Unité de Pneumologie, le Prof. D. Rodenstein et le docteur G. Liistro, qui ont toujours trouvé dans leur agenda très chargé des plages à me consacrer pour un conseil, un avis ou encore mener quelques mesures.

Il y eut également une série de collaborations avec des mémorands et des stagiaires, chacune d'elle étant l'occasion de contacts enrichissants, d'apport d'idées et d'énergie. Il y a eu, par ordre chronologique, Benoît David, Christof Stieger (ETH-Zürich), Benoît Maricau, Hamoud Merrad et Daton Medenou.

J'ai bénéficié de l'aide du personnel de l'unité LEI, tant académique que scientifique et technique, toujours prêt à aider, à donner de son temps.

« Last but not least » comme dit l'expression, il y a enfin mon épouse, qui m'a permis de dégager le temps nécessaire, et mes enfants, qui ont dû accepter pendant bien longtemps les contraintes inhérentes à ce projet. Ils vont tous pouvoir pousser un intense soupir de soulagement.

# Développement et applications de méthodes de mesure différentielle à haute résolution

## Résumé

Au cours de cette recherche, nous avons combiné des méthodes de mesures de paramètres physiques à des méthodes de traitement du signal pour extraire par voie indirecte des signatures de paramètres physiologiques. Ces mesures ont pour particularité d'être *différentielles* et *à haute résolution*, ce qui a permis d'appréhender des variations de signaux extrêmement ténues sous l'effet du phénomène étudié. Le traitement numérique a permis, à partir d'un signal brut très bruité, de séparer l'information utile de tout ce qui peut être considéré comme de la dérive ou des artefacts.

Les mesures ont été effectuées à l'aide de systèmes alliant, d'un point de vue électrique, faible consommation et faible bruit, dans la perspective de réaliser des équipements utilisables à terme en mode ambulatoire. Ensuite, elles ont été transmise sur ordinateur dans un environnement temps réel.

Deux grandes classes d'applications ont été étudiées. Bien qu'ultérieure chronologiquement, nous allons d'abord présenter la mesure combinée des rythmes cardiaques et respiratoires, ainsi que la détection d'apnées. La méthode est particulièrement peu contraignante puisqu'elle repose sur la mesure de pression dans un brassard destiné à l'évaluation de la tension artérielle, porté au poignet et faiblement gonflé. Nous avons évalué la période des signaux obtenus après décomposition par filtrage en ondelettes, ces signaux étant une image soit du rythme cardiaque, soit de l'activité respiratoire. Nous avons montré qu'il y a une bonne concordance entre les valeurs provenant de nos traitements numériques et celles pouvant être considérées comme des références. Nous avons également mis en évidence des signature d'apnées centrales et obstructives.

Par ailleurs, nous avons également cherché à caractériser l'état de l'épiderme à partir de ses propriétés de transmission thermique, ce qui fut l'occasion d'une activité intense au niveau de l'identification d'un modèle de propagation de la chaleur. Nous avons établi quels sont les paramètres qui semblent être les mieux corrélés à l'état de l'épiderme. La retombée la plus importante à nos yeux est une généralisation de la méthode de Prony, qui permet de faire clairement la part des choses entre l'influence des conditions initiales et celle de la commande appliquée au système.

Bien que développés dans le cadre d'applications bien définies, nous pensons que les méthodes de mesures et les algorithmes numériques mis au point sont susceptibles de trouver des débouchés dans d'autres branches de l'art de l'ingénieur.

# Development and applications of high resolution differential measurement methods

## Abstract

During this research, we combined physical parameters measurement methods with numerical signal processing in order to extract indirectly physiological parameters signatures. The measures are characterized by the fact that they are *differential* and done at a *high resolution*. This permitted to acquire very low level signal variations caused by the physiological phenomenon of interest. Numerical processing permitted, starting from a very noisy raw signal, to separate the relevant information from what could be considered as noise and artifacts.

Those measurement were done on systems combining, from an electrical point of view, low power consumption and low noise, in the perspective of developing ambulatory devices. Afterwards, data were acquired on a computer using a real-time environment.

Two main applications have been studied. Although coming later chronologically, we will first mention combined evaluation of cardiac and respiratory rhythms, together with apnoea detection for sleep study and monitoring. The method induced low discomfort, as the sensor is a cuff used for blood pressure measurement, wrapped around the wrist and inflated at a low pressure. After the decomposition of the original signal in a number of frequency bands using wavelet filtering, we evaluated the resulting signals period, those signals representing respectively cardiac or respiratory rhythms. We showed there is a good agreement between the values provided by our numerical processing and reference values. We also established how to detect the signature of central and obstructive apnoeas.

We also tried to characterize epidermis state and to detect its alterations through its heat transfer properties. This gave rise to intense activity in order to determine a heat propagation model. We established the parameters which seems the most correlated to epidermis state. The most important result is a generalization of the Prony method, permitting to clearly distinguish terms due to initial conditions from terms resulting from an applied command.

Developed in a well defined framework, we still think our measurement methods and the numerical algorithms we worked out may also be applied to other engineering domains.

# Table des matières

## *Première partie. Contenu*

**1 Introduction** **3**
**1.1 Thème et contenu du doctorat** . . . 3
**1.2 Chronologie et publications** . . . 5
**1.3 Mémoires de fin d'études** . . . 6

**2 Plan** **7**
**2.1 Présentation générale** . . . 7
**2.2 Instrumentation** . . . 7
**2.3 Monitoring cardiorespiratoire** . . . 7
**2.4 Caractérisation cutanée par voie thermique** . . . 8

## *Deuxième partie. Instrumentation*

**1 Technique de mesure** **13**
**1.1 Introduction** . . . 13
**1.2 Température différentielle** . . . 13
**1.3 Mesure en pont** . . . 14
*1.3.1 Mode commun* . . . 14
*1.3.2 Bruits à basse fréquence* . . . 15
*1.3.3 Bruits à large bande* . . . 15
*1.3.4 Bruit généré par les capteurs* . . . 15

**2 Carte mono-canal** **17**
**2.1 Cahier des charges** . . . 17
*2.1.1 Analyse de l'existant* . . . 17
*2.1.2 Objectifs du design* . . . 19

**2.2 Schéma général** . . . 24
**2.3 Réalisation** . . . 24
*2.3.1 Générateur triangulaire stabilisé en amplitude* . . . 24
*2.3.2 Amplificateur d'Instrumentation* . . . 27
*2.3.3 Démodulateur synchrone* . . . 29
*2.3.4 Filtre passe-bas* . . . 30
*2.3.5 Tension de référence* . . . 30
*2.3.6 Calcul théorique de la sensibilité* . . . 31
*2.3.7 Calcul du bruit* . . . 31
**2.4 Communication avec la carte Lab-PC** . . . 31
**2.5 Résultats** . . . 33
*2.5.1 Consommation électrique* . . . 33
*2.5.2 Stabilité à court terme* . . . 33
*2.5.3 Emploi avec un accéléromètre* . . . 33
**2.6 Calibration du thermomètre différentiel** . . . 34

**3 Carte multi-canaux** **35**
**3.1 Modifications apportées** . . . 35
*3.1.1 Synchronisation de l'oscillateur* . . . 35
*3.1.2 Augmentation du nombre de canaux d'entrées* . . . 37
*3.1.3 Mesure de l'amplitude* . . . 38
**3.2 Réalisation** . . . 38

**4 Acquisition temps réel sous Linux** **41**
**4.1 Introduction** . . . 41
**4.2 Définition du cadre** . . . 42
*4.2.1 Séparation des fonctionnalités* . . . 42
*4.2.2 Architecture distribuée* . . . 42
**4.3 Présentation de Linux** . . . 43
**4.4 Développements précédents** . . . 44
**4.5 La solution sous Linux** . . . 45
*4.5.1 Tâche d'acquisition* . . . 45
*4.5.2 Traitement numérique* . . . 45
**4.6 Device Driver sous Linux** . . . 46
*4.6.1 Modules* . . . 47
*4.6.2 Accès mémoire* . . . 48
*4.6.3 Application temps réel : RT-Linux* . . . 49
*4.6.4 Organisation générale du gestionnaire* . . . 50
*4.6.5 Philosophie générale* . . . 52

**4.7 Programme d'acquisition en mode utilisateur** . . . . . 52
*4.7.1 Thread d'acquisition* . . . . . 55
*4.7.2 Thread d'émission* . . . . . 56
*4.7.3 Thread de collecte* . . . . . 56
*4.7.4 Thread principal, ou de gestion* . . . . . 56
*4.7.5 Configuration* . . . . . 58
**4.8 Couplage noyau – utilisateur** . . . . . 58
*4.8.1 Les erreurs en mode noyau* . . . . . 59
*4.8.2 Les erreurs en mode utilisateur* . . . . . 59
**4.9 Résultats** . . . . . 60
**4.10 Architecture client-serveur** . . . . . 61
*4.10.1 Motivations* . . . . . 61
*4.10.2 Systèmes clients-serveurs* . . . . . 61
*4.10.3 Opérations à implémenter* . . . . . 61
**4.11 Retombées possibles** . . . . . 63
*4.11.1 Micro-centrales de mesure* . . . . . 63
*4.11.2 Logiciels d'acquisition* . . . . . 65
**4.12 Conclusion** . . . . . 67

## *Troisième partie. Monitoring cardiorespiratoire*

**1 Voie thermique** **73**
**1.1 Premiers essais** . . . . . 73
*1.1.1 Traitement par ondelettes* . . . . . 74
*1.1.2 Exemple de résultat* . . . . . 74
*1.1.3 Filtrage par ondelettes* . . . . . 76
*1.1.4 Hypothèses formulées* . . . . . 78
**1.2 Seconde étape** . . . . . 79

**2 Rythmes cardiaques et respiratoires** **81**
**2.1 Introduction** . . . . . 81
**2.2 Contexte : la polysomnographie** . . . . . 81
**2.3 Prévention de la mort subite du nourrisson.** . . . . . 84
**2.4 Améliorations possibles** . . . . . 84
**2.5 Mécanismes physiques** . . . . . 84
**2.6 Dispositif instrumental** . . . . . 85

**3 Analyse par ondelettes** **87**
**3.1 Introduction** . . . . . 87

**3.2 Pourquoi les ondelettes ?** . . . 87
**3.3 Géométrie fonctionnelle** . . . 89
*3.3.1 Produit scalaire* . . . 89
*3.3.2 Norme d'une fonction* . . . 89
*3.3.3 Écart entre deux fonctions* . . . 89
**3.4 Transformée de Fourier** . . . 90
**3.5 Caractéristiques de base des ondelettes** . . . 90
**3.6 Ondelette de Haar** . . . 91
**3.7 Formalisation** . . . 95
*3.7.1 Multirésolution* . . . 97
**3.8 Fonctions d'échelle et ondelettes** . . . 100
*3.8.1 Construction par l'équation de raffinement* . . . 101
*3.8.2 Construction utilisant la transformée de Fourier* . . . 102
**3.9 Transformée rapide en ondelette** . . . 104
*3.9.1 Décomposition pyramidale* . . . 104
*3.9.2 Propriétés des filtres* . . . 106
*3.9.3 Propriétés de la Transformée* . . . 107
*3.9.4 Approximation* . . . 107
**3.10 Applications** . . . 107
*3.10.1 Compression du signal* . . . 107
*3.10.2 Filtrage adaptatif* . . . 108

**4 Monitoring : technique de la manchette** **111**
**4.1 Contexte** . . . 111
**4.2 Analyse par ondelettes** . . . 112
**4.3 Première solution : travail sans décimation** . . . 113
*4.3.1 Détermination du rythme cardiaque* . . . 115
*4.3.2 Algorithme `fp extrema.m`* . . . 118
*4.3.3 Résultats* . . . 119
*4.3.4 Discussion* . . . 120
*4.3.5 Vérification de la méthode* . . . 122
*4.3.6 Détection des apnées obstructives* . . . 122
*4.3.7 Conclusion* . . . 123
**4.4 Rythme respiratoire : travail par reconstruction** . . . 124
*4.4.1 Résultats précédents* . . . 124
*4.4.2 Mesure avec sonde œsophagienne* . . . 124
*4.4.3 Conclusions* . . . 126
**4.5 Seconde solution : travail avec décimation** . . . 126

*4.5.1 Détection du rythme respiratoire* . . . 129
*4.5.2 Détection d'apnées* . . . 131
**4.6 Comparaison des deux approches** . . . 134
**4.7 Conclusion** . . . 137

## *Quatrième partie. Caractérisation thermique de l'épiderme*

**1 Contexte** **143**
**1.1 Introduction : tests dermatologiques** . . . 143
**1.2 Physiologie et caractéristiques de la peau** . . . 144
**1.3 Propriétés thermiques du corps humain** . . . 146
**1.4 Description des mesures** . . . 147
**1.5 Utilisation des résultats** . . . 147
**1.6 Présentation des méthodes utilisées** . . . 148

**2 Classification par modélisation** **151**
**2.1 Contexte** . . . 151
**2.2 Modélisation de type boîte noire** . . . 151
**2.3 Modélisation par cellules R–C** . . . 152
**2.4 Modélisation par l'équation de la diffusion** . . . 153

**3 Transformée cepstrale** **155**
**3.1 Introduction** . . . 155
**3.2 Définition** . . . 155
**3.3 Traitement cepstral** . . . 157
**3.4 Résultats** . . . 159
**3.5 Discussion** . . . 161
**3.6 Classification par coefficients cepstraux** . . . 162
**3.7 Discussion** . . . 162

**4 Méthode de Prony généralisée** **165**
**4.1 Introduction** . . . 165
**4.2 Identification des processus** . . . 166
*4.2.1 Cadre de référence* . . . 166
*4.2.2 Modélisation* . . . 166
*4.2.3 Critère de performance* . . . 167
*4.2.4 Détermination des paramètres linéaires* . . . 168
**4.3 Modélisation par somme d'exponentielles** . . . 169
*4.3.1 Méthode non-linéaire* . . . 169

*4.3.2 Méthode de Prony* . . . 170
*4.3.3 Comparaison* . . . 172
**4.4 Extension de la méthode de Prony** . . . 173
*4.4.1 Comment la nommer ?* . . . 173
*4.4.2 Identification de réponse indicielle* . . . 173
*4.4.3 Signal constant par morceaux* . . . 174
*4.4.4 Algorithme de Prony générique* . . . 177
*4.4.5 Réponse indicielle et gain statique* . . . 180
*4.4.6 Pôles complexes* . . . 181
*4.4.7 Implémentation : `fp pronysq.m` et `fp pronyeq.m`* . . . 182
*4.4.8 Illustration : `fp pronysq`* . . . 184
*4.4.9 Illustration : `fp pronyeq`* . . . 188
*4.4.10 Commentaires* . . . 190
**4.5 Discussion** . . . 191
**4.6 Résumé** . . . 192

**5 Application : Modélisation thermique** **193**
**5.1 Introduction** . . . 193
**5.2 Traitement numérique** . . . 193
**5.3 Résultats : sujet n°2** . . . 195
**5.4 Résultats : sujet n°1** . . . 199
**5.5 Perspectives** . . . 201
**5.6 Conclusions** . . . 202

## *Conclusions*

## *Annexes*

**A Algorithmes** **209**

**B Colophon** **231**

**C Environnement de travail** **233**

**Références** **235**

*Première partie*

---

# ***Contenu***

Chapitre 1

# Introduction

## 1.1 Thème et contenu du doctorat

L'idée à la base de notre thèse était de développer des applications d'une méthode de mesure de température différentielle à haute résolution-seuil associée à des capteurs passifs. Cette méthode a été développée par le Prof. M. Van Ruymbeke de L'Observatoire Royal de Belgique et utilisée dans des applications de type géophysique.

Nous avons décidé d'effectuer cette transposition en y apportant notre contribution personnelle. Pour cela, nous avons décidé de suivre un certains nombre d'orientations spécifiques par rapport à la méthode de départ, à savoir :

- développer une électronique d'instrumentation à très faible consommation électrique, afin que les réalisations soient, à terme, utilisables dans un contexte de monitoring ambulatoire;
- ajouter directement dans la chaîne de mesure un convertisseur de type analogique-numérique à haute résolution-relative et à haute linéarité, afin de permettre par la suite de mettre en évidence des signaux de faible amplitude par des méthodes de décomposition appropriées reposant sur des techniques de traitement numériques;
- établir la faisabilité d'applications dans le domaine thermique, mais sans nécessairement nous y limiter.

Après avoir mis en œuvre le matériel initial, nous avons développé un premier prototype instrumental. D'emblée, il est apparu que les signaux acquis étaient susceptibles de présenter des dérives très importantes, et que seul un traitement numérique approprié permettrait de mettre clairement en évidence les phénomènes étudiés.

La première classe d'application que nous avons développée concernait la caractérisation par voie thermique de l'épiderme. Nous avons utilisé une méthode d'identification

active consistant à relever l'onde de température générée à la surface de l'épiderme par une source de puissance thermique, après un parcours sur une faible distance.

Pour étudier la relation entre l'entrée et la sortie, à savoir d'une part le flux thermique injecté et d'autre part l'élévation de température concomitante, nous avons commencé par une approche de type Identification des Processus conventionnelle. Il est apparu que cette approche n'était pas très satisfaisante. En effet, les méthodes classiques se sont révélées peu aptes à modéliser correctement le comportement de nos signaux.

Disposant de courbes associées à différents états, nous avons commencé par essayer de les différencier en extrayant certaines caractéristiques des courbes. Cette démarche ne nous semblait guère parlante : en effet, quelle signification physique accorder à la variation des coefficients d'un modèle ? Résulte-t-elle bien d'une différence significative des propriétés du milieu de transmission ou s'agit-il d'un artefact de calcul ?

Afin de mettre en évidence la présence éventuelle de chemins de propagations multiples entre la source de puissance thermique et le point de mesure, nous avons aussi utilisé la méthode cepstrale. Dans cette approche, des délais de propagation multiples se marquent par une impulsion correspondante dans la transformée cepstrale du signal. Nous avons également développé une méthode d'identification où l'identification d'un modèle s'effectue en minimisant la distance entre le cepstre du signal mesuré et celui prédit par le modèle.

À la recherche de sources de perturbations, nous avons essayé de détecter la présence de signaux liés au rythme respiratoire. Par un filtrage passe-bande particulier (filtrage en ondelettes sans décimation), il a été possible de mettre en évidence de tels signaux, avec une très faible amplitude. Par la suite, il est apparu que ces signaux étaient liés à des phénomènes de modulation de la pression appliquée sur le capteur par la peau. Cette modulation entraînait une variation du couplage thermique, d'où la présence d'une signature au niveau thermique.

Nous avons poursuivi cette voie en utilisant cette fois des capteurs de pression. Cela permit de développer la seconde classe d'application, à savoir la mesure combinée des rythmes cardiaques et respiratoires.

Nous avons pour cela évalué la pression dans un brassard de mesure de la tension artérielle faiblement gonflé. Le signal obtenu contenait une trace du rythme respiratoire que nous avons extraite par un traitement reposant sur la transformée en ondelettes. Par la suite, nous avons également pu mettre en évidence des signatures d'apnées.

Au cours de la dernière étape du doctorat, nous avons amélioré les différents points développés jusqu'alors :

– au niveau de l'électronique d'instrumentation, nous avons réalisé un second prototype, une des modifications majeures étant l'ajout de canaux d'entrée supplémentaires, le but recherché étant de réaliser plusieurs mesures en une seule manipulation ou encore d'étudier la répartition spatiale d'une quantité physique;

- au niveau logiciel, nous avons mis en œuvre une solution tournant sous le système d'exploitation Linux. Cela a permis de travailler dans un cadre temps réel, de rendre le processus d'acquisition extrêmement robuste, et de répartir les différents traitements sur plusieurs machines;
- pour la modélisation de la réponse thermique, nous avons mis au point une généralisation de la méthode de Prony, permettant de travailler sur une somme de réponses indicielles décalées dans le temps. Ce modèle permet également d'obtenir directement la réponse de régime à un signal d'entrée périodique à partir d'une identification effectuée avec des données appartenant à la zone transitoire de la relation entrée-sortie. À l'analyse, il est apparu que les matrices de régression étaient mal conditionnées, le système étudié ayant un de ses pôles très voisin de l'unité. Cela explique probablement la variabilité des coefficients des modèles obtenus au cours des étapes précédentes. Dans le cas de cette approche, la détermination des amplitudes se base sur une matrice de régression où les colonnes sont nettement différentes, cette matrice est donc mieux conditionnée.
- pour le traitement des signaux de pression de l'application de monitoring cardio-respiratoire, nous avons modifié le traitement par ondelettes, ce qui a permis de mettre plus clairement en évidence les signaux respiratoires.

## 1.2 Étapes chronologiques et publications

Voici en résumé les grandes étapes de notre travail de recherche :

- 1993 : communication [1];
- 1996 :
  - Mémoire de fin d'études de B. David, « Modélisation et Identification Thermique des Caractéristiques Thermiques de la peau humaine par Thermométrie à Haute Résolution », que j'ai encadré;
  - communication [2];
- 1997 : communication [3];
- 1997 - 1998: travail de fin d'études de Ch. Stieger, « Recherches de nouvelles méthodes pour le monitoring respiratoire », que j'ai encadré;
- 1998 : communication [4];
- 1998 - 1999, recherches de méthodes de mesure du rythme respiratoire et de détection d'apnées basées sur la détection du pouls paradoxal, en collaboration avec le Dr. G. Liistro, de l'unité de pneumologie. Parmi les étapes marquantes, citons :
  - mémoire de fin d'études de B. Maricau, « Détection de problèmes respiratoires : Développement d'un environnement d'acquisition et de traitement de données », que j'ai encadré;
  - communication [5]. Une version plus étoffée en a été publiée dans [6].

### Liste chronologique des publications

[1] Pascal Dupuis and Christian Eugène. A high resolution micropower acquisition system for passive sensors. In *Proc. 6$^{th}$ IMEKO TC-4 Symposium*, Intelligent Instrumentation for Remote and On-site Measurement, pages 309–314, Brussels, Belgium, 12–13 May 1993. IBRA-BIRA. ISBN 90-9006205-X.

[2] Pascal Dupuis, Christian Eugène, and Benoit David. Thermal characterisation of the human skin using differential thermometry. In *Proc. joined IEEE–IMTC'96 and IMEKO/TC–7 Conference*, Quality Measurements: The Indispensable Bridge between Theory and Reality, pages 1458–1463, Brussels, Belgium, 4–6 Jun 1996. IEEE. IEEE catalog n°96CH3596, ISBN 0-7803-3312-83, doi: 10.1109/IMTC.1996.507612.

[3] Pascal Dupuis and Christian Eugène. Identification of the human skin through high resolution thermometry : cepstral method. In *Proc. 14$^{th}$ IMEKO World Congress*, New measurements – challenges and *visions*, pages 232–237, Tampere, Finland, 1–6 Jun 1997. Finnish Society of Automation.

[4] Pascal Dupuis, Christian Eugène, and Christof Stieger. New way to detect respiratory apnoea by high resolution cuff pressure measurement. In *Proc. 10$^{th}$ IMEKO TC-4 Symposium*, Development in Digital Measuring Instrumentation, pages 543–547, Naples, Italy, 17–18 Sep 1998.

[5] Pascal Dupuis and Christian Eugène. Combined detection of respiratory and cardiac rhythms disorders by high resolution differential cuff pressure measurement. In *IMTC/99, The 16$^{th}$ IEEE Instrumentation and Measurement Technology Conference*, pages 109–113, Venice, Italy, 24–26 May 1999. ISBN 0-7803-5276-9, doi: 10.1109/IMTC.1999.776729.

[6] Pascal Dupuis and Christian Eugène. Combined detection of respiratory and cardiac rhythms disorders by high resolution differential cuff pressure measurement. *IEEE Trans. on Instrumentation and Measurement*, 49(3): 498–502, June 2000. doi: 10.1109/19.850383.

## 1.3 Mémoires de fin d'études

Au cours de mes mandats d'assistants, j'ai encadré trois travaux de fin d'étude en relation avec ma thèse.

[7] Benoît David. Modélisation et identification thermique des caractéristiques thermiques de la peau humaine par thermométrie à haute résolution, 1996.

[8] Christof Stieger. Recherches de nouvelles méthodes pour le monitoring respiratoire, 1998.

[9] Benoît Maricau. Détection de problèmes respiratoires : Développement d'un environnement d'acquisition et de traitement de données, 1999.

Chapitre 2

# Plan

## 2.1 Présentation générale

Comme le lecteur pourra s'en rendre compte en lisant la description chronologique, le déroulement de la recherche a comporté de nombreux aller-retour entre trois pôles : instrumentation, expérimentation et développements méthodologiques. Le but final a été d'établir la faisabilité d'un certain nombre d'applications, les aspects de développement associés étant au service de cette finalité.

Dans la présentation qui suit, nous avons par souci de clarté regroupé les différents points abordés par thèmes. De ce fait, l'ordre ne sera pas chronologique. Commençons par esquisser le contenu de chacun de ces thèmes.

## 2.2 Instrumentation

Nous reprenons ici tous les développements liés à l'acquisition de signaux, aussi bien du point de vue matériel que logiciel. Après avoir analysé le système dont nous sommes partis, nous présenterons une carte d'acquisition mono-canal, dont nous analyserons les performances. Pour enrichir les possibilités d'acquisition, nous avons ensuite développé une carte d'acquisition multi-canaux ainsi qu'un environnement logiciel de type temps réel et distribué.

## 2.3 Monitoring cardiorespiratoire

L'histoire de cette recherche commence par une interrogation : est-il possible qu'un signal lié au rythme respiratoire puisse perturber des acquisitions thermiques cutanées

à haute résolution? Ayant pu mettre en évidence un tel signal, nous avons par la suite établi qu'il s'agissait d'un signal de pression. Nous expliquerons quelle est l'origine physiologique de ce signal, et par quel traitement nous avons pu l'extraire. De ce point de vue, nous avons utilisé deux méthodes. La première était basée sur l'hypothèse qu'il n'était pas possible d'effectuer un traitement par ondelettes « classiques » de type temps-réel. Ayant par la suite trouvé dans la littérature un moyen de contourner cette difficulté, nous avons pu mettre plus clairement en évidence les divers phénomènes respiratoires.

## 2.4 Caractérisation cutanée par voie thermique

Nous présenterons tout d'abord des méthodes existantes de caractérisation de l'épiderme, ainsi que leur contexte d'utilisation. Nous avons essayé de mesurer les caractéristiques de propagation thermique par une méthode active, en relevant l'onde de température induite par une injection de puissance thermique. Pour effectuer la modélisation de la relation entrée-sortie, nous avons essayé une série de méthodes d'identification et développé deux méthodes, l'une basée sur la transformée cepstrale et l'autre consistant en une généralisation de la méthode de Prony. Cette dernière méthode a montré que dans ce cas, il est préférable de ne pas se focaliser sur les coefficients d'un modèle, qui sont susceptibles d'être bruités, mais plutôt sur des propriétés plus globales, en particulier sa réponse de régime. C'est cette approche qui a donné la différenciation la plus sensible des différentes courbes obtenues.

*Deuxième partie*

---

# *Instrumentation : aspects matériel et logiciel*

Un des buts de notre recherche était de développer une méthodologie de mesure reposant sur une acquisition performante de signaux électriques. Ayant mis en œuvre une solution linéaire, a haute résolution, et à faible bruit additif, nous avons pu par la suite extraire des signaux utiles masqués par de la dérive et du bruit. Ce sont ces aspects d'acquisition et d'instrumentation associée qui constituent le fil conducteur de cette partie.
Chronologiquement, nous avons d'abord développé une carte d'acquisition destinée à évaluer avec une résolution élevée le déséquilibre d'un pont de Wheatstone. En combinant de façon adéquate des thermistances et des résistances, le pont peut fournir une tension de sortie image d'une température différentielle. Pour exploiter cette carte, il était nécessaire de la piloter par ordinateur : nous avons également écrit une application dans ce but. Au fil du temps, ce système s'est révélé insuffisant par manque de canaux d'entrée et pas assez fiable : nous avons donc conçu une carte multi-canaux et repris les développements logiciels dans un environnement de programmation de type Unix. Nous en expliquerons quels en sont les bénéfices et retombées possibles du point de vue instrumental.

Chapitre 1

# Technique de mesure

## 1.1 Introduction

Le point de départ de ce travail était l'exploitation d'une méthode de mesure différentielle à haute résolution, associée à un pont de Wheatstone constitué de thermistances et de résistantes. L'application initiale visait à exploiter le déséquilibre de ce pont, ce qui représentait une température différentielle. Nous allons commencer par présenter et analyser sommairement ces différentes notions, avant d'exposer dans les chapitres ultérieurs comment nous avons fait évoluer ce projet initial.

## 1.2 Température différentielle

Les températures « usuelles » sont définies au mieux avec deux décimales : le point triple de l'eau est à 273.16 K, ... Il s'agit de températures *absolues* définies par rapport à un zéro absolu, qui est le point de départ de l'échelle de température en Kelvins.

De ce fait, il n'est pas physiquement raisonnable de prétendre annoncer une température absolue comportant plus de deux décimales. Par contre, une résolution plus importante devient possible si on considère la *différence* de température entre deux capteurs. Pour cela, il convient d'y associer un système de mesure dont la sortie est une image directe de cette différence. On pourrait envisager d'étalonner le couple de capteurs en mesurant pour chacun d'eux leur température absolue et en déduire par différence la sensibilité du système de mesure. Cependant, il y aurait soustraction numérique de deux valeurs très proches, ce qui peut donner lieu à une erreur relative très importante. En pratique, on travaillera directement par voie différentielle, en écrivant une expression linéarisée de la sortie du système en fonction d'une part de la température absolue des capteurs et d'autre part de leur écart, ce qui constitue la grandeur utile, aussi appelée mesurande.

La température différentielle s'obtient donc par *interpolation*, et c'est ainsi qu'elle prend son sens physique.

## 1.3 Mesure en pont

La voie royale pour évaluer par voie électrique un désappariement entre des capteurs à variation de résistance consiste à les placer dans un pont de Wheatstone. Autour d'un point de fonctionnement, la tension de déséquilibre sur la diagonale détectrice est une image linéaire (ou linéarisable) de la différence de la grandeur à laquelle le capteur réagit.

Il y a un certains nombres de problèmes bien connus à maîtriser au niveau d'un pont pour arriver à exploiter sa sortie avec une résolution élevée. Passons-les en revue :

- influence du mode commun;
- bruits à basse fréquence
- bruits à large bande
- bruit généré par les capteurs

### 1.3.1 Mode commun

D'une manière générale, la tension de déséquilibre est une fraction de la tension de mode commun, à savoir la tension moyenne à laquelle est portée chaque sortie du pont. Si l'amplificateur qui suit le pont ne rejette pas suffisamment ce mode commun, il en résulte une tension de sortie qui n'est pas liée au phénomène étudié. Pour quantifier l'importance de cette tension, on parle de CMRR (Common Mode Rejection Ratio). Une façon de réduire son influence consiste à utiliser des amplificateurs dont la valeur de ce paramètre est aussi élevée que possible.

Nous avons choisi une voie différente :

- les résistance fixes sont choisies de façon à ce que leur valeur soit environ égales à celles des thermistances dans la plage de température utile;
- le pont est alimenté par deux tensions de signes opposés.

En fait, il y a une alimentation principale ainsi qu'un système à amplificateur opérationnel de gain $-1$ fournissant la seconde alimentation du pont. Cela augmente légèrement la complexité car il y a un amplificateur opérationnel de plus. L'avantage est que les diagonales détectrices sont à une tension proche de la masse, et que la tension de mode commun est ainsi très réduite.

### 1.3.2 Bruits à basse fréquence

Il s'agit par exemple de dérive thermique sous l'effet des fluctuations journalières de la température du lieu d'expérimentation, ce qui peut influencer de plusieurs façons les circuits électroniques. Nous inclurons également les perturbations liées à la fréquence du réseau de distribution électrique.

La parade classique à ces problèmes est fournie par la méthode de la détection synchrone. Un signal alternatif, portant le nom de signal analysant, est injecté dans le dispositif à caractériser. La grandeur de sortie est une tension ou un courant généré sous l'effet de ce signal. Cette sortie, contaminée par du bruit, est ensuite modulée à la fréquence du signal analysant et filtrée par un passe-bas de faible largeur. Le signal utile, étant modulé à sa propre fréquence, génère une composante continue ou à basse fréquence qui se retrouve intacte à la sortie du filtre. Les composantes parasites basse fréquence présentes dans le signal amplifié se retrouvent transportées autour de la fréquence modulante et sont rejetées par le passe-bas.

### 1.3.3 Bruits à large bande

Dans ce cas, nous considérons plutôt le bruit électronique en général, que ce soit en terme de tension ou de courant. En ce qui concerne le concerne, il n'y a pas d'approche standard. Le mieux que l'on puisse faire est d'analyser soigneusement l'influence des différents termes de bruits, et, en considérant plusieurs approches différentes, de trouver celle qui donne la contribution totale la plus basse.

### 1.3.4 Bruit généré par les capteurs

De ce point de vue, la consommation et le bruit évoluent en sens inverse l'un de l'autre. Un capteur assimilable à une résistance de faible valeur aura une consommation élevée et un bruit électronique faible; toute réduction de consommation se fera par une augmentation de la résistance et donc au détriment du bruit. Dans cette optique, il convient de trouver un compromis acceptable.

En outre, dans le cas des capteurs thermiques, il convient également de limiter l'auto-échauffement, celui-ci étant lié à la puissance électrique dissipée sous l'effet de la tension d'alimentation du pont.

Chapitre 2

# Électronique d'acquisition : carte mono-canal

## 2.1 Cahier des charges

### 2.1.1 Analyse de l'existant

Le « thermomètre » dont nous sommes parti utilisait le principe de la détection synchrone, et fonctionnait comme suit :

- un oscillateur CMOS générait un signal carré, d'amplitude fixe et de fréquence ajustable via une résistance de réglage;
- ce signal attaquait le pont de capteurs, constitué de deux thermistances et de deux résistance fixes;
- le signal sur la diagonale détectrice était amplifié via un amplificateur différentiel à ampli-op;
- ce signal amplifié passait par un système à gain commuté, piloté par le signal carré. De la sorte, un redressement synchrone était obtenu;
- un étage final de filtrage assurait enfin un lissage passe-bas du signal, pour ne conserver que l'information à basse fréquence.

Tel que conçu, la sortie de ce système était purement analogique. Nous avons effectué nos premiers test en utilisant un enregistreur papier ! En ce qui concerne la calibration, elle consistait à mettre en série une résistance de 1 Ω en série avec une thermistance, et d'en déduire la sensibilité globale.

En travaillant avec un appareil de mesure de précision, cet appareil a permis de mettre en évidence des phénomènes liés aux marées terrestres et aux cycles astronomiques, ces mesures étant effectuées avec une résolution de 20 bits et une bande passante

d'une fraction de Hertz, l'analyse s'effectuant par voie spectrale (repérage de raies dont la fréquence correspond à des phénomènes astronomiques comme l'alternance jour-nuit, les cycles lunaires, les cycles solaires).

Souhaitant travailler dans la filière de l'électronique micro-puissance, nous avons commencé par remplacer les ampli-ops de cette carte, de type LM324, par des LP324, une version à faible consommation. Il est apparu un problème suite à la réduction de « slew rate ». Considérons la figure 2.1. Il y a trois courbes décalées. Celle centrée sur

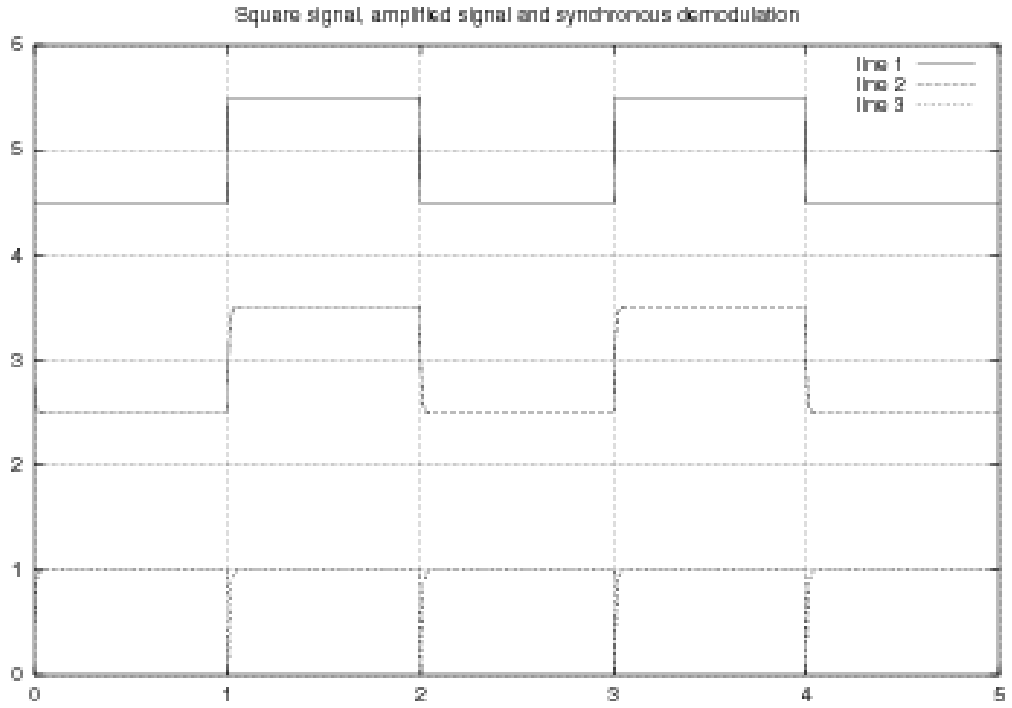



FIG. 2.1: Effet du slew rate sur le redressement synchrone

5 est le signal du générateur; celle sur 3 est le signal amplifié : l'effet de la limitation de la pente est de faire apparaître un flanc qui n'est pas vertical lors des transitions de niveau.

Après démodulation synchrone, représentée en bas du graphe, ce flanc donne lieu à des impulsions qui réduisent la valeur moyenne du signal. Ces impulsions peuvent être assimilées à des triangles dont la hauteur est celle du signal et dont la base lui est proportionnelle car liée par une relation linéaire. Leur surface est donc une fonction quadratique de l'amplitude du signal non-perturbé. Autrement dit, l'effet est d'introduire un terme de distorsion quadratique. Concrètement, il convenait de trouver une autre forme d'onde pour supprimer cet effet.

### 2.1.2 Objectifs du design

Après avoir étudié ce prototype, nous avons voulu mettre au point une version améliorée de ce « thermomètre ». Nous avons établi comme suit les caractéristiques que devrait posséder notre version :

1. faible consommation;
2. réponse linéaire;
3. possédant un convertisseur analogique/digital intégré;
4. haute résolution.

L'idée sous-jacente est d'appliquer cette méthodologie à des études de type biomédical, dans des applications qui puissent éventuellement être effectuées en mode ambulatoire. Le rôle de la haute résolution est de permettre la mise en évidence, par des techniques adéquates, de signaux très faibles, et apparemment indiscernables du bruit. Cette résolution élevée doit bien entendu aller de pair avec une bonne linéarité et une approche adéquate des phénomènes des bruits d'origine électronique.

En connectant directement le convertisseur à la source de signal, nous évitons de transmettre à distance un signal analogique. Par ailleurs, cela rend le système plus facilement autonome, puisqu'il peut ainsi être relié directement par une interface numérique à un microprocesseur ou à une micro-centrale de mesure.

Voyons à présent quelle réponse nous avons pu apporter aux différents points cités.

#### Réponse linéaire

Les impulsions dans la sortie démodulée étant intrinsèquement liée au fait que la commutation a lieu à un moment où le signal est différent de zéro, nous avons choisi de mettre en œuvre d'autres formes d'ondes. Un signal sinusoïdal étant difficile à générer avec une grande pureté[1], nous avons fait le choix d'utiliser des signaux triangulaires. Ceux-ci peuvent en effet s'obtenir très simplement par intégration d'un signal carré. Cette approche présente un certain nombres d'avantages par rapport aux signaux carrés :

- plus de problèmes de slew-rate ni de dépassement, l'ampli-op travaillant toujours en contre-réaction;
- commutation possible lors des passages par zéro;
- contenu harmonique plus faible, ce qui est favorable d'un point de vue compatibilité électromagnétique;
- valeur efficace plus faible, ce qui réduit les problèmes de consommation et d'auto-échauffement dans les capteurs.

1. ce genre de circuit fait appel à des ponts non-linéaires pour en stabiliser l'amplitude.

Mais, comme un intégrateur possède toujours un peu de dérive, il fallait trouver un moyen de l'éliminer et de stabiliser ce signal en amplitude et en fréquence. La solution que nous avons implémentée a été de réguler la fréquence pour obtenir une amplitude constante.

### Convertisseur D/A intégré

Le rôle de ce circuit est de convertir localement le signal, pour en permettre une transmission digitale. Cette approche offre une meilleur immunité au bruit, puisqu'il n'y a pas de perte de rapport signal/bruit dû à la transmission à distance d'un signal analogique. En effet, une telle ligne de transmission peut présenter une susceptibilité au bruit induit, notamment par couplage électromagnétique avec des sources de perturbations.

Nous avons d'abord pensé à utiliser des convertisseurs double-rampe. Ces circuits peuvent atteindre des résolutions limite de 16, voire 18 bits. Le facteur limitatif est la difficulté de détecter le passage par zéro lors de la phase d'intégration de la référence. En effet, une résolution élevée implique que l'on soit à même de détecter une variation de tension aussi petite que $V_{ref}/2^n$, où $n$ est la résolution. Par ailleurs, le temps de conversion n'est pas constant, ce qui rend impossible l'analyse fréquentielle d'un signal échantillonné en continu.

Bien que déjà connus depuis un certain temps, d'autres types de convertisseurs commençaient à être disponibles commercialement : les sigma-delta. Le principe de base est de réaliser une modulation du signal d'entrée, en y ajoutant (sigma) ou en retranchant (delta) la valeur d'une tension de référence.

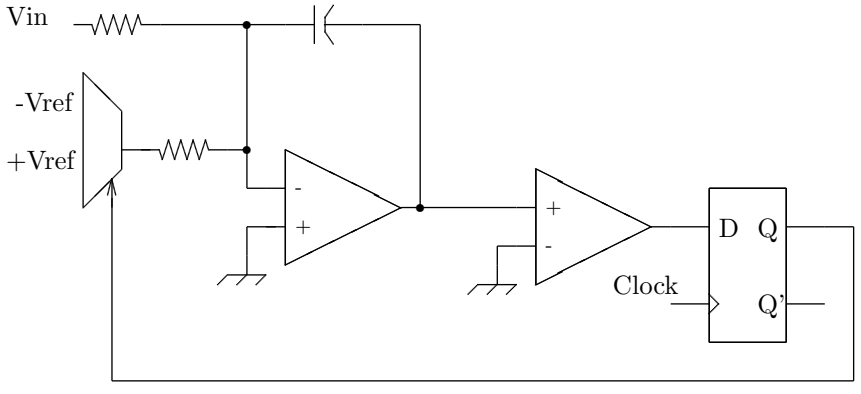



FIG. 2.2: principe de base du sigma-delta

Le schéma le plus simple d'une tel circuit est donné dans la figure 2.2. L'ampli-op intègre la somme de la tension d'entrée et d'une tension de correction. Sa sortie est comparée par rapport à zéro à intervalles réguliers via le D flip-flop, ce qui conditionne la rétroaction de façon à maintenir à zéro la valeur moyenne de la sortie de l'intégrateur. La rétroaction est donc *discrète* et *quantifiée*.

Voyons ce qui se passe pour une tension d'entrée négative de faible valeur. Un chronogramme est donné dans la figure 2.3. Lors de la première comparaison, la tension est supérieure à zéro, ce qui donne une sortie positive pour le comparateur, interprétée comme un "1" logique. Lors de la phase suivante, on intégrera +Vref et la sortie de l'intégrateur présentera une pente négative.

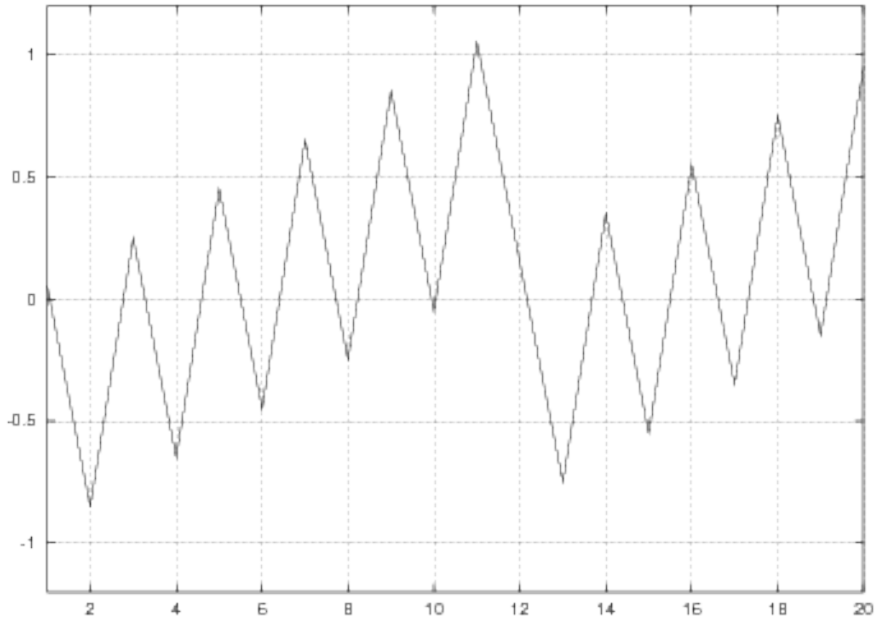
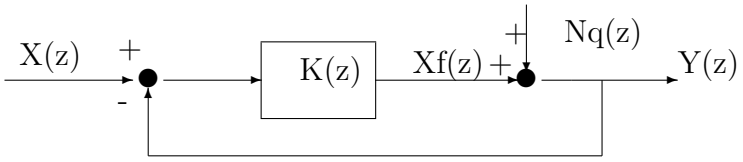



FIG. 2.3: évolution temporelle de la tension de sortie de l'intégrateur

Ce cycle élémentaire va être reproduit un certain nombre de fois; on constate une dérive positive des sommets consécutifs. Après 5 cycles de type "1 – 0", la $11^{\text{ième}}$ valeur est à nouveau "1", pour compenser la charge produite par la tension d'entrée, que nous avons choisie égale à $-1/10^{\text{ième}}$ de Vref. L'information sur la tension d'entrée se trouve en fait dans la densité de "1" et de "0". Cette information de densité est obtenue en prenant la moyenne du signal sur une fenêtre d'observation recouvrant plusieurs cycles.

Le signal passe donc par un filtre numérique passe-bas, qui va transformer le signal binaire de fréquence d'échantillonnage élevée en un signal à haute résolution et à fréquence d'échantillonnage plus faible, ce qui assure l'équivalence de l'information contenue dans chacun d'eux.

Une de ses particularités est que le bruit de quantification n'est pas distribué de manière uniforme sur la bande passante. Nous pouvons modéliser de manière générale la boucle de modulation de ce circuit par le schéma suivant :




Dans cette figure, $K(z)$ représente la fonction de transfert du filtre, et $Nq(z)$ le bruit de quantification. La relation entre $Y(z)$ et $X(z)$ s'écrit :

$$Y(z) = \frac{K(z)}{1 + K(z)} X(z) + \frac{1}{1 + K(z)} Nq(z) \tag{2.1}$$

pour un intégrateur du premier ordre, réalisé avec des capacités commutées, dont la fonction de transfert est

$$K(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{2.2}$$

cette expression se simplifie en

$$Y(z) = z^{-1}X(z) + (1 - z^{-1})Nq(z) \tag{2.3}$$

ou, en fonction de la fréquence

$$Y(z) = z^{-1}X(z) + z^{-\frac{1}{2}}\, 2\, j\, \sin(\frac{\pi\, f}{f_s})\, Nq(z). \tag{2.4}$$

Cette boucle présente deux particularités :

1. la fonction de transfert de l'entrée est unitaire;
2. la fonction de transfert du bruit est nulle à basse fréquence et maximale à la moitié de la fréquence d'échantillonnage via le terme en sinus.

Ce modulateur « colore » le bruit (noise shaping), ce qui réduit sa contribution dans la bande passante du signal. En pratique, le modulateur présentera une topologie différente, comportant une boucle du deuxième ou du troisième ordre, dont l'effet est d'accentuer ce phénomène. De ce fait, pour une résolution donnée, le rapport de suréchantillonnage est plus faible, ce qui simplifie la conception du circuit.

Le terme « convertisseur suréchantillonné » reflète cette opération consistant à perdre de la résolution temporelle au profit de la résolution sur le résultat.

Il s'agit d'une famille de convertisseurs en pleine expansion, et faisant l'objet de nombreuses recherches. Un problème typique est le suivant [48] : certaines valeurs de la tension d'entrée donnent lieu à des cycles répétitifs au niveau du modulateur, qui

dans les topologies d'ordre plus élevé peuvent perdurer alors que le signal d'entrée a changé. En pratique, cela se marque par une densité de bruit non-uniforme, présentant une série d'impulsions en fonction de la valeur du niveau d'entrée.

D'une manière générale, ce circuit comprend donc :

- un ou plusieurs intégrateurs, dont les performances ne sont pas cruciales : il est possible d'en corriger l'erreur de dérive.
- un ou plusieurs comparateurs. En cas d'erreur sur une comparaison, l'excédent de charge sera corrigé au cours des cycles suivants.
- des interrupteurs, afin de réaliser notamment des circuits à capacités commutées.
- des circuits logiques : flip-flop, registres, ...
- une unité de commande et de communication.

En pratique, ces différents constituants s'intègrent très bien en technologie CMOS, il en résulte une faible consommation électrique. À titre d'exemple, le circuit que nous utilisons, de type CS5507 (Crystal Semiconductor, à présent racheté par Cirrus), consomme 300 µA pour une résolution de 16 bits à 20 Hz.

### Haute résolution

Évaluons sommairement comment atteindre une résolution de 20 µK :

- la sensibilité d'une thermistance est de l'ordre de 4 %/K
- la tension d'alimentation du pont équivalente (après détection synchrone) vaut 1.23 Volts;
- la tension sur la diagonale détectrice du pont de Wheatstone vaut, pour un rapport de résistances unitaire

$$V_{out} = \frac{1}{4}\,\frac{dR}{R}\,V_{in} \tag{2.5}$$

Avec ces données numériques, la variation de tension de sortie correspondant à la résolution recherchée est de

$$V_{out} = 0.25 * 0{,}04 * 20\,10^{-6} * 1{,}23 \simeq 250\,nV \tag{2.6}$$

Expliquons brièvement cette valeur de 1.23 V : le pont est alimenté d'un côté par une tension triangulaire de 2,47 V de valeur de crête, et de l'autre côté par une tension de même amplitude mais en opposition. La valeur en crête-à-crête est donc de 4.94 V. Après redressement synchrone, nous « perdons » une des polarités, l'amplitude crête-à-crête est divisée par deux. Après filtrage, la valeur basse fréquence vaut la moitié de la valeur de crête, d'où l'amplitude de 1.23 V.

Le problème avec une tension à détecter aussi faible est le bruit généré par l'électronique. Nous exposerons par la suite la solution que nous avons implémentée pour résoudre au mieux ce problème.

**Faible consommation**

La faible consommation a été obtenue en utilisant des circuits intégrés spécifiques. Pour la partie digitale, nous avons choisi des circuits de la série CMOS de la série 74HC; le cas du convertisseur a déjà été évoqué. Pour la partie analogique, nous avons utilisé une série de composants fabriqués par la firme Linear Technology :

- LTC 1078 et 1079 : circuits à deux ou quatre amplificateurs opérationnels
- LTC 1043 : ce circuit comprend 4 interrupteurs commandés par une source commune. Il est prévu pour être utilisé dans des circuits à capacités commutées, et comprend un système d'annulation de la charge injectée à chaque commutation.

## 2.2 Schéma général

Voici à présent le schéma de principe complet (figure 2.4). Le pont est alimenté par deux signaux triangulaires en opposition. De ce fait, la diagonale détectrice, à proximité de l'équilibre, se trouve à une tension faible, ce qui évite les problèmes de mode commun. Le signal triangulaire est de pente ajustable et d'amplitude constante, ce qui est obtenu par un contrôle de le fréquence d'oscillation.

La sortie du pont arrive sur un ampli d'instrumentation de gain variable par décade, allant de 10 à 1000. La sortie de cet ampli passe par le redresseur synchrone, commandé pour commuter aux passages par zéro du signal amplifié (le pont est purement résistif). La sortie de ce redresseur, qui est unipolaire, passe ensuite dans un filtre de Bessel, d'ordre 3 et de fréquence de coupure de 3 Hz, ce qui permet de conserver les signaux d'origine biologique auxquels nous nous intéressons, et de rejeter le reste.

Ce signal à basse fréquence est évalué par le convertisseur sigma-delta, piloté par un PC assurant la lecture et l'analyse des données ainsi que l'affichage et le stockage des résultats.

## 2.3 Réalisation

Le schéma complet du prototype que nous avons développé se trouve dans la figure 2.5. La partie analogique comprend 6 ampli-op, l'amplificateur d'instrumentation et un comparateur. Passons-en plus en détail les différents constituants.

### 2.3.1 Générateur triangulaire stabilisé en amplitude

Ce système occupe la moitié gauche de la figure 2.5. Il s'articule autour d'une boucle à verrouillage de phase. La sortie de cette boucle passe par deux flip-flop de type J-K,

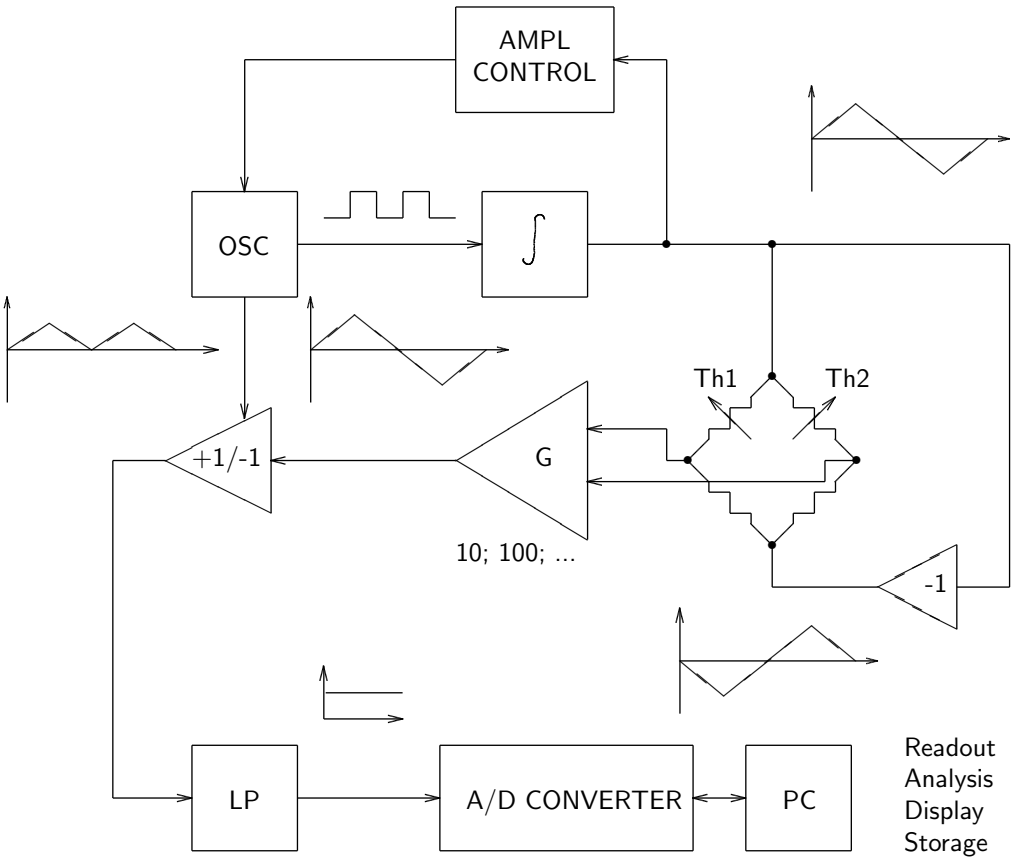



FIG. 2.4: Schéma de principe de la tête de mesure

qui fournissent deux signaux en quadrature : $P$ et $Q$. Le principe est d'effectuer un cycle (0, 0); (1, 0); (1, 1); (0, 1); ... Les transitions de l'un par rapport à l'autre sont donc décalées d'un échantillon. Le signal $P$ est intégré, ce qui donne un des signaux de sortie; l'autre sortie est obtenue par un montage inverseur, de gain $-1$. Ces deux signaux sont alternativement comparés à une tension de référence : après un basculement du comparateur, c'est l'autre signal qui est pris en compte. De la sorte, nous obtenons deux impulsions pour un cycle du signal triangulaire; ce sont ces impulsions qui servent de référence à la boucle à verrouillage.

La rétroaction s'effectue sur la fréquence, pour obtenir, à pente fixe, une amplitude constante.

De plus, pour minimiser les problèmes de dérive, nous avons alimenté les circuits logiques associés en +5 et -5 V : de cette façon, la valeur moyenne du signal carré attaquant l'intégrateur est nulle.

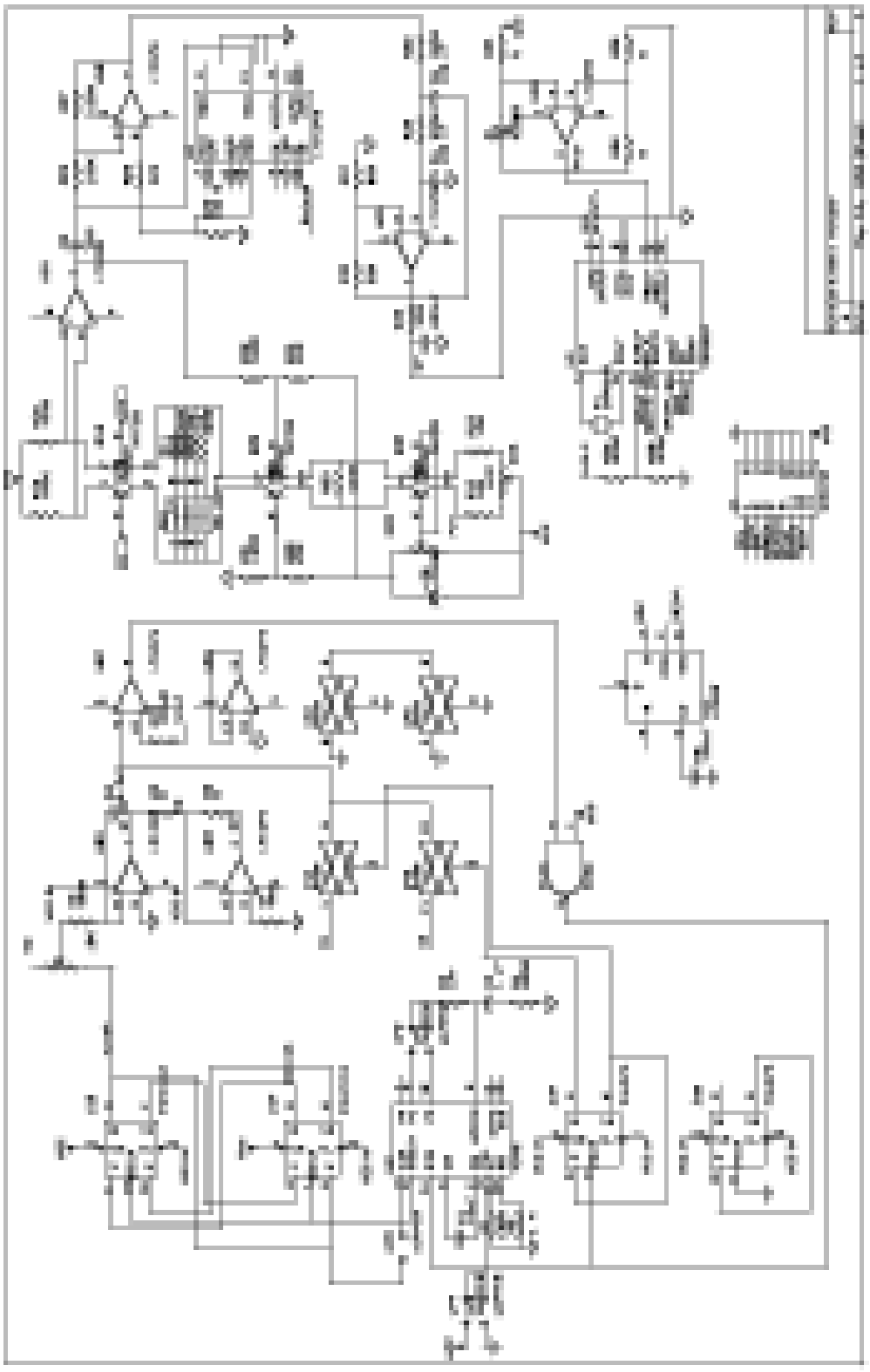


FIG. 2.5: Carte d'acquisition mono-canal

### 2.3.2 Amplificateur d'Instrumentation

Pour ne pas charger le pont, ce qui réduirait la sensibilité, il est nécessaire d'utiliser une topologie à deux ou trois ampli-op. Étudions-en les performances au niveau du bruit sans nous intéresser aux performances statiques; en effet, les problèmes de décalage de zéro et similaires seront éliminés par la détection synchrone.

Le pont a été dimensionné comme suit : considérant des thermistances dont la valeur nominale est de 30 kΩ à 25 °C, à la température de la peau (29 – 33 °C), elles auront une valeur approximative de 22 kΩ. Nous avons donc pris une résistance fixe associée de 22 kΩ, ce qui donne une impédance vue par l'amplificateur de 11 kΩ.

Considérons quatre configurations typiques :

1. OP-27, amplificateur opérationnel à faible bruit, consomme environ 5 mA;
2. OP-90, amplificateur opérationnel de type micropuissance, dont la consommation est de 20 µA;
3. OP-220, amplificateur opérationnel double, de type micropuissance, dont chaque élément consomme entre 60 et 100 µA;
4. MAT02, paire de transistors bipolaires, fonctionnant sous courant de polarisation de 10 µA par transistor.

Soit un circuit comportant un amplificateur, pour lequel les sources de bruit peuvent s'écrire comme une source de tension $e_n$ et de courant $i_n$ à l'entrée de celui-ci. Associant cet amplificateur à un circuit dont la résistance interne est $R_s$, nous pouvons en calculer le facteur de bruit (à 1 kHz) comme suit :

$$NF = 10\ \log_{10}\big(1 + \frac{e_n^2 + (i_n\,R_s)^2}{4\,k\,T\,R_s}\big)[dB] \tag{2.7}$$

Cela donne

| Circuit | $e_n$ [nV/Hz$^{1/2}$] | $i_n$ [pA/Hz$^{1/2}$] | NF [dB] |
|---|---|---|---|
| OP-27 | 3 | 0.4 | 0.6 |
| OP-90 | 90 | 0.8 | 16.7 |
| OP-220 | 40 | 0.09 | 10 |
| MAT-02 | 6 | 0.2 | 0.9 |

Nous constatons que la configuration avec des MAT-02 offre des performances similaires à celles d'un ampli-op de précision, pour une consommation électrique 250 fois plus faible.

Le circuit d'amplification est réalisé comme indiqué dans la figure 2.6, et apparaît dans la quatrième rangée verticale du schéma 2.5. Ce circuit s'articule autour de trois

paires de transistors appariés et d'un amplificateur opérationnel. La paire du bas, en base commune, sert de source de courant (et est représentée comme telle) et est alimentée par une référence de tension non-représentée. $V_1$ et $V_2$ sont les deux entrées, l'ampli-op fournissant la tension de sortie.

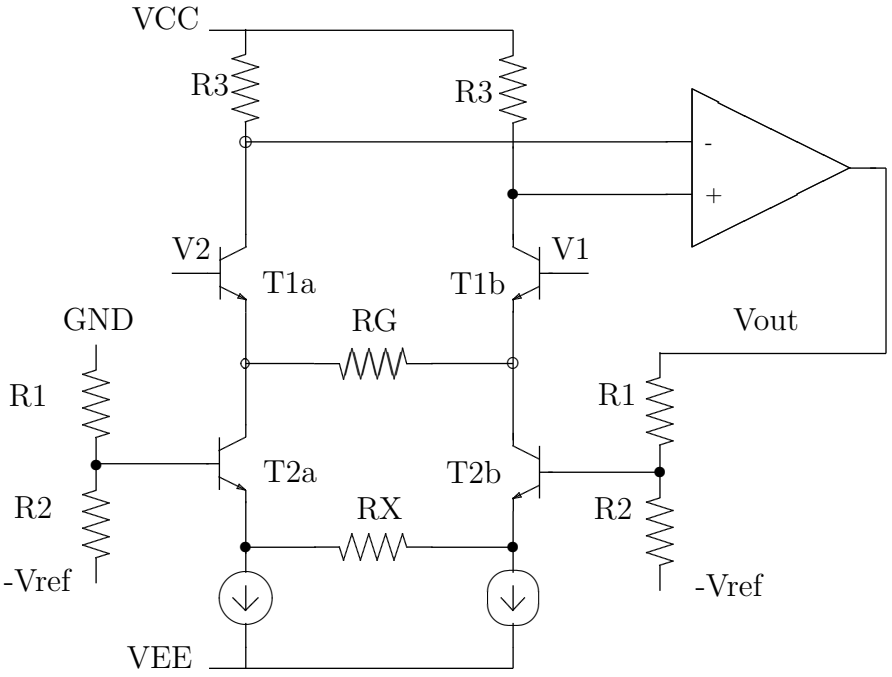



FIG. 2.6: Amplificateur différentiel à transistors appariés

Le gain de ce montage se calcule aisément en petit signal, moyennant l'hypothèse que le courant d'émetteur est sensiblement égal au courant de collecteur (le gain en courant est au minimum de 200), et que les transistors fonctionnent en paire différentielle.

Les courants collecteurs dans la seconde paire sont

$$(I_c,T_{2a}) = -\frac{R2}{R1+R2}\,\frac{V_{out}}{R_X} \tag{2.8}$$

$$(I_c,T_{2b}) = \frac{R2}{R1+R2}\,\frac{V_{out}}{R_X} \tag{2.9}$$

Dans la première paire, le courant collecteur sera la somme du courant collecteur de la première paire et du courant circulant dans $R_G$ :

$$(I_c,T_{1a}) = (I_c,T_{2a}) + \frac{V_2 - V_1}{R_G} \tag{2.10}$$

$$(I_c,T_{1b}) = (I_c,T_{2b}) - \frac{V_2 - V_1}{R_G} \tag{2.11}$$

L'ampli-op force ces deux expressions à être égales, d'où nous déduisons

$$V_{out} = (1 + \frac{R1}{R2}) \frac{R_X}{R_G} (V_2 - V_1) \tag{2.12}$$

Il s'agit donc bien d'un amplificateur différentiel, comportant deux facteurs de gain. Nous avons choisi le rapport entre $R_1$ et $R_2$ égal à 10; pris une valeur fixe de résistance pour $R_X$ et rendu $R_G$ réglable via une série d'interrupteurs de type "dip-switch".

### 2.3.3 Démodulateur synchrone

Il s'agit en fait d'un circuit à gain commuté. Il apparaît en haut à droite de la figure 2.5. Pour plus de clarté, les deux phases sont représentées séparément dans les figures 2.7 et 2.8.

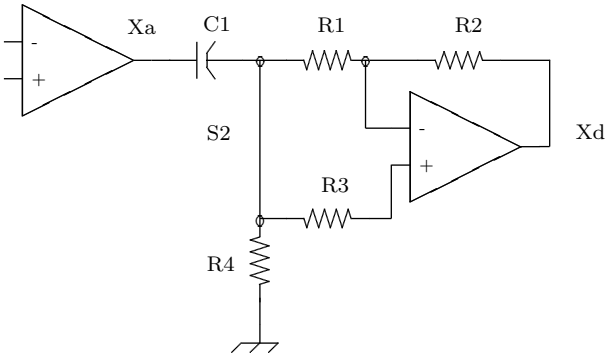



FIG. 2.7: Démodulateur : gain +1

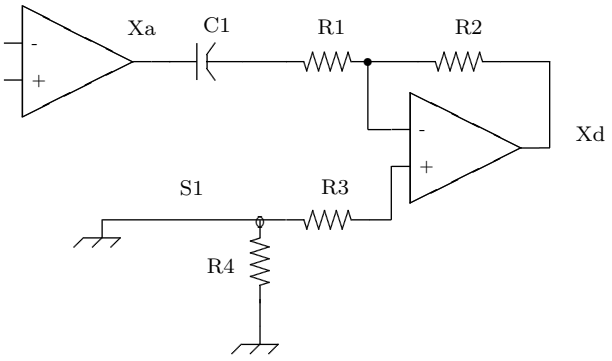



FIG. 2.8: Démodulateur : gain -1

Les valeurs de éléments sont :

- 100 kΩ pour $R_1$, $R_2$ et $R_4$;
- 49.9 kΩ pour $R_3$;

- 1 µF pour $C_1$;
- $S_1$ et $S_2$ sont les connexions établies par la fermeture de l'interrupteur correspondant du LTC1043.

$X_a$ est la tension amplifiée de l'étage précédent; $X_d$ la tension démodulée.

Dans la première configuration, il n'y a pas de chute de tension sur $R_1$, de ce fait il n'y en a pas non plus sur $R_2$. La tension de sortie est donc égale à la tension sur l'entrée positive, celle-ci étant identique à la tension d'entrée : le gain est unitaire. La résistance $R_4$ sert de charge pour la capacité, puisqu'il n'y a pas de courant dans $R_1$ ni dans $R_3$.

Dans la seconde configuration, nous avons un ampli inverseur de gain $-1$, d'impédance d'entrée égale à $R_1$.

Le rôle de la capacité, en association avec l'impédance d'entrée constante, est donc d'effectuer un filtrage passe-haut pour éliminer les composantes de décalage et à basse fréquence (50 Hz), pour que le signal présent soit aussi proche que possible d'un triangle afin que la commutation ait bien lieu aux passages par zéro.

La valeur de $R_3$ est telle que l'impédance vue par chacune des entrées de l'amplificateur opérationnel soit identique, de façon à minimiser l'effet des courants de polarisation.

### 2.3.4 Filtre passe-bas

Nous avons choisi un filtre de Bessel d'ordre 3, afin de minimiser la distorsion. Ce filtre est réalisé avec un amplificateur opérationnel suivant le principe du VCVS (Voltage-Controlled Voltage-Source), et réalise une amplification supplémentaire de 1.27. Il s'agit du circuit marqué “U6B”, au milieu de la dernière rangée de la figure 2.5.

### 2.3.5 Tension de référence

Une tension de 2,47 V est générée à partir d'une référence de 1,23 V par doublement de sa tension via un ampli-op; une résistance auxiliaire connectée à l'alimentation négative assure un démarrage correct. À l'allumage, le circuit fonctionne en réaction positive et est susceptible de partir indifféremment en saturation négative ou positive. Cette résistance assure un démarrage vers la saturation positive, ce qui polarise correctement la référence. À partir de ce moment, la résistance dynamique de cet élément devient faible, et l'ampli-op se retrouve globalement en contre-réaction négative. Cet élément marqué “U6B” apparaît au bas de la dernière rangée de la figure 2.5.

### 2.3.6 Calcul théorique de la sensibilité

Nous avons à présent en main tous les éléments pour calculer la sensibilité théorique du circuit avec un gain de 10 :

1. le sigma-delta travaille avec une référence de 2,47 V. En bipolaire, un échantillon correspond à une tension de $2.47/2^{16}$ V, ou encore 37.7 µV.
2. le filtre de Bessel et l'amplificateur ont un gain combiné de 12,7;
3. la plus petite quantité détectable vaut donc environ 3 µV;
4. il y correspond une différence de température de

$$\frac{310^{-6}}{0.04 * 0.25 * 1.23} \simeq 240\mu K \tag{2.13}$$

### 2.3.7 Calcul du bruit

En ce qui concerne le bruit, la bande passante étant de 3 Hz, la valeur efficace du bruit vaut donc

$$e_t = \sqrt{[(4\,k\,T\,R_s) + e_n + R_s\,i_n]\,B} \simeq 26nV \tag{2.14}$$

Si nous considérons une valeur crête-à-crête de 6 $\sigma$, avec un gain de 10, le demi-pas de quantification vaut environ 10 fois la valeur en crête-à-crête du bruit, ce qui est admissible. En gain de 100, ce nombre tombe à 1, ce qui signifie que le dernier bit risque d'être inexploitable tel quel.

## 2.4 Communication avec la carte Lab-PC

Cette carte d'acquisition devant être pilotée par ordinateur, nous avons utilisé une carte Lab-PC de la firme National Instruments dans ce but, celle-ci disposant de 24 lignes d'entrées/sorties digitales.

Nous avons d'abord utilisé la ligne série du convertisseur en mode pilotage externe, ce qui requérait d'effectuer une scrutation continuelle de la patte $\overline{\text{DRDY}}$ afin de détecter la fin des conversions.

Pour diminuer la charge de l'ordinateur, nous avons ensuite essayé d'utiliser le mode d'acquisition synchronisé de la carte Lab-PC : dans ce mode, un nouveau transfert digital a lieu sur le flanc d'un signal d'horloge, ce transfert étant géré par une routine d'interruption. Dans ce cas, l'horloge série était générée par le convertisseur et en phase avec son horloge interne, ce qui minimise les problèmes d'interférence.

Cette solution n'a pas fonctionné dans l'environnement de développement LabWindows : il est apparu que le latence de la *première* interruption était de l'ordre de 50 µs, ce qui était trop élevé, un cycle d'horloge à 32 kHz durant 30 µs.

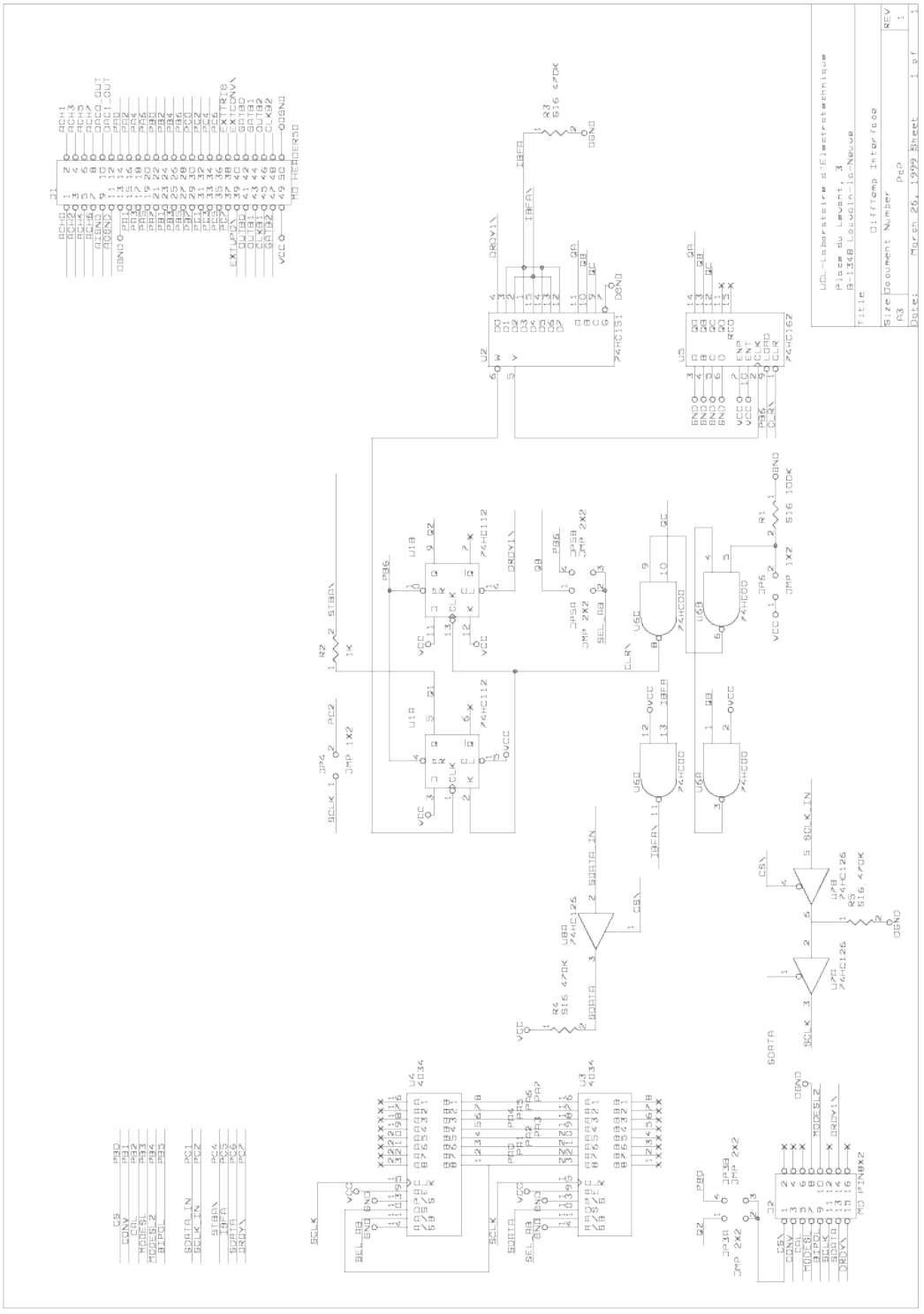


FIG. 2.9: Interface tête de mesure/carte Lab-PC

Cela se passait avant nos travaux sous Linux, nous avons contourné le problème en construisant un petit interface réalisant la conversion série/parallèle. De cette façon, la carte Lab-PC vient lire deux mots de huit bits à la fin d'une transmission.

Le plan de cette carte est donné dans la figure 2.9. La conversion série/parallèle s'effectue par deux circuits de type 4034.

## 2.5 Résultats

### 2.5.1 Consommation électrique

En fonctionnement continu, nous avons mesuré une consommation de 2 mA. Combiné à une pile alcaline de 9 V, d'une capacité de 1 Ah, l'autonomie sera de 500 heures, ou encore 3 semaines.

### 2.5.2 Stabilité à court terme

La figure 2.10 montre la stabilité à court terme du système. Les deux thermistances ont été placées côte-à-côte dans une enceinte régulée à 37 °C. L'enregistrement s'est effectué avec un gain de 100. Les résultats oscillent autour d'une valeur moyenne : en augmentant le temps d'intégration, il est donc possible de réduire l'influence du bruit et d'accroître la résolution. La déviation standard est d'un-demi LSB, et l'écart maximum par rapport à la moyenne de 1,7 LSB.

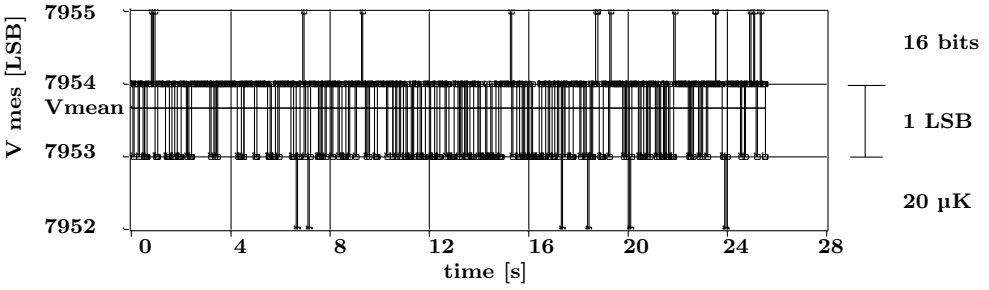



Fig. 2.10: Stabilité à court terme du thermomètre différentiel

### 2.5.3 Emploi avec un accéléromètre

Le système que nous avons développé est utilisable avec des capteurs placés en pont, pourvu qu'ils soient purement résistifs, et avec une impédance d'au moins 10 kΩ par capteur, afin de préserver le caractère micro-puissance.

Nous l'avons testé avec un accéléromètre piezo-résistif, et obtenu, par un processus de moyenne, une stabilité équivalente à 20 bits. Des tests de vibration ont montré une bonne concordance entre la fréquence d'excitation et le spectre du signal mesuré.

Nous avons utilisé ce capteur pour réaliser un inclinomètre, et obtenu un écart-type de 0.1 degré entre le cosinus de l'angle et le résultat mesuré, ce qui correspond à l'incertitude du système de positionnement.

## 2.6 Calibration du thermomètre différentiel

Une calibration complète en a été réalisée comme suit : les deux capteurs ont été placés dans des blocs de polystyrène expansé en présence d'un transistor utilisé comme élément chauffant et d'un autre capteur servant à mesurer et à stabiliser la température absolue.

Une série de mesures ont été effectuées, en balayant par pas de 1 °C, entre 29 et 37 °C pour l'un des capteurs, et en prenant 8 pas couvrant la plage d'entrée du convertisseur pour l'autre capteur. De la sorte, nous avons obtenu une soixantaine de points.

Une mesure consiste en une période de stabilisation de 5 minutes, suivie d'une période de prise de résultats durant également 5 minutes, la valeur finale étant la moyenne sur 64 échantillons. Ces résultats sont valables pour des échelons positifs (augmentation de température); la période de stabilisation pour un échelon négatif étant un peu plus longue (15 minutes), ce qui mène à une durée de 20 minutes pour la prise de résultat.

Nous avons complètement automatisé cette opération d'échantillonnage, celle-ci durant en effet 12 heures !

Au terme de cette acquisition, il a été possible d'obtenir les résultats suivants:

- gain 10 : sensibilité de 230 µK, déviation standard entre les pentes des différentes relations : 0.7%;
- gain 100 : sensibilité de 23 µK, déviation standard de 2%;
- réjection du mode commun, c.-à-d. indépendance par rapport à la température globale : 56 dB en gain 10, et 65 dB en gain 100.

Chapitre 3

# Électronique d'acquisition : carte multi-canaux

## 3.1 Modifications apportées

### 3.1.1 Synchronisation de l'oscillateur

Dans le premier prototype, l'oscillateur était construit autour d'une régulation de fréquence pour maintenir fixe l'amplitude.

Au niveau des convertisseurs de type sigma-delta, la réjection d'une fréquence parasite est excellente si celle-ci est un sous-multiple de la fréquence pilotant le convertisseur.

Pour éviter tout risque d'interférence via des couplages au niveau des circuits électroniques entre le circuit de génération des ondes triangulaires et le convertisseur, nous avons décidé de piloter ce générateur par une fréquence dérivée par comptage de la fréquence fondamentale du convertisseur. Il restait à résoudre le problème de la stabilisation de la dérive de l'intégrateur.

Par ail leur, le modèle précédent ne permettait pas d'arrêter cet oscillateur. Dans une logique de minimisation de la consommation électrique, nous avons souhaité pouvoir l'arrêter et mettre l'alimentation du pont à la masse.

De plus, pour simplifier l'interface et réduire la consommation[1], nous avons choisi d'alimenter les circuits logiques entre 0 et 5 V : de ce fait, l'onde carrée à intégrer n'est plus de valeur moyenne nulle.

Nous avons résolu ces problèmes via le circuit dont le schéma est donné dans la figure 3.1.

1. pour les circuits CMOS, la consommation est une fonction quadratique de la tension d'alimentation

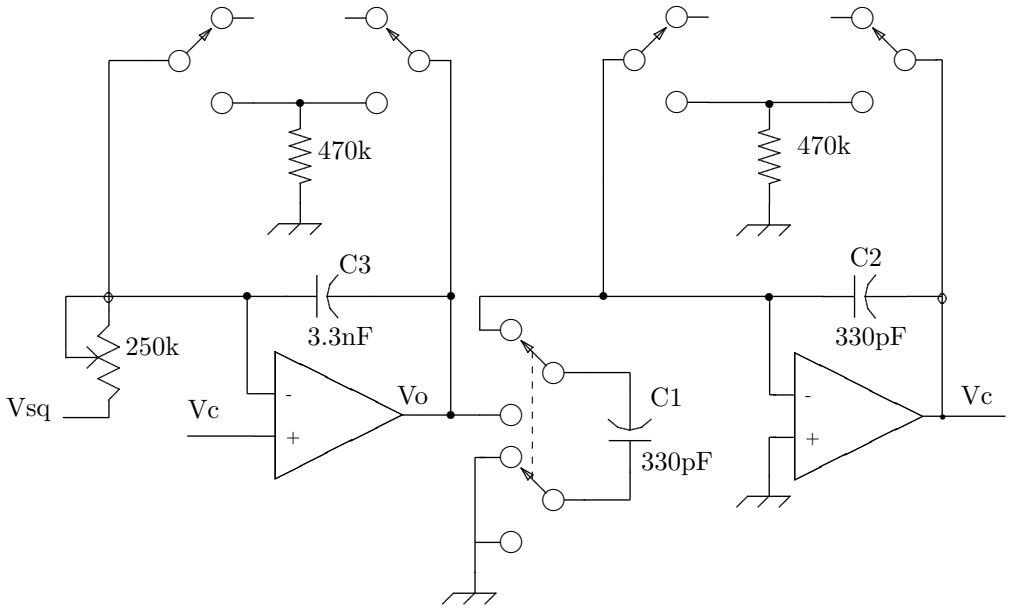



FIG. 3.1: Oscillateur asservi à compensation de dérive

Ce circuit reçoit un signal périodique carré, de moyenne non-nulle, via $V_{sq}$. Le courant d'entrée de l'intégrateur vaut la différence entre cette tension et une tension de contrôle, $V_c$, fournie par l'autre ampli-op.

les deux interrupteurs couplés sont commandés par un signal en quadrature avec $V_{sq}$. Ce signal présente donc des commutations au moment où le signal triangulaire devrait être nul. La tension de sortie $V_o$ charge la capacité $C_1$. Au moment de la commutation, la charge résiduelle de cette capacité est transférée à l'entrée du second ampli-op connecté en amplificateur de charge. Si $V_o$ est nulle au moment de la commutation, il n'y a pas de transfert de charge, et la tension $V_c$ reste constante. Sinon, du fait que les capacités sont égales, $V_c$ va faire un saut dont l'amplitude vaudra l'opposé de $V_O$. Étant donné que la tension sur la capacité de l'intégrateur ne peut varier instantanément, le saut de tension d'amplitude $-V_o$ sur l'entrée positive est répercuté sur la sortie, celle-ci va donc tendre vers zéro à l'instant du transfert. Après quelques cycles, la tension de correction va devenir constante et la tension $V_0$ sera bien nulle aux instants de commutation. De la même façon, toute tendance à la dérive sera également corrigée.

Le rôle des quatre interrupteurs situés en haut du circuit est de permettre de bloquer l'oscillateur. Une fois commutés dans l'autre position, l'ampli-op travaillant en amplificateur de charge devient un suiveur de tension et va présenter à sa sortie une tension nulle. De ce fait, l'intégrateur va devenir un simple amplificateur inverseur, de gain zéro (la résistance de contre-réaction est un court-circuit) : sa sortie va également s'annuler, et de ce fait le pont ne sera plus alimenté. Le rôle des deux résistances de 470 kΩ est d'éviter que les sorties des interrupteurs restent flottantes lorsqu'ils sont inutilisés.

Ces quatre interrupteurs sont commandés par un signal provenant du sigma-delta et pouvant prendre 3 états :

- "0" logique : convertisseur actif, l'horloge série est externe;
- au milieu des tensions d'alimentations de la partie digitale : le convertisseur est à l'arrêt complet, y compris son oscillateur;
- "1" logique : le convertisseur est actif et génère l'horloge série.

Ainsi, le pont n'est alimenté que lorsque l'oscillateur du convertisseur est actif. Sans cela, quel que soit le signal présent à l'entrée de l'intégrateur, celui-ci finirait par arriver en saturation, amenant la pleine tension d'alimentation aux bornes du pont.

### 3.1.2 Augmentation du nombre de canaux d'entrées

Les campagnes de mesure thermique réalisées avec le premier prototype nécessitaient un temps considérable. Une mesure commençait par le placement du capteur et une phase de stabilisation de 20 minutes, suivie de l'acquisition proprement dite, durant environ 15 minutes. Au total, la mesure consécutive sur quatre sites prenait plus de deux heures.

Afin de réduire ce délai et de travailler dans les mêmes conditions de couplage, nous avons souhaité augmenter le nombre de canaux d'entrées. Une autre retombée possible est la possibilité d'effectuer une cartographie thermique, en étudiant une répartition de température.

Il s'est avéré que la solution la plus simple, étant donné qu'il s'agit de phénomènes à variation lente, consistait à placer un multiplexeur à l'entrée de l'amplificateur d'instrumentation. Nous avons également envisagé d'autres solutions :

- multiplexage au niveau de l'alimentation des capteurs, $n$ branches en parallèle, une seule alimentée : dans cette configuration, le régime thermique des capteurs n'aurait pas été constant;
- multiplexage à l'entrée du détecteur synchrone : cette configuration requiert un amplificateur par canal;
- multiplexage au niveau de l'acquisition (sigma-delta multicanaux) : cette solution aurait nécessité de dupliquer complètement les étages d'amplification, redressement synchrone et filtrage.

Pour conserver un pilotage aussi simple que possible de ce système, nous avons choisi un multiplexeur de type LTC1390 (Linear Technology). Il s'agit d'un multiplexeur à commande série 3 fils, avec interface bidirectionnelle.

Le câblage des commandes se fait comme suit :

- les signaux SCLK et $\overline{\text{CS}}$, qui représentent respectivement l'horloge série et le sens du transfert, proviennent de la carte Lab-PC et sont communs au multiplexeur et au convertisseur;

– la ligne de donnée provenant de l'interface Lab-PC aboutit sur la patte DATA 1 du multiplexeur, tandis que la patte DATA 2 est connectée à la patte SDATA du sigma-delta. Ce mécanisme permet d'implémenter une ligne série bidirectionnelle.

Le sens de transfert des données à l'intérieur du multiplexeur dépend de la valeur de $\overline{\text{CS}}$ :

– un "1" logique signifie que l'interface série du convertisseur n'est pas sélectionnée. Dès lors, on peut écrire la valeur du prochain canal à activer ou désactiver complètement le multiplexeur. Cette situation deviendra effective au prochain changement d'état de $\overline{\text{CS}}$. Les données transitent de DATA 1 vers DATA 2, mais l'interface comprend un registre à décalage, qui « consomme » les quatre premiers bits reçus et transmet les suivant, permettant ainsi de cascader plusieurs multiplexeurs.
– un "0" logique est une demande de lecture sur le convertisseur; dès lors les données transitent de DATA 2 vers DATA 1.

Cette solution nous paraît simple et élégante, puisque l'interface est réduite à son expression la plus simple.

### 3.1.3 Mesure de l'amplitude

Ayant changé radicalement la méthode de pilotage de l'oscillateur, il restait à résoudre le problème de l'évaluation de l'amplitude : en effet, celle-ci est à présent ajustable, et nous devons nous prémunir contre des phénomènes de dérive lente.

Pour cela, nous avons fait subir à la tension d'alimentation du pont les mêmes opérations que le signal amplifié, à savoir redressement synchrone et filtrage. Cela n'a pas nécessité de composants supplémentaires; il restait en effet un interrupteur libre sur le LTC1043.

Cette tension continue, image de l'amplitude de l'alimentation du pont, est devenue la référence du sigma-delta. Ainsi, nous obtenons une mesure *ratio-métrique*, puisque le convertisseur sigma-delta évalue cette fois-ci le rapport entre la tension diagonale et la tension d'alimentation.

## 3.2 Réalisation

Le schéma complet du nouveau prototype que nous avons développé se trouve dans la figure 3.2. Voyons-en à présent les changements majeurs.

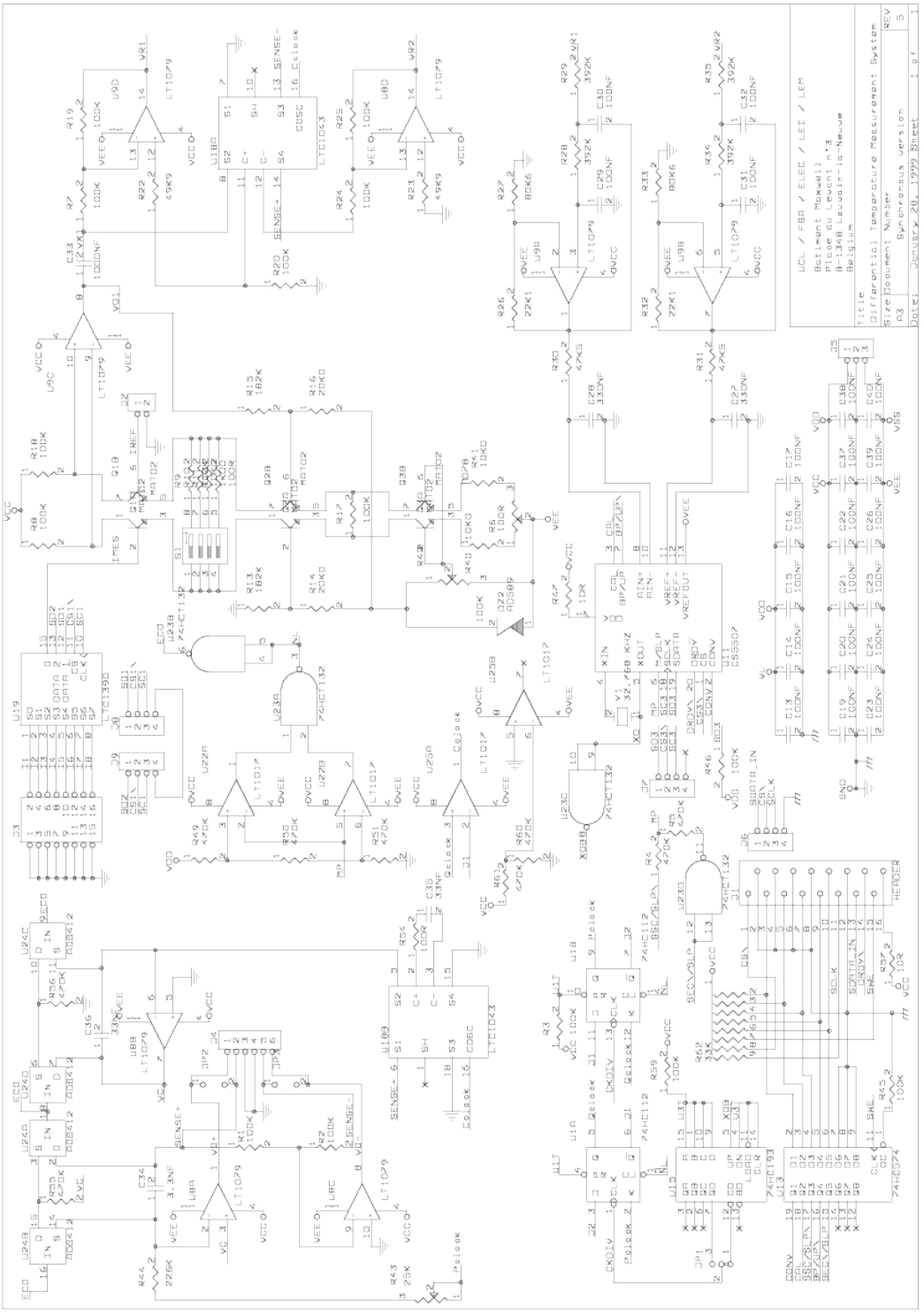


FIG. 3.2

1. la structure de l'oscillateur a été modifiée, il est à présent prévu pour fonctionner sur un sous-multiple de la fréquence du quartz du convertisseur, obtenu par un compteur (74HC193);
2. l'oscillateur est corrigé via un circuit auxiliaire, et peut être arrêté via 4 interrupteurs (AD6412);
3. la sortie analogique comprend à présent une rétroaction (SENSE) pour compenser les chutes de tension dans les connections ou pour pouvoir ajouter un élément-tampon (transistor) en sortie;
4. il y a 8 entrées analogiques, arrivant sur un multiplexeur (CS1390);
5. la sortie analogique est également démodulée et filtrée pour fournir la référence du convertisseur;
6. l'adaptation de niveau entre la partie digitale (0 – 5 V) et le LTC1043 (± 5 V) se fait via un comparateur LT1017; pour accélérer la vitesse de commutation, ses entrées sont pilotées par un signal logique et son complément;
7. un circuit auxiliaire surveille l'activité du sigma-delta et coupe la sortie analogique lorsque le convertisseur est à l'arrêt;
8. pour réduire les interférences, un verrou (74HC574) est interposé entre les signaux arrivant de la carte Lab-PC et le convertisseur. De cette façon, même en cas d'activité sur le bus, il n'y a pas de signal transmis sur la carte d'acquisition.

En ce qui concerne les composants, analogiques, il y a donc 8 ampli-op et 4 comparateurs (dont un n'est pas utilisé), ce qui est un nombre similaire à ce que l'on avait sur le premier prototype.

Au niveau de l'interface entre la carte d'acquisition et la carte Lab-PC, la seule modification notable a été de rendre le bus série bidirectionnel. Avec le prototype précédent, ce bus ne fonctionnait que dans le sens sigma-delta vers série/parallèle au niveau des données, et fonctionnait en concordance avec le mode du sigma-delta au niveau horloge.

Dans ce cas-ci, la carte Lab-PC reçoit puis envoie des données, ce qui est géré par un circuit de type 74HC126.

Cette carte a été complètement testée et s'avère tout à fait fonctionnelle.

Chapitre 4

# Aspects logiciels : acquisition de données temps réel sous Linux

## 4.1 Introduction

Nos premiers travaux ont visé à développer un prototype de système d'acquisition dont l'élément central était un convertisseur de type sigma-delta. Sans un pilotage adéquat, il était impossible d'exploiter ses résultats. Le laboratoire possédant des cartes d'acquisition de type Lab-PC de la firme National Instruments, nous avons commencé par écrire un programme dans l'environnement LabWindows.

Cet environnement provient de cette même firme, et est destiné à fournir un certain nombre de facilités de programmation pour exploiter les diverses cartes d'acquisitions dont elle assure la commercialisation.

Bien vite, nous avons ajouté des possibilités allant au-delà du simple pilotage du dispositif, et il est devenu difficile de gérer la source à cause de son volume. Par ailleurs, le fait de devoir mélanger des fonctions de bas niveaux (gestion des entrées-sorties digitales) et de haut niveau (gestion de l'interface graphique) n'allait certainement pas dans le sens de la clarté du code.

De plus, le convertisseur n'ayant pas de mémoire tampon, il fallait pouvoir garantir une latence maximale entre le signalement par le convertisseur de la fin d'une conversion et la lecture du résultat, c.-à-d. travailler dans un cadre de programmation qui soit strictement temps-réel, même si a priori la cadence d'acquisition est très faible : 20 échantillons/sec au maximum pour le sigma-delta, soit 320 bits/sec à transmettre.

Nous avons souhaité poursuivre le développement d'un logiciel d'acquisition général dans un environnement multi-tâche et temps-réel, à savoir Linux. L'occasion s'en est présentée dans le cadre du développement d'un système de monitoring des rythmes

cardiaques et respiratoires, où nous avions d'une part à acquérir des signaux électriques et d'autre part à effectuer un filtrage pour en extraire les rythmes cardiaques et respiratoires.

## 4.2 Définition du cadre

En nous engageant dans cette voie, nous avons poursuivi une série d'objectifs que nous allons passer en revue.

### 4.2.1 Séparation des fonctionnalités

En génie logiciel, l'approche dite « orientée objet » rencontre un succès croissant lorsqu'il s'agit de d'implémenter des fonctionnalités très complexes. L'idée maîtresse est de ne plus analyser les fonctionnalités sous l'angle de vue des étapes à effectuer, mais à considérer le cycle de vie des données ainsi que leurs transformations. Pour cela, l'application est divisée en modules appelés objets communicant les uns avec les autres. Chaque module est responsable d'un aspect particulier du cycle de vie, et ils sont conçus de telle sorte que le fonctionnement de l'un ait peu de retentissement sur les autres.

Dans le même ordre d'idée, nous avons voulu séparer clairement l'acquisition en une série de niveaux bien distincts :

- gestion de bas niveau de la carte Lab-PC par programmation de ses registres internes tout en offrant une interface plus évoluée masquant les détails de fonctionnement;
- processus générique d'acquisition de données, avec une attention particulière pour les aspects robustesse et fiabilité;
- interface utilisateur : pilotage de l'acquisition, analyse et visualisation des résultats de conversion.

### 4.2.2 Architecture distribuée

En outre, Il nous est apparu utile de scinder l'application globale en deux parties suivant une approche de type client-serveur. Il y avait en effet deux tâches bien distinctes à effectuer, soumises à des contraintes très différentes :

- la collecte des données, qui ne demandait guère de ressources au niveau du processeur, mais dont on devait pouvoir garantir la fiabilité;

– le traitement numérique de ces données, requérant une puissance de calcul suffisante pour pouvoir effectuer un filtrage au vol, mais avec des contraintes moins strictes en ce qui concerne le temps de réponse : avoir un affichage un peu sautillant n'est guère problématique.

Pour réaliser une telle décomposition, il fallait au minimum que le système d'exploitation soit multi-tâches, ce qui n'était pas le cas de LabWindows.

Après avoir retracé l'évolution de Linux, nous expliquerons plus en détail quelles étaient les limitations que nous avons voulu dépasser et comment Linux a permis d'atteindre ces objectifs. Nous présenterons ensuite plus en détail le logiciel d'acquisition, et nous conclurons en parlant d'une façon plus générale des bénéfices que peut apporter Linux au niveau de l'instrumentation sur ordinateur.

## 4.3 Présentation de Linux

En 1991 [56], un étudiant de l'université d'Helsinki du nom de Linus Benedict Torvalds s'achète un PC avec un processeur Intel 386. Souhaitant en étudier le fonctionnement, il se lança dans l'aventure Unix. À l'époque, il existait des systèmes commerciaux comme Unix Sytem V R4 (AT&T Bell Laboratories), SCO Unix, Xenix et un système « gratuit »[1], FreeBSD, celui-ci étant une version libre d'accès du système d'exploitation BSD de l'université de Berkeley.

Le développement de FreeBSD étant assez « fermé » (limité à un groupe de développeurs de référence), Linus commença à écrire un ensemble de programmes sous Minix, un système d'exploitation didactique développé par Andries Tanenbaum. Puis il ajouta des fonctionnalités, récrivit certaines parties, ... En août 1991, il diffusa sur Internet son embryon de système d'exploitation, de manière libre, sous le nom de Linux (contraction de Linus et de Unix).

Des développeurs s'intéressèrent à ce clone d'Unix, envoyèrent à Linus des corrections ainsi que des programmes implémentant de nouvelles fonctionnalités : le mouvement était lancé.

Ce modèle de développement est assez particulier, puisqu'il s'agit d'une mise en commun de compétences, de temps et d'énergie. Une synthèse intitulée « The cathedral and the bazaar »[2], compare le développement fermé (the cathedral) à celui,

1. Remarquons l'ambiguïté de ce terme, traduction de l'anglais "free". Comme le dit R. Stallman : « It's "free" like in "free speech", not "free beer" ». "Free" est à comprendre comme liberté d'accès au logiciel *et à ses sources*, tout en admettant qu'une personne ou une société puisse demander une contribution financière pour les frais de distribution ou les services prestés en relation avec ce logiciel, par exemple l'adaptation ou la modification à la demande et pour le compte d'un client. Par corollaire, cette liberté d'accès peut aussi signifier la gratuité, puisque la transmission du logiciel est explicitement autorisée.

2. http://www.earthspace.net/~esr/writings/cathedral-bazaar/cathedral-bazaar.html

ouvert, de Linux (the bazaar). En résumé, le développement de type cathédrale est assuré par un nombre restreint de personnes qui décident le comment et le pourquoi de l'évolution. Une nouvelle version contient en général un nombre de modifications assez significatif, mais sera séparée de la précédente par un laps de temps important. Par opposition, le bazar est la confrontation d'idées et de solutions. Plusieurs solutions seront apportées à un problème, et la meilleure en terme d'impact ou de performance sera conservée. Les versions apparaissent à un rythme rapide, et sont destinées à tester de façon incrémentale les nouvelles caractéristiques.

## 4.4 Développements précédents

Notre logiciel d'acquisition a d'abord été écrit pour un environnement spécialisé commercialisé par la firme National Instruments : LabWindows. Il s'agit d'un système d'exploitation de type DOS étendu, avec accès transparent à la totalité de la mémoire, alors que par défaut un programme DOS ne peut accéder qu'à 640 K de mémoire pour son code et ses données. Parmi les caractéristiques principales, nous pouvons citer :

- système fondamentalement mono-tâche;
- facilités de programmation pour les cartes d'acquisition de National Instruments;
- facilités de développement d'interfaces graphiques (panneaux, menus déroulant, indicateurs, ...).

L'approche générale est de type « programmation événementielle » : typiquement, la tâche principale d'un programme est d'attendre qu'un événement se produise pour ensuite y répondre. De cette façon, des routines d'acquisition peuvent tourner en arrière-plan sans qu'il y ait de tâches à l'avant-plan consommant des ressources du processeur. Néanmoins, cette technique a ses limitations, notamment à cause de l'interface graphique : il faut en effet continuer à mettre à jour le curseur en fonction des déplacements de la souris, activer les menus déroulants, ...

Le programme que nous avons développé présentait des ratés aléatoires, sans que nous ayons jamais pu mettre le doigt sur la cause de ces erreurs avec le débogueur fourni. Par ailleurs, nous n'avons pu obtenir de support du personnel de la firme, notre version du logiciel ayant été déclarée obsolète.

Enfin, nous aurions souhaité pouvoir utiliser dans notre programme les possibilités de génération d'interruption de cette carte Lab-PC. Pour cela, il aurait fallu disposer d'informations sur la manière de programmer ces opérations. Ce type de renseignements était disponible via une clause de confidentialité ("non-disclosure agreement"). À l'époque où nous en avons effectué la demande, nous avons reçu une fin de non-recevoir : « votre version du produit est trop ancienne ».

Le compilateur du logiciel Lab-PC ne pouvant travailler que sur une seule source, il avait fallu créer un programme de taille très importante, mélangeant des fonctionnalités très hétérogènes.

Ceci illustre à nos yeux quelques-uns des problèmes fondamentaux liés aux logiciels commerciaux :

- obsolescence rapide, nécessité de renouveler régulièrement sa licence et de mettre à jour ses applications pour « profiter » des améliorations. De même, dans le cas d'application générant des fichiers binaires, une nouvelle version introduit en général des incompatibilités vis-à-vis de la précédente. Dans le cas d'un traitement de texte, cela implique qu'à partir du moment où une majorité de correspondants ont effectué une mise à jour, on est soi-même obligé de faire de même afin de pouvoir continuer à échanger des textes.
- impossibilité pour le programmeur qui en possède les compétences nécessaires de résoudre des problèmes ou d'apporter des améliorations car il n'y a pas d'accès aux sources.
- manque de suivi à long terme.

## 4.5 La solution sous Linux

Dans le cas de Linux, nous avons pu implémenter les objectifs que nous avons cités dans la section 4.2.

### 4.5.1 Tâche d'acquisition

L'ensemble des fonctionnalités est reparti sur un ensemble de niveaux :

- gestion de bas niveau (Device Driver) de la carte, par accès direct aux registres internes de celle-ci. Cette partie est directement intégrée au noyau et bénéficie d'une priorité élevée pour communiquer avec la carte;
- programme d'acquisition en mode utilisateur, communicant avec le gestionnaire à travers un interface standardisé;

L'avantage principal d'une telle approche réside tout d'abord dans la simplicité, puisque chaque niveau implémente un ensemble restreint de fonctionnalités bien définies. Chaque couche peut donc être testée simplement et indépendamment des autres. De plus, la généricité de l'approche fait que les diverses couches peuvent être exploitées séparément. Utiliser une autre source de mesure ou réaliser d'autres traitements n'implique que de modifier une partie de l'application.

### 4.5.2 Traitement numérique

Le cadre de développement comprend des primitives de communication par réseau. De cette façon, l'architecture distribuée s'implémente très simplement en utilisant des

fonctionnalités de base. Cette approche permet de réaliser un gain de temps durant la phase de mise au point.

## 4.6 Device Driver sous Linux

Nous avons trouvé sur le site du LLP (The Linux Lab Project[3], un site consacré à l'acquisition de données sous Linux) un gestionnaire de carte Lab-PC pour Linux, développé par un australien du nom de Glenn Moloney. Ce driver était loin d'être achevé, mais il avait le mérite d'exister. La partie la plus complète concernait le volet analogique. Au niveau des entrées/sorties digitales, il n'était possible que d'utiliser le mode direct, c.-à-d. des accès sans aucune possibilité de synchronisation.

Dans un premier temps, nous y avons ajouté les primitives dont nous avions besoin, à savoir une routine d'interruption permettant d'effectuer des transferts digitaux en mode synchronisé. Nous avons ensuite crée une « abstraction » du convertisseur sigma-delta, pour le faire apparaître comme une entité à part entière. L'idée a été de masquer les détails de fonctionnements liés au ports digitaux d'entrée/sortie, et de présenter une interface de plus haut niveau : calibrer le sigma-delta, effectuer une mesure, ...

Ce genre de module utilise un mécanisme de communication générique connu sous le nom de "Virtual File System" ou « Système Virtuel de Fichiers » (SVF). Lors des premiers développements de Linux, il est en effet apparu intéressant de gérer plusieurs types de systèmes de fichiers (minix, ms-dos, ext, ...). Le concept de SVF, apparu en 1986 dans SunOS, facilite ce genre d'opération : il donne à chaque périphérique la possibilité de définir ses méthodes d'accès à travers une interface standardisée. De cette façon, du point de vue d'un programme utilisateur, il n'y a qu'une sorte d'interface. Une souris, un modem, un générateur de caractères aléatoires ou le texte d'un programme possèdent ainsi que la même représentation, il s'agit de fichier sur lequel on peut lire ou écrire des données. Les détails de gestion sont réglés dans le noyau, par un ensemble de fonctions constituant un module.

À partir d'un *descripteur de fichier*, l'accès se fait via un petit nombre de primitives de haut niveau :

- `open()` et `close()` pour l'initialisation et la libération de la ressource;
- `read()` et `write()` pour les transferts de donnée;
- `select()` pour obtenir des informations sur le statut du dispositif;
- `ioctl()` (input/output control) pour les opérations de configuration.

Les fichiers associés sont en fait des fichiers spéciaux, n'occupant pas de place sur le disque dur[4], mais contenant l'information nécessaire pour que le noyau puisse utiliser

3. http://www.llp.fu-berlin.de/
4. à l'exception d'une structure de description appelée i-noeud

l'interface du module. Dans le cas de notre gestionnaire, ces fichiers se trouvent sous le répertoire `/dev/labpc` :

- `labpc` est l'interface générale, dont l'ouverture provoque l'initialisation de toutes les structures internes du module;
- `adc` est l'interface vers l'ADC, considéré comme un seul dispositif même lorsqu'il travaille en mode multiplexé;
- `dac0` et `dac1` correspondent aux deux DAC;
- `countera0` à `counterb2` sont associés aux 6 compteurs disponibles sur la carte
- `porta`, `portb`, `portc` sont les 3 ports digitaux d'un circuit de type 8255
- `porte` est une extension que nous avons ajoutée, permettant d'accéder à un dispositif complexe via une seule interface. Nous avons utilisé cette possibilité pour masquer complètement toute la gestion du sigma-delta, qui requiert un pilotage des différentes accès des ports ainsi la gestion d'un interface série. En lisant sur cette interface, le programme utilisateur obtient directement les résultats de conversion.

### 4.6.1 Modules

Typiquement, dans les premières versions de Linux, un gestionnaire de dispositif s'organisait comme suit :

- un appel au code d'initialisation devait être ajouté à l'endroit adéquat pour pouvoir être activé au démarrage;
- ce code d'initialisation enregistrait les fonctions nécessaires dans le SVF;
- le reste du programme contenait une implémentation de ces fonctions. Développer un gestionnaire nécessitait donc de recompiler un noyau, de l'installer et de redémarrer la machine. Lors d'une phase de développement, ceci complique singulièrement la mise au point.

Des programmeurs implémentèrent la notion de *module dynamique*. Il s'agit d'un fichier objet contenant le code d'un gestionnaire de périphérique. Ce code peut être inséré et retiré du noyau, parfois de façon automatique grâce à un programme appelé `kerneld`. Les avantages de cette approche sont nombreux :

- lorsque l'on dispose d'un parc de machines avec plusieurs types de périphériques, il n'est plus nécessaire de compiler un noyau avec les options pour supporter chacun des périphériques. Il suffit d'avoir un noyau minimaliste, et d'utiliser au cas par cas les modules requis.
- lorsque certains périphériques ou systèmes de fichier ne sont utilisés que très occasionnellement, le support via modules permet d'économiser de la mémoire. Bien que cet argument puisse sembler moins pertinent aujourd'hui, il reste pleinement d'application dans le cas de micro-centrales de mesures comme nous l'évoquerons dans le paragraphe 4.11.1.

– enfin, lors d'une phase de développement, il est beaucoup plus commode d'enlever et de recharger un module que de devoir redémarrer la machine.

Le support pour la carte Lab-PC a été conçu comme un module dynamique à l'origine, et nous avons poursuivi les développements dans cette voie.

### 4.6.2 Accès mémoire

Les premiers systèmes UNIX ont implémenté la notion de mémoire virtuelle pour les processus, de façon à éviter qu'une application puisse modifier indûment les données d'une autre application. Cela requiert une collaboration du microprocesseur, qui doit posséder un jeu d'instructions spécialisées dans la gestion de la mémoire, qui doivent notamment implémenter la notion de vérification de droit d'accès[5].

Linux ne déroge pas à ce schéma, les caractéristiques de la mémoire accessible en mode noyau et en mode utilisateur différent comme suit:

**mémoire noyau :**
- adresses *réelles* (les pointeurs correspondent à l'adresse physique);
- mémoire toujours présente;
- le code peut accéder à la totalité de la mémoire;
- les pages de mémoire appartiennent au noyau. Si une des routines, après avoir demandé l'allocation d'une zone de mémoire, ne la libère pas lorsqu'elle ne l'utilise plus, cette zone ne pourra plus être réutilisée par le noyau. À la longue, le système peut arriver à court de mémoire.

**mémoire utilisateur :**
- adresses *virtuelles* (mécanisme de traduction par le microprocesseur);
- des pages mémoires peuvent être transférées dans un fichier d'échange;
- un processus ne peut accéder qu'à sa propre zone de mémoire;
- les pages de mémoires appartiennent à un processus. Lorsqu'il se termine, toute la mémoire allouée est automatiquement récupérée par le noyau.

Ces contraintes impliquent que, pour l'écriture d'un module, il convient d'être économe en mémoire, et d'être particulièrement prudent afin de ne pas modifier d'autres variables du noyau. Il faut également éviter de déréférencer un pointeur de valeur nulle, ce qui entraîne un blocage du noyau et nécessite le redémarrage de la machine. Ces contraintes sont de toute façon applicable pour tout développement de code du noyau. Une programmation soignée permet d'éviter ce genre de problème.

5. à l'usage, cela posa d'autres problèmes : comment faire lorsque plusieurs processus doivent *partager* une zone de mémoire? Cela fut résolu dans UNIX System V par des mécanisme appelé "Shared Memory" et "Inter-Process Calls" (IPC)

### 4.6.3 Application temps réel : RT-Linux

Le noyau est chargé de répartir équitablement le temps processeur disponible entre les différents processus actifs. Dans Linux, cela se fait d'une façon fort simple : à chaque processus est attaché un compteur. Un *ordonnanceur de tâches* est activé à intervalles réguliers. Chaque fois qu'un processus a tourné durant une tranche définie de temps, le compteur associé est décrémenté. S'il arrive à zéro, le processus est alors bloqué. Lorsqu'il n'y a plus de processus actif dont le compteur associé est non-nul, l'ordonnanceur les recharge tous, en donnant des valeurs plus élevées aux processus les plus prioritaires.

Lors des premières utilisations, il est apparu que le délai de réponse entre une demande de traitement provenant de la carte Lab-PC et la réponse du gestionnaire pouvait fluctuer largement, en particulier lors des accès disques. En fait, rien ne *garantit* que le gestionnaire pourra disposer « à temps » d'une fenêtre d'exécution pour répondre à la requête. Nous nous sommes alors tourné vers une solution de type temps réel.

On trouve dans la littérature informatique plusieurs notions d'application temps réel :

1. la première idée consiste à exiger que l'application tourne en moyenne à une vitesse suffisante pour ne pas qu'il apparaisse de délai de transmission *perceptible*. Dans cette optique, ne pas satisfaire occasionnellement à cette contrainte est admissible. Cette notion concerne plus particulièrement des logiciels de type génération de musique, de video, ...
2. la définition la plus exigeante d'un système temps-réel est un système qui *garantit* un délai de réponse maximal à une série d'*événements* quelle que soit la situation de la machine. Le non-respect de cette contrainte est considéré comme une défaillance du système. L'exigence provient du fait que la machine doit, alors qu'elle est en train de répondre à un événement, pouvoir répondre à un autre événement.
3. d'autres nuances sont possibles entre ces deux extrêmes, par exemple garantir le délai de réponse à *un* type d'événement, ...

Dans la version standard de Linux, l'ordonnanceur de tâches garantit *en moyenne* une répartition optimale des ressources du processeur. Par ailleurs, il peut arriver que certaines parties du noyau monopolisent le processeur pendant des délais supérieurs à 20 millisecondes, comme par exemple le gestionnaire de disques durs. En effet, pour simplifier la gestion des interruptions, une routine d'interruption en cours d'exécution bloque la prise en compte de nouvelles interruptions.

Supposons en effet qu'une routine d'interruption commence à mettre à jour un ensemble de données, ou initie une communication avec un dispositif. Si une autre interruption survient, et que la routine associée modifie les mêmes données ou envoie d'autres informations au même dispositif, il y aura *incohérence* au niveau des données

manipulées ou transmises. Pour éviter ce problème, il convient d'utiliser des mécanismes garantissant un accès exclusif, et que *toutes* les routines respectent le protocole mis en place. Le mécanisme d'interruption unique décrit au paragraphe précédent permet d'éviter ce genre de complication[6].

Une équipe du New Mexico Institute of Technology a développé une version « temps réel » de Linux[7], dans laquelle il est possible de donner à certaines interruptions le droit d'en interrompre d'autres considérées comme moins prioritaires, si celles-ci sont en cours d'exécution. Par ailleurs, le gestionnaire de temps à l'intérieur du noyau tourne avec une résolution bien supérieure à la résolution standard, qui est de 100 Hz sur une machine à base de processeur i386.

Au niveau programmation, les seules différences concernent :

- la routine d'enregistrement d'une fonction d'interruption;
- la routine et la représentation interne de la base de temps;
- certaines fonctions du noyau, notamment les fonctions d'affichage de messages sur la console, ne peuvent pas être utilisées à l'intérieur d'une routine d'interruption temps réel. En effet, un accès à ces fonctions alors qu'elles sont en cours d'utilisation, provoquerait la corruption de leurs données.

Pour mesurer la latence d'interruption, la routine d'interruption de notre gestionnaire met à "1" un des accès de la carte Lab-PC au début de son exécution, et le remet à "0" à la fin. En visualisant à l'analyseur logique les signaux, nous avons pu établir que le délai entre une demande d'interruption et l'activation de la routine est de 5 à 10 µs, quelle que soit l'activité de la machine. Ce test a été effectué sur un ordinateur dont le processeur est de type i486, à 66 MHz.

### 4.6.4 Organisation générale du gestionnaire

Le fonctionnement du gestionnaire comprend schématiquement trois parties principales :

- une partie d'initialisation et de configuration;
- une partie de communication, à travers le SVF, avec les programmes utilisateurs;
- un gestionnaire d'interruption, chargé de répondre de façon adéquate aux demandes d'interruption en provenance de la carte.

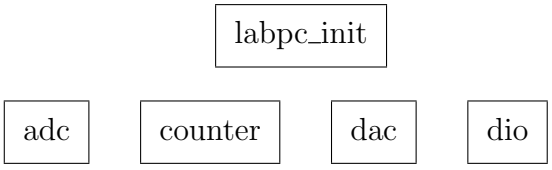
6. avec l'avènement de machines multiprocesseurs, cette approche est devenue très contraignante, car elle implique qu'un seul processeur peut tourner en mode noyau. Depuis la série de noyaux Linux 2.2, cette restriction a été partiellement levée, afin de mieux répartir la charge entre les processeurs. Cela a impliqué l'ajout de primitives pour éviter que plusieurs processeurs ne soient actifs simultanément dans la même routine d'interruption.

7. http://luz.nmt.edu/ rtLinux

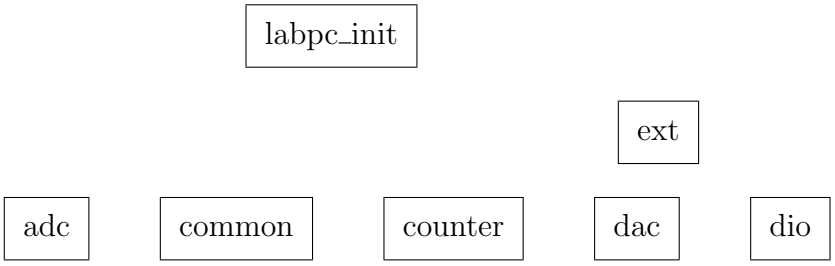
Du point de vue du gestionnaire, la carte Lab-PC consiste en une série de registres, accessibles en lecture ou en écriture. Afin de diminuer la charge du processeur et de réduire le nombre de communications à travers le SVF, le gestionnaire contient un petit nombre de tampons internes, qui servent à stocker provisoirement les données en transit entre la carte et les programmes utilisateurs. Les données ne sont dont pas transférées une à une mais par blocs.

### Organisation

Le gestionnaire est constitué de modules spécialisés. Au départ, la hiérarchie était la suivante :




Nous l'avons modifiée comme suit :




Nous y avons rajouté un module d'extension qui vient « coiffer » le module dio : les 3 ports digitaux sont utilisés pour piloter le sigma-delta.

Le rôle des différents constituants est :

**labpc_init** Initialisation générale, y compris celle des autres modules. Contient le point d'entrée principal de la routine d'interruption, qui appellera tour à tour celle de chacun des autres modules.

**adc** Gestionnaire de l'ADC, interface SVF et routine d'interruption

**common** Une série de routines communes, principalement pour la gestion des tampons : allocation et libération.

**counter** Gestionnaire des compteurs accessibles à l'utilisateur.

**dac** Gestionnaire du DAC.

**dio** Gestionnaire (bidirectionnel) du DIO (entrées/sorties digitales).

**ext** Il permet de faire apparaître une entité complexe (le sigma-delta et sa carte d'interface) comme un seul élément homogène. L'idée est de bâtir une *extension* de la carte, en cachant certains détails internes de fonctionnement. En pratique, cette extension fait elle-même appel aux routines situées en dessous. Elle contient son propre gestionnaire d'interruption, qui peut être substitué à celui du DIO.

### 4.6.5 Philosophie générale

Comme nous l'avons mentionné plus haut, une des motivations qui nous ont poussé à effectuer ces développements était d'obtenir un système aussi robuste que possible. Dans les faits, cela a été implémenté de plusieurs façons :

- en analysant le plus complètement possible l'état des registres afin de détecter toute situation anormale et de pouvoir y donner suite.
- en fournissant des mécanismes de synchronisation, d'une part entre les différentes parties du gestionnaire, pour qu'il n'y ait aucune ambiguïté sur l'état interne de celui-ci; et d'autre part avec les programmes utilisateurs, afin de permettre une collaboration active entre les deux.
- en fournissant des informations supplémentaires, comme par exemple l'instant d'acquisition, afin de permettre aux programmes utilisateurs de vérifier que la cadence d'acquisition est bien respectée.

À force de répéter des acquisitions, nous avons pu progressivement localiser les sources de problèmes, et y apporter une réponse, permettant à l'acquisition de se poursuivre dans les meilleures conditions possibles. Pour alléger le corps du texte, le fonctionnement de ces algorithmes se trouve en annexe. Nous avons essayé d'y décrire de façon simple et de documenter les parties que nous avons modifiées le plus par rapport à l'implémentation de départ.

L'approche générale suivie consiste à détecter un maximum de situations problématiques et à y donner suite de manière adéquate. Lorsque ce n'est pas possible ou que la situation ne s'améliore pas, l'acquisition en cours est stoppée, afin de ne pas bloquer indéfiniment l'ordinateur.

## 4.7 Programme d'acquisition en mode utilisateur

Le gestionnaire de la carte Lab-PC, conçu comme un module du noyau, répond à un certain nombre de contraintes et de restrictions :

- la nécessité d'économie de mémoire implique une taille limitée des tampons destinés à stocker les données intermédiaires;

– par ailleurs, le SVF est prévu comme interface entre le noyau et des programmes utilisateurs, il n'est donc pas possible[8] pour une routine en mode noyau d'effectuer directement des transferts disques, puisqu'il faut passer par un autre SVF, celui associé au fichier dans lequel seront stockés les résultats.

Le programme d'acquisition viendra donc en complément du module de gestion, afin d'implémenter l'algorithme de mesure. Son rôle sera principalement d'organiser les transferts de données entre la carte et d'autres dispositifs : soit des fichiers, pour un stockage local des données, soit un socket, pour une communication à distance.

Traditionnellement, un programme contient un seul chemin d'exécution : au niveau machine, cela se caractérise par un pointeur unique vers l'instruction courante et l'état des registres du processeur, appelé contexte d'exécution. Dans le cadre d'un système d'acquisition multi-canaux, cette approche est mal adaptée. En effet, il convient de surveiller un ensemble de canaux, de lire à temps les données disponibles, de réagir à un problème en relançant le système, ... Dans cette optique, il aurait fallu balayer consécutivement l'ensemble des canaux d'entrée[9], déterminer à chaque instant quelle serait l'étape suivante, ... Cette méthode, dite de scrutation ("polling"), est extrêment lourde et inefficace en ce qui concerne l'utilisation du CPU.

Au niveau Unix, il existe une méthode de duplication de processus via la fonction `fork()`. Cette fonction permet de créer deux processus, appelés le père et le fils, partageant le même code opératoire, et ayant juste après l'appel à cette fonction les mêmes données initiales, à l'exception de l'identificateur de processus et de la valeur de retour de `fork()`. Par la suite, chaque processus travaillera sur ses données propres, c.-à-d. que les modifications effectuées par l'un ne seront pas visibles par l'autre.

Cette fonction est typiquement utilisée dans les programmes de type serveur : après son initialisation, le serveur se met en attente d'une ouverture de connexion. Lorsqu'une telle demande survient, le programme fait un appel à `fork()`. Le fils va traiter cette connexion, tandis que le père va se remettre en attente. De cette façon, on économise le temps d'initialisation du fils.

Dans notre cas, cette solution ne convenait pas pour plusieurs raisons :

– les deux processus créés deviennent indépendants l'un de l'autre. Si le fils modifie une variable globale, cette modification n'est pas reflétée chez le père. La notion de statut global n'est donc pas implémentable.
– la synchronisation se fait par signaux. Ce procédé n'est pas très simple à mettre en œuvre; en particulier, si plusieurs signaux sont délivrés au même processus, l'ordre dans lequel ils seront pris en compte est indéterminé. Par ailleurs, si un processus bloque un signal et que, par la suite, d'autres processus lui envoient

8. pour les noyaux jusqu'à et y compris la série 2.2

9. utilisés en mode non-bloquant, c.-à-d. qu'une lecture sur le descripteur de fichier doit entraîner le transfert demandé ou renvoyer immédiatement un code d'erreur indiquant que cet appel aurait normalement été bloquant.

plusieurs exemplaires de ce signal, un seul exemplaire en sera reçu au moment où ce signal est débloqué. Il n'est donc pas possible d'utiliser de façon fiable ce mécanisme pour savoir par exemple combien d'enfants sont actifs.

– la communication de données peut se faire par un mécanisme portant le nom de tube ("pipe"), qui est en fait un tampon unidirectionnel de capacité finie. Par défaut, les accès à ce tampon sont *bloquants* c.-à-d. que si le tampon est vide (resp. plein), une opération de lecture (resp. d'écriture) sera bloquée jusqu'à ce qu'elle soit à nouveau possible, ce qui enlève la possibilité de réagir « instantanément » à un problème. Pour éviter ce problème, il est de toute façon nécessaire de gérer soi-même un tampon alloué dynamiquement.

Nous avons résolu le problème de la gestion simultanée de plusieurs canaux par la méthode du « multithreading ». Le mot « thread » peut se traduire par fil. Dans le jargon informatique, il est parfois rendu par « enfilade ». Ne connaissant pas de traduction qui nous semble appropriée, nous continuerons à l'utiliser tel quel.

Un thread est une unité d'exécution, comprenant du code, un espace d'adressage, un identificateur de processus unique, ... Le multithreading consiste à avoir plusieurs unités d'exécution, tournant en pseudo-parallélisme, partageant l'espace d'adressage et le code, différenciés par un identificateur et une pile pour les variables locales uniques. Cette technique de programmation, normalisée au sein de la norme POSIX, permet de mieux définir le point de départ des unités concurrentes et de partager de l'information entre elles. Cette norme spécifie également une série de primitives de sérialisation (pour éviter des accès incohérents aux données) et de synchronisation. C'est ainsi que la fonction principale peut attendre explicitement la fin de l'exécution d'un thread, et que des threads peuvent se mettre en attente d'événement.

En pratique, nous avons conçu le programme d'acquisition comme un automate fini à 7 états. La routine principale, qui constitue également un thread, a pour objet de gérer un ensemble de threads spécialisés. Cette approche est nettement plus simple d'un point de vue conceptuel que la solution de la boucle unique dans laquelle il faut déterminer à chaque instant quelle sera la prochaine étape à exécuter. Les 7 états sont :

**NOT_INIT** tant que les données ne sont pas correctement initialisées;

**INIT** lorsque tout est prêt pour lancer une acquisition;

**RUNNING** lorsque tous les threads auxiliaires sont actifs;

**SUSPEND** lorsque un des threads a détecté un problème et que les autres n'ont pas encore été prévenus;

**RESTART** lorsque tous les threads ont été suspendus et sont prêts à redémarrer un acquisition;

**FAILURE** en cas d'erreur impossible à corriger;

**FINISH** une fois que tous les threads ont terminé leur travail.

Voyons à présent les différents threads présents dans le programme.

### 4.7.1 Thread d'acquisition

Son rôle est d'assurer le stockage des données en provenance de la carte. Pour cela, il gère une liste de tampons, liste qui peut accueillir des nouveaux membres, alloués dynamiquement, si nécessaire. De cette façon, la majeure partie du stockage a lieu dans la mémoire utilisateur.

Les étapes sont :

1. si le programme est à l'étape **INIT** ou à l'étape **RESTART**, initialisation du canal d'acquisition (sigma-delta ou convertisseur A/D) et incrémentation de la variable globale `nr_running`. Se mettre en attente d'un changement de l'état global (via la variable de synchronisation *status_cond*).
2. tant qu'il y a des données à lire :
   - dans le cas où l'état est **SUSPEND**, **FAILURE** ou **FINISH**, arrêt du thread;
   - lecture d'un tampon via le descripteur de fichiers associé au convertisseur; en cas de problème, faire passer l'état général à **FAILURE** et sortir du thread;
   - calcul de la différence de temps par rapport au tampon précédent. Si cette différence n'est pas dans une marge bien précise, afficher un message d'erreur et passer à l'étape **SUSPEND**;
   - tester si le tampon suivant est libre, si ce n'est pas le cas, en allouer un nouveau dynamiquement;
   - détermination de l'état du gestionnaire par un appel à la fonction `select()`. En cas d'absence de réponse dans un délai imparti ou de condition d'exception, passer à l'état **FAILURE**.
3. renvoyer une valeur de retour signifiant la fin de l'acquisition.

Cette routine gère donc une série de problèmes, comme par exemple un délai anormal entre deux tampons consécutifs.

La fonction `select()` fait partie du SVF et permet d'interroger le statut d'un gestionnaire de fichier. Elle permet de répondre à une des trois questions :

1. y a-t-il des données prêtes en lecture ?
2. peut-on écrire via le descripteur de fichier ?
3. y a-t-il une situation spéciale ? La signification de « spécial » dépend de l'application et n'implique pas nécessairement un dysfonctionnement.

De plus,cet appel permet également de spécifier un délai d'attente maximal.

L'appel à `select()` est bloquant, c.-à-d. qu'il ne se termine que dans trois cas (le statut « prêt pour écriture » n'ayant ici aucun sens) :

1. lorsque des données deviennent disponibles en lecture : la boucle redémarre et elles sont alors immédiatement lues;

2. lorsqu'une situation exceptionnelle est détectée à l'intérieur du module de gestion, ce qui nécessite de relancer le processus d'acquisition;
3. lorsque, endéans un délai maximum que nous avons fixé au temps nécessaire pour acquérir deux échantillons de plus que le nombre contenu dans un tampon, il n'y a pas de données disponibles.Là encore, il est nécessaire de relancer l'acquisition, parce que cela peut être l'indice que le convertisseur est arrêté.

### 4.7.2 Thread d'émission

Ce thread est très similaire au thread d'acquisition, sauf que bien entendu le sens du transfert va du disque dur vers le convertisseur.

### 4.7.3 Thread de collecte

Celui-ci est chargé de vider les tampons d'acquisition, de convertir les résultats, exprimés dans le code du convertisseur, et de les transférer dans le tampon de sortie.

Les routines d'acquisition peuvent fonctionner à des rythmes différents, par exemple un échantillon sur le sigma-delta pour 3 échantillons sur le convertisseur A/D. Dans cette situation, la routine effectuera une lecture sur les tampons associés au sigma-delta, suivi de 3 lectures sur les tampons A/D. À l'issue de chaque lecture, une mise à jour des tampons sera effectuée si nécessaire, les tampons vidés pouvant être à nouveau utilisés par la routine d'acquisition.

La fonction de conversion est nécessaire pour tenir compte du nombre de bits et de l'encodage des convertisseurs. En effet, les convertisseurs peuvent par exemple exprimer leur résultat dans un encodage de type « complément à deux », alors qu'un résultat variant en continu par décalage du zéro est plus simple à manipuler. Cette fonction assure également la traduction du code binaire en chaînes de caractères. Ce format est en effet plus portable, de nombreux logiciels acceptant en entrée des colonnes de nombres séparés par des espaces.

### 4.7.4 Thread principal, ou de gestion

Son rôle est de procéder à l'initialisation de l'environnement, puis de lancer les différents threads. Ensuite, il veillera à synchroniser l'ensemble des threads, en déterminant le nombre de threads actifs. Lorsque l'acquisition sera terminée, il fermera tous les descripteurs de fichiers ouverts, afin de libérer les ressources utilisées et de les rendre disponibles pour l'utilisation suivante.

Son algorithme peut être schématisé comme suit :

– démarrer avec un statut global **NOT_INIT**;

- interprétation des arguments donnés en ligne de commande;
- lecture d'un fichier de configuration, spécifiant les canaux à utiliser, les fréquences d'échantillonnage, le nombre de tampons, ...;
- appel des fonctions de configuration pour les différents dispositifs. Ces fonctions ouvrent des descripteurs de fichiers, lisent des variables globales contenant les informations de configuration, puis traduisent cette configuration via une série d'appels de type contrôle d'entrée/sortie (`ioctl`);
- ouverture du descripteur de fichier qui recevra les résultats, celui-ci étant un fichier ou un socket;
- ouverture des descripteurs de fichiers pour les données à émettre. Les premières lignes peuvent commencer par le caractère `#` pour signifier qu'il s'agit de lignes de commentaire à ignorer;
- répéter *NumSeq* fois :
  - calibrer le sigma-delta (séquence de calibration interne du zéro et du gain) et le convertisseur D/A;
  - faire passer le statut à **INIT**;
  - répéter indéfiniment :
    - mettre *nr_running* et *nr_runnable* à zéro;
    - si on utilise le sigma-delta, initialiser les tampons associés, configurer et lancer le thread associé, incrémenter *nr_runnable*;
    - répéter cette opération pour le convertisseur A/D;
    - répéter cette opération pour le convertisseur D/A;
    - lancer le thread de collecte des données, incrémenter une fois de plus *nr_runnable*;
    - tant que *nr_running* est inférieur à *nr_runnable* et que le statut est **INIT** ou **RESTART**, attendre via une variable de synchronisation;
    - faire passer le statut général à **RUNNING**, et le signaler via cette même variable de synchronisation;
      - répéter indéfiniment :
      - si le statut est **FINISH**, lire la valeur de retour de chaque thread, et repasser dans la boucle principale;
      - si le statut est **FAILURE**, annuler tous les threads, fermer tous les descripteurs de fichiers et sortir du programme;
      - si le statut est **SUSPEND**, annuler tous les threads, demander au gestionnaire de reprendre l'acquisition, faire ensuite passer le statut à **RESTART**, et revenir dans la boucle précédente;
      - si le statut est **RUNNING**, se mettre en attente de changement via une variable de synchronisation.
- fermeture de tous les descripteurs de fichiers ouverts.

### 4.7.5 Configuration

La configuration générale de ce programme se fait dans deux modules. Le premier, `crys_config.c`, analyse le fichier de configuration, à la recherche de lignes du style :
`nom_de_la_variable = valeur`
Il essaye ensuite de convertir la chaîne de caractères `valeur` dans le format de la variable.

Cette solution est assez souple, puisqu'il suffit d'un éditeur de texte pour changer la configuration globale, et que le format est compréhensible par un programmeur.

Le second, `crys_init`, réalise la configuration des différents canaux d'acquisition en fonction de la valeur données précédemment aux variables globales. En fait, seules les fonctions contenues dans ce module ont besoin d'utiliser un modèle détaillé des dispositifs, via les commandes de type `ioctl()` qui leur sont spécifiques. [10]

En définitive, le programme principal est relativement générique, puisqu'il ne fait d'accès aux données que via des descripteurs de fichier. Cette généricité implique qu'il est assez simple d'adapter ce programme à d'autres situations, ou à d'autres dispositifs d'acquisition.

Cette généricité est en grande partie due à la simplicité de l'interface. Par contraste, les fonctions de l'interface d'acquisition de LabWindows sont bien plus nombreuses : il y a un groupe de fonctions spécifiques pour chaque dispositif. Dans certains cas, il y a même des sous-groupes différents suivant que l'accès se fasse en lecture ou en écriture !

## 4.8 Couplage entre le gestionnaire noyau et les programmes utilisateurs

La séparation des fonctionnalités entre le noyau et le mode utilisateur permet une approche différenciée des événements.

Le gestionnaire en mode noyau doit pouvoir répondre à tout événement généré par la carte Lab-PC sous forme d'une demande d'interruption. Certains sont normaux et attendus, tandis que d'autres indiquent des problèmes. Dans l'optique de rendre le système aussi robuste que possible, nous avons essayé de détecter et de traiter correctement chaque situation susceptible de survenir, même si a priori il n'y a aucune raison qu'elle se produise.

Le programme utilisateur a une vision beaucoup plus générale de l'acquisition : soit elle se déroule normalement, soit elle doit être relancée, soit il y a un problème majeur qui rend la poursuite de l'opération impossible.

10. il s'agit d'une mesure de précaution, pour éviter de donner erronément à un dispositif un code de commande destiné à un autre.

### 4.8.1 Les erreurs en mode noyau

Les appels systèmes contenus dans le gestionnaire suivent la philosophie UNIX en ce qui concerne les erreurs : un code de retour strictement inférieur à zéro indique une condition d'erreur, tandis qu'un code supérieur ou égal à zéro est renvoyé en cas de succès. Les conditions d'erreur sont définies via des constantes symboliques : `EINVAL` indique un paramètre incorrect, `ENOMEM` le manque ou l'impossibilité d'allouer de la mémoire, `EIO` pour une erreur d'entrée/sortie, ...

Au niveau de la routine ext_read, deux cas un peu spéciaux sont prévus lorsque le système se réveille après avoir attendu qu'un tampon soit disponible :

1. lorsqu'un signal (logiciel) a été délivré au processus ayant fait appel à ext_read, la valeur de retour est un code spécial, `ERESTARTNOINTR`, demandant au noyau de reprendre, si c'est possible, l'appel système qui a été interrompu. De cette façon, le processus utilisateur est réactivé, peut prendre en compte le signal, et ensuite se remettre en attente comme si rien ne s'était passé.
2. lorsque le réveil s'effectue par expiration du délai d'attente, ce qui est l'indication d'un problème au niveau du sigma-delta, la valeur de retour est `ESTARTSYS`, indiquant que la mesure doit être relancée. Dans le même temps, il y a réveil du processus utilisateur qui serait en attente d'exception dans l'appel système `select()`.

Dans la routine ext_interrupt, les problèmes pris en considération sont :

- interruption provenant d'une source qui n'a pas été activée : il s'agit d'un problème bénin, mais qui doit néanmoins être traité. Dans ce cas, on reprogramme le DIO pour qu'il ne génère plus cette interruption;
- après avoir rempli un tampon, le suivant n'est pas disponible en écriture : c'est l'indication que le programme utilisateur ne parvient pas à lire suffisamment rapidement les données. Dans ce cas, il y a une tentative de s'en sortir, lorsqu'il n'y a qu'un seul canal d'acquisition, en incrémentant, dans la structure `Guard_t`, le champ relatif au nombre d'échantillons perdus. Sinon, il y a également réveil du processus utilisateur qui serait en attente d'exception.

### 4.8.2 Les erreurs en mode utilisateur

Les erreurs générées en mode noyau sont prises en considération à une série d'endroits dans la routine do_acquire, qui constitue le thread d'acquisition :

- si une mesure ne peut pas démarrer ou redémarrer, suite à une opération de type `IOCTL`, il y a une problème fondamental et le processus général est arrêté;

- si une lecture sur un descripteur de fichiers via la fonction `read()` renvoie un code d'erreur :
  - si le code d'erreur est différent de `ESTARTSYS`, c'est qu'il y a également un problème fondamental, et il faut arrêter l'acquisition. Sinon, la mesure doit être relancée, le mécanisme de synchronisation entre threads est activé dans ce sens.
- après avoir lu un bloc de données, le délai par rapport au bloc précédant est calculé. Si ce délai n'est pas dans l'intervalle imparti, la mesure doit être relancée.
- si le programme doit allouer dynamiquement de l'espace mémoire pour un nouveau tampon et que le système ne peut lui en fournir, il est impossible de continuer : en conséquence, l'acquisition est arrêtée.
- avant de lire le bloc suivant, la routine fait un appel à `select()`. Là encore, certains problèmes peuvent se produire :
  - `select()` renvoie un code d'erreur, indiquant qu'un des paramètres de l'appel n'est pas correct. Cette situation résulte d'une erreur de programmation, et l'acquisition est arrêtée;
  - il y a eu expiration du délai d'attente : la mesure doit être relancée;
  - il y a eu une situation exceptionnelle, comme expliqué au paragraphe 4.8.1 : la mesure doit être relancée;
  - il n'y a pas de données en lecture, ce qui résulte d'un problème fondamental de programmation ou au niveau du gestionnaire : l'acquisition doit être arrêtée.

## 4.9 Résultats

Pour vérifier la robustesse de ce programme, nous l'avons fait tourner durant 48 heures, et nous avons pu en vérifier le bon fonctionnement : l'intervalle entre le démarrage et l'arrêt du programme correspondait bien au nombre d'échantillons multiplié par la fréquence d'échantillonnage.

Voici en chiffre quelques données relatives à un enregistrement sur 13 heures, soit bien plus que nécessaire pour acquérir les données concernant une nuit de sommeil :

- nombre d'échantillons : 1008000
- intervalle entre deux marques temporelles : 12 échantillons, soit $12/20.2 = 594$ ms
- nombre total de marques temporelles : 84000
- nombre de marques différentes de 594 ms : 0

En ce qui concerne les enregistrements aussi bien de nature thermique que respiratoires que nous avons effectués, aucune défaillance ne s'est produite, alors que certaines acquisitions ont impliqué de déplacer le matériel (ordinateur, carte sigma-delta et accessoires) à Woluwé (UcL – Saint-Luc). Par comparaison, l'application tournant sous LabWindows nous a fait plus d'une fois faux bond.

## 4.10 Architecture client-serveur pour le traitement des données

### 4.10.1 Motivations

Lorsque nous avons développé le traitement par ondelettes pour les applications de monitoring cardio-respiratoire, il est apparu qu'il était nécessaire de disposer d'une puissance de calcul importante pour traiter en un temps raisonnable les fichiers de données. Dans le même temps, il fallait pouvoir garantir que le processus d'acquisition se fasse avec une fiabilité élevée. Lors du passage sous Linux, nous avons eu l'idée de scinder clairement ces deux tâches, via une approche de type client-serveur. Cette scission a été opérée en faisant tourner deux logiciels sur des machines distinctes, les données étant transférée de l'une à l'autre par liaison réseau.

### 4.10.2 Systèmes clients-serveurs

Dans l'environnement Unix, il est classique de séparer des fonctionnalités ayant un thème en commun. Dans le cas du courrier électronique par exemple, l'application permettant de consulter le contenu de sa boîte au lettre est le "Mail User Agent", tandis que la transmission de courrier s'effectue via un "Mail Transfer Agent". Il s'agit en effet de fonctionnalités très différentes : le MUA est essentiellement un programme interactif, permettant de consulter un fichier dont l'emplacement est fixe. Quant au MTA, il tourne en permanence, donc sans intervention d'opérateur, et communique avec d'autres MTA en fonction des courriers à émettre et à recevoir. Son fonctionnement implique donc de pouvoir localiser le serveur adéquat, d'essayer d'établir une connexion, de recommencer si nécessaire sans pour cela saturer ni la machine sur laquelle il tourne ni le réseau, ...

Un système client-serveur consiste à avoir deux agents communiquant entre eux via un protocole fixé à l'avance, articulés autour d'un thème commun, et gérant des aspects relativement incompatibles de ce thème.

### 4.10.3 Opérations à implémenter

Dans notre cas, nous devions assurer deux types d'opération, chacune soumise à des contraintes propres difficilement conciliables :

1. acquisition et sauvegarde locale des résultats, ce qui nécessite peu de puissance de la part du processeur, mais requiert un temps de réponse garanti afin de ne perdre aucune donnée;

2. filtrage et traitement par ondelettes, ainsi que visualisation en temps réel des différentes traces : ici, la puissance de calcul requise pour travailler en pseudo-temps réel est très importante, tandis qu'un délai aléatoire a peu de conséquences sur la bonne marche du système.

La solution retenue a été d'affecter deux machines à cette tâche. La première contient la carte d'acquisition, tourne sous un noyau temps réel, avec une puissance de calcul modeste (processeur i486 à 66 Mhz) mais néanmoins amplement suffisante pour assurer le bon fonctionnement du programme d'acquisition. La seconde est beaucoup mieux équipée en puissance de calcul (2 processeurs PII à 333 MHz), ainsi qu'en possibilités graphiques (serveur X, applications graphiques en 800x600 pixels).

De la sorte, il a été possible d'obtenir d'une part une excellente fiabilité au niveau de l'acquisition, et d'autre part un traitement extrêmement rapide de nos données : à titre d'exemple, la partie filtrage d'une séquence d'une durée de 15 minutes, à une fréquence d'échantillonnage de 20 Hz, soit 18 000 points, ne demande que quatre secondes.

Le transfert s'effectue via une interface de programmation portant le nom de socket, permettant d'établir une communication fiable entre deux applications. Les données passent du noyau au programme d'acquisition, qui les stocke temporairement en mémoire et les transmet à une autre application qui effectue d'une part la sauvegarde dans un fichier sur le disque dur, et d'autre part gère une extrémité du socket, servant à écrire les données. Des clients peuvent venir s'y connecter pour lire le résultat des acquisitions. Afin de limiter la charge de la machine, leur nombre a été limité à 5.

En ce qui concerne la visualisation proprement dite, la solution implémentée a été de réaliser un programme dans le langage Java et utilisé dans un navigateur sous forme d'application embarquée ("applet"). Dans cette approche, l'environnement logiciel contient toute une série de fonctionnalités, en particulier au niveau graphique, ce qui a facilité le travail de programmation.

Les avantages de cette approche client-serveur sont :

1. découplage des aspects acquisition et traitement : ils peuvent même être effectués sur des sites différents. De plus, un serveur de traitement peut être utilisé pour plusieurs acquisitions effectuées simultanément.
2. gain de fiabililité, puisque la machine d'acquisition est totalement dévolue à cette tâche. En cas de perte de connexion, les données sont stockées en local, et le traitement peut être différé.
3. en termes de coût, l'ordinateur effectuant l'acquisition n'ayant besoin que d'un niveau modeste de performances, sera d'un coût moindre. Éventuellement, on pourra récupérer une machine considérée comme « dépassée » pour d'autres applications.

## 4.11 Retombées possibles

### 4.11.1 Micro-centrales de mesure

#### Systèmes à micro-contrôleurs

Typiquement, le développement d'un système d'acquisition autonome se fait autour d'un circuit de type micro-contrôleur. Ce circuit est accompagné de mémoire, d'interfaces d'entrée/sortie, d'interfaces de communication, ...

Leur programmation s'effectue en général en langage d'assemblage, ou parfois via un compilateur, en langage de plus haut niveau, comme le C. Dans les deux cas, il convient en général de programmer une série de fonctionnalités, comme une routine d'acquisition suffisamment paramétrable, un protocole de transfert fiable via l'interface de communication, une routine de gestion générale, ...

Comme il est souvent difficile ou onéreux de trouver le code adéquat, une partie non négligeable du travail de développement consiste à programmer et à tester ce dispositif. À ce stade, l'utilisation d'un simulateur ou d'un émulateur (ICE) s'avère indispensable. Cette partie du travail est plutôt fastidieuse, en particulier lorsque le micro-contrôleur travaille de pair avec un ordinateur. Dans ce cas, il faut en effet effectuer les développements sur les deux plate-formes, vérifier que les formats binaires soient compris de la même façon sur chacune d'entre elles, ...

Par ailleurs, le code est en général extrêmement spécifique, suivant par exemple qu'il s'agit d'un système d'acquisition générique, d'un système d'acquisition impliquant le pilotage de dispositifs externes, d'un système de contrôle et de commande, ... Il n'est donc pas évident de pouvoir récupérer du code provenant d'un autre développement car cela demande un travail d'adaptation non négligeable. De plus, tout changement de micro-contrôleur implique au minimum de s'assurer de la compatibilité du code : il peut y avoir de nouvelles instructions, d'autres registres à programmer, ...

Lors d'un travail de fin d'études que nous avons supervisé en 1991 concernant un tensiomètre artériel ambulatoire, les étudiants avaient choisi de réaliser une micro-centrale d'acquisition, programmée en C. Ils ont du développer eux-mêmes la totalité du code sur deux plate-formes. En particulier, pour la transmission des résultats de mesure, il s'est avéré nécessaire d'implémenter un protocole travaillant par paquets, avec vérification individuelle de chaque bloc et retransmission en cas d'erreur.

#### Système à microPC

À l'heure actuelle, les progrès techniques, notamment au niveau de la miniaturisation des circuits intégrés, ont rendu possible la réalisation de « microPC » (“matchbox

computer"), qui ont notamment été utilisés dans des réalisations comme « le plus petit serveur WEB au monde »[11]. Ces solutions s'articulent autour d'un processeur de type i486 fabriqué par AMD, avec 4 mega de RAM, une unité de stockage de masse équivalente à un disque dur de 200 mega, un port série, un port parallèle, ...

Une version de Linux correctement configurée et allégée au maximum peut tourner de façon tout à fait confortable sur cette plate-forme. Ceci nous ouvre donc la possibilité de disposer de micro-centrales équipées d'un système d'exploitation multi-tâches possédant un support réseau. De la sorte, il n'est plus nécessaire de reprogrammer de façon spécifique les logiciels développés précédemment ou disponibles.

Dans cette configuration, un certain nombre de primitives intéressantes pour réaliser des acquisitions sont disponibles :

- Gestion d'une horloge interne. Un programme peut donc savoir quand a démarré ou a eu lieu une acquisition. De plus, il existe des programmes permettant de se synchroniser sur un serveur externe, ce qui évite les dérives d'horloge sur le long terme.
- Exécution périodique d'un programme via "cron"[12]. Cette application permet de spécifier de façon très complète quand une tâche doit être exécutée, par exemple le premier lundi du mois ou encore tous les quart d'heure de 8 heures à 18 heures du lundi au vendredi, ...
- Attente de données, monitoring d'événements : cela se fait via l'appel système `select()` déjà évoqué. Dans ce cas, il n'y a pas de ressources processeur consommées durant l'attente.
- Système multi-tâches et multithreading, ce qui simplifie la gestion simultanée de plusieurs voies d'acquisition;
- Gestion et protection de la mémoire; un processus ne peut ni modifier la mémoire d'un autre processus ni utiliser comme référence un pointeur invalide, ce qui sont des erreurs de programmation communes. Sans protection de la mémoire, ce genre d'erreur est difficile à mettre en évidence autrement que par ses effets; lorsqu'il y a protection de la mémoire, l'application fautive est arrêtée immédiatement, avec génération d'un fichier spécial permettant de retrouver l'instruction problématique.
- Au niveau de la consommation électrique, le noyau fonctionne totalement par interruptions, et arrête le processeur s'il n'y a pas de tâches qui peuvent tourner, par exemple si elles sont toutes en attente d'événement. De ce fait, la consommation est minimisée, ce qui maximise la durée de vie des batteries. Par contre, d'autres systèmes d'exploitation comme DOS entrent dans une boucle de scrutation, avec maintien de la pleine consommation.
- Communication via modem : `PPP`, le Point-to-Point Protocol. Le client permet d'une part de s'interfacer simplement avec un modem en utilisant le protocole

11. World's Smallest Web Server, http://wearables.stanford.edu
12. du grec « chronos » : il s'agit d'un programme de gestion temporelle de tâches.

« HAYES », plus connu sous le nom de « AT » parce que toutes les commandes commencent par cette chaîne de caractères; d'autre part de mettre à jour la table de routage pour rendre accessible de façon transparente la machine avec laquelle la connexion a été établie. Derrière le modem peut se trouver une ligne téléphonique, ou encore une liaison radio, ...

– Communication directe vers une autre machine : outre le protocole PPP, il existe un autre protocole appelé SLIP (Serial Line Interface Protocol), dont une version spécifique (PLIP) fonctionne sur le port parallèle (le port d'imprimante standard), donnant ainsi des taux de transmission plus élevés.
– Protocole TCP/IP : ce protocole permet à plusieurs applications et à plusieurs machines de partager un même support de transmission; de plus, la transmission se fait par paquet avec détection des erreurs de transmission.
– ...

Dans ce contexte, le développement portera essentiellement sur la *configuration* des différents utilitaires et non plus sur la *réécriture* (et le test) de code. De ce fait, il sera possible de profiter d'une large gamme de programmes (sous forme de code source) et d'applications existantes.

Dans les programmes que nous avons développés, nous avons essayé au maximum de séparer la partie générique du code de ce qui était spécifique à la carte Lab-PC : par exemple, dans le module `dio.c`, tous les accès spécifiques se sont encapsulés dans des fonctions ou des macro. Pour adapter ce code à un autre type de carte d'acquisition, il n'y aura que le contenu de ces fonctions spécifiques à changer, tout le reste pouvant être réutilisé. Il y a donc un gain de temps et une économie de moyens à mettre en œuvre.

### 4.11.2 Logiciels d'acquisition

Le programme utilisateur est relativement générique, en ce sens que nous avons également séparé la gestion globale de l'acquisition du code spécifique à la carte Lab-PC. Par exemple, toutes les opérations de configuration de cette carte sont effectuées dans des routines appelées au cours de l'initialisation ou à la fin du code.

De plus, étant donné que le code accède aux données à travers des descripteurs de fichier, travailler avec d'autres cartes ou d'autres dispositifs d'acquisition se fera sans difficulté : il suffit simplement d'ouvrir les descripteurs de fichiers adéquats.

Travailler en ayant à sa disposition les sources du système d'exploitation et des librairies de haut niveau offre un certain nombre de facilités :

– il est possible de comprendre exactement comment sont implémentées et fonctionnent les primitives;

- en cas de doute sur la façon de programmer une application, il est possible de regarder dans d'autres modules pour comprendre comment d'autres programmeurs ont réalisé cette programmation;
- au cas où il apparaît un problème, dû à des conditions d'utilisation tout à fait particulières, il est possible d'effectuer des investigations au niveau du code et de comprendre où se situe la source.

Pour illustrer ces propos, voici une anecdote survenue au cours de l'encadrement d'un travail de fin d'études consacré à une évaluation de méthodes de mesure de distance par ultrasons. Pour pouvoir travailler avec la carte Lab-PC, nous avions développé un modulateur/démodulateur, pour que l'émission et l'acquisition aient lieu en bande de base. À partir d'une horloge à 39 kHz et d'un compteur, nous générions un signal d'horloge à 13 kHz pour l'ADC et à 6,5 kHz pour le DAC. En effet, le principe était de déterminer l'enveloppe du signal, requérant une acquisition en phase et en quadrature. Au niveau des spécifications, l'ADC peut travailler jusqu'à une fréquence de 100 kHz, tandis que pour le DAC, seul un délai maximal entre l'ordre d'écriture et la stabilisation de la sortie analogique était mentionné.

Comme le traitement numérique nécessitait plusieurs calculs de corrélation, la durée des séquences était de 256 échantillons. De ce fait, un train d'impulsions durait un peu moins de 40 millisecondes. En analysant la variance de plusieurs résultats consécutifs, nous nous sommes aperçu que nous n'avions pas une distribution gaussienne, mais que environ un résultat sur 10 était plus grand d'une unité que la moyenne. En comparant une séquence normale à une séquence incorrecte, nous avons fini par trouver que, effectivement, en réception, à partir d'un instant quelconque, tous les échantillons présentaient un décalage unitaire par rapport à leur homologue appartenant à une séquence correcte.

Nous avons fait l'hypothèse que, au cours de l'acquisition, une autre routine d'interruption pouvait être activée, et de ce fait retardait à ce point la prise en compte de la demande d'interruption du DAC, que celui-ci « perdait » une impulsion d'horloge. L'ADC quant à lui n'était pas affecté par ce problème puisque disposant d'une mémoire tampon (FIFO).

Quelles étaient les sources d'interruption susceptibles d'être activées? Il n'y avait pas d'accès disques, nous avions pris soin de ne rien modifier à l'affichage, de ne pas toucher au clavier ni à la souris, ... Il restait l'horloge interne du DOS, activée 18 fois par secondes, c.-à-d. toutes les 55 millisecondes. Nous avons donc adjoint à notre programme une petite routine, appelée juste avant le démarrage d'une mesure, et surveillant en continu la valeur de l'horloge interne. Lorsqu'elle en détectait une modification, cette routine rendait le contrôle à notre programme, qui démarrait donc systématiquement juste après la fin de cette routine d'interruption d'horloge. À partir de ce moment, nous n'avons plus obtenu de séquence incorrecte !

Ceci illustre la difficulté de trouver les causes de problèmes dans un code dont on ne dispose pas des sources. Par ailleurs, la solution est purement symptomatique et n'est

pas applicable en général, puisqu'il reste la contrainte de la fenêtre de travail inférieure à 55 millisecondes. Dans l'environnement LabWindows, la seule solution possible aurait été d'acquérir une autre carte analogique, spécialisée dans la génération de signaux, et disposant de ses propres tampons ou à tout le moins de FIFO pour permettre au système de continuer à fonctionner correctement dans l'hypothèse qu'une interruption n'a pas été traitée à temps.

La conclusion de ceci est qu'il est nécessaire, pour réaliser des acquisitions en général, de disposer d'un système qui permette de garantir strictement un délai de réponse maximal à un événement. La solution que nous avons implémentée permettait de contourner le problème, en travaillant exclusivement durant des intervalles de temps où d'autres interruptions ne pouvaient pas survenir, mais cette solution reste limitée à des mesures durant moins de 55 millisecondes.

Par comparaison, nous avons effectué des « stress test » sur notre programme d'acquisition, en faisant tourner simultanément sur la même machine le programme de traitement (filtrage par ondelette) ainsi que le programme de visualisation (applet Java tournant dans Netscape, lui-même tournant sous X Window). Dans la version tournant sous Linux standard, nous avons eu des échantillons perdus, ceci étant dû au fait que le gestionnaire ne pouvait pas arriver à tourner endéans les 50 millisecondes que dure une acquisition sur le sigma-delta. Par contre, avec la version de Linux temps réel, nous avons pu vérifier, via un analyseur logique, que la latence d'interruption était bornée, et que notre programme continuait de fonctionner correctement même en conditions de charge du processeur très élevée.

## 4.12 Conclusion

De tout ceci nous pourrons retenir :

- une tâche d'acquisition sur ordinateur est par essence de type temps réel et nécessite un support adéquat de la part du système d'exploitation;
- nous avons développé un programme d'acquisition multi-canaux extrêmement robuste, dont la fiabilité est égale à celle de la machine sur laquelle il tourne;
- ce programme peut être facilement réutilisé avec d'autres dispositifs d'acquisition, ou sur une plate-forme de type microPC pour réaliser des systèmes embarqués autonomes.

*Troisième partie*

---

# ***Applications de type monitoring cardiorespiratoire***

La recherche dans le cadre de l'évaluation du rythme respiratoire a comporté une séries d'étapes dont nous allons simplement esquisser ici le fil conducteur.

Nous avons d'abord voulu savoir si le rythme respiratoire pouvait intervenir comme signal parasite au niveau thermique. La réponse s'étant avérée positive, nous nous sommes ensuite attelés au développement d'une méthodologie où ce paramètre serait l'objet de la mesure.

Ce signal était particulièrement difficile à mettre en évidence; par la suite nous avons pu établir qu'il ne résultait pas d'un effet thermique mais bien d'un effet mécanique, par modulation de la pression exercée sur le capteur thermique. Nous avons alors repris la recherche en nous basant directement sur un signal de pression. Cette voie s'est révélée plus prometteuse, car nous avons pu mettre en évidence dans ce signal non seulement la présence du rythme respiratoire, mais également du rythme cardiaque ainsi que d'apnées.

La prise de mesure sur le patient étant très peu contraignante, cette méthode d'évaluation nous est apparue intéressante dans le cadre de l'étude du sommeil (polysomnographie), qui requiert l'enregistrement de nombreux paramètres physiologiques, mais où les paramètres les plus importants sont liées à la respiration et au rythme cardiaque.

L'extraction du signal s'est faite par une méthode de décomposition faisant appel aux ondelettes, une notion dont nous expliquerons les principes. Cette décomposition se fait en multirésolution, à savoir que les différentes composantes sont à des résolutions différentes. Pour disposer de suffisamment de points pour effectuer des analyses en fréquence, nous avons considéré deux approches, l'une basée sur une approximation de la reconstruction, l'autre sur une reconstruction correcte. Nous présenterons les résultats obtenus en suivant chacune de ces voies, ainsi que les diverses réalisations mises au point dans le cadre de cette filière.

Chapitre 1

# Voie thermique

## 1.1 Premiers essais

Au cours d'une des phases de recherche, nous avons cherché à savoir si le rythme respiratoire pouvait donner lieu à une signature spécifique dans le signal d'origine thermique, l'idée étant de considérer un tel signal comme un *artefact*.

Dans un premier temps, nous avons procédé à des enregistrements thermiques à haute résolution (20 µK) au niveau de l'avant-bras gauche, en demandant au volontaire de respirer à un rythme imposé : le spectre de la température contenait un pic à la même fréquence. Nous avons ensuite évalué le rythme respiratoire de manière simple : le volontaire tenait dans la main droite un bouton-poussoir, et appuyait au cours des phases d'inspirations. Ce signal binaire contenait la fréquence et la phase du phénomène respiratoire.

À ce niveau de résolution, le signal thermique présentait assez bien de dérive, mais en analysant le spectre en fréquence, nous avons pu mettre en évidence une zone dans laquelle les composantes principales du rythme respiratoire se retrouvaient.

Il convenait donc d'appliquer un filtrage passe-bande au signal thermique pour en extraire la composante générée par le rythme respiratoire. Nous avons appliqué des filtres de Butterworth, et pu mettre en évidence une oscillation dont l'amplitude était de l'ordre de 1 mK.

Pour nous assurer qu'il ne s'agissait pas d'un simple artefact, c.-à-d. que le signal ne pouvait résulter d'un filtrage passe-bande appliqué à un bruit blanc, nous avons étudié ce qui se passait en cas d'apnées centrales résultant d'un arrêt volontaire de la respiration. Dans ce cas, le signal filtré était corrélé avec le rythme respiratoire, disparaissant pendant l'apnée et réapparaissant en même temps que la respiration.

### 1.1.1 Traitement par ondelettes

Afin d'obtenir un traitement plus standardisé, nous avons utilisé une première méthode de filtrage par ondelette. Nous y reviendrons plus longuement dans le chapitre 3. Nos motivations étaient d'obtenir un traitement systématique, c.-à-d. susceptible de prendre en compte la totalité des composantes du signal, et d'autre part de ne pas effectuer une périodisation du signal, ce qui aurait été le cas si nous avions appliqué l'analyse de Fourier.

### 1.1.2 Exemple de résultat

La figure 1.1 donne la température différentielle par rapport à un point fixe au niveau de l'avant-bras gauche, avec un gain de 100 pour l'amplificateur d'instrumentation soit une résolution de 20 µK.

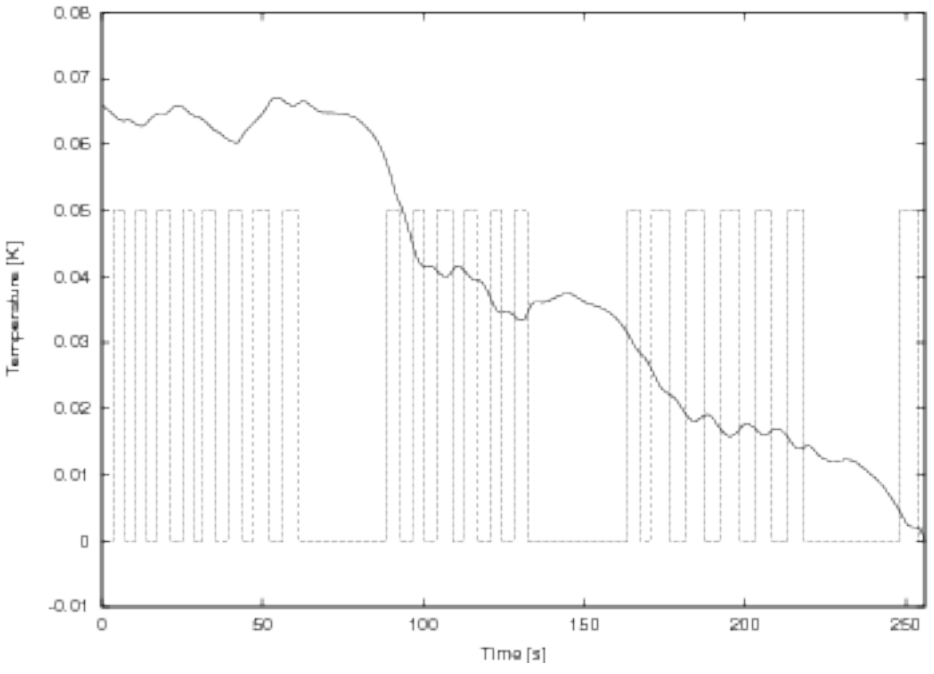



FIG. 1.1: Comparaison température différentielle – phase respiratoire

Le signal en créneau indique la phase respiratoire, l'autre signal est la température différentielle. Par commodité, ce signal a été translaté pour se terminer en zéro. La gamme de variation est de 70 mK.

Comme il s'agit d'un signal à variation lente, un échantillon en sortie correspond à une moyenne sur 10 points consécutifs, ce qui correspond à une fréquence d'acquisition effective de 2 Hz.

Une apnée centrale volontaire survient après 8 cycles de respiration normale. Ensuite, nous avons un seconde puis une troisième apnée, chacune précédée de 6 cycles normaux.

Globalement, le signal de température présente une tendance générale à la dérive. Il ne semble pas y avoir de connexion directe entre les deux phénomènes, à savoir température différentielle et phase respiratoire.

Voyons à présent ce signal, après filtrage par ondelettes (figure 1.2). Nous illustrons un niveau de détail, ce qui correspond à un filtrage passe-bande centré sur la fréquence du rythme respiratoire. Nous avons utilisé dans ce cas des ondelettes de Daubechies d'ordre 8. Ces résultats ont été obtenus par le logiciel Octave, en utilisant une série de routines écrites par P. Carrette, de l'unité Auto, permettant de générer les coefficients et de réaliser les opérations de filtrage.

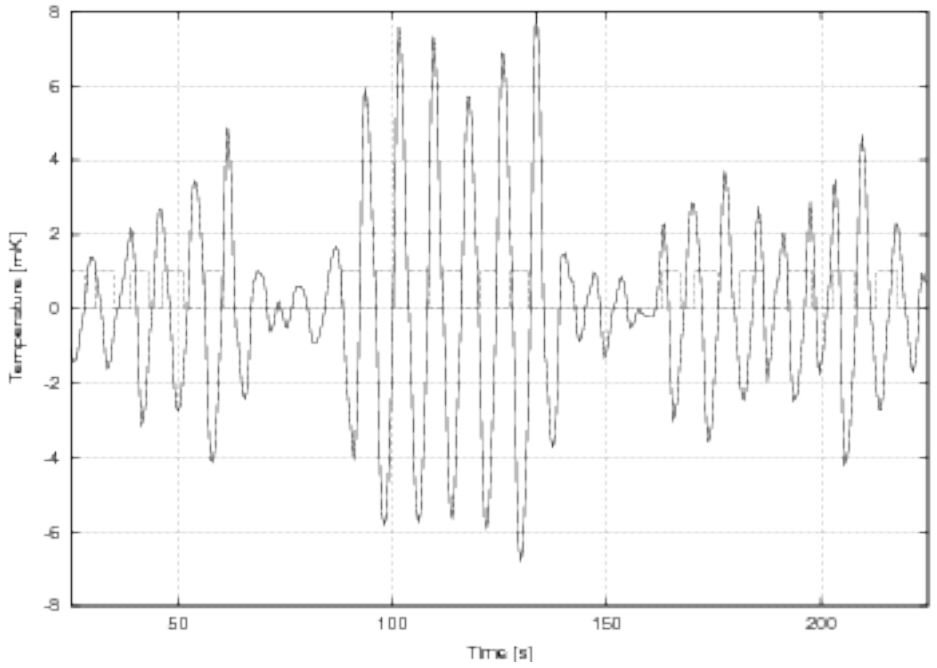



Fig. 1.2: Mise en évidence du rythme respiratoire dans le signal thermique.

Nous constatons qu'il apparaît un signal après filtrage, qui est en opposition avec le signal respiratoire lorsqu'il est présent, et dont l'amplitude décroît fortement durant les apnées : l'amplitude crête-à-crête ne représente plus que 25 % de l'amplitude durant les cycles normaux. Les oscillations résiduelles sont liées à la réponse impulsionnelle du filtre et correspondent à l'extinction des oscillations présentes avant le début de l'épisode d'apnée.

### 1.1.3 Filtrage par ondelettes

L'analyse par ondelette est une méthode de filtrage systématique fort intéressante. Une présentation plus complète en est donnée au chapitre 3, nous allons simplement esquisser ici les propriétés majeures de cette technique. La différence la plus significative par rapport à l'analyse de Fourier est que les signaux analysants sont limités dans l'espace des fréquences (comme le sont les sinusoïdes de l'analyse de Fourier) mais également dans l'espace temporel, alors que les sinusoïdes conviennent mal pour détecter un changement dans un signal, à cause de leur durée semi-infinie ou infinie.

Les coefficients des ondelettes permettent d'analyser le contenu fréquentiel, mais également de détecter et de traiter des changements abrupts dans les caractéristiques temporelles.

Les ondelettes sont choisies pour former une base orthogonale dans un espace muni d'un produit scalaire défini de manière adéquate. De cette façon, des signaux complexes peuvent être décomposés en signaux plus simples et indépendants. Les propriétés de linéarité et l'énergie totale sont également préservées. Un certain nombre de familles sont disponibles pour des applications numériques, elles se différencient au niveau du compromis entre leur domaine d'existence et leur régularité.

Le point de départ est une ondelette de base, appelée ondelette « mère » ou génératrice qui est appliquée (convoluée) au signal à analyser. Cette ondelette mère agit comme un filtre passe-haut et donne les « détails », c.-à-d. la partie du signal variant rapidement. Un filtre complémentaire associé à une fonction dite « d'échelle » fournit une approximation ou un lissage du signal. Le signal initial est la somme de l'approximation et du détail.

Les deux filtres ont une fréquence de coupure valant la moitié de la fréquence d'échantillonnage et leur réponse en fréquence est symétrique par rapport à cette valeur. Comme l'énergie du détail est concentrée dans la partie supérieure des fréquences tandis que celle de l'approximation l'est dans la partie inférieure, la représentation est redondante et ils peuvent être décimés par un facteur deux sans perte d'information, de par le théorème d'échantillonnage de Nyquist.

Pour poursuivre l'analyse, l'ondelette de départ est dilatée d'un facteur 2 en temporel et réappliquée au détail obtenu précédemment. Cela donne un nouveau niveau de détail qui est cette fois de type passe-bande, ainsi qu'une nouvelle approximation. La dilatation temporelle de l'ondelette ne nécessite pas de recalculer de nouveaux coefficients : en effet, la décimation est équivalente à un ré-échantillonnage, donnant la relation correcte entre la fréquence d'échantillonnage et la fréquence de coupure des filtres.

Cette construction récursive assure un filtrage systématique de la totalité du spectre, avec une résolution adaptée à la bande de fréquence des filtres consécutifs : le rapport entre le facteur de qualité et la bande passante reste constant. Cette méthode porte

le nom d'algorithme de transformation pyramidale; appliquée à un signal comportant $2^J$ points, le résultat contient le même nombre d'échantillons à cause des étapes de décimation intermédiaires. La complexité numérique est de $O(J\,2^J)$.

Pour retrouver le nombre de point initial, il convient de resynthétiser par filtrage inverse le signal correspondant à la bande passante recherchée. Cette approche nécessite (du moins le pensions-nous lors de cette phase de la recherche) d'avoir complété l'acquisition.

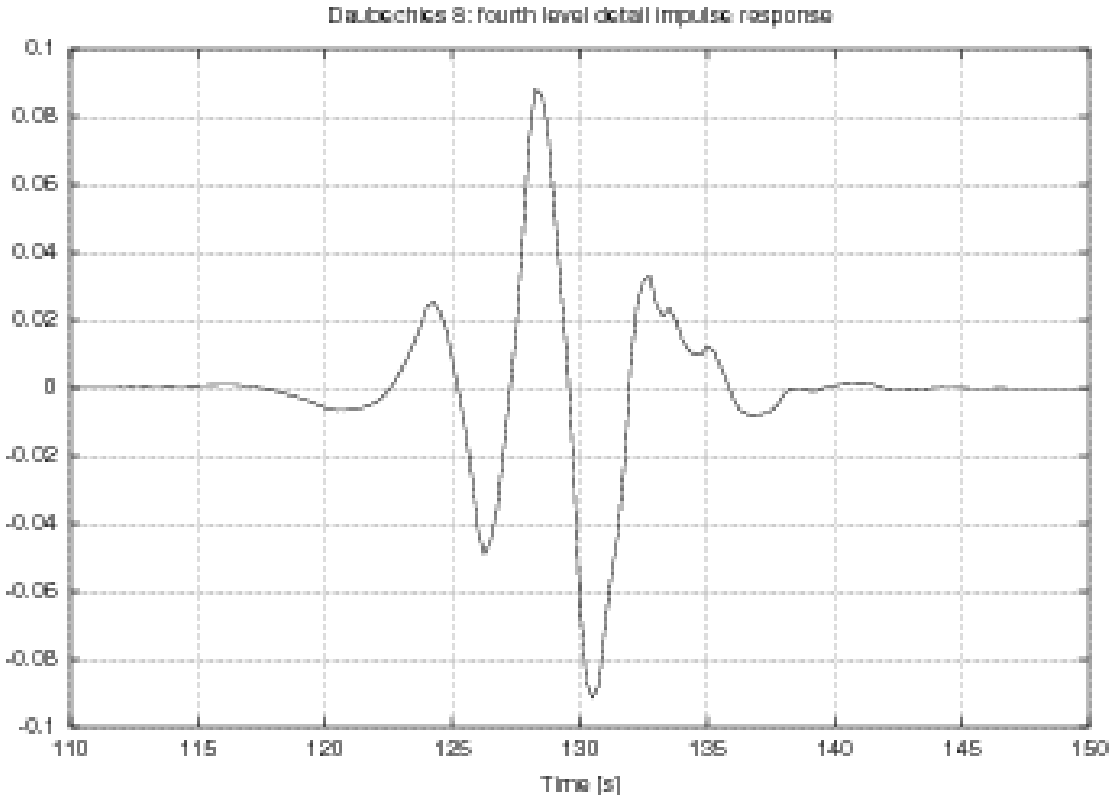



FIG. 1.3: Réponse impulsionnelle du quatrième niveau de détail

Afin de mieux comprendre le principe de ce filtrage, la figure 1.3 contient la réponse impulsionnelle du filtre appliqué pour obtenir le signal illustré dans la figure 1.2. Cette figure a été obtenue en considérant un fenêtre temporelle de 256 secondes, comportant uniquement une impulsion en son milieu. Le signal a été filtré, puis une composante a été sélectionnée et reconstruite à la résolution initiale.

Nous avons utilisé WaveLab pour obtenir ces figures : il s'agit d'un ensemble de routines permettant d'effectuer des analyses par ondelettes. L'implémentation est légèrement différente de celle de P. Carrette citée précédemment, il y a dans ce cas une compensation du délai introduit par les filtres. De ce fait, la réponse présente un pic principal en $t = 128$ secondes. Cette réponse est de moyenne nulle et illustre une caractéristique importante des ondelettes : leur durée est limitée.

La réponse fréquentielle des quatre premiers niveaux de détail est donné dans la figure 1.4. Le premier niveau est un passe-haut, tandis que les suivants sont de type passe-bande et comportent un lobe principal ainsi que des lobes secondaires d'amplitude plus faible.

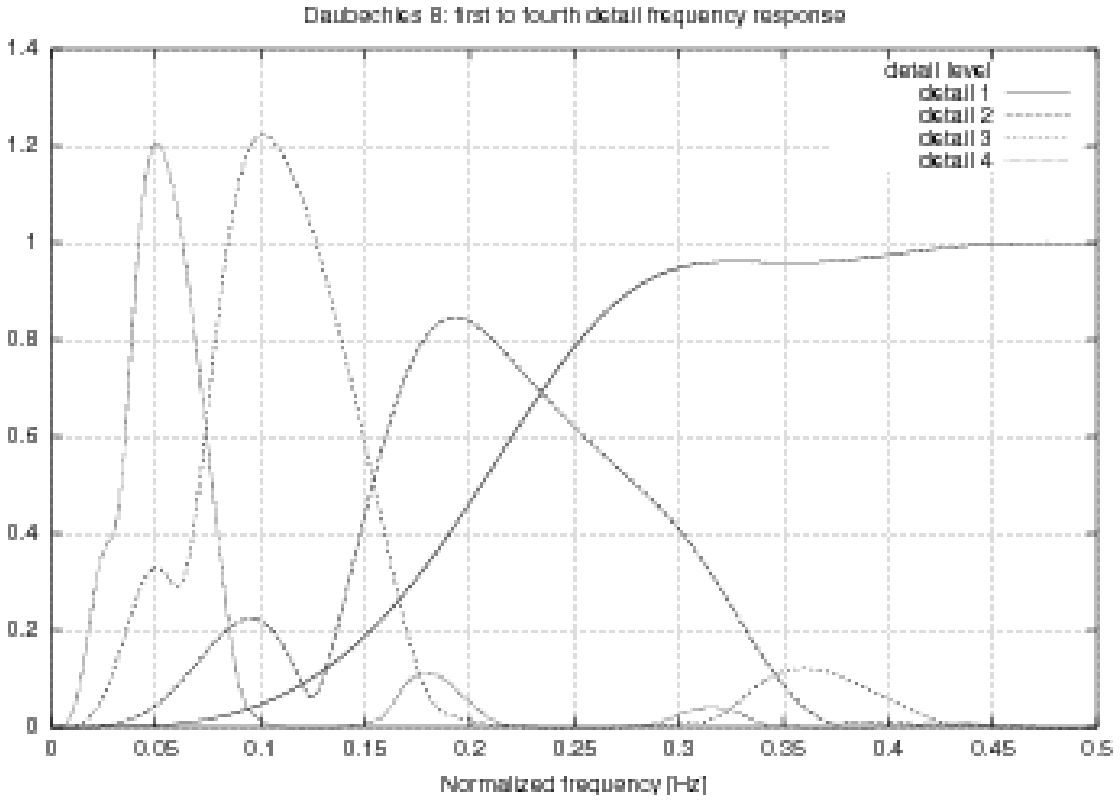



FIG. 1.4: Réponses en fréquence pour Daubechies (8)

### 1.1.4 Hypothèses formulées

Pour expliquer ce signal qui nous apparaissait a priori comme un effet thermique, nous avions envisagé, après contact avec des médecins, quelques hypothèses permettant de lier le rythme respiratoire à une modulation de la température cutanée :

- le rapport entre la surface et le volume du corps se modifie au cours des phases respiratoires, modulant ainsi la densité de pertes de chaleur;
- les alvéoles pulmonaires sont le lieu d'échanges thermiques entre le sang et l'air inspiré, entraînant une oscillation concomitante de la température du sang artériel;
- le taux d'oxygène disponible dans le sang est modulé par l'activité respiratoire, ce qui entraînerait une variation de l'activité métabolique.

## 1.2 Seconde étape

Au cours d'un autre phase de la recherche, nous avons repris ces travaux pour essayer de mieux affiner la connaissance de ce phénomène.

Après quelques essais infructueux, nous nous sommes aperçu que le signal ne pouvait être mis en évidence que lorsque l'adhésif qui fixait le capteur thermique à l'avant-bras encerclait complètement celui-ci. En cas de fixation partielle, sur la moitié de l'avant-bras par exemple, il n'a pas été possible de mettre en évidence la moindre trace de rythme respiratoire dans le signal de température.

Nous en avons conclu qu'il s'agissait peut-être d'un phénomène de pression. Pour nous en assurer, nous avons effectué une mesure de pression, dans un brassard servant à évaluer la tension artérielle, maintenu faiblement gonflé. Après analyse, un signal corrélé au rythme respiratoire est clairement apparu. Nous avons dès lors poursuivi la recherche dans cette voie, ce que nous allons développer dans les chapitres suivants.

Chapitre 2

# Monitoring des rythmes cardiaques et respiratoires

## 2.1 Introduction

Ce chapitre présente une approche nouvelle de l'évaluation des rythmes cardiaques et respiratoires. Ces deux paramètres physiologiques sont de première importance, il existe de nombreuses méthodes et dispositifs pour les évaluer.

Nous avons développé une approche où les variations de pression artérielle liées à ces deux phénomènes se répercutent sur un capteur de pression. À partir de ce signal, nous extrayons des composantes qui contiennent l'un ou l'autre de ces rythmes.

Cette approche est susceptible d'offrir des retombées dans deux domaines, à savoir la polysomnographie et la prévention de la mort subite du nourrisson. Nous allons présenter ces deux contextes ainsi que les résultats obtenus. La recherche s'est surtout concentrée dans le premier domaine.

Nous avons été mis en contact avec le docteur G. Liistro, de l'Unité de Pneumologie du Professeur D. Rodenstein, Faculté de Médecine, qui ont accepté de collaborer à une série de campagnes de mesures.

## 2.2 Contexte : la polysomnographie

Un certain nombre de problèmes de fatigue ou de somnolence diurne excessive peuvent être liés à la mauvaise qualité du sommeil. Parmi les causes possibles, nous pouvons citer :

– les troubles du rythme circadien : les heures où le patient arrive à dormir sont décalées par rapport aux heures « normales »;

- les troubles de rythme du sommeil : la personne s'endort très difficilement, ou n'arrive pas à se rendormir après une phase d'éveil dans le courant de la nuit;
- certaines pathologies se manifestent par des endormissements très rapides dans le courant de la journée

Parfois, le problème est lié à des perturbations du rythme respiratoire. Ces troubles entraînent une mauvaise oxygénation du sang, ce qui se traduit par un sommeil qui n'est pas réparateur.

Pour détecter ou pour traiter de manière adéquate les troubles du sommeil, le médecin peut faire appel à un examen portant le nom de polysomnographie. Comme ce nom l'indique, cette méthode consiste à effectuer une série d'enregistrements de paramètres physiologiques durant le sommeil. Les paramètres d'intérêt visent à étudier d'une part le déroulement des différentes phases de sommeil, et d'autre part à détecter une série de problèmes spécifiques.

Le médecin aura également recours à ce genre d'examen lorsqu'une somnolence diurne ou un mauvais état général ne peuvent a priori être attribués à aucune autre cause organique.

Au niveau des phases du sommeil, l'étude vise à quantifier le temps passé dans chacune de ces phases, leur déroulement successif, leur fréquence d'occurrence, ...

Au niveau des problèmes respiratoires, le médecin recherchera :

1. la présence d'apnées centrales. Ce type d'apnée correspond à une pause de l'activité respiratoire, liée au système nerveux autonome. Le médecin cherchera à connaître leur fréquence et leur durée : il s'agit en effet d'un phénomène normal, qui peut survenir quelques fois par heure, pour de courtes périodes (une dizaine de secondes). Si leur fréquence et/ou leur durée deviennent trop importantes, un traitement adéquat pourra être suivi. En effet, elles mènent à une désaturation en hémoglobine du sang. Si ce phénomène se répète trop souvent, il s'accompagne d'une mauvaise récupération au niveau musculaire.
2. les apnées obstructives. Elles se produisent en cas de blocage des voies aériennes supérieures. Elles s'accompagnent d'un violent effort inspiratoire, sans qu'il n'y ait de flux d'air associé. Ce genre d'apnée perturbe le sommeil, et peut provoquer le réveil du patient. Dans certains cas, ce problème peut se résoudre par le port de masque respiratoire connecté à une pompe, de sorte que l'inspiration se fasse à une pression légèrement supérieure à la pression atmosphérique. De ce fait, on évite l'écrasement des voies respiratoires supérieures responsable de l'apnée obstructive.
3. le ronflement. Bien que ce phénomène n'empêche pas le patient de dormir, il demande un effort inspiratoire plus important, ce qui peut se traduire par une moins bonne récupération musculaire. De plus, le ronflement s'accompagne en général d'apnées obstructives.
4. les mouvements périodiques de jambes.

La polysomnographie est un examen qui se pratique dans un laboratoire spécialisé dans l'étude du sommeil. Il requiert pour le patient de porter un certain nombre de capteurs, qui seront placés par des membres du personnel spécialement formé pour cela. En voici une liste qui n'est pas exhaustive :

1. électrodes au niveau du crâne pour un enregistrement de type EEG;
2. thermistances devant la bouche et devant chaque narine, pour obtenir une estimation du flux d'air;
3. électrodes sur la figure pour un électro-oculogramme;
4. au niveau de la respiration, plusieurs systèmes sont envisageables, visant à détecter les mouvements de la cage thoracique et ceux de l'abdomen : sangles dont on mesure la déformation, électrode pour mesurer l'impédance, ...
5. sonde pléthysmographique pour mesurer la saturation du sang;
6. électrodes sur la poitrine pour un électrocardiogramme;
7. éventuellement, on peut chercher à évaluer la variation de pression artérielle par un dispositif de type Finapress
8. enregistrement audio et vidéo

L'ensemble des signaux électriques sera conditionné par un équipement spécialisé, puis stocké sur un support. La tendance à l'heure actuelle est de travailler en numérique, de compresser le signal et d'utiliser comme support des disques optiques; précédemment, l'enregistrement s'effectuait par voie analogique sur bandes magnétiques qui étaient relues en accéléré pour le dépouillement des résultats.

Les phase du sommeil se déduisent de l'EEG et de l'EOG; le rythme respiratoire est évalué sur base des mouvements respiratoires accompagnés d'un flux d'air; les apnées centrales se détectent sur base d'un arrêt des mouvements respiratoires. Les apnées obstructives sont relativement simples à détecter sur base des mouvements respiratoires; toutefois, les épisodes d'augmentation de résistance à l'inspiration (sans apparition d'apnée) ne peuvent être identifiés correctement que via une mesure de l'effort inspiratoire, ce qui peut être obtenu par une sonde œsophagienne. Néanmoins, l'utilisation de ce genre de sonde contribue à augmenter le coût total et rend l'examen relativement inconfortable.

Le ronflement est détecté via les enregistrements audio, tandis que les mouvements pourront être observés sur l'enregistrement vidéo.

Après dépouillement des données acquises durant le repos du patient, une évaluation assez précise de la qualité de son sommeil pourra être établie. Ces données concernent en général un épisode de sieste dans l'après-midi, et environ huit heures de sommeil nocturne.

Une des difficultés majeures liées à ce genre d'examen est la difficulté d'identifier correctement les événements qui occasionnent une mauvaise qualité du sommeil : typiquement, il s'agit d'apnées obstructives.

En résumé, il s'agit d'un examen fort complexe, coûteux en matériel, en personnel et en temps de traitement. De plus, il s'agit d'une situation peu confortable pour le patient, ce qui est paradoxal puisque l'on en vient à perturber ce qui l'objet principal de l'investigation : le sommeil.

## 2.3 Prévention de la mort subite du nourrisson.

Cette prévention repose sur la détection des apnées et le déclenchement d'une alarme lorsque ce phénomène survient.

Parmi les dispositifs couramment utilisés de nos jours, il existe un matelas muni de capteurs qui détecte les mouvements du bébé qui est couché dessus. Les mouvements de la cage thoracique sont enregistrés, et une alarme est déclenchée en cas de pause de trop longue durée. Par contre, dans le cas d'apnée obstructive, les mouvements thoraciques sont toujours présents, de sorte que ce type d'apnée n'est pas détectable par cette approche.

Ici aussi, un système simple et robuste, capable de détecter et de différencier les états respiratoires, peut offrir des retombées intéressantes.

## 2.4 Améliorations possibles

Dans le contexte de la polysomnographie, les médecins sont à la recherche d'alternatives à ce genre d'examen ou de solutions plus légères. En particulier, ils seraient intéressés par un système permettant de poser un premier diagnostic : soit le patient ne présente pas de troubles significatifs du sommeil, et la cause de ses symptômes de somnolence est à chercher ailleurs; soit il y a des troubles marqués du sommeil, et un examen plus approfondi est nécessaire pour mieux en cerner les causes.

Deux pistes de développement sont susceptibles d'intéresser les médecins :

- rendre le système de mesure plus simple et moins invasif, de façon à ce que son port soit plus confortable pour le patient : de cette façon, le sommeil sera moins perturbé, ce qui rendra le résultat plus représentatif de la réalité;
- rendre le système ambulatoire : il pourra ainsi être utilisé par le patient à son domicile, ce qui évite une hospitalisation et va également dans le sens d'une meilleure représentativité.

## 2.5 Mécanismes physiques

La modulation de la pression artérielle par le rythme respiratoire est un phénomène connu des médecins sous le nom de *pulsus paradoxus* ou « pouls paradoxal »[30]. Le

poumon gauche exerce une pression sur le cœur et en modifie le volume de remplissage. La quantité de sang éjectée à chaque battement dépends donc de manière inversement proportionnelle du volume d'air contenu dans les poumons. Il s'ensuit une variation concomitante de la pression artérielle.

Le pouls paradoxal a été analysé dans trois stades distincts de respiration[29] : respiration normale, ronflement et apnée obstructive. Le résultat de cette étude est que l'amplitude du pouls paradoxal augmente lors des épisodes de ronflement et est maximale en cas d'apnée obstructive. Cette étude ne prend pas en considération les apnées centrales, mais nous nous attendons à ce qu'il n'y ait pas de changement dans cette situation : en effet, la pression alvéolaire reste constante. Ce point a été confirmé par nos observations.

Il existe des dispositifs susceptibles de suivre l'évolution de la pression artérielle comme le Finapress : il s'agit d'une manchette portée à l'index, dont la pression est maintenue constante par un système de rétroaction. L'intensité de cette rétroaction est une image des variations de pression artérielle. Néanmoins, ce genre de dispositif est relativement contraignant à porter et mal toléré par les patients.

Nous avons pensé à utiliser un système de pléthysmographie pour détecter les apnées. Le problème est que la courbe de désaturation du sang présente un phénomène d'hystérèse, et que la durée de l'apnée doit être relativement importante (plus de 30 secondes) avant que la désaturation ne commence à être perceptible.

## 2.6 Dispositif instrumental

Pour être susceptible d'intéresser les médecins, il faut donc arriver à développer un système permettant de caractériser au mieux la qualité du sommeil, minimisant l'inconfort du patient et susceptible d'être utilisé en ambulatoire, pour un examen à domicile. Au terme de l'enregistrement, un traitement adéquat doit permettre de localiser et de caractériser les épisodes problématiques.

Nous avons effectué nos premiers relevés de pression via une manchette normalement utilisée pour la mesure de pression artérielle. Cette manchette était portée au-dessus de la pliure du coude, et maintenue à une pression de 10 mmHg par gonflage préalable.

Pour rendre les mesures moins inconfortables, nous avons ensuite utilisé une manchette de plus petite dimension[1] portée au niveau du poignet : nous avons obtenu des résultats similaires.

La pression est évaluée via un capteur de pression différentiel à jauges de contraintes de type Sensym SCX05DN. L'aspect différentiel résulte de la présence de deux entrées sur le capteur. L'une d'elle est laissée à l'air libre, tandis que l'autre est connectée par

1. en fait, il s'agit d'une manchette à usage pédiatrique

un tuyau flexible à la manchette. De cette façon, nous accédons à la différence entre la pression à l'intérieur de la manchette et la pression atmosphérique.

Nous avons évalué la sensibilité par comparaison avec un capteur de précision de marque Honeywell. Nous avons effectué 15 mesures entre 0 et 25 mmHg, et obtenu la sensibilité par régression linéaire : 0.14 Pa/LSB, ou 0.001 mmHg.

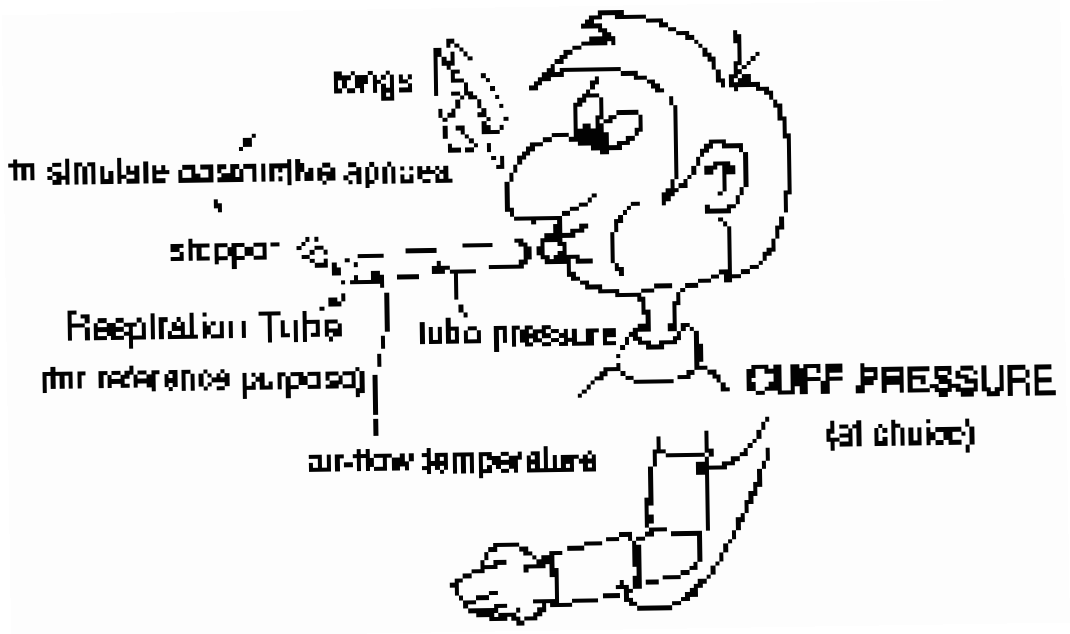



FIG. 2.1: Dispositif de mesure

Le dessin de la figure 2.1 a été effectué par Ch. Stieger. Nous le surnommons « Gaston », et il illustre la configuration utilisée.

Pour caractériser notre système, nous avions besoin de disposer également d'un évaluation du rythme respiratoire : nous avons utilisé pour cela une thermistance placée dans un tuyau à travers lequel le patient respire. La température dans la thermistance ne donne pas le flux thermique, mais permet néanmoins de situer la phase respiratoire dans laquelle le patient se trouve.

La pince sur le nez[2] force la respiration à s'effectuer exclusivement par la bouche.

Un bouchon permet d'obturer le tube afin de simuler une apnée obstructive. Nous avons également mesuré la pression dans le tube, qui est au cours de ces épisodes d'apnées le reflet de la pression intra-alvéolaire. Nous avons exploité cette information pour détecter la présence de ces événements.

2. en pratique, il ne s'agit pas d'une pince « crocodile » comme le suggère le dessin, mais plutôt d'un pince-nez du type de ceux utilisés en piscine.

Chapitre 3

# Analyse par ondelettes

## 3.1 Introduction

Ce chapitre est exclusivement consacré à la présentation de la méthode que nous avons utilisée pour mettre en évidence les signatures cardiaques et respiratoires contenues dans le signal de pression. Étant donné l'ampleur de cette présentation, nous avons préféré y consacrer un chapitre à part, les résultats obtenus faisant l'objet du chapitre suivant. L'intérêt de cette présentation est d'exposer les détails internes des transformations en ondelette, ce qui permettra de justifier les deux approches présentées ultérieurement (travail sans et avec décimation).

## 3.2 Pourquoi les ondelettes ?

Les signaux que nous avons à analyser sont non-stationnaires, bruités, présentent de la dérive et des artefacts. De plus, étant générés par des mécanismes physiologiques, leur fréquence n'est pas constante mais varie en fonction de l'activité métabolique.

Tous ces arguments ne plaident pas en faveur de l'analyse spectrale par la méthode de Fourier. En effet, y retrouver l'information temporelle est difficile : les coefficients complexes contiennent la valeur principale de la phase, c.-à-d. qu'elle est comprise entre $-\pi/2$ et $\pi/2$.

Pour un signal réel, la phase doit être une fonction continue de la fréquence. Il est donc possible de la reconstituer correctement, via des algorithmes de "phase unwrapping" : si le nombre de discontinuités est limité, il est possible de les enlever et de reconstruire un signal continu en extrapolant aux endroits des discontinuités.

Une autre solution consiste à dériver numériquement par rapport à la fréquence, remplacer les points de discontinuité (présentant une valeur absolue supérieure à $\pi/2$) par une interpolation entre les points voisins, et ensuite calculer l'intégrale de ce signal lissé.

Ces méthodes ne sont pas applicables si plusieurs points consécutifs présentent des discontinuités, ou si le signal est quasiment nul (ou très petit, donc susceptible d'être bruité) dans une bande de fréquence.

De plus, l'analyse spectrale de type Fourier présente un certain nombre de problèmes typiques lorsque les fréquences présentes dans le signal ne sont pas des harmoniques de la fréquence d'acquisition : élargissement des pics, lobes secondaires, ...

En outre, le calcul des composantes spectrales se fait par l'algorithme de Transformée Discrète, qui repose sur l'hypothèse implicite que le signal est de durée infinie et de période égale à la fenêtre d'observation. Cet effet se marque notamment dans la transformée inverse : le signal obtenu est la meilleure approximation *périodique* du signal de départ.

Avant d'en arriver à l'exposé sur les ondelettes, nous allons commencer par rappeler quelques notions de bases.

*Remarque* : la lecture de publications consacrées aux ondelettes rappelle à l'auteur une expérience similaire dans un autre domaine : prenez trois livres consacrés aux roses anciennes et comparez la description d'un même rosier. Dans certains cas, on en arrive à se demander si les auteurs parlent réellement de la même variété. Cela résulte de la variabilité de la forme de la plante, de la taille et du coloris de ses fleurs en fonction des conditions de culture et notamment de la température. Cultivé en région parisienne, un rosier pourra être très différent d'un exemplaire de la même variété, cultivé en Californie.

Il en est de même de la littérature consacrée aux ondelettes. La plupart des auteurs parlent de signaux de base appelés $\varphi$ et $\psi$, tandis que d'autres, trouvant ces lettres trop semblables, parlent de $V$ et $W$. En ce qui concerne les coefficients intervenant dans les transformées directes et inverses, ils peuvent être normalisés de façon à ce que les deux opérations s'expriment de façon similaire, ou au contraire présenter une asymétrie. Dans ce cas, la transformée directe peut être écrite sous une forme plus synthétique exempte de facteurs $\sqrt{2}$, mais la transformée inverse devra introduire une correction adéquate.

Nous avons essayé de synthétiser une présentation claire et cohérente dans ce domaine, et avons pris le parti de choisir la version normalisée des coefficients. Que le lecteur ne s'inquiète pas si certaines formules ou certaines données semblent différentes de ce qu'il a déjà rencontré : cette différence représente un facteur $\sqrt{2}$.

## 3.3 Géométrie fonctionnelle

### 3.3.1 Produit scalaire

Le produit scalaire de deux fonctions d'une seule variable se défini comme

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f \cdot \bar{g} \, \mathrm{d}t \tag{3.1}$$

Ce produit scalaire donne une image de la similitude existant entre deux fonctions. En statistique, ce terme vaut la valeur de la fonction de corrélation pour un décalage nul. En physique, si l'une des variables représente un effort et l'autre un flux, l'intégrant est une puissance et le résultat une énergie.

### 3.3.2 Norme d'une fonction

La norme d'ordre 2 est le produit scalaire d'une fonction avec elle-même, c.-à-d.

$$\|f\|^2 = \langle f, f \rangle = \int_{-\infty}^{+\infty} |f(x)|^2 \, \mathrm{d}t \tag{3.2}$$

Lebesgue a défini $\mathrm{L}^2(\mathbf{R})$ comme l'ensemble des fonctions dont la norme est finie. Ces fonctions sont qualifiées de mesurables.

### 3.3.3 Écart entre deux fonctions

La norme de la différence entre deux fonctions permet de définir une classe d'équivalence : $f \equiv g \Rightarrow \|f - g\|^2 = 0$.

Il s'agit d'une classe d'équivalence et non d'une distance : en effet, la propriété d'unicité $d(x,y) = 0 \Rightarrow x = y$ n'est pas satisfaite. Considérons par exemple

$$f(x) = \begin{cases} \sin(x) & 0 \leq x \leq \pi \\ 0 & \text{ailleurs} \end{cases}$$

$$g(x) = \begin{cases} f(x) & x \neq 1 \\ 10 & x = 1. \end{cases}$$

L'écart $\|f - g\|^2 = 0$, et pourtant les fonctions ne sont pas identiques. On a une classe d'équivalence en ce sens que la différence entre les deux fonctions doit être de mesure nulle : il peut s'agir de discontinuités isolées, mais certainement pas de fonctions de type delta de Dirac.

## 3.4 Transformée de Fourier

La transformée continue de Fourier d'une fonction d'une variable qui est conventionnellement appelée le temps, est une fonction d'une autre variable, définie par l'intégrale

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)\ \exp(-j\omega t)\ \mathrm{d}t. \tag{3.3}$$

Cette fonction représente la corrélation entre le signal étudié et une sinusoïde de durée infinie.

Cette relation est inversible et permet d'exprimer une fonction sous la forme d'une somme infinie de sinus et de cosinus.

Un des problèmes de cette approche est que les fonctions de base sont à support global : dans une reconstruction, chaque point résulte de la superposition d'une infinité de sinusoïdes. Dans un signal à durée limitée, les points de valeur nulle en dehors de l'intervalle où le signal est défini sont obtenus par compensation mutuelle de toutes les composantes.

La transformée d'une fonction non-périodique est une fonction continue. Cette représentation est redondante; par exemple un simple créneau limité dans le temps a une représentation quasiment illimitée en fréquence car sa décroissance est de type $1/f$. Dans un tel cas, la sélectivité en fréquence est donc médiocre.

Une fonction périodique possède une transformée discrète. Une telle fonction peut être vue comme la convolution d'une fonction de base, de largeur $T$, avec un train infini d'impulsions, séparées de la même largeur. Le résultat est un spectre discret, constitué de deltas de Dirac. Dans ce cas, la sélectivité en fréquence est excellente, mais la dimension temporelle est gommée : le temps ne permet que de définir à quel endroit du cycle on se trouve.

## 3.5 Caractéristiques de base des ondelettes

Depuis le début des années 1990, une autre méthode d'analyse a commencé a gagner de la popularité, à savoir la transformée en ondelettes. Historiquement, la découverte de ces fonctions remonte à Haar, en 1910 [67]. Une série d'avancées se sont produites successivement dans diverses branches scientifiques, mais il faudra attendre les années 1990 pour voir se dégager une théorie unifiée résultant de la convergence de différents champs d'investigations.

Une ondelette $\psi$ est un signal défini dans le cadre de $\mathrm{L}^2(\mathbf{R})$ [1] Par ailleurs, une ondelette doit satisfaire une condition d'admissibilité au niveau de sa transformée (continue)

1. les fonctions de base de l'analyse de Fourier n'appartiennent pas à cet espace : leur *puissance* est finie.

de Fourier:

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} \, \mathrm{d}\omega < \infty \tag{3.4}$$

où $\hat{\psi}(\omega)$ représente la transformée de Fourier de $\psi$.

Cette condition d'admissibilité implique que la fonction soit de moyenne nulle : en effet $\hat{\psi}(\omega)$ doit valoir zéro.

Une ondelette doit également être limitée dans le temps et dans l'espace des fréquences : les moyennes $\bar{x}$ et $\bar{\omega}$ et les variances $\sigma_x$ et $\sigma_\omega$ doivent être finies. Ces quantités sont définies comme suit :

$$\bar{x} = \frac{1}{\|\psi\|^2} \int_{-\infty}^{+\infty} x|\psi(x)|^2 \, \mathrm{d}x \tag{3.5}$$

et

$$\sigma_x^2 = \frac{1}{\|\psi\|^2} \int_{-\infty}^{+\infty} (x - (\bar{x}))|\psi(x)|^2 \, \mathrm{d}x; \tag{3.6}$$

la définition est similaire pour $\bar{\omega}$ et $\sigma_\omega$.

Pour satisfaire à cette condition, deux comportements sont possibles :

- soit n'être différent de zéro que sur un intervalle fini;
- soit avoir une décroissance de type exponentielle en dehors d'un intervalle fini.

## 3.6 Illustration : l'ondelette de Haar

Avant de poursuivre l'énoncé, illustrons ces notions avec la première ondelette, découverte par Haar [67]. Elle se définit par

$$H(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{ailleurs} \end{cases}$$

Considérons $V_n$, l'ensemble des fonctions constantes sur les intervalles définis par $[j\ 2^{-n}, (j+1)\ 2^{-n}[$, avec $j$,$n \in \mathcal{Z}$. Soit une fonction f « en escalier », comportant $2^m$ points, définie sur l'intervalle [0, 1[. Cette fonction représente par exemple l'approximation d'une fonction à variation continue et à durée limitée. L'intervalle [0, 1[ indique que le temps a été normalisé par rapport à la durée totale.

À partir de l'ondelette de départ, nous générons des répliques par des opérations de translation et de dilatation :

$$H_{j,\,k}(x) = 2^{j/2}\, H(2^j\, x - k), \qquad j,\, k \in \mathcal{Z} \tag{3.7}$$

$$H_{j,\,k}(x) = \begin{cases} 2^{j/2} & 2^{-j}k \le x < 2^{-j}(k+1/2) \\ -2^{j/2} & 2^{-j}(k+1/2) \le x < 2^{-j}(k+1) \end{cases}$$

La fonction de base $H_{0,\,0}$ appartient à $V_1$; les répliques $H_{j,\,k}$ sont dans $V_{j+1}$. À titre d'illustration, prenons une fonction f à quatre points appartenant à $H_2$ et écrivons-la sous forme d'une combinaison linéaire des $H_{j,\,k}$ :

$$f \in H_2 = \sum_{j=-\infty}^{1} \sum_{k=0}^{1} c(j,\,k) H_{j,\,k} \tag{3.8}$$

Étant donné qu'il s'agit d'une approximation, la « meilleure » valeur pour les coefficients $c(j,\,k)$ est donnée par la projection orthogonale de $f$ sur $H_{j,\,k}$ :

$$c(j,\,k) = \frac{< f,\, H_{j,\,k} >}{< H_{j,\,k},\, H_{j,\,k} >} \tag{3.9}$$

Commençons par le niveau le plus fin, à savoir $j = 1$. Nous avons

$$\begin{aligned} c(1,0) &= \frac{\int_0^{1/2} f(x)\; H_{1,\,0}(x)\; \mathrm{d}x}{\int_0^{1/2} H_{1,\,0}(x)\; H_{1,\,0}(x)\; \mathrm{d}x} \\ &= \frac{\sqrt{2}(f(0)\int_0^{1/4} \mathrm{d}x - f(1/4)\int_{1/4}^{1/2} \mathrm{d}x)}{2(\int_0^{1/4} \mathrm{d}x + \int_{1/4}^{1/2} \mathrm{d}x)} \\ &= \frac{\sqrt{2}}{4}(f(0) - f(1/4)) \end{aligned}$$

nous trouvons de même

$$c(1,1) = \frac{\sqrt{2}}{4}(f(1/2) - f(3/4))$$

Nous constatons que le facteur de normalisation $2^{j/2}$ a pour effet de simplifier les calculs; en effet l'intégrale apparaissant au dénominateur est de valeur unitaire.

Calculons ensuite le résidu par rapport à ce niveau, à savoir

$$r_1(x) = f(x) - \sum_k c(1,\,k) H_{1,\,k}(x).$$

Nous obtenons

$$
\begin{aligned}
r_1(0) &= f(0) - (\frac{\sqrt{2}}{4}(f(0) - f(1/4)))(\sqrt{}(2)) = 1/2(f(0) + f(1/4)) \\
r_1(1/4) &= f(1/4) - (\frac{\sqrt{2}}{4}(f(0) - f(1/4)))(-\sqrt{}(2)) = 1/2(f(0) + f(1/4)) \\
r_1(2/4) &= f(2/4) - (\frac{\sqrt{2}}{4}(f(2/4) - f(3/4)))(\sqrt{}(2)) = 1/2(f(2/4) + f(3/4)) \\
r_1(3/4) &= f(3/4) - (\frac{\sqrt{2}}{4}(f(2/4) - f(3/4)))(-\sqrt{}(2)) = 1/2(f(2/4) + f(3/4))
\end{aligned}
$$

Nous constatons que ce résidu appartient à $V_1$ : en effet, il n'y a plus qu'une seule transition, en $x = 1/2$. Pour cette raison, nous considérerons que le résidu ne contient que deux valeurs, $r(0)$ et $r(1/2)$.

Poursuivons la décomposition, en prenant la projection sur $H_{0,\,0}$ du résidu $r_1$ :

$$
\begin{aligned}
c(0,0) &= \frac{\int_0^1 r_1(x)\; H(x)\; \mathrm{d}x}{\int_0^1 H(x)\; H(x)\; \mathrm{d}x} \\
&= \frac{(r_1(0) \int_0^{1/2} \mathrm{d}x - r_1(1/2) \int_{1/2}^{1/1} \mathrm{d}x)}{\int_0^{1/2} \mathrm{d}x + \int_{1/2}^1 \mathrm{d}x} \\
&= \frac{(r_1(0) - r_1(1/2))}{2}
\end{aligned}
$$

Le résidu $r_0$ vaut :

$$
\begin{aligned}
r_0(0) &= r_1(0) - 1/2(r_1(0) - r_1(1/2))(1) = 1/2(r_1(0) + r_1(1/2)) \\
r_0(1/2) &= r_1(1/2) - 1/2(r_1(0) - r_1(1/2))(-1) = 1/2(r_1(0) + r_1(1/2))
\end{aligned}
$$

Ce résidu est à présent constant sur l'intervalle $[0,\, 1[$ et vaut simplement la moyenne du signal initial. Ce résidu pourrait à son tour être décomposé en fonction de $H_{-1,\,0}$, $H_{-2,\,0}$, $H_{-3,\,0}$, ... Nous trouverions que les résidus $r_{-j}$ valent $2^{-j}\; r_0$. Cette décomposition comporte une infinité de termes, à décroissance exponentielle.

Plutôt que de décomposer un signal à durée limitée en une série infinie de termes, on préfère limiter la décomposition. Partant de $V_2$, nous avons tout d'abord obtenu les coefficients $c_1$. Les résidus $r_1$ appartiennent à $V_1$. L'information contenue dans les coefficients $c_1$ constitue donc les détails nécessaires pour passer du niveau $V_1$ au niveau $V_2$. Si nous considérons la génération de ces coefficients $c_1$ comme une opération de projection que nous notons $Q_1$, nous pouvons associer les résidus à une autre opération

de projection, $P_1$, qui lui est orthogonale. En effet, $(I - Q_1).Q_1$ vaut $(Q_1 - Q_1) = 0$ car l'opérateur de projection appliqué deux fois est égal à lui-même.

Appelant $W_1$ l'espace généré par la base de $Q_1$, à savoir $H_1$, nous en concluons que $V_2$ se décompose en deux constituantes, $V_1$ et $W_1$, dont les opérateurs de projection sont orthogonaux. Il en résulte que les bases respectives sont orthogonales, et nous avons

$$V_2 = V_1 \oplus W_1 \tag{3.10}$$

Le symbole $\oplus$ dénote la somme directe, c.-à-d. que chaque élément de $V_2$ s'écrit de façon unique comme la somme d'un élément de $V_1$ et d'un élément de $W_1$.

La base associée à l'opérateur de projection $P_1$ donnant les résidus est appelée « fonction d'échelle ». Dans le cas de l'ondelette de Haar, il s'agit d'un créneau de largeur unitaire, à partir duquel on génère des répliques par translation et dilatation.

Par récurrence, en décomposant $V_i = V_{i-1} \oplus W_{i-1}$, nous pouvons écrire que

$$V_n = \bigoplus_{k=-\infty}^{n-1} W_k = V_0 \bigoplus_{k=0}^{n-1} W_k \tag{3.11}$$

Nous interprétons cela de deux manières :

- une fonction en escaliers appartenant à $V_n$ peut se décomposer exactement en une série (infinie) de fonctions de bases, obtenues par dilatation et translation de l'ondelette de départ $H$;

ou

- une fonction appartenant à $V_n$ est la somme (finie) de détails obtenus par dilatation et translation de $H$, ainsi que d'approximations appartenant à $V_0$, obtenus par translation de la fonction d'échelle.

Cet exemple a permis également d'introduire la notion de multirésolution : la fonction initiale comportait quatre points. Nous avons obtenu une information équivalente via deux coefficients $c_1$ et deux résidus $r_1$, soit un total inchangé de quatre points. Ces résidus ont à leur tour été décomposés en un coefficient $c_0$ et une approximation (un résidu) $r_0$.

Le nombre de points présents à chaque étape n'est donc pas constant, mais présente une progression logarithmique, tandis que le nombre total de points est invariant. Intuitivement, la résolution est adaptée en fonction du nombre de détails contenus dans le signal.

Pour résumer les notions importantes :

- Nous avons considéré $L^2(\mathbf{R})$, l'ensemble des fonctions d'énergie finie. Typiquement, de telles fonctions sont à durée limitée.

– Nous avons décomposé $V_n$, l'ensemble des fonctions constantes sur des intervalles de largeur $2^{-n}$, en une somme de sous-espaces indépendants. Ces sous-espaces ont pour bases des versions dilatées et translatées d'un signal de départ appelé ondelette, répondant à un certain nombre de critères mathématiques.
– La représentation de la fonction à résolution constante, est équivalente à une représentation constituée d'une suite de détails et d'une approximation finale, chacun des éléments possédant une résolution spécifiquement adaptée.

## 3.7 Formalisation

Nous allons présenter avec un peu plus de détails les principales notions mathématiques sous-jacentes à la théorie des ondelettes.

Le nom « ondelette » évoque les trois caractéristiques principales de ces signaux, qui se déduisent de l'appartenance à $L^2(\mathbf{R})$ et de la condition d'admissibilité 3.4 :

– ils sont à durée limitée (dans le temps comme dans l'espace des fréquences);
– en temporel, ils sont de moyenne nulle et ils oscillent autour de 0.
– en fréquentiel, ils peuvent être considérés comme un filtre passe-bande.

À partir d'une ondelette de départ, appelée $\psi_0$, on définit une famille de fonctions par des opérations de *dilatation* et de *translation* :

$$\psi_{a,\,b}(x) = \frac{1}{\sqrt{|a|}}\,\psi\Big(\frac{x-b}{a}\Big) \quad \text{avec } a,\, b \in \mathbf{R}, a \neq 0 \tag{3.12}$$

La dilatation dépend de $a$ tandis que la translation est donnée par $b$. Sous cette forme, $b$ est à la même « échelle » que $x$. Le premier terme est un facteur de normalisation.

En langage mathématique, de tels signaux sont appelés *à support compact*. Cette notion signifie que l'ensemble de départ $L^2(\mathbf{R})$ est « couvert » (généré) par un nombre *fini* de fonctions de base. Cela résulte de la durée limitée : en tant que signal analysant, une ondelette donnera de l'information sur une fenêtre temporelle et sur une fenêtre fréquentielle, contrairement aux fonctions de base de la transformée de Fourier où nous avons le paradoxe suivant : chaque signal de base recouvre la totalité de l'espace temporel. Les valeurs nulles d'un signal à durée limitée se trouvant en dehors de l'intervalle où le signal existe, sont obtenues par compensation mutuelle d'une infinité de signaux de base.

À titre d'exemple de fonction satisfaisant les conditions d'admissibilité, nous pouvons citer le « sombrero », illustré dans les figures 3.1 et 3.2 :

$$\psi(x) = (1 - 2x^2)\; e^{-x} \tag{3.13}$$

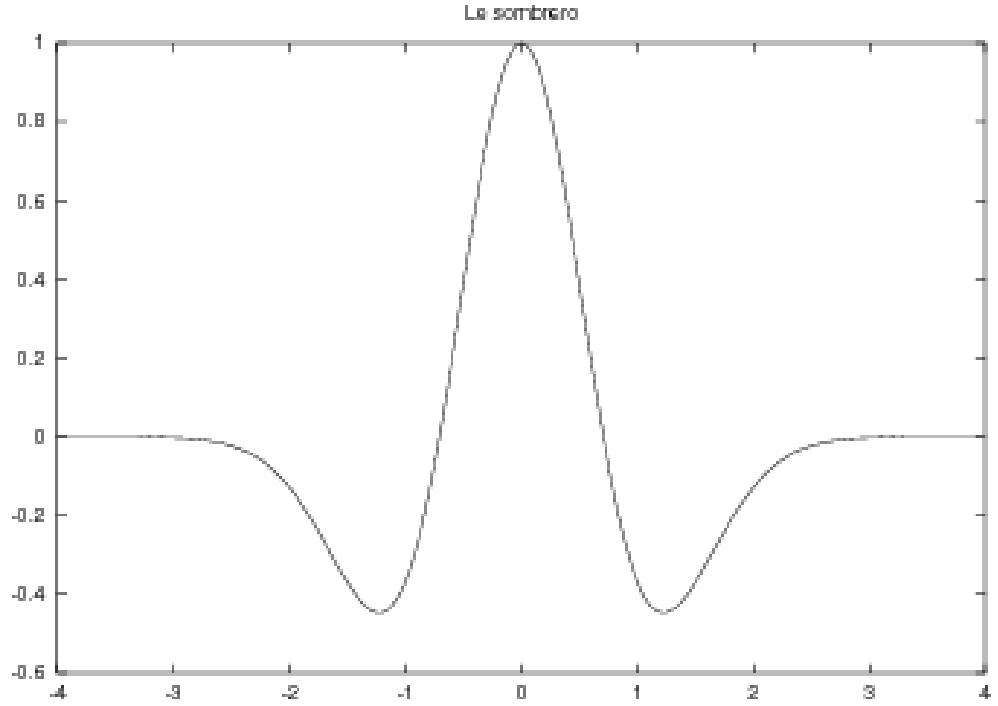



FIG. 3.1: Le sombrero

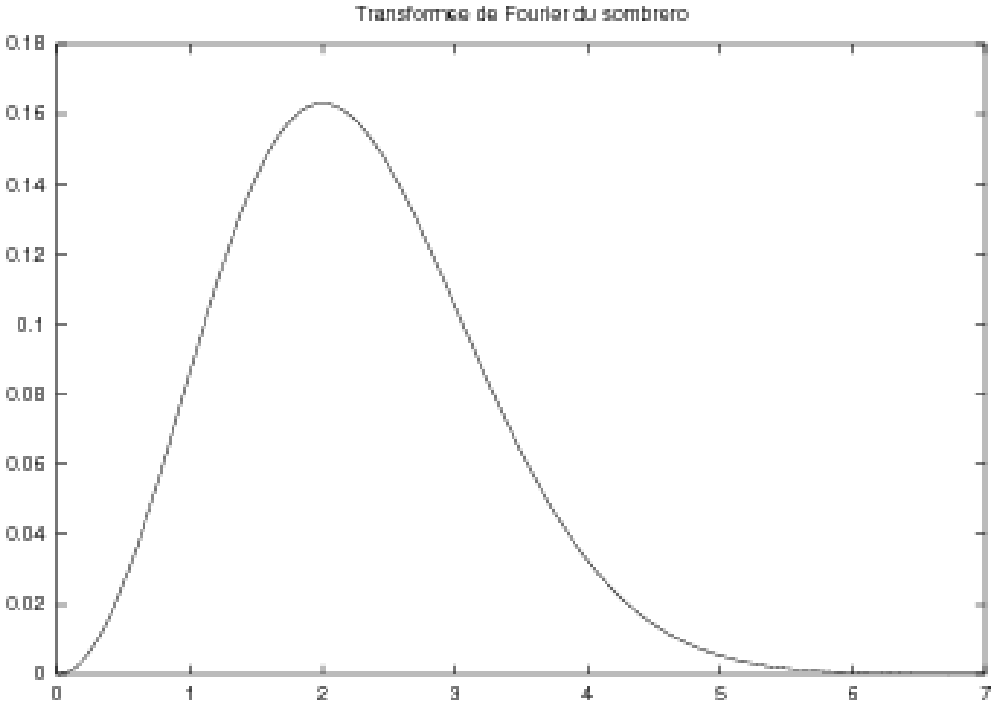



FIG. 3.2: Transformée de Fourier

Cette fonction est continue et dérivable, il s'agit de la dérivée seconde d'une gaussienne. La version temporelle est limitée à l'intervalle $[-3, 3]$ et la version fréquentielle est comprise entre $[-2\pi, 2\pi]$.

La transformée en ondelette $W$ d'une fonction $f$ par une ondelette $\psi$ est une autre fonction, paramétrée par les variables $a$ et $b$.

$$W(a, b) = \langle f, \psi_{a, b} \rangle \tag{3.14}$$

Ces coefficients indiquent combien la fonction, prise à l'échelle $a$ et observée à proximité de $b$, « ressemble » à l'ondelette. Une façon de représenter cette fonction est de tracer un graphique à deux dimensions dont les axes sont respectivement $a$ et $b$. Chaque point est marqué en gris, la grandeur de $W(a, b)$ étant transformée en niveau d'intensité compris entre zéro (blanc) et un (noir). La valeur zéro représente un coefficient nul tandis que la valeur un est attribuée au coefficient d'amplitude maximale.

Si la condition d'admissibilité (3.4) est satisfaite, cette transformation est inversible :

$$f = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W(a, b)\ \psi_{a, b}\ \frac{\mathrm{d}a\mathrm{d}b}{a^2} \tag{3.15}$$

Il est donc possible de manipuler la transformée d'un signal (filtrage, élimination de certains niveaux) puis de resynthétiser un signal temporel.

On parlera de transformée continue si $a$ et $b$ peuvent prendre a priori n'importe quelle valeur. Dans le cas de valeurs discrètes, un choix commun consiste à utiliser un découpage dyadique : $a = 2^{-j}$ et $b = k\ a$ avec $j, k \in \mathbf{Z}$.

Parmi les utilisations typiques de la transformée continue, citons la détection et la localisation de singularités, la caractérisation de fonctions (en étudiant la convergence des coefficients) et l'étude de signaux dont les propriétés varient dans le temps.

### 3.7.1 Multirésolution

#### Approximations et fonctions d'échelle

Dans le cadre des transformées discrètes, Mallat [70] a introduit la notion de multirésolution, consistant à définir une séquence de sous-espaces. Ces sous-espaces répondent à une définition : dans l'exemple sur les ondelettes de Haar, il s'agissait des fonctions constantes sur des intervalles bien définis. Plus prosaïquement, ces espaces sont générés par des fonctions de base, et les propriétés suivantes indiquent les conditions auxquelles elles doivent satisfaire :

1. $V_j \subset V_{j+1}$ ; les caractéristiques de $V_{j+1}$ sont également applicables à $V_j$.
2. $v(x) \in V_j \Leftrightarrow v(2x) \in V_{j+1}$ ; le passage d'une dimension à la suivante implique une dilatation par deux de l'échelle temporelle.

3. $v(x) \in V_0 \Leftrightarrow v(x+1) \in V_0$; l'espace de base est invariant par rapport à une translation unitaire. Dans le cas de Haar, cette condition est satisfaite, puisque par définition $V_0$ est l'ensemble des fonctions constantes sur l'intervalle $[k, k+1[$; $k \in \mathcal{Z}$.
4. $\bigcup_{j=-\infty}^{+\infty} V_j$ est dense sur $L^2(\mathbf{R})$ et $\bigcap_{j=-\infty}^{+\infty} V_j = \{\mathbf{0}\}$. Cette condition est plus technique. La notion de densité implique l'existence de suites de Cauchy convergentes, et donc la possibilité d'obtenir une approximation avec un écart (une erreur) arbitrairement petit. L'intersection vide implique qu'il n'y ait pas de redondance entre les définitions des sous-espaces successifs.
5. il existe une fonction d'échelle ("scaling function") $\varphi \in V_0$ telle que l'ensemble $\{\varphi(x-l) \,|\, l \in \mathbf{Z}\}$ forme une base de $V_0$.

Comme l'indique l'avant-dernière condition, ces sous-espaces servent à approximer de plus en plus finement l'espace de départ $L^2(\mathbf{R})$, c.-à-d. réduire la distance $\epsilon_n^2 = \|f - f_n\|^2$ où $f$ est une fonction appartenant à $L^2(\mathbf{R})$ et $f_n$ son approximation de niveau $n$.

Puisque $V_0$ est inclus dans $V_1$, n'importe quelle fonction de $V_0$ peut s'écrire comme une combinaison linéaire des fonctions de $V_1$. On a en particulier :

$$\varphi(x) = 2 \sum_k h_k \, \varphi(2x-k) \tag{3.16}$$

Cette équation récursive porte le nom d'équation de dilatation ou de raffinement. Par la suite, nous verrons comment l'utiliser pour générer des familles d'ondelettes.

**Détails et ondelettes.**

Introduisons à présent les sous-ensembles $W_j$ [2] : il s'agit d'espaces tels que

$$V_{j+1} = V_j \oplus W_j \tag{3.17}$$

Ces espaces contiennent les « détails » nécessaires pour passer d'un niveau d'approximation au suivant. Comme indiqué dans la section sur l'analyse de l'ondelette de Haar, n'importe quelle approximation (y compris $L^2(\mathbf{R})$, obtenu comme limite de $V_n$ lorsque $n \to \infty$) peut donc s'écrire comme somme directe des sous-espaces $W_j$.

Une ondelette $\psi$ est une fonction telle que l'ensemble $\{\psi(x-l) \,|\, l \in \mathbf{Z}\}$ forme une base de $W_0$. Puisque $\psi \in W_0 \subset V1$, nous pouvons utiliser les mêmes arguments que ceux qui ont mené à l'équation de raffinement et écrire

$$\psi(x) = 2 \sum_k g_k \, \varphi(2x-k) \tag{3.18}$$

2. à ne pas confondre avec $W_{a,\,b}$

### Multirésolution orthogonale

Une analyse en multirésolution est orthogonale si

$$W_0 \perp V_0 \tag{3.19}$$

ce qui est équivalent à

$$\langle \psi(x),\, \varphi(x-l) \rangle = 0 \qquad l \in \mathcal{Z} \tag{3.20}$$

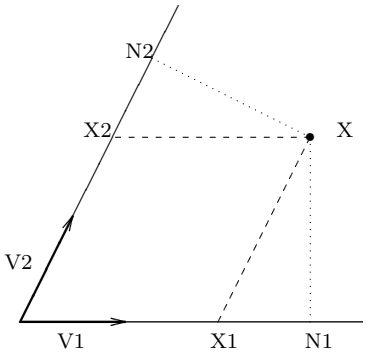



FIG. 3.3: propriétés d'orthogonalité.

Illustrons l'importance de la notion d'orthogonalité. La figure 3.3 représente un point $X$ situé dans un espace où les vecteurs de base sont représentés en traits gras. Les coordonnées de $X$ dans cet espace s'obtiennent en construisant un parallélogramme, dont les sommets opposés sont occupés respectivement par l'origine et par le point $X$, et dont les côtés sont parallèles aux vecteurs de base. Les coordonnées s'obtiennent en prenant le rapport entre la longueur d'un côté ($X1$ ou $X2$) et celle du vecteur de base qui lui est parallèle ($V1$ ou $V2$). Les points $N1$ et $N2$ représentent les projections orthogonales de $X$ sur les axes de coordonnées. Étant donné que la base n'est pas orthogonale, le parallélogramme (en traits discontinus) n'est pas identique au quadrilatère (en pointillé) limité par les projections de $X$ sur les axes : les projections orthogonales, qui s'obtiennent par un produit scalaire, ne donnent pas les coordonnées dans la base $V1, V2$.

### Fonctions d'échelle et ondelettes orthogonales

Une ondelette est orthogonale si

$$\langle \psi(x),\, \psi(x-l) \rangle = \delta_l \qquad l \in \mathcal{Z} \tag{3.21}$$

La définition d'une fonction d'échelle orthogonale est similaire.

Une manière possible d'obtenir une ondelette orthogonale à partir d'une fonction d'échelle orthogonale est d'avoir

$$g_k = (-1)^k \, \overline{h_{1-k}} \tag{3.22}$$

Remarque : la première lettre apparaissant comme indice de $h$ est le chiffre '1' en minuscule.

Une démonstration complète requiert d'étudier la périodicité de la transformée de Fourier de $g$ et $h$. Nous allons simplement montrer que cette relation entraîne bien la nullité du produit scalaire lorsque, comme c'est le cas avec les ondelettes que nous utiliserons, le nombre de coefficients est pair :

$$\langle \psi(x),\, \varphi(x) \rangle = \langle \sum_k g_k \, \varphi(2x-k),\, \sum_m h_m \, \varphi(2x-m) \rangle \tag{3.23}$$

$$= \sum_k g_k \, \sum_m h_m \, \langle \varphi(2x-k),\, \varphi(2x-m) \rangle \tag{3.24}$$

De par l'orthogonalité de la fonction d'échelle, les termes avec $k \neq m$ sont nuls et ceux pour lesquels $k = m$ valent 1. La double somme se réduit finalement à

$$\sum_k (-1)^k \, \overline{h_{1-k}} \, h_k \tag{3.25}$$

Il s'agit d'une somme de produits. Chacun des terme de la somme apparaîtra deux fois, avec les facteurs $(-1)^k$ et $(-1)^{1-k}$. Les parités de $k$ et $1-k$ étant opposées, la somme se composera de termes se compensant deux à deux, le résultat final sera donc bien nul.

## 3.8 Construction de fonctions d'échelle et d'ondelettes

En fonction de l'application envisagée, on recherche un certain nombre de propriétés chez les ondelettes utilisées :

1. support compact
2. coefficients rationnels (ou dyadiques, pour optimiser les calculs sur ordinateurs)
3. symétrie
4. régularité
5. nombre de moments nuls : on définit le moment d'ordre $p$ d'une ondelette comme

$$\mathcal{N}_p = \int_{-\infty}^{+\infty} x^p \, \psi(x) \mathrm{d}x \quad \text{avec } p \in \mathbf{N} \tag{3.26}$$

   Le nombre de moments nuls est égal au degré du polynôme qu'une ondelette peut approximer sans erreur.

6. existence d'une expression analytique.

En pratique, une ondelette représentera un compromis entre ces différentes propriétés.

### 3.8.1 Construction par l'équation de raffinement

Cette approche consiste à exploiter l'équation 3.16 récursivement : connaissant les points en 0, 1, 2, ... on calcule ensuite les points en 1/2, 3/2, 5/2, ... puis en 1/4, 3/4, ... Par cette approche, la valeur de la fonction n'est connue qu'en des points dyadiques (multiples de $2^j$, $j \in \mathcal{Z}$).

Illustrons cela par un exemple : la fonction d'échelle pour les ondelettes de Daubechies [63] d'ordre 4 se base sur le jeux de coefficients[3] $c_0 = \frac{1}{4}(1+\sqrt{3})$, $c_1 = \frac{1}{4}(3+\sqrt{3})$, $c_2 = \frac{1}{4}(3-\sqrt{3})$ et $c_3 = \frac{1}{4}(1-\sqrt{3})$. Les coefficients $c_i$ valent deux fois les coefficients $h_i$ de l'équation de raffinement. Cette fonction d'échelle est nulle pour $x \leq 0$ et $x \geq 3$. Nous pouvons calculer les valeurs en $x = 1$ et $x = 2$ :

$$\varphi(1) = \frac{1}{4}\left(3+\sqrt{3}\right)\varphi(1) + \frac{1}{4}\left(1+\sqrt{3}\right)\varphi(2) \tag{3.27}$$

$$\varphi(2) = \frac{1}{4}\left(1-\sqrt{3}\right)\varphi(1) + \frac{1}{4}\left(3-\sqrt{3}\right)\varphi(2) \tag{3.28}$$

Cette équation, du type $\varphi = L\,\varphi$, admet comme solution les vecteurs propres de $L$ : dans le cas de $D4$, prenons $\varphi(1) = \frac{1}{2}(1+\sqrt{3})$ et $\varphi(2) = \frac{1}{2}(1-\sqrt{3})$. À partir de ce moment, il est possible de trouver les valeurs dyadiques de $\varphi$ par récursion. Par exemple, nous avons

$$\begin{pmatrix} \varphi(\frac{1}{4}) \\ \varphi(\frac{5}{4}) \\ \varphi(\frac{9}{4}) \end{pmatrix} = \begin{bmatrix} c_0 & 0 & 0 \\ c_2 & c_1 & c_0 \\ 0 & c_3 & c_2 \end{bmatrix} \begin{pmatrix} \varphi(\frac{1}{2}) \\ \varphi(\frac{3}{2}) \\ \varphi(\frac{5}{2}) \end{pmatrix} \tag{3.29}$$

Une des particularités de cette construction est qu'elle est obtenue par *itération* d'un opérateur : la courbe qui en résulte présente des caractéristiques fractales : observée à une certaine échelle, cette courbe comporte des similitudes avec la vue à une autre échelle.

La fonction d'échelle de base est représentée à la figure 3.4. On peut notamment observer que le motif compris entre 0.5 et 1 se retrouve entre 0.25 et 0.5, ainsi qu'entre 0.125 et 0.25, ...

3. il serait trop long d'expliquer comment sont générés ces coefficients. Une série de propriétés auxquelles ils doivent répondre n'ont pas été exposées.

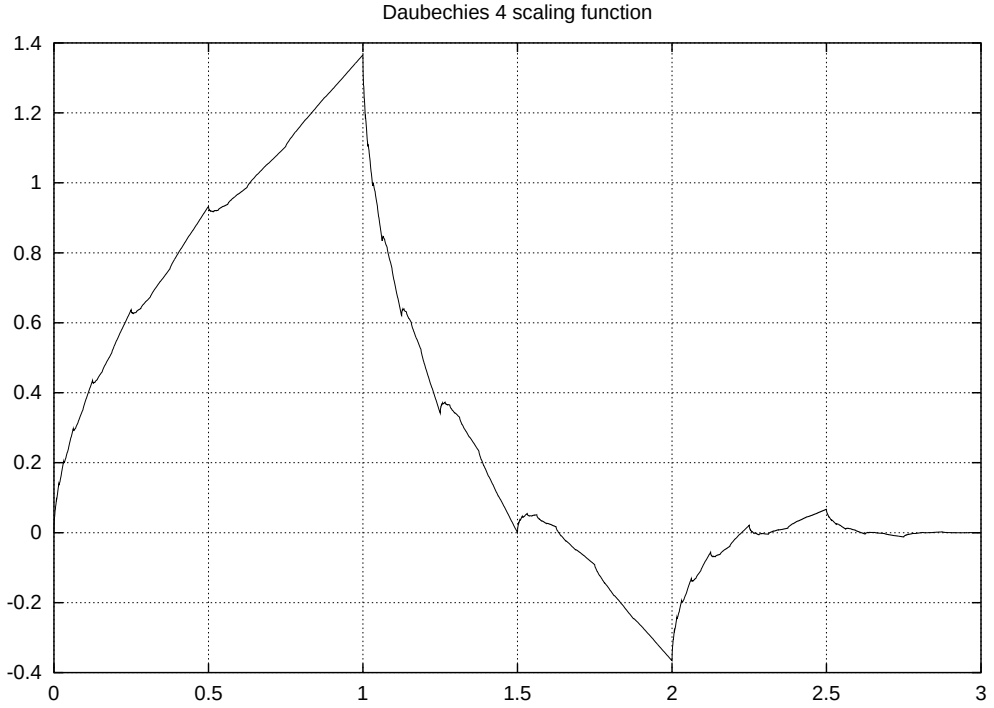



FIG. 3.4: Fonction d'échelle de Daubechies d'ordre 4

La figure 3.5 illustre l'ondelette de Daubechies d'ordre 4, obtenue en appliquant la définition 3.18, avec la valeur des coefficients $g_k$ telle que donnée dans 3.22, à savoir

$$\psi(x) = c_3\ \varphi(2x) - c_2\ \varphi(2x-1) + c_1\ \varphi(2x-2) - c_0\ \varphi(2x-3) \tag{3.30}$$

### 3.8.2 Construction utilisant la transformée de Fourier

La transformée de Fourier de l'équation de raffinement donne :

$$\hat{\varphi}(\omega) = \sum_k h_k \int \varphi(2x-k)\ \exp(-j\ \omega\ x)\ \mathrm{d}x$$

avec le changement de variables $y = 2x - k$, il vient :

$$\begin{aligned}
\hat{\varphi}(\omega) &= \sum_k h_k \int \varphi(y)\ \exp(-j\ \omega\ \frac{y+k}{2})\ \mathrm{d}y/2 \\
&= \frac{1}{2}\sum_k h_k\ \exp\left(-j\ k\ \omega/2\right) \int \varphi(y)\ \exp(-j\ \omega\ /2)\ \mathrm{d}y \\
&= \frac{1}{2}(\sum_k h_k\ \exp\left(-j\ k\ \omega/2\right))\ \hat{\varphi}(\frac{\omega}{2})
\end{aligned}$$

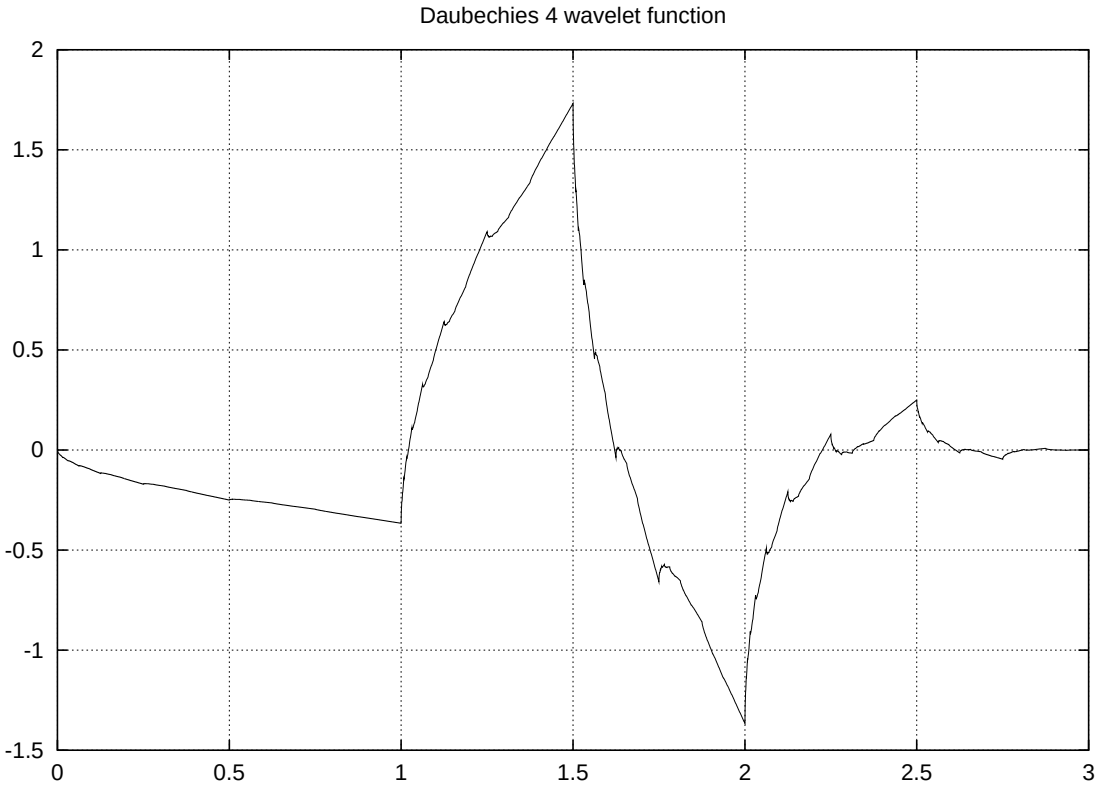



FIG. 3.5: Ondelette de Daubechies d'ordre 4

Posant $P(\omega) = \frac{1}{2}\sum_k h_k \ \exp(-j\ k\ \omega)$, nous pouvons appliquer cette formule récursivement :

$$\hat{\varphi}(\omega) = \left[\prod_{k=1}^{N} P(\frac{\omega}{2^k})\right] \ \hat{\varphi}(\frac{\omega}{2^N}) \tag{3.31}$$

dont la limite vaut

$$\hat{\varphi}(\omega) = \left[\prod_{k=1}^{\infty} P(\frac{\omega}{2^k})\right] \ \hat{\varphi}(0) \tag{3.32}$$

À condition que les termes $\hat{\varphi}(0)$ et $P(0)$ soient distincts de zéro, nous pouvons donc calculer directement la transformée de Fourier de la fonction d'échelle, sur base uniquement des coefficients $h_k$. Dans ce cas, nous ne sommes pas limités à un échantillonnage dyadique.

## 3.9 Transformée rapide en ondelette

### 3.9.1 Décomposition pyramidale

Nous allons à présent voir qu'il n'est pas nécessaire de calculer explicitement les fonctions d'échelles et d'ondelettes pour obtenir la transformée d'un signal, lorsque la décomposition en multirésolution ainsi que les fonctions de base sont orthogonales.

Une fonction $v_j \in V_j$ peut s'écrire comme la somme d'une fonction $v_{j-1} \in V_{j-1}$ et d'une fonction $w_{j-1} \in W_{j-1}$

$$v_j(x) = \sum_k \lambda_{j,\,k}\,\varphi_{j,\,k}(x) = v_{j-1}(x) + w_{j-1}(x) \tag{3.33}$$

$$= \sum_l \lambda_{j-1,\,l}\,\varphi_{j-1,\,l}(x) + \sum_l \gamma_{j-1,\,l}\,\psi_{j-1,\,l}(x) \tag{3.34}$$

Par définition des coefficients, nous avons

$$\lambda_{j-1,\,l} = \langle v_j,\ \varphi_{j-1,\,l}\rangle \tag{3.35}$$

Appliquant la relation de dilatation 3.16, il vient successivement :

$$\begin{aligned}\varphi_{j-1,\,l} &= \sum_k h_k\ \sqrt{2^{j-1}}\ \varphi(2(2^{j-1}x - l) - k)\\ &= \sqrt{2}\ \sum_k \sqrt{2^j}\ \varphi(2^j x - 2l - k)\\ &= \sqrt{2}\ \sum_k \varphi_{j,\,2l+k}\end{aligned}$$

Ceci nous mène à

$$\begin{aligned}\lambda_{j-1,\,l} &= \sqrt{2}\,\langle v_j,\ \sum_k h_{k-2l}\ \varphi_{j,k}\rangle\\ &= \sqrt{2}\langle \sum_k \lambda_{j,\,k}\ \varphi_{j,\,k},\ \sum_m h_m\ \varphi_{j,\,2l+m}\rangle\\ &= \sqrt{2}\sum_k \lambda_{j,\,k}\ \sum_{m-2l} h_{m-2l}\ \langle \varphi_{j,\,k},\ \varphi_{j,\,m}\rangle\end{aligned}$$

Dans cette expression, chacun des membres ne contient que des termes du niveau $j$. Ayant fait le choix de fonctions de base orthonormées, les produits scalaires donnent zéro si $m \neq k$ et un pour $m = k + 2l$. Le résultat final vaut donc

$$\lambda_{j-1,\,l} = \sqrt{2}\sum_k h_k\ \lambda_{j,\,k+2l} \tag{3.36}$$

Nous obtenons de manière similaire

$$\gamma_{j-1,\ l} = \sqrt{2} \sum_k g_k\ \lambda_{j,\ k+2l} \tag{3.37}$$

Le facteur $\sqrt{2}$ est un facteur de normalisation, de façon à maintenir constante l'énergie totale.

Numériquement, cette opération peut être implémentée comme un filtrage, les coefficients du filtre étant égaux aux coefficients ayant généré la fonction d'échelle et l'ondelette. Toutefois, il y a une une particularité : après avoir filtré le signal $\lambda_{j,\ k}$ par $h_k$, nous devons ensuite appliquer une opération de *décimation*[4] pour tenir compte du facteur $+2l$ : seuls les termes d'indice pair sont conservés.

Appliquée récursivement, cette méthode porte le nom de transformée en ondelette rapide, ou encore de décomposition pyramidale : le point de départ est un signal de longueur $2^n$. Ce signal est filtré deux fois, par $g_k$ pour donner un « détail », et par $h_k$ pour donner une « approximation ». Le nombre de point est réduit de moitié par décimation. À la sortie des filtres, nous avons donc deux signaux de longueur $2^{n-1}$, de sorte que la longueur totale est inchangée. On applique ensuite cette opération sur l'approximation, fournissant un détail et une approximation de niveau inférieur, jusqu'à obtenir une approximation ne contenant plus qu'un seul point.

Graphiquement, la représentation traditionnelle de cette opération est illustrée dans la figure 3.6.

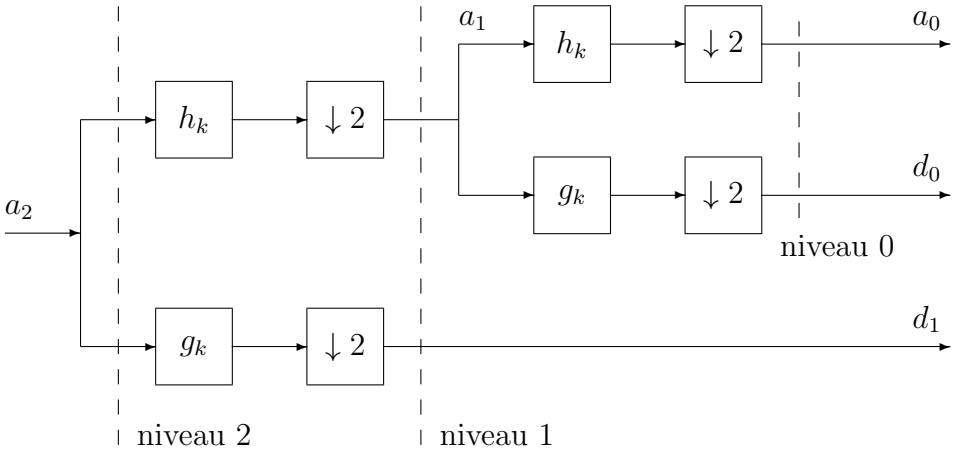



FIG. 3.6: Transformée Rapide en Ondelette

4. l'étymologie de ce terme est une punition appliquée chez les soldats romains aux légions qui avaient failli au combat : un homme sur 10, pris au hasard, était exécuté. Cela permettait de ne pas trop réduire les effectifs de la légion. Quant aux survivants, leur meilleure chance de survie à long terme était d'être victorieux lors des combats ultérieurs.

Nous pouvons à présent interpréter les coefficients $h_k$ et $g_k$. Posant $v_j = \varphi_{j,\,0}$ dans 3.33, c.-à-d. ne considérant qu'un seul terme dans la décomposition, les termes $\lambda_{j-1,\,l}$ qui résultent de la projection de $\varphi_{j,\,0}$ sur $\varphi_{j-1,\,l}$ sont tout simplement égaux à $h_k$. Ces coefficients qui servent à générer la fonction d'échelle valent le produit scalaire de la fonction d'échelle dilatée par deux avec la fonction elle-même.

Au niveau de la reconstruction, les propriétés d'orthogonalité rendent l'opération symétrique :

$$\lambda_{j,\,k} = \sqrt{2} \sum_l h_{k-2l}\ \lambda_{j-1,\,l} + \sqrt{2} \sum_l g_{k-2l}\ \gamma_{j-1,\,l} \tag{3.38}$$

Dans ce cas, l'opération de filtrage est précédée d'une interpolation ("upsampling") consistant à intercaler un zéro entre deux échantillons consécutifs.

Il reste le problème des conditions limites. Par définition, la transformée s'applique à un nombre fini de points. En extrayant artificiellement un bloc de données, on crée des discontinuités aux extrémités du signal. Les ondelettes étant particulièrement efficaces pour détecter une singularité, cela va donner lieu à une perturbation des coefficients. Plusieurs solutions sont possibles:

1. ignorer la zone perturbée;
2. remplacer les extrémités par la moyenne entre les valeurs extrêmes
3. symétriser le signal par rapport aux extrémités.

### 3.9.2 Propriétés des filtres

De par leur construction (3.22), les filtres $h_k$ et $g_k$ sont symétriques par rapport à la fréquence : si nous notons par $H(\omega)$ et $G(\omega)$ leur réponse en fréquence et supposons que celle-ci est normalisée ($f_s = 1$), $H(\omega)$ est un passe-bas dans la bande $[0, \pi/2]$ et $G(\omega)$ un passe-haut dans la bande $[\pi/2, \pi]$.

Après une étape, l'approximation suivante correspond au filtre $H(2\omega)H(\omega)$, passe-bas dans la bande $[0, \pi/4]$ tandis que le détail résultant de $H(2\omega)G(\omega)$ correspond à un passe-bande dans $[\pi/4, \pi/2]$. La décomposition consiste donc à diviser les intervalles fréquentiels en blocs de largeur $[2^j\,\pi, 2^{j+1}\,\pi]$. Cette analyse se fait par octave, avec des filtres dont le facteur de qualité, rapport entre la fréquence centrale et la largeur, reste constante. La réponse de quatre filtres consécutifs est illustrée dans la figure 1.4, page 78.

### 3.9.3 Propriétés de la Transformée

Cette transformation conserve la norme (identité de Parseval)

$$\|f\| = \sqrt{\sum_{j,\ k} \gamma_{j,\ k}^2} \quad j,\ k \in \mathcal{Z} \tag{3.39}$$

$$\|f\| = \sqrt{\sum_{k} \lambda_k^2 + \sum_{j,\ k} \gamma_{j,\ k}^2} \quad k \in \mathcal{Z},\ j > 0 \tag{3.40}$$

Il est donc possible de décomposer de manière unique un signal quelconque en une somme d'ondelettes dilatées et translatées : dans ce cas, on caractérisera le signal par les coefficients des ondelettes.

L'élément $\gamma_{j,\ k}$ représente l'information autour de la fréquence $2^j$ et du temps $2^{-j}\ k$, la somme sur k représentant le *détail* au niveau $2^{-j}$. On traduit cette propriété en disant que les coefficients représentent une analyse temps-fréquence du signal.

C'est cette propriété qui est à la base du succès des ondelettes. En effet, le temps est une variable à part entière, et une modification temporelle des caractéristiques d'un signal donnera lieu à une modification de la partie correspondante de sa transformée en ondelette.

### 3.9.4 Approximation

Un résultat important dépend de la propriété de densité citée dans la section 3.7.1 consacrée à la décomposition en multirésolution. Cette propriété implique qu'il est possible d'approximer une fonction quelconque avec une précision arbitrairement petite.

Nous avons évoqué la nature fractale de certaines ondelettes. Bien que paraissant a priori très irrégulières, elles permettent cependant d'approximer exactement des polynômes dont le degré est inférieur au nombre de moments nuls tels que définis dans l'équation 3.26.

La figure 3.7 illustre cela : il s'agit du graphe de $\varphi(x) + 0.5\varphi(x-1)$. Nous pouvons observer l'existence d'un tronçon rectiligne entre les abscisses 2 et 3.

## 3.10 Applications

### 3.10.1 Compression du signal

Dans le domaine du traitement de l'image, un des problèmes fondamentaux est la place occupée par une image : celle-ci est en effet une collection de pixels, caractérisés

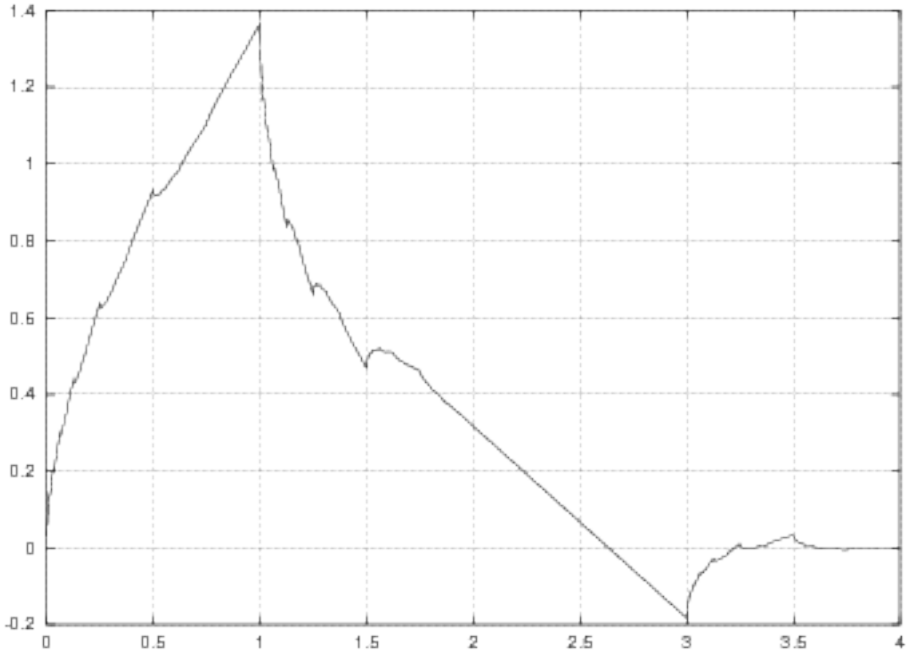



FIG. 3.7: $\varphi(x) + 0.5\varphi(x-1)$

par un nombre de niveaux de gris, ou par des coordonnées dans un système chromatique (RGB, ...)

La compression sans perte d'une image consiste à essayer de prédire un point par rapport à ses voisins, et à encoder la différence entre la prédiction et la valeur réelle. Si le prédicteur fonctionne bien, cette différence sera le plus souvent petite et pourra être encodée sur un nombre restreint de bits.

Des rapports de compression plus élevés sont atteignables si on accepte de perdre un peu d'information. Le principe est de décomposer l'image dans une base (transformée en cosinus, ondelettes, ...) et de mettre à zéro les coefficients inférieurs à un seuil prédéterminé. À la reconstruction, les coefficients manquant feront apparaître un « bruit ». Néanmoins, si ce bruit n'est pas *perceptible* (parce que à haute fréquence par exemple), la qualité du signal recomposé sera jugée équivalente à celle du signal original.

### 3.10.2 Filtrage adaptatif

Les ondelettes se sont révélées particulièrement utiles pour analyser des signaux contaminés par du bruit, mais susceptibles de présenter des variations locales très importantes. Un filtrage classique de type passe-bas éliminera le bruit se trouvant en dehors de la bande passante du signal, mais atténuera significativement les impulsions.

Donoho [65] a développé une méthode de filtrage adaptatif basée sur la transformée en ondelettes. Sa méthode consiste à évaluer la variance du bruit à chaque niveau de détail, et à annuler les coefficients de norme inférieure à un seuil dépendant de cette variance. Les coefficients conservés sont ceux qui ont le plus de chance d'être corrélés avec le signal d'entrée. Dans les zones de variations locales importantes, *tous* les coefficients prendront une valeur élevée et seront conservés : en conséquence, le signal reconstruit présentera un bruit plus faible, mais les impulsions présentes seront conservées correctement. Cette méthode s'est révélée particulièrement bien adaptée dans le cas de signaux contenant de nombreux pics, comme par exemple une courbe de spectrophotométrie.

Chapitre 4

# Monitoring cardiorespiratoire par la technique de la manchette

## 4.1 Contexte

Rappelons brièvement le contexte : il s'agit d'évaluer par voie indirecte le rythme respiratoire ainsi que ses accidents. À partir du signal de pression dans une manchette de tension artérielle, nous allons montrer qu'il est possible d'évaluer les rythmes respiratoires et cardiaques.

La figure 4.1 représente les signaux obtenus au cours d'une expérience avec le dispositif de mesure tel que défini dans le chapitre 2, section 2.6. Dans ce cas, le patient a simplement simulé un certain nombres d'états respiratoires spécifiques, entrecoupés de phases de respiration normale.

Le graphe contient trois signaux : celui du bas (AF) représente une image indirecte du flux d'air, via la température d'une thermistance située devant la bouche. Le second signal (OP) représente la pression dans une sonde œsophagienne. Cette sonde a été positionnée de façon à maximiser le couplage avec le signal cardiaque. Le troisième signal est celui dans la manchette.

Le signal AF donne la référence de rythme respiratoire; le signal OP contient principalement la signature du rythme cardiaque, ainsi que des composantes liées au rythme respiratoire. Le signal dans la manchette, CP, est le signal à analyser.

Dans le segment illustré, nous avons successivement les événements suivants :

- jusqu'à trente secondes, la respiration est normale;
- entre trente et cinquante secondes intervient un épisode de ronflement; on remarquera le changement de morphologie au niveau de la pression oesophagienne;

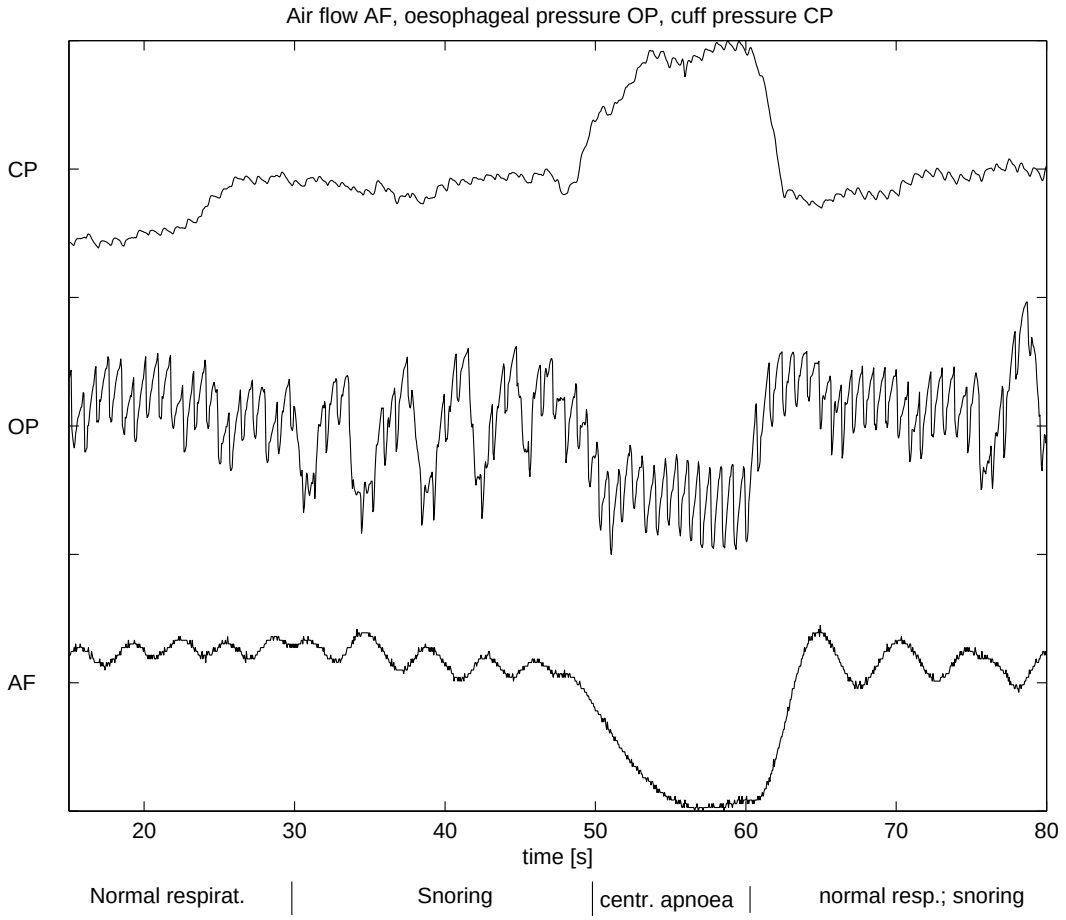



FIG. 4.1: Signaux bruts

– entre cinquante et soixante secondes se produit un arrêt de la respiration, de façon à simuler une apnée centrale;
– après soixante secondes, nous avons trois cycles respiratoires normaux, suivis d'un nouvel épisode de ronflement.

Par commodité, les trois signaux ont été normalisés de façon à présenter une amplitude crête-à-crête d'une unité, et décalés.

## 4.2 Analyse par ondelettes

Dans une première phase des travaux, nous avons cherché à utiliser les différents niveaux de détail comme autant de filtres passe-bande.

Souhaitant travailler dans des conditions de type temps réel, nous avons rencontré un problème : chaque niveau de détail correspond à une décimation temporelle par

rapport au niveau précédent. Le tableau suivant indique de manière approximative les bandes de fréquence des divers filtres :

| indice | approximation | détail |
|---|---|---|
| 0 | 0 – 10 Hz | |
| -1 | 0 – 5 Hz | 5 Hz – 10 Hz |
| -2 | 0 – 2.5 Hz | 2.5 Hz – 5 Hz |
| -3 | 0 – 1.25 Hz | 1.25 Hz – 2.5 Hz |
| -4 | 0 – 0.625 Hz | 0.625 Hz – 1.25 Hz |

Chaque filtre couvre une octave. Néanmoins, étant donné que la réponse temporelle est finie, la réponse fréquentielle n'est pas de type créneau : il y a un lobe principal et des lobes secondaires.

En ce qui concerne les notations utilisées dans les travaux publiés, nous avions donné l'indice 0 au niveau initial, 1 après le premier filtrage, et ainsi de suite. Cet indice reflète en fait le nombre de filtrages effectués. En toute rigueur, on considère dans la littérature sur les ondelettes qu'il y a $2^j$ points au niveau $j$. Si nous considérons une fenêtre d'observation de 256 points, le signal original est au niveau 8, et nous avons 8 approximations, allant du niveau 7 au niveau 0 qui correspond à la valeur moyenne du signal. La notation $-1$ que nous proposons ici est relative au niveau de départ et indique que l'on se situe un niveau plus bas, après application d'une décimation par deux.

Le problème lié à la décimation est l'introduction d'un délai. Considérons un filtre de délai $\tau$. Au premier niveau, il faut $\tau$ points en entrée pour obtenir en sortie le premier élément de la réponse. Au niveau suivant, il faut attendre $2\tau$ éléments supplémentaires, puis $4\tau$ et ainsi de suite, soit $(2^j - 1)\tau$ en tout après $j$ niveaux de filtrage.

Dans la première solution que nous allons présenter, nous pensions que le délai au niveau de la reconstruction était tout simplement doublé par rapport à cette valeur, puisqu'il fallait effectuer une nouvelle fois ce processus en sens inverse. Dès lors, nous avions essayé de réduire ce délai par un artifice.

## 4.3 Première solution : travail sans décimation

Après nos premiers essais, nous avons conclu qu'il était nécessaire de disposer du signal filtré avec la même résolution temporelle que le signal original, pour avoir la possibilité d'effectuer une analyse sur un nombre suffisant de points. Nous reviendrons plus en détail sur ce sujet, lorsque nous parlerons de la résolution sur la mesure du rythme cardiaque.

Nous avons procédé comme suit : considérant que les points pairs et impairs à la sortie du filtre contiennent la même information fréquentielle, nous avons pris comme

approximation du signal reconstruit le signal sans décimation. De cette façon la résolution temporelle reste constante.

Pour maintenir les propriétés de couverture par octave, nous avons adopté le schéma de filtrage suivant : à la sortie des filtres, les points sont séparés en deux groupes suivant que leur index soit pair ou impair. Sur chacun de ces vecteurs, correspondant formellement à une décimation par deux de la sortie du filtre, on ré-applique les filtres passe-bas et passe-haut, et on répète la même séparation.

Au premier niveau de filtrage, on obtient deux vecteurs; au second niveau, quatre; et ainsi de suite. Chaque groupe de vecteurs est ensuite recombiné pour donner une approximation du détail ou de l'approximation au niveau de décomposition correspondant, possédant la même résolution que le signal initial.

En pratique, cette opération est pratiquée « en place », et s'effectue via des manipulations adéquates au niveau des indices. Numériquement, la complexité est plus élevée, puisque à chaque étape, nous appliquons les deux filtres à des signaux de longueur identique à celle du signal de départ.

L'algorithme a été implémenté dans le cadre de la boîte à outil « WaveLab ». Il s'agit d'une collection de routines et d'utilitaires composant un laboratoire sur les ondelettes : outre les opérations de transformées directe et inverse, on y trouve également une série de démonstrations sur les propriétés fondamentales d'une série d'ondelettes, ainsi que des exemples d'applications.

Nous avons créé un algorithme, `FWT_NP` (Forward Wavelet Transform, Not Periodized). L'algorithme standard, `FWT_PO`, permet en effet de tenir compte des conditions aux limites, par exemple en périodisant artificiellement le signal étudié. Nous avons pris le parti de considérer que, travaillant dans un cadre temps réel, notre système devait pouvoir fonctionner en une seule passe, sans avoir à utiliser au début du traitement des points se trouvant à la fin du bloc en cours de traitement.

Pour illustrer la différence entre notre traitement et le traitement avec décimation et reconstruction, les graphiques 4.2 et 4.3 montrent un exemple de réponse impulsionnelle correspondant à chacune des approches.

Ces graphiques ont été obtenu en prenant un signal impulsionnel sur 256 points, valant partout zéro sauf au $128^{\text{ième}}$ point, où la valeur est unitaire.

La différence fondamentale entre les deux approches réside au niveau de la reconstruction d'un signal :

- avec décimation, il est possible de reconstruire chaque niveau de détail à la résolution initiale, puis d'*additionner* chacune des traces : le résultat sera le signal de départ. Les propriétés de linéarités de la reconstruction font que la somme des reconstructions partielles correspond bien à une reconstruction globale.
- avec l'approximation sans décimation, cette opération n'est a priori pas possible, puisque les points intermédiaires ne correspondent qu'à une approximation de la reconstruction.

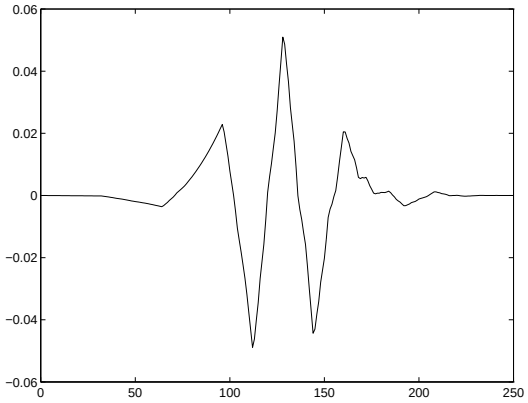



FIG. 4.2: Filtrage par ondelette — avec décimation : réponse impulsionnelle ($5^{\text{ième}}$ détail – Daubechies ordre 6 )

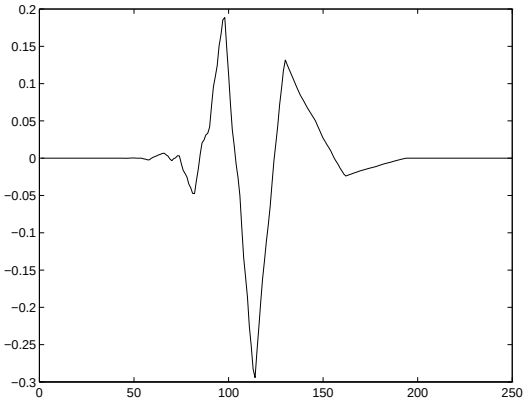



FIG. 4.3: Filtrage par ondelette — sans décimation : réponse impulsionnelle ($5^{\text{ième}}$ détail – Daubechies ordre 6)

### 4.3.1 Détermination du rythme cardiaque

En examinant les niveaux de détail $-3$ et $-4$ de la pression dans la manchette CP de la figure 4.1, il est apparu un signal composé d'impulsions comme illustré dans la figure 4.4. La période de ce signal, prise comme la distance moyenne entre les impulsions, laisse à penser qu'il s'agit bien d'une trace à la fréquence cardiaque.

Comme on peut s'en apercevoir, ce signal présente une composante principale, des harmoniques dont on perçoit la présence au sommet des impulsions, ainsi que des artefacts. L'amplitude crête-à-crête de ce signal varie de 10 à 20 centièmes de millimètre Hg. Parfois, cette amplitude peut n'être que de 5 centièmes de millimètres Hg, ce qui correspond à environ 50 codes du convertisseur.

Afin de déterminer plus précisément la période de base de ce signal, nous en avons d'abord pris la figure d'auto-corrélation, comme illustré dans la figure 4.5.

Techniquement, il s'agit de la fonction d'auto-corrélation *non-biaisée*: l'élément d'indice $k$ résulte de la moyenne du produit de deux séquences de longueur $N-k$, $N$ étant la longueur de la séquence observée, divisé par $N-k$.

#### Détermination de la résolution

Avant de poursuivre l'exposé, nous allons motiver plus en détail les choix qui ont été opérés au niveau du traitement numérique.

Au niveau médical, les praticiens attendent une évaluation du rythme cardiaque avec une résolution de 1 battement par minute, ce qui correspond à environ à 1.7 mHz.

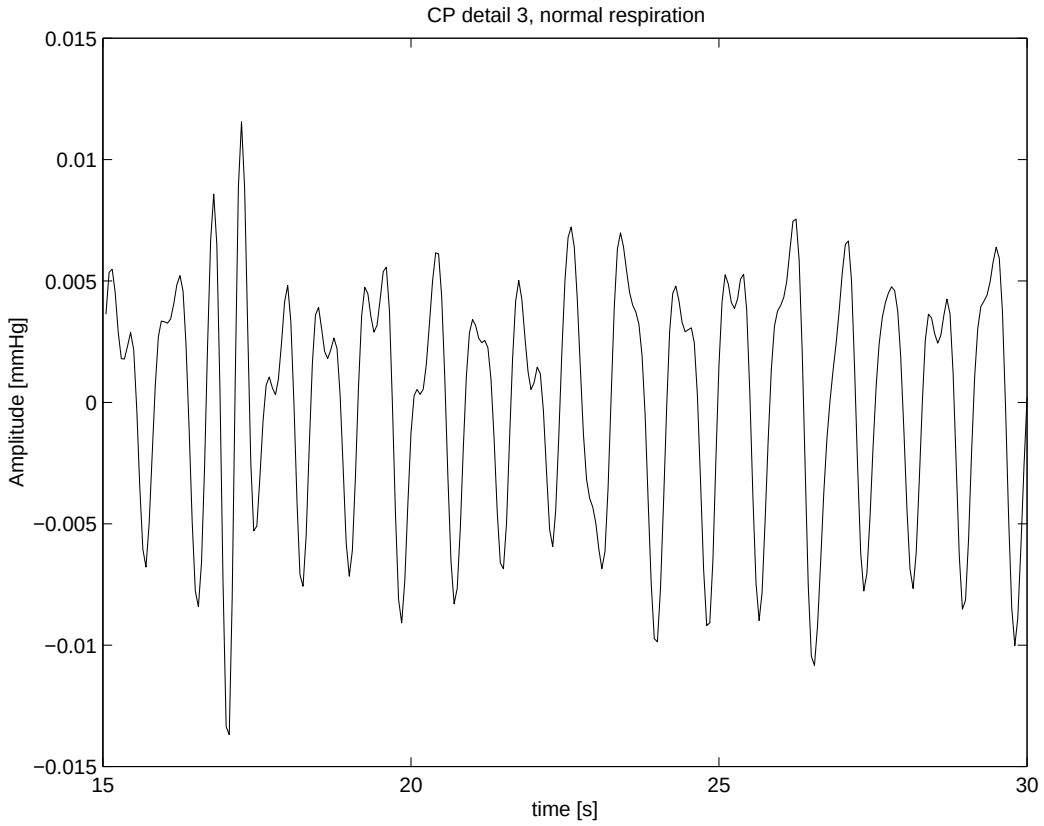



FIG. 4.4: signal de pression dans la manchette, après filtrage passe-bas centré sur le rythme cardiaque.

En travaillant directement via l'analyse de Fourier, nous obtenons une résolution *fréquentielle* qui est inversement proportionnelle à la largeur de la fenêtre d'observation. En particulier, pour suivre l'évolution du rythme cardiaque, nous avons utilisé des fenêtres d'observation de 5 secondes, ce qui donne une résolution de 0.2 Hz : cette quantité représente 20% de la valeur de base du rythme normal à l'état de veille (60 – 70 bpm).

Si nous essayons de déterminer directement la période, nous avons une incertitude de 50 ms due à la période d'échantillonnage. Cette quantité équivaut à 5% de la période typique du signal cardiaque. Nous nous sommes en effet limité à une fréquence d'acquisition de 20 Hz, ce qui est amplement suffisant pour observer les signaux respiratoires et cardiaques. Augmenter cette fréquence reviendrait à augmenter le nombre de points à stocker et à traiter.

Nous allons mettre en œuvre une technique qui porte parfois le nom de super-résolution, consistant à obtenir une estimée de la période avec une résolution supérieure à celle que l'on obtiendrait par la méthode de Fourier. L'approche classique repose sur l'identification des pôles d'une relation auto-régressive, en calculant les coefficients puis les racines d'un polynôme en $\mathcal{Z}$.

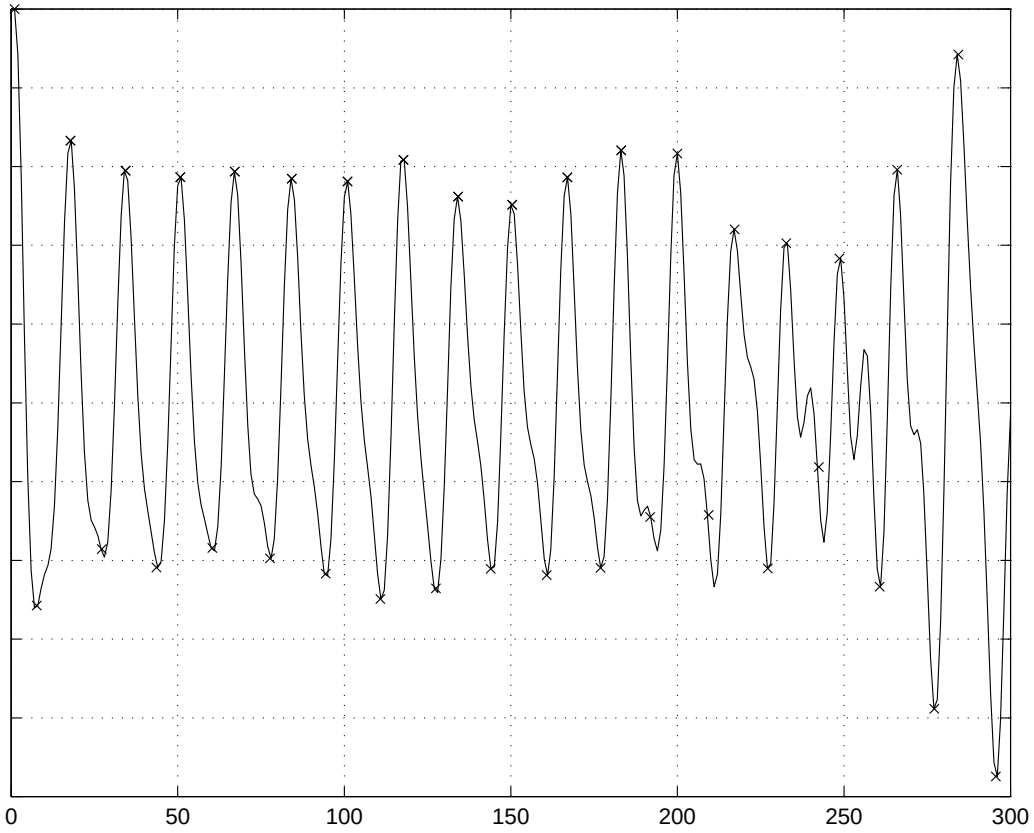



FIG. 4.5: Fonction d'auto-corrélation de la figure 4.4

**Exposé du problème**

Le problème de base est d'extraire « la » période de base d'un signal observé sur un nombre restreint de périodes. D'une manière générale, un certain nombre de difficultés peuvent se manifester :

- bruit et artefacts : tout mouvement, toute vibration transmise à la manchette donnera lieu à une trace. Dans les situations d'apnée, il se produit également des variations de pression très rapides, comme nous le verrons par la suite;
- présence d'harmoniques : de par la présence des lobes secondaires, certaines harmoniques peuvent ou non contaminer le signal. Ces harmoniques donneront éventuellement lieu à un battement, qui se manifestera par une variation lente de l'amplitude, voire même par une variation de la distance entre des passages par zéro consécutifs.
- problème de résolution temporelle, comme expliqué au point précédent.

Éventuellement, une information complémentaire peut être recherchée dans la variation d'intervalle entre deux battements cardiaques : nous avons observé que ce genre de mesure n'est pas possible sur nos signaux, à cause des artefacts.

Nous avons estimé la période via un algorithme que nous avons mis au point. L'idée de base est d'augmenter la résolution sur la position des extrema de la figure d'auto-corrélation, et de faire clairement la part des choses entre les zones non-bruitées et les zones contaminées par le bruit.

Rendre cet algorithme robuste n'a pas été une tâche aisée : en effet, nous avons identifié un certain nombre de situations pathologiques susceptibles de produire des résultats aberrants. Nous avons procédé par raffinements successifs, et nous pensons être arrivé à une situation où les causes d'erreurs sont correctement localisées et traitées de manière adéquate.

### 4.3.2 Algorithme `extrema.m`

Cet algorithme étant assez long et complexe, nous avons préféré en expliquer schématiquement les différentes étapes plutôt que d'inclure le code source :

1. localisation rapide et sommaire des changements de signe : ils répondent au critère $x_n * x_{n+1} < 0$. On n'effectue pas d'extrapolation sur leur position : en effet, la zone où le signal principal est de valeur nulle est également la zone où le bruit est le plus susceptible de se manifester;
2. détection des zones où le signal passe à proximité de zéro : le but est d'éliminer les tronçons où la fonction d'auto-corrélation serait tangente ou quasi-tangente à l'axe horizontal;
3. estimation grossière de la demi-période : on prend la moitié de la valeur médiane de la distance entre les extrema consécutifs de même signe;
4. estimation grossière de l'amplitude, en prenant la déviation standard de la valeur absolue des 5 premiers échantillons, c.-à-d. dans la zone où se trouve le premier pic de la fonction;
5. localisation de l'extremum du premier segment; si sa valeur absolue est au moins égale à 20 % de l'amplitude déterminée au point précédent, elle est retenue comme amplitude du maximum (resp. minimum) des segments;
6. première passe de localisation : repérage de l'extremum de chaque segment compris entre deux passages par zéro consécutifs. L'indice de cet extremum est retenu si son amplitude vaut au moins 20 % de l'amplitude déterminée précédemment. Le but de cette boucle est de ne pas retenir les extrema de petite amplitude, qui sont dus typiquement aux harmoniques et se marquent par une oscillation plus rapide autour de l'axe horizontal.
7. calcul de la différence entre les maxima (resp. minima) consécutifs, on retient une période $dmax$ (resp. $dmin$) valant le médian de ces différences.
8. détection de la présence d'harmoniques : si un des critères
   - $dmax \simeq 2dmin$
   - $dmin \simeq 2dmax$

est atteint, c'est le signe d'une contamination harmonique importante, on retient alors comme estimée le maximum entre *dmin* et *dmax*, qui donne la fréquence la plus basse;

9. Pour le premier segment, détection du premier extremum (normalement situé en première position), et de la pente;
10. Boucle principale : pour chaque segment,
    (a) repérage de l'extremum et des zéros qui l'encadrent;
    (b) séparation en deux sous-segments, respectivement à gauche et à droite de l'extremum localisé;
    (c) détermination de cas pathologiques :
        – un extremum situé à moins d'un quart de période d'un passage par zéro;
        – un extremum suivi d'un passage par zéro puis d'un extremum de même signe et d'amplitude supérieure, le tout dans un intervalle plus petit qu'une demi-période;
        – un extremum suivi d'un changement de concavité, puis d'un autre extremum de même signe, le tout également situé dans un intervalle de moins d'une demi-période.

        Si un de ces cas se produit, l'extremum courant est abandonné au profit de l'extremum de même signe suivant;
    (d) construction d'un intervalle symétrique encadrant l'extremum local, dont la concavité est correcte;
    (e) recherche de la parabole de régression passant par les points retenus. Lorsque les coefficients de cette parabole sont calculés, la position de son sommet est ensuite déterminée, et une vérification de la cohérence est obtenue en comparant la concavité de la parabole avec celle de l'extremum. Si ce test réussit, on conserve l'abscisse de la position du sommet, cette abscisse étant un nombre réel et non plus un nombre entier;
    (f) s'il n'est pas possible de calculer cette parabole, la position (entière) de l'extremum est conservée;
    (g) la distance, interpolée ou non, entre l'extremum actuel et son prédécesseur de même type est calculée;
11. le résultat final s'obtient en prenant le médian des distances entre maxima consécutifs, ainsi que le médian des distances entre minima consécutifs. Les avantages du médian est d'une part que le résultat fait partie de l'échantillonnage, et d'autre part que cette méthode est peu sensible aux éléments erronés ("outliers").

### 4.3.3 Résultats

Le résultat de cet algorithme, appliqué aux données de la figure 4.4 est en fait visible dans la figure 4.5 : les positions des extrema interpolés sont repérées par des

petites croix. Leur abcisse est donnée dans la table 4.1, qui compare les estimations de distance entre extrema basées sur les indices entiers ainsi que celle basée sur notre méthode d'interpolation.

Cet exemple a été choisi parce que nous pensons qu'il illustre un certain nombre de problèmes typiques :

- on remarque que les deux minima situés de part et d'autre de l'indice 200 se trouvent approximativement au milieu des passages par zéro, alors que les positions relevées directement sur la courbe brute sont nettement asymétriques; on remarque également que les petites oscillations près de l'indice 250 n'ont pas été retenues dans le calcul des extrema interpolés.
- on note que les distances évaluées sur base des positions des extrema de la courbe oscillent aléatoirement entre 16 et 17 (en rejetant la première distance entre minima, 21). Avec notre méthode d'interpolation, nous obtenons des valeurs médianes respectivement de 16.67 pour les minima et de 16.56 pour les maxima, d'où nous déduisons comme valeur moyenne de l'estimation 16.62, ce qui correspond à un rythme cardiaque de 72.2 bpm.

Bien que dans ce cas la moyenne des distances entières (16.5) ne soit pas très différente de notre évaluation, nous avons pu observer des comportements assez différents, en particulier en présence d'harmoniques ou d'artefacts. Au niveau des déviations standard, elles sont de 0.51 dans le cas entier, et de 0.3 avec notre interpolation.

### 4.3.4 Discussion

La fonction d'auto-corrélation d'un signal analogique échantillonné valant l'échantillonnage de la fonction d'auto-corrélation de ce signal, les extrema interpolés correspondent donc aux positions limites que l'on aurait relevées pour une résolution temporelle infinie. Nous avons admis implicitement que le voisinage sur lequel on calcule la parabole de régression est suffisamment petit pour qu'un développement en série limité à l'ordre deux donne le même extremum que celui de la fonction d'auto-corrélation réelle. Le traitement statistique consistant à prendre la valeur médiane permet de réduire l'influence des divers parasites.

Le gain de cette méthode est de pouvoir déterminer la position des extrema avec une résolution meilleure qu'une unité (de période d'échantillonnage), ceci se traduit par une augmentation concomitante de la résolution sur la période évaluée du rythme cardiaque.

L'inconvénient en est la lourdeur de l'algorithme : en effet, chaque tronçon de courbe doit être méticuleusement analysé afin d'en détecter et d'en traiter les anomalies éventuelles.

| index min | distance | min interp. | dist. interp. | index max | distance | max interp. | dist. interp. |
|---|---|---|---|---|---|---|---|
| 7 | | 7.69 | 19.49 | 18 | | 17.77 | 16.56 |
| | 21 | | | | 16 | | |
| 28 | | 27.18 | 16.43 | 34 | | 34.34 | 16.45 |
| | 16 | | | | 17 | | |
| 44 | | 43.61 | 16.78 | 51 | | 50.79 | 16.32 |
| | 17 | | | | 16 | | |
| 61 | | 60.39 | 17.28 | 67 | | 67.11 | 17.10 |
| | 17 | | | | 17 | | |
| 78 | | 77.67 | 16.75 | 84 | | 84.21 | 16.79 |
| | 17 | | | | 17 | | |
| 95 | | 94.42 | 16.38 | 101 | | 100.99 | 16.75 |
| | 16 | | | | 17 | | |
| 111 | | 110.80 | 16.67 | 118 | | 117.75 | 16.42 |
| | 17 | | | | 16 | | |
| 128 | | 127.48 | 16.50 | 134 | | 134.17 | 16.20 |
| | 16 | | | | 16 | | |
| 144 | | 143.98 | 16.77 | 150 | | 150.37 | 16.60 |
| | 17 | | | | 17 | | |
| 161 | | 160.74 | 16.16 | 167 | | 166.97 | 16.18 |
| | 16 | | | | 16 | | |
| 177 | | 176.90 | | 183 | | 183.15 | |

TAB. 4.1: Position des minima et des maxima, et distances de la fig. 4.5

### 4.3.5 Vérification de la méthode

Les médecins nous ont proposé de comparer la fréquence dans le signal provenant de la sonde œsophagienne à celui trouvé dans la manchette : en effet, les mesures ont été réalisées avec la sonde dans le tiers supérieur de l'œsophage, là où le couplage avec le signal cardiaque est maximal. Au cours des examens type polysomnographie, on place plutôt cette sonde dans le tiers inférieur, afin de favoriser la mesure du rythme respiratoire.

Pour les courbes de la figure 4.1, nous avons obtenu, au $4^{\text{ième}}$ niveau de détail, 76.3 bpm pour CP et 76.8 bpm pour OP, en utilisant les mêmes filtres ainsi que notre algorithme. L'écart entre ces deux évaluations est de 1/2 bpm, alors que les deux courbes contiennent des artefacts. Sur les expériences que nous avons menées, l'écart entre ces deux évaluations est resté de manière générale inférieur à 1 bpm, ce qui nous conforte dans l'idée que cette méthode de la manchette permet d'accéder effectivement au rythme cardiaque.

### 4.3.6 Détection des apnées obstructives

Cette apnée se produit suite à un affaissement des voies respiratoires supérieures, d'où il résulte une perte de charge très importante, voire un blocage du flux d'air.

La difficulté spécifique au niveau du monitoring est que, contrairement à l'apnée centrale, cette apnée s'accompagne d'un mouvement de la cage thoracique. Le patient a en effet tendance à inspirer très vigoureusement, mais en pure perte. Les méthodes classiques basées sur l'enregistrement des mouvements respiratoires ne feront donc pas de différence entre cet épisode apnéique et une situation de respiration normale.

En polysomnographie, la détection de l'apnée obstructive requiert donc au minimum :

- l'enregistrement du flux d'air;
- la détection des mouvements de la cage thoracique;
- éventuellement, la mesure de la résistance à l'inspiration, définie comme la différence de pression entre celle régnant dans les voies respiratoires et l'atmosphère, rapportée au flux d'air qui en résulte.

Suite à une série d'observations, nous avons repéré une signature typique au niveau de nos signaux. En effet, lorsqu'on évalue le rythme cardiaque via les détails $-3$ et $-4$, on obtient deux estimations qui tendent à être très proches l'une de l'autre, en régime de respiration normale.

Par contre, durant les épisodes obstructifs, il se produit des mouvements respiratoires à un rythme plus rapide, ayant un effet assez prononcé au niveau de la pression dans la manchette. De ce fait, le signal au niveau de détail $-4$ comporte à la fois

des composantes du rythme cardiaque ainsi que du rythme respiratoire. Par contre, le niveau de détail $-3$ n'est quasiment pas contaminé.

Cette observation est illustrée dans la table 4.2. Durant cette séquence d'une minute, le patient a simulé une apnée obstructive allant de 20 à 50 secondes.

| time [sec] | detail -3 [bpm] | detail -4 [bpm] |
|---|---|---|
| 00 | 87.7 | 87.6 |
| 05 | 85.7 | 85.6 |
| 10 | 84.5 | 83.8 |
| 15 | 84.5 | 84.2 |
| 20 | 87.3 | 40.4 |
| 25 | 93.7 | 63.3 |
| 30 | 95.1 | 22.7 |
| 35 | 93.6 | 20.3 |
| 40 | 93.7 | 21.7 |
| 45 | 87.8 | 17.6 |
| 50 | 86.5 | 77.7 |
| 55 | 89.5 | 92.0 |
| 60 | 93.9 | 93.7 |

Tab. 4.2: Exemple de détection des apnées obstructives.

Nous pouvons constater qu'avant l'épisode, les deux estimations sont assez similaires, à une unité près. Durant l'apnée, au niveau $-3$, le rythme cardiaque présente une accélération, ce qui est un réflexe typique causé par la baisse de la pression partielle en oxygène dans le sang. Par contre, au niveau $-4$, on assiste à une diminution très importante, et la valeur la plus basses (17.6 bpm) serait franchement alarmante s'il s'agissait réellement d'un rythme cardiaque ! Nous déduisons de ce désaccord entre les deux estimées que le signal au niveau $-4$ contient une autre contribution, provoquée par l'apnée. Nous pouvons également constater qu'à la fin de l'épisode, les deux estimations reconvergent vers des valeurs similaires.

### 4.3.7 Conclusion

Les résultats obtenus par cette méthode de filtrage sans décimation ont montré son efficacité en ce qui concerne l'évaluation du rythme cardiaque. Par contre, nous n'avons pu obtenir de résultats réellement convaincants au niveau respiratoire, si ce n'est par voie indirecte au cours des épisodes d'apnées obstructives. Dans certaines courbes, il a été possible d'observer visuellement un signal similaire au signal AF, en particulier au niveau des détails $-5$ et $-6$, mais ce genre d'observation n'a pu être généralisée. Nous avons donc repris le problème, mais cette fois ci avec une méthode plus rigoureuse basée sur une reconstruction correcte et non plus une approximation.

## 4.4 Évaluation du rythme respiratoire : travail par reconstruction

### 4.4.1 Résultats précédents

Nos premières recherches au niveau du rythme respiratoire se sont focalisées sur des signaux filtrés sans décimation. Cette approche a donné des résultats probants au niveau du rythme cardiaque, mais pas au niveau respiratoire; en particulier l'algorithme d'extraction de la période n'a que très rarement permis de mettre en évidence une fréquence dont la valeur était cohérente par rapport au rythme respiratoire dont la trace était visible dans le signal thermique lié au flux d'air.

Nous allons commencer par illustrer les résultats obtenus par cette approche, puis exposer une autre méthode qui semble nettement plus prometteuse.

Dans les figures 4.6 et 4.7, nous illustrons les signaux acquis ainsi que leur version filtrée : on trouve de haut en bas :

1. la pression à l'intérieur du tube;
2. sa température AF, image du flux d'air;
3. la pression dans la manchette CP;
4. le niveau de détail $-4$ de CP, contenant principalement le rythme cardiaque;
5. les niveaux de détail $-5$ ou $-6$ ainsi que $-7$ de CP, où on peut apercevoir des traces du rythme respiratoire.

Le premier épisode illustre une apnée centrale : on constate une nette diminution du signal au niveau des détails 6 et 7 de la pression dans la manchette (CP).

Le second épisode concerne une apnée obstructive : la signature dans ce cas est l'impulsion présente dans le détail 5, consécutive à la brusque variation de pression artérielle survenant au cours de l'apnée obstructive.

### 4.4.2 Mesure avec sonde œsophagienne

Dans cet exemple, nous allons reprendre les signaux illustrés dans la figure 4.1, que nous allons ensuite analyser à d'autres niveaux de détail, de façon à essayer de mettre en évidence la signature du rythme respiratoire.

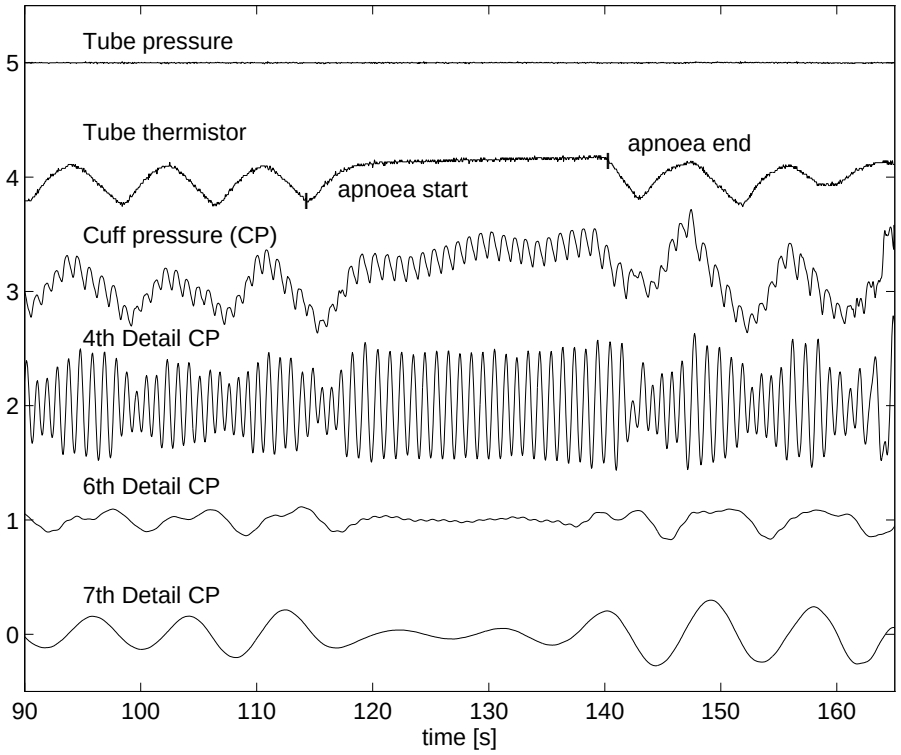



FIG. 4.6: Séquence comprenant une apnée centrale

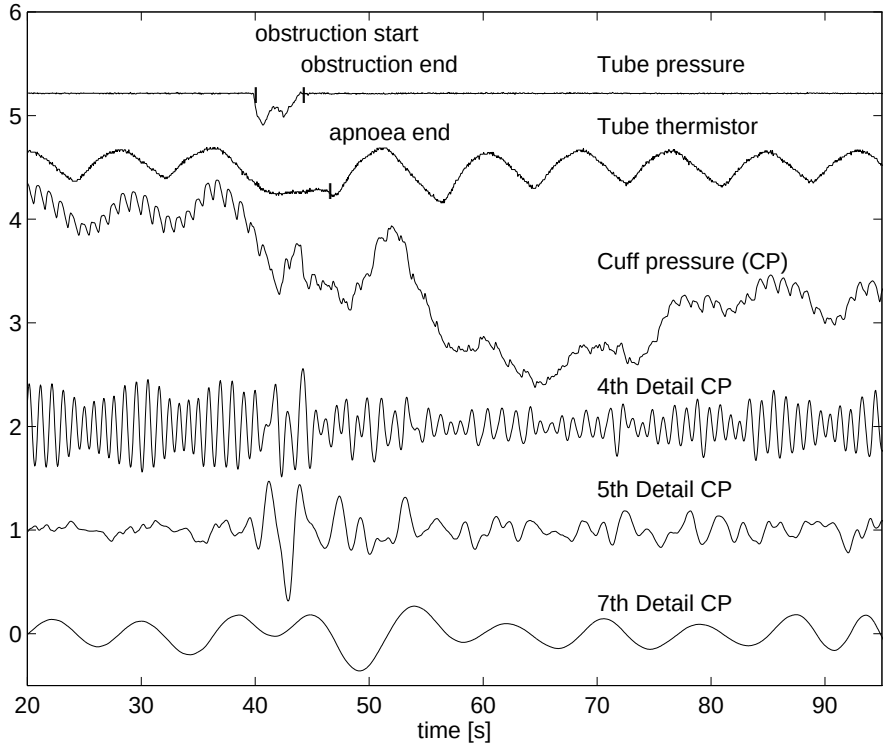



FIG. 4.7: Séquence avec une apnée obstructive

Pour rappel, cet épisode de 50 secondes se compose d'une phase de ronflements, d'une apnée centrale et d'une phase de reprise de respiration. Au niveau de la manchette, le seul signal directement visible était constitué d'oscillations de faible amplitude générées par les battements cardiaques. Le saut de niveau survenant entre 50 et 65 secondes a une valeur approximative de 1 mmHg. En ce qui concerne le signal respiratoire, aucune trace ne peut a priori être mise en évidence dans la pression de la manchette, tandis que ce signal est à peine discernable dans le signal de pression œsophagienne.

La figure 4.8 reprend ces mêmes signaux, mais au niveau de détail $-6$, sauf pour AF qui reste inchangé. Le rythme respiratoire, présent lors de la première phase, apparaît plus clairement après la fin de l'apnée, ce qui correspond à des respirations d'amplitude plus importante. Les impulsions qui encadrent l'épisode apnéïque sont générées par les flancs correspondant à la variation brusque de pression artérielle.

Si par contre nous nous intéressons à l'approximation basse fréquence, la variation de ligne de base est extrêmement claire dans le signal de la manchette, comme illustré dans la figure 4.9. Nous pouvons observer une similitude entre les variations de CP et celles de OP durant l'apnée, mais sans plus.

### 4.4.3 Conclusions

Si, visuellement, nous avons des *indices* de présence du signal respiratoire dans les différentes traces obtenues après filtrage passe-bande et passe-bas, il n'en reste pas moins que cette approche n'est pas satisfaisante pour plusieurs raisons :

- le rapport signal/bruit est en général faible;
- il est difficile de faire la part des choses entre une réponse réelle générée par un événement et des épisodes d'oscillations correspondant à la réponse impulsionnelle des filtres, et déclenchées par une variation brusque de pression, résultant d'une apnée ou d'un artefact.

## 4.5 Seconde solution : travail avec décimation

Le lecteur pourrait trouver étrange qu'après avoir développé une méthode d'approximation sans décimation et montré qu'elle permettait effectivement d'accéder au rythme cardiaque, nous en revenions à une méthode basée sur la décimation.

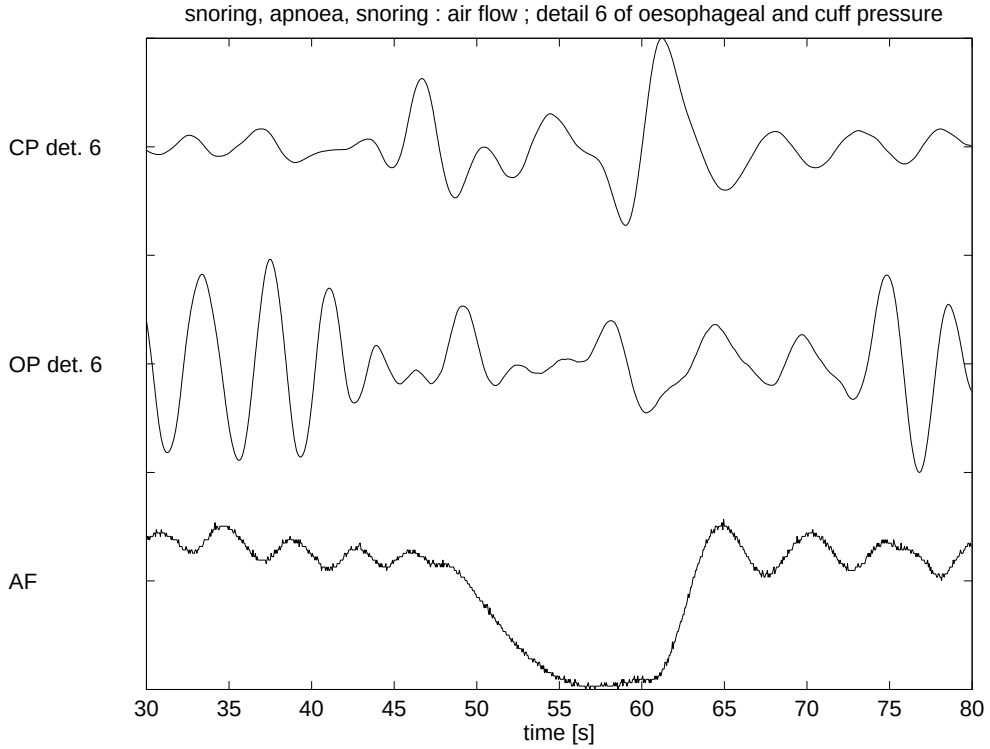



FIG. 4.8: Après filtrage autour du rythme respiratoire

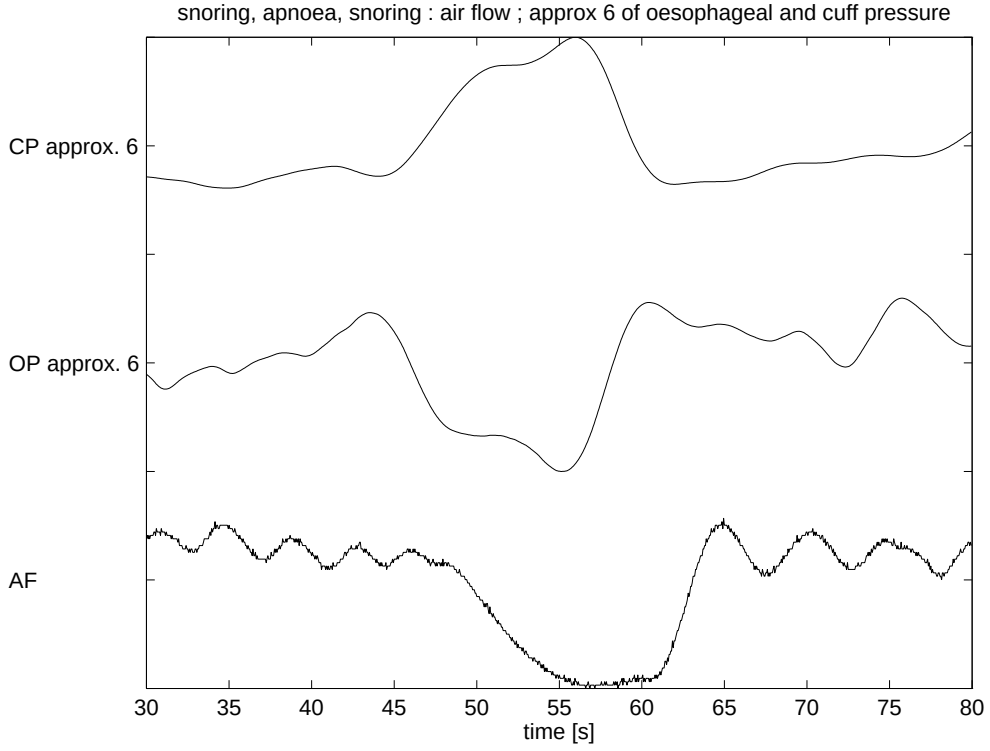
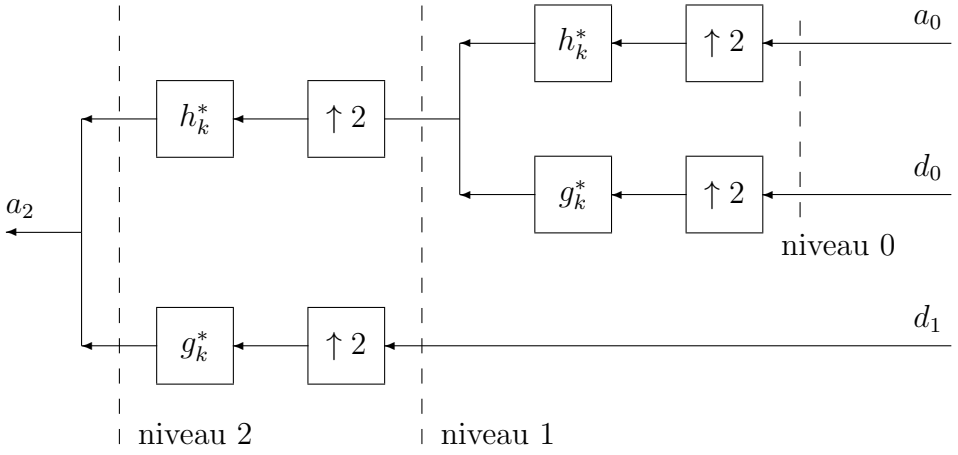



FIG. 4.9: Filtrage passe-bas, approximation de niveau $-6$

Nos motivations, lorsque nous avions développé la méthode sans décimation, était de pouvoir disposer d'une méthode de filtrage utilisable en temps réel, c.-à-d. dont la fréquence des échantillons en sortie était identique à celle des échantillons en entrée, et travaillant sans périodisation artificielle du signal. En effet, cette périodisation nécessite d'avoir complété l'acquisition d'une séquence d'échantillons considérée comme formant une suite périodique circulaire, c.-à-d. répondant à la propriété $x_{N+1} = x_1$.

En lisant des documents sur le filtrage par ondelettes, nous avons découvert qu'il est possible de faire fonctionner l'étage de reconstruction finale à la même fréquence que celle du signal d'entrée, via un artifice appelé "the splitting trick".

Considérons en effet la reconstruction :




Pour remonter d'un niveau, on commence par doubler la fréquence des signaux : un zéro est inséré entre deux points consécutifs. On applique ensuite les filtres de reconstruction $g_k^*$ et $h_k^*$, et on additionne les deux contributions : la fréquence de sortie est donc effectivement doublée. À la fin de la pyramide, la fréquence est identique à celle du signal original.

Considérons à présent la sortie des filtres de reconstruction intermédiaires. Lorsque n est pair, nous avons pour le filtre $h^*$ :

$$y(n) = h^*(0)\ a(n) + h^*(2)\ a(n-1) + \ldots \tag{4.1}$$

tandis que pour n impair il vient

$$y(n) = h^*(1)\ a(n) + h^*(3)\ a(n-1) + \ldots \tag{4.2}$$

Du fait de l'entrelacement par des zéros, nous voyons que les points pairs ou impairs résultent du filtrage de l'approximation par les coefficients de même parité au niveau du filtre. Tout se passe comme si nous avions *deux* filtres de reconstruction, $h_e^*$ et $h_o^*$, constitués respectivement des coefficients pairs et impairs.

Disposant de $a(n)$, $a(n-1)$, $\ldots a(n-k/2)$, nous pouvons donc fournir en une opération *deux* valeurs de la reconstruction. En résumé, pour deux échantillons avant décomposition, nous obtenons un échantillon de l'approximation ainsi qu'un échantillon de détail, que nous pouvons traiter puis utiliser pour reconstruire deux échantillons.

De ce fait, nous pouvons par exemple calculer plusieurs niveaux de détails, que nous reconstruisons ensuite à la fréquence initiale : nous obtiendrons ainsi autant de version passe-bande de notre signal, mais cette fois nous avons une propriété très importante : la *somme* de plusieurs détails *reconstruits* est égale à la *reconstruction* de la *somme* de plusieurs détails. À la limite, la somme de toutes les reconstructions de détails ainsi que de la reconstruction de l'approximation donne le signal de départ.

Cette façon de travailler permet en fait d'élargir la bande passante utile; ce qui permet par exemple de « capturer » plusieurs harmoniques d'un même signal, ou encore d'obtenir une meilleure reconstitution si la fréquence varie au cours du temps ou correspond à la zone d'intersection entre deux bandes de fréquence : sans cela, une trace du signal aurait été observée dans deux bandes de fréquence adjacentes.

### 4.5.1 Détection du rythme respiratoire

Considérons le graphique 4.10. Il correspond à une portion d'un enregistrement de plus longue durée (15 minutes), réalisée sur un patient en position étendue, qui a simulé un certain nombre de phases de sommeil.

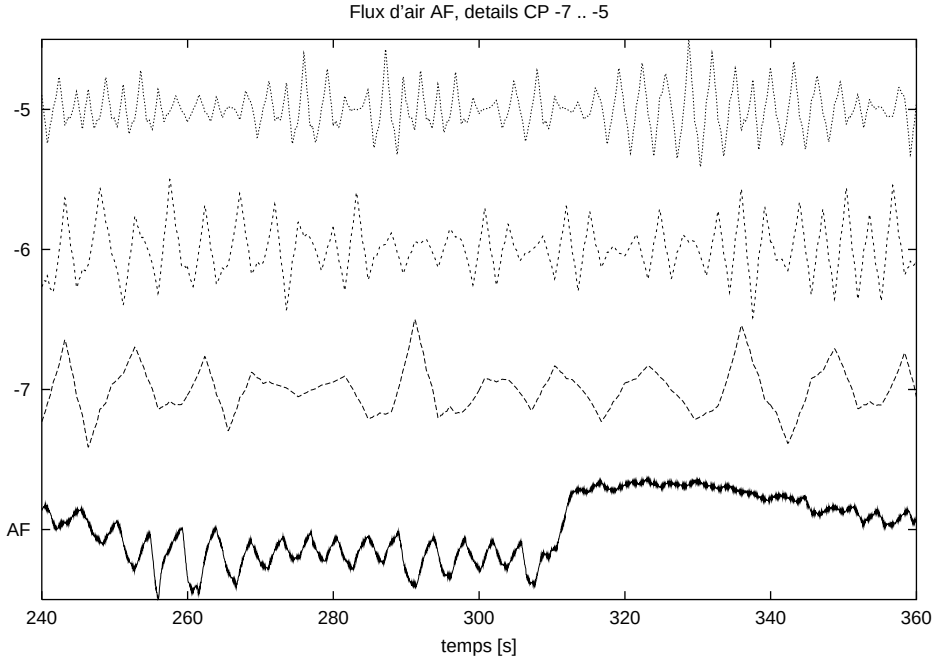



FIG. 4.10: respiration calme, traces distinctes

Au niveau des conditions d'acquisition, nous avons

- fréquence d'acquisition 20 Hz;
- pression de gonflage 10 mmHg;
- gain de l'amplificateur différentiel : 1000;
- résolution sur la mesure de pression : 1 $10^{-3}$ mmHg.

L'épisode illustré démarre 4 minutes après le début de l'enregistrement et dure deux minutes. Il correspond à une phase de respiration calme, avec un soupir vers 310 secondes. Ce soupir a eu pour effet de saturer le capteur thermique, masquant durant quelques instants le signal respiratoire au niveau AF, représenté en bas du graphe. Les trois traces supérieures correspondent à la pression dans la manchette, au niveau de détails $-7$ à $-5$, obtenus par reconstruction avec décimation. Pris individuellement, le niveau $-6$ présente une vague ressemblance avec le signal respiratoire, tandis que les deux autres sont à des fréquences différentes.

Pour donner une idée, les amplitudes des traces, en terme de code du convertisseur, sont :

- niveau -7 : déviation standard 5, crête-à-crête environ 25;
- niveau -6 : déviation standard 7, crête-à-crête environ 25;
- niveau -5 : déviation standard 4, crête-à-crête environ 30;

En termes de pression, les déviations standard sont de 4, 7, et 5 $10^{-6}$ mmHg ou encore 0.7, 0.9, et 0.55 Pa.

Ces résultats permettent également de justifier le recours à la haute résolution : en effet, le signal utile, en terme de code du convertisseur, représente à peine quelques unités. Si la résolution avait été moindre, il n'aurait pu être séparable du bruit de quantification.

Une fois combinés, la situation s'est nettement améliorée, comme on peut s'en rendre compte dans la figure 4.11.

Dans ce cas, les informations fréquentielles sont mieux corrélées. À titre d'exemple, entre les abscisses 280 et 300 secondes, nous comptons 4.5 cycles respiratoires complet au niveau AF, ce qui correspond à un rythme d'environ 0.225 Hz. Notre algorithme `extrema.m`, qui, rappelons-le, a été optimisé dans un cadre légèrement différent, nous donne comme estimée de la fréquence 0.226 Hz, ce qui correspond à 0.5 % de notre estimation. La période correspondante est de 4.42 secondes.

En ce qui concerne l'amplitude, la déviation standard du signal combiné vaut à présent environ 10 codes, et la valeur crête-à-crête varie entre 20 et 35 codes.

Nous voyons que nous pouvons dans ce cas reconstruire un signal sur 3 octaves, regroupant 3 bandes de détail, ce qui nous a donné une reconstruction plus fidèle. En

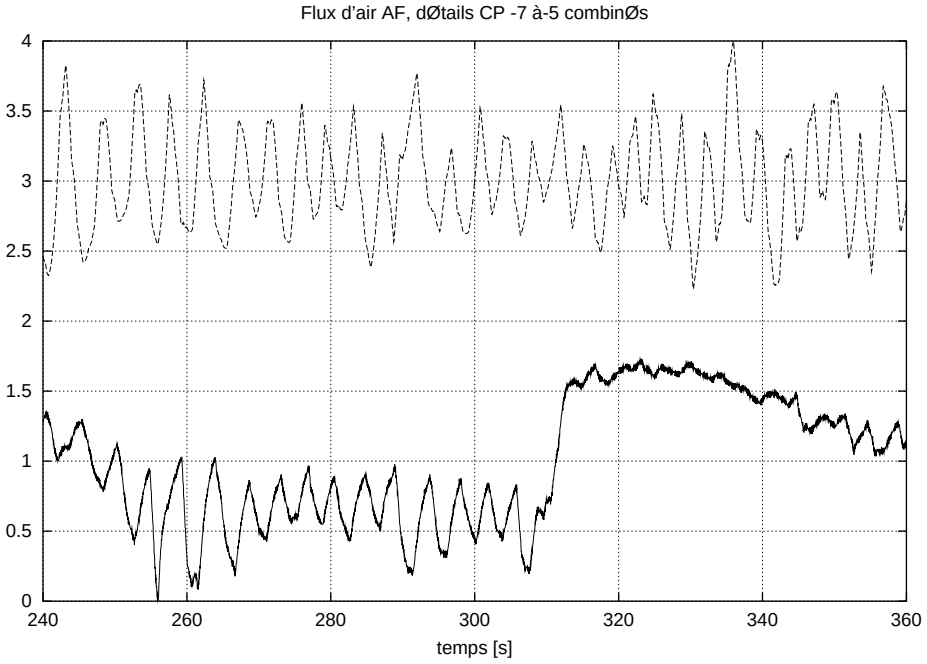



FIG. 4.11: respiration calme, traces combinées

effet, les trois bandes contiennent des traces générées à la base par un même phénomène : on peut donc les combiner, ce qui permet d'augmenter le rapport signal/bruit.

Les signaux illustrés ont été obtenus avec un filtre de Daubechies d'ordre 6 : ce filtre semble donner un bon compromis entre la régularité des signaux reconstruits et la complexité numérique. Avec des filtres d'ordre 4, le signal est plus irrégulier; tandis qu'avec un filtre d'ordre 8, il ne semble pas y avoir d'amélioration notable de la qualité.

### 4.5.2 Détection d'apnées

Nous allons présenter des extraits de deux séquences, afin d'illustrer la possibilité de détecter la signature d'une apnée centrale. La séquence portant le numéro quatre a été utilisée dans la section précédente pour illustrer la détection du rythme respiratoire, celle portant le numéro deux a été réalisée dans des conditions similaires, et comporte le même genre d'événements.

Considérons la figure 4.12, relative à la séquence n°2 : le rythme respiratoire est normal, puis survient une apnée centrale vers 670 s., d'une durée approximative de 30 s. Les signaux illustrés sont :

- $AF$, le flux d'air évalué via une thermistance;
- $CP$, la pression dans la manchette;

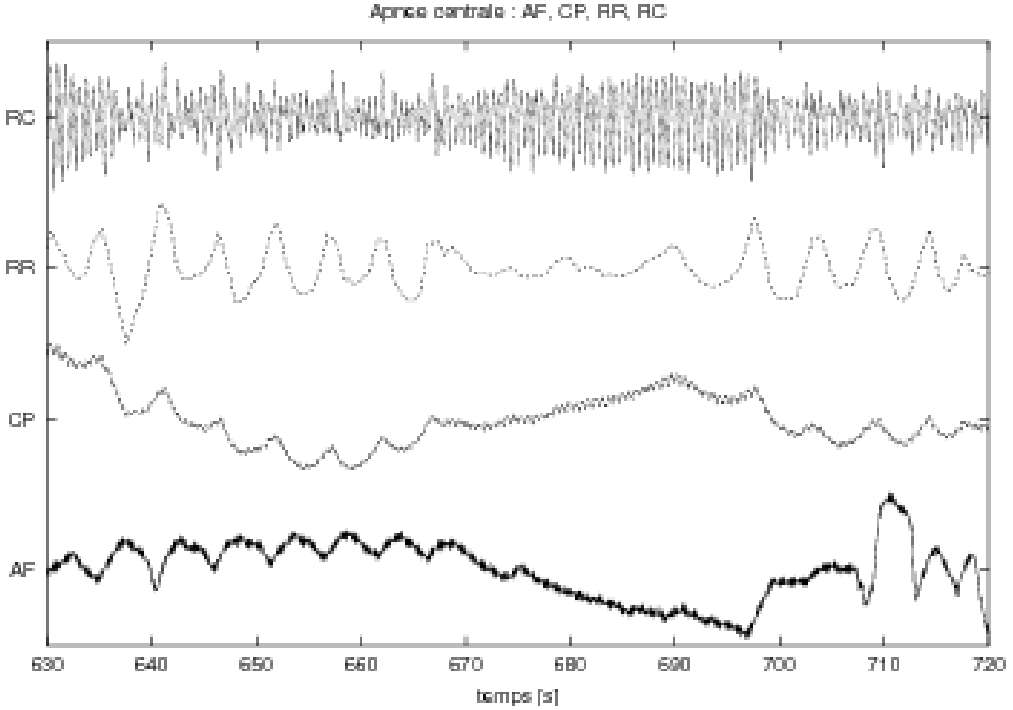



FIG. 4.12: Détection d'apnée : AF (flux d'air), CP (pression manchette), RR (rythme respiratoire) et RC (rythme cardiaque), séquence n°2

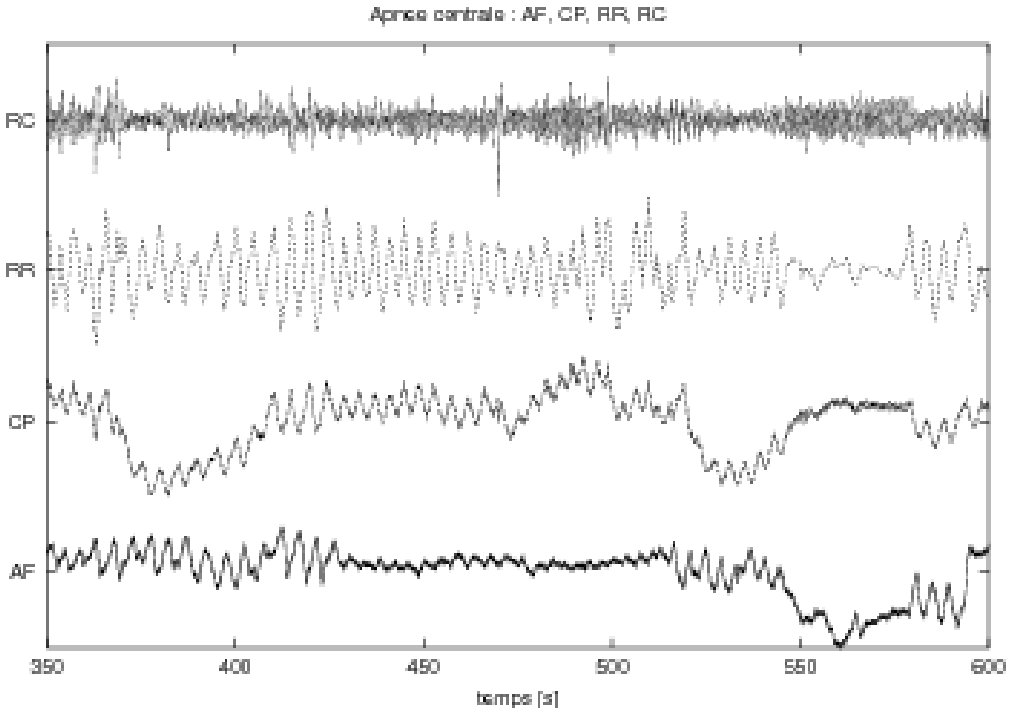



FIG. 4.13: Détection d'apnée : AF (flux d'air), CP (pression manchette), RR (rythme respiratoire) et RC (rythme cardiaque), séquence n°4

- $RR$ : le rythme respiratoire. Il s'agit comme précédemment de la somme des détails aux niveaux $-7$ à $-5$, dont la déviation standard est d'environ 10 codes;
- $RC$ : le rythme cardiaque. Il s'agit ici de la somme des détails aux niveaux $-4$ à $-2$ : nous avons constaté que, de cette façon, le signal obtenu est nettement moins bruité.

Nous observons deux phénomènes concomitants :

- l'onde de respiration, qui est bien régulière, disparaît quasiment lors de l'apnée; les variations qui apparaissent sont à attribuer à des artefacts de mouvement;
- durant la phase de respiration, l'amplitude du rythme cardiaque est assez variable, tandis que durant l'apnée elle est nettement plus constante.

Pour la figure 4.13, nous avons la suite d'événements suivants :

- à partir de 6 minutes, un épisode de ronflement, d'une durée de 1 minute. On remarquera une amplitude plus importante dans le signal thermique, mais cela ne semble pas donner lieu à une signature distincte dans le rythme respiratoire.
- ensuite, deux minutes de respiration normale, avec quelques respirations plus importantes vers la fin.
- une apnée centrale démarre à l'abscisse 9 minutes et dure 30 secondes.
- le signal se termine par une respiration normale.

Ici encore, il y a une nette diminution de l'amplitude du signal RR durant l'épisode d'apnée, et une stabilisation de l'amplitude du signal RC.

Nous n'avons pas pu mettre en évidence de modulation d'amplitude du rythme cardiaque; nous avons cherché une modulation classique de type AM:

$$x(t) = A(1 + \cos(\omega_m t)) \cdot (cos(\omega_s t)) \tag{4.3}$$

Nous avons utilisé une méthode itérative utilisant la méthode de Prony : nous avons supposé que l'approximation est une fonction de deux paramètres non-linéaires, $\omega_m$ et $\omega_s$. La fonction génératrice dépend de trois paramètres liés, $\omega_s - \omega_m$, $\omega_s$, et $\omega_s + \omega_m$. La méthode itérative a consisté à minimiser l'erreur, ce qui implique de calculer le gradient et la hessienne du vecteur d'erreur. Au niveau des résultats, nous nous attendions à trouver pour $\omega_s$ la fréquence cardiaque et pour $\omega_m$ la fréquence respiratoire, mais cela n'a pas été le cas.

Ce genre d'observation a pu être répétée sur d'autres courbes obtenues dans les mêmes conditions d'enregistrement. Les filtres en ondelettes ont donc montré leur utilité pour séparer les signaux respiratoires et cardiaques, et permettre d'évaluer la période de chacune de ces grandeurs.

## 4.6 Comparaison des deux approches

La méthode de filtrage sans décimation résulte d'un accident de parcours, nous avons pensé à ne pas la présenter. Néanmoins, en comparant systématiquement certaines courbes, nous nous sommes rendu compte de ce que les fonctions de transfert des deux méthodes ne sont pas identiques.

Nous nous sommes concentré sur le rythme cardiaque. Pour un des enregistrements où nous disposions des pressions dans la manchette et œsophagienne, nous avons comparé une série de courbes :

1. aux niveaux $-3$ (figure 4.14)et $-4$ (figure 4.15), fonction d'auto-corrélation pour les signaux avec et sans décimation, représentés respectivement en traits continus et interrompus;
2. modélisation par la méthode de Prony standard de la somme des détails 2 et 3, figure 4.16. Le modèle contient 32 sinusoïdes, la courbe resynthétisée est un pointillé;
3. modélisation par la méthode de Prony standard de la fonction d'auto-corrélation de la somme des détails 2 et 3, figure 4.17. Le modèle contient 16 sinusoïdes;
4. contenu en fréquence, associé aux deux cas précédents, figure 4.18. Les lignes continues correspondent à la pression dans la manchette tandis que celles en traits interrompus correspondent à la pression œsophagienne. Les deux familles d'amplitude ont été normalisées par rapport à leur maximum respectif. Les fréquences sont exprimées en battements par minutes.

Nous avons effectué un certain nombre d'observations. Bien qu'elles ne concernent qu'une seule expérience, nous pensons qu'il s'agit de pistes qui méritent d'être explorées sur un plus grand nombre de campagnes de mesures.

En ce qui concerne la fonction d'auto-corrélation, nous avons constaté que la méthode non-biaisée introduit une légère distorsion. Il en résulte que les fréquences présentes peuvent être légèrement modifiées. Une méthode de fenêtre glissante travaillant soit par redondance cyclique, soit sur la suite du signal, ne présente pas cet effet.

Pour la modélisation par une somme de sinusoïdes des signaux, nous avons constaté qu'il convient de prendre 32 fréquences pour les signaux bruts et 16 pour les fonctions d'auto-corrélation, afin d'obtenir visuellement une bonne concordance entre le signal de départ et le signal resynthétisé. Nous avons utilisé la méthode de Prony standard, ce qui fournit en même temps la composition fréquentielle du signal. Par rapport à la méthode de Fourier, les fréquences obtenues sont indépendantes les unes des autres.

Les courbes d'auto-corrélation sont nettement plus régulières que les courbes brutes. En ce qui concerne les contenus fréquentiels, les phénomènes de modulation sont beaucoup moins marqués.

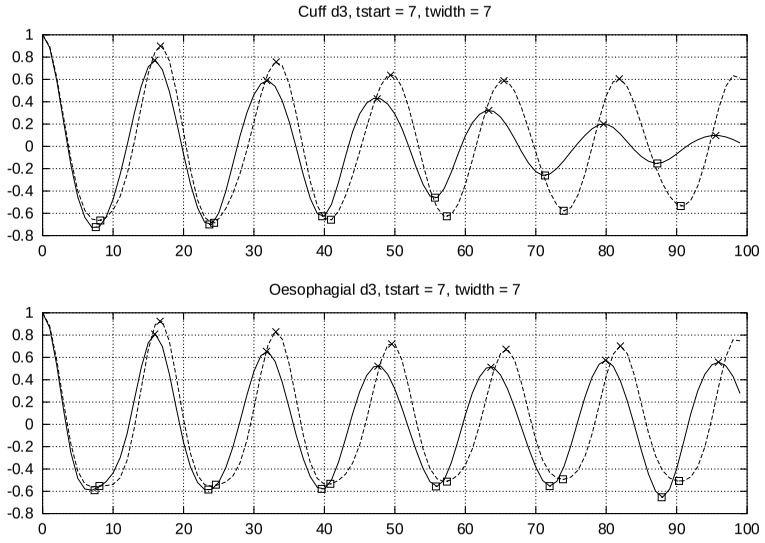



FIG. 4.14: Détail 3

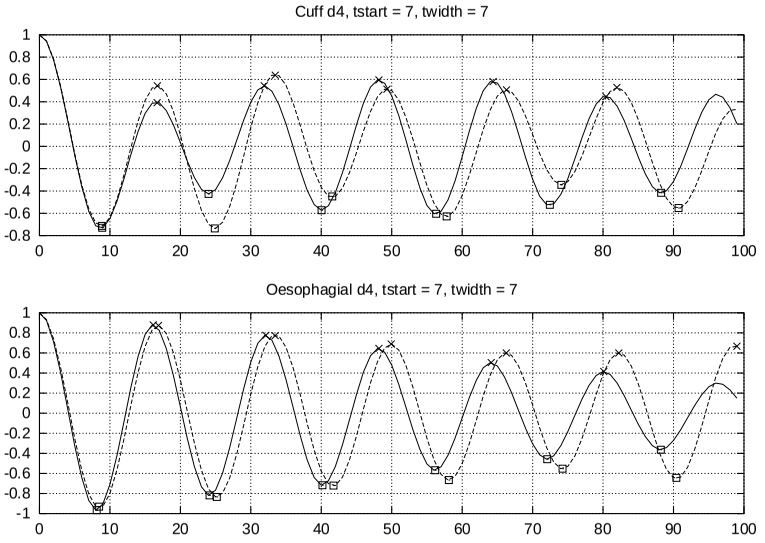



FIG. 4.15: cepstre du signal de sortie

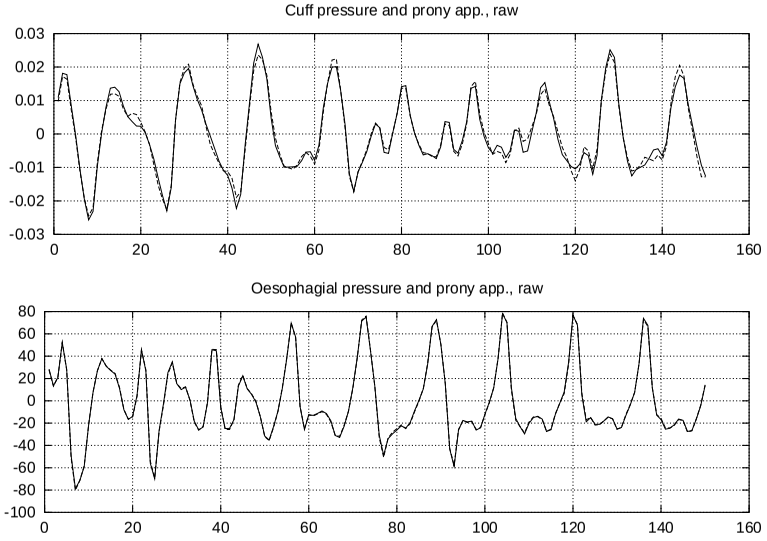



FIG. 4.16: cepstre de la réponse impulsionnelle

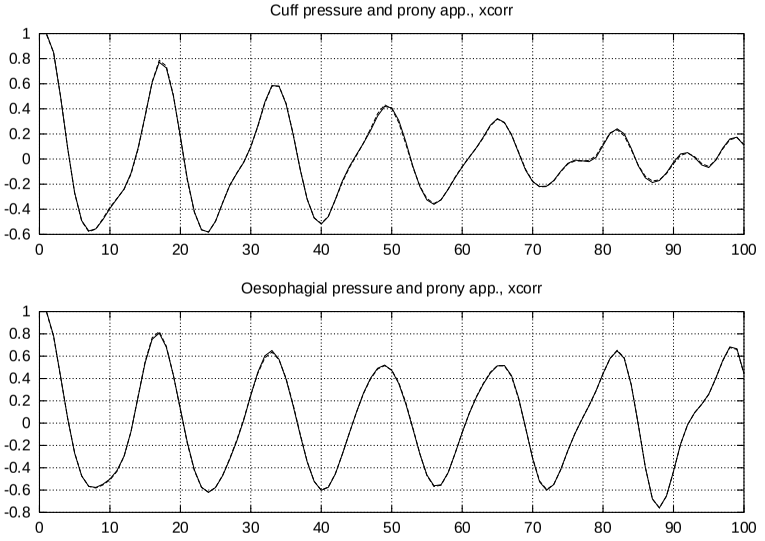



FIG. 4.17: cepstres partiels

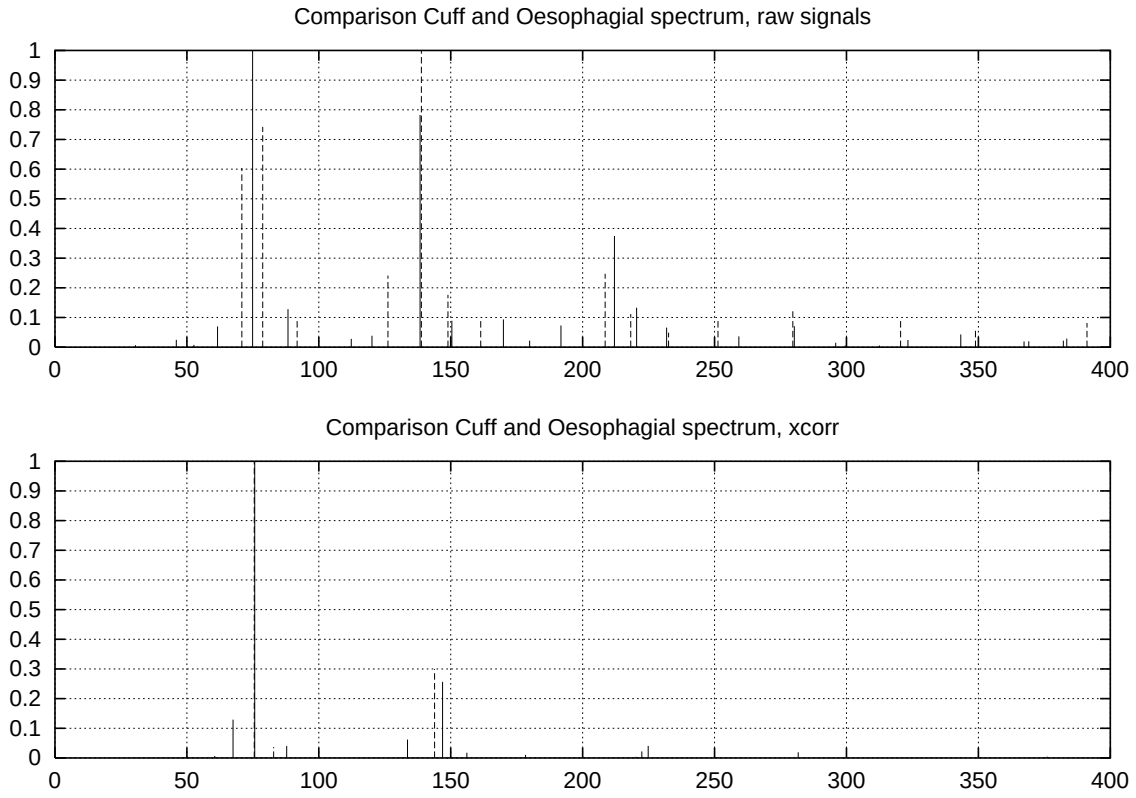



FIG. 4.18: Contenu fréquentiel des figures 4.16 et 4.17.

Pour ce qui est des spectres illustrés dans la figure 4.18, nous constatons que le spectre du signal brut est difficile à interpréter : en effet, on observe une série de composantes aux alentours de la fréquence cardiaque.

Cette approche nous a permis de constater que les signaux sont parfois modulés : sur la séquence analysée, certaines composantes sont parfois groupées aux alentour de 75 Hz et distantes d'une quinzaine de Hz. La question de la détermination de « la » fréquence prend ici tout son sens. En ce qui concerne le contenu fréquentiel des courbes d'auto-corrélation, ce phénomène disparaît et la concordance entre les fréquences principales des deux familles de courbe est excellente.

Les fonctions d'auto-corrélations des méthodes avec et sans décimation ne sont pas identiques, ce qui indique que les filtres ne sont pas équivalents. Il serait intéressant d'analyser plus en détail les deux approches, en particulier du point de vue de la présence de modulation et de la sensibilité de la fréquence principale par rapport à cette modulation.

## 4.7 Conclusion

La combinaison d'une méthode d'acquisition à haute résolution et d'une méthode de filtrage par ondelettes nous a permis d'obtenir une évaluation combinée de plusieurs paramètres physiologiques. L'avantage de cette approche est que le dispositif instrumental est particulièrement simple et peu contraignant, et utilisable en mode ambulatoire.

La haute résolution s'est révélée indispensable afin de mettre en évidence des signaux de très faible amplitude.

La méthode de séparation par ondelettes nous a permis d'obtenir plusieurs vues de notre signal et de rechercher une série de signatures spécifiques, permettant de faire la différence entre plusieurs types d'état respiratoires. Seul le ronflement ne semble pas donner lieu à une manifestation distincte.

*Quatrième partie*

---

# ***Caractérisation thermique de l'épiderme***

Ce chapitre est plus particulièrement consacré à des applications de la méthode thermométrique à haute résolution. Nous nous sommes intéressé à une classe d'application consistant à établir une corrélation entre certaines caractéristiques de la peau humaine et ses propriétés de transmission thermique.
Nous passerons une série de méthodes en revue, que nous avons utilisées pour comparer et différencier les réponses thermiques. Dans cette partie, les développements ont essentiellement été d'ordre méthodologique. Utilisant l'Identification des Processus, nous avons d'abord utilisé des méthodes standard. Ayant eu du mal à utiliser et à interpréter les résultats fournis, nous avons mis au point une extension de la méthode de Prony, qui a pu fournir une classification que nous estimons être plus robuste des différentes réponses thermiques.

Chapitre 1

# Contexte

## 1.1 Introduction : tests dermatologiques

Nous allons présenter une application de la méthode thermométrique à haute résolution dans le domaine de la dermatologie. Le point de départ fut une lecture des travaux du docteur D. Van Neste [28, 27, 26], qui à l'époque évaluait les modifications des propriétés de l'épiderme sous l'effet d'agents irritants.

Pour ce faire, une quantité définie d'histamine était injectée dans la couche superficielle de l'épiderme. Par la suite, les mesures consistaient à comparer une zone saine de référence avec une zone se situant à proximité du point d'injection, autour duquel s'était formé un érythème.

Ce genre de mesure apportait des renseignements sur la variation de température provoquée par le produit irritant, mais ne pouvait fournir aucune indication au niveau des modifications des *caractéristiques thermiques* de l'épiderme. Nous avons pour cela imaginé une autre approche, consistant à relever la réponse thermique à une onde de température générée par une source contrôlée de flux de chaleur.

Cette approche a été utilisée avec comme finalité les tests d'allergie. Ces tests consistent à appliquer sur la peau des agents auxquels le sujet est susceptible de développer une réaction allergique. Les zones de peau ainsi traitées peuvent se modifier : les phénomènes vont d'une simple rougeur passagère à une réaction cutanée : éruption, desquamation, ....

Ces réactions sont liées d'une part à la modification de l'état des vaisseaux capillaires situés à proximité de la zone d'application, et d'autre part à une modification de la teneur en eau de l'épiderme.

Classiquement, le test de réaction est visuel et sensitif. Le médecin observe s'il y a apparition d'une rougeur ainsi que son intensité (modérée - moyenne - importante). Il estimera également la modification au toucher de l'état de surface de la peau.

Ces tests sont donc plutôt qualitatifs, et reposent sur l'acuité visuelle et sensitive du médecin.

Nous avons donc eu l'idée, suite à des contacts avec d'autres médecins, de développer une méthodologie permettant d'évaluer de manière plus précise les modifications des propriétés de l'épiderme générées par le produit de test.

Avant de présenter les différents points mentionnés, nous allons d'abord donner quelques éléments de la physiologie de la peau.

## 1.2 Physiologie de la peau et mesure de ses caractéristiques fonctionnelles.

La peau, interface entre l'intérieur et l'extérieur de l'organisme, exerce une double fonction de barrière. D'une part, elle prévient la fuite de liquide interne et, de part son opacité aux infrarouges, limite les pertes de chaleur par rayonnement. Par ailleurs, elle empêche la pénétration excessive de substances extérieures vers l'organisme.

Sa position d'interface en fait un lieu d'échange thermique important. La vasodilatation est un mécanisme de transfert de chaleur vers l'extérieur, qui repose sur trois effets :

- augmentation de la conductivité thermique par augmentation de la vascularisation;
- apport de chaleur par le sang;
- apport de chaleur provenant des couches sous-jacentes du derme.

Un autre mécanisme est la transpiration qui permet d'évacuer une quantité de chaleur par vaporisation d'eau, cette eau étant excrétée au niveau de la peau par les glandes sudorales.

Ces mécanismes sont sous contrôle du système nerveux sympathique. Bien qu'elle dispose de sa propre régulation de température, la peau participe à la régulation de la température centrale. Cette régulation propre est organisée sous forme de cycles circadiens, la valeur instantanée de la température étant un reflet de l'état général de l'organisme : veille, repos, niveau d'activité physique, ...

Sa position d'avant-poste l'expose à subir un certain nombre d'agressions provenant du milieu extérieur : chaud, froid, taux d'humidité trop faible, ... Une réponse appropriée à ces agressions (vasodilatation, vasoconstriction, transpiration) lui permet de maintenir son intégrité et ses caractéristiques fonctionnelles.

L'épiderme se compose d'une couche de cellules mortes, ou stratum corneum. Cette couche se renouvelle régulièrement par desquamation et remplacement par des cellules provenant par différenciation de cellules sous-jacentes appelées kératinocytes.

La conductance électrique du stratum corneum est relativement faible, tandis que celle des kératinocytes et du derme sous-jacent est nettement plus importante, l'eau étant responsable de cet état de conduction électrique.

La peau est également parcourue par un réseau très dense de vaisseaux sanguins de très petit diamètre, les capillaires. Par diffusion à travers la paroi des cellules, des éléments nutritifs lui sont apportés (oxygène, glucose, ...) tandis que les déchets liés au métabolisme (gaz carbonique, ...) s'y déversent et repartent dans la circulation sanguine veineuse.

La peau n'est pas totalement étanche, et une substance chimique appliquée à sa surface peut diffuser à travers et entraîner une réaction. Un type particulier en est la réaction inflammatoire, qui se caractérise d'abord par une vasodilatation intense (érythème) ainsi que par une sensation de brûlure ou de démangeaison. En cas d'exposition prolongée, les cellules de l'épiderme peuvent dégénérer et prendre un aspect anormal : lésion inflammatoire, eczéma, prurit, ...

Ce mécanisme d'irritation a d'abord été utilisé par des plantes et des animaux, pour se protéger contre d'éventuels prédateurs ou, au contraire, pour induire chez une proie une réaction tellement violente qu'elle mène à la paralysie de celle-ci, une proie immobile étant plus facile à capturer.

Un exemple-type de plante irritante est l'ortie. Ses feuilles et ses tiges sont garnies de poils urticants, longues cellules creuses, à la pointe acérée renforcée de silicate. Lors d'un contact avec la peau, la pointe la pénètre et s'y brise, en injectant dans la blessure une substance protéique contenant notamment de l'histamine, de l'acétylcholine ainsi qu'un suc contractant les muscles lisses. C'est ce cocktail acétylcholine-histamine qui provoque la sensation de brûlure.

Au cours de son histoire, l'homme a appris à synthétiser un nombre toujours croissant de substances chimiques différentes nécessaires à son activité. Les égyptiens étaient passés maîtres dans l'art de la conservation des corps par momification. Les grecs avaient développé toute une pharmacopée basée sur l'utilisation de plantes médicinales. Mais la chimie a dû attendre la révolution industrielle pour connaître un essor considérable. Depuis, elle est omniprésente dans notre société, sous diverses formes : pétrochimie, chimie organique, produits phytosanitaires, biochimie, agro-alimentaire, ...

Or, certaines substances chimiques présentes dans des produits à usage quotidien sont des allergènes potentiels. Comment déterminer d'une manière fiable ce caractère? L'extrapolation des résultats obtenus sur modèle animal est aléatoire, le type de réaction pouvant varier fortement d'une espèce à l'autre, voire d'un individu à l'autre.

Le test d'allergie classique chez l'être humain consiste à faire pénétrer dans l'épiderme une quantité minime du produit étudié à l'aide d'une lancette et à relever les symptômes cliniques apparaissant consécutivement. Ces symptômes vont d'une rougeur fugace à la formation d'un érythème plus ou moins étendu, avec une papule à l'endroit

d'injection. La papule est due à une accumulation d'eau libre tandis que l'érythème provient d'une vasodilatation intense.

Un score clinique simple peut être attribué par observation de la taille de l'érythème, après le temps nécessaire à l'apparition d'une réaction significative.

Le docteur D. Van Neste, dans sa thèse d'agrégation de l'Enseignement Supérieur [26], présente trois méthodes non-invasives d'évaluation de l'état fonctionnel de la peau. Il s'agit :

- de la mesure de la capacitance de l'épiderme, reflet de la quantité d'eau présente à proximité de la sonde;
- de la mesure de la perte transépidermique en eau (TEWL). Celle-ci donne une image de la quantité d'eau quittant à chaque instant l'épiderme par évaporation;
- de la mesure de flux sanguin par vélocimétrie laser Doppler. Dans ce type d'expérience, un rayon laser, modulé à une certaine fréquence, est envoyé sur la peau. Il la pénètre sur une distance d'environ 1–1.5 mm, se réfléchit sur les cellules présentes et revient vers un détecteur. Au cas où la cellule est en mouvement, la fréquence de l'onde réfléchie est une image de la vitesse de celle-ci (effet Doppler). On obtient de la sorte une indication sur le nombre moyen de cellules présentes ainsi qu'une image de leur vitesse moyenne.

Le protocole d'expérimentation consiste tout d'abord à mesurer ces valeurs sur peau saine. Le produit étudié est ensuite appliqué. L'histamine ([27]) est injectée par une lancette, tandis que le lauryl sulfate (SLS, [28]), produit irritant « de référence » est appliqué sous occlusion (patch). Ce produit se retrouve dans un certain nombre de détergents à usage industriel ou domestique. Après un laps de temps adéquat, la réponse de l'organisme est évaluée par les différentes méthodes évoquées ci-dessus. Dans le cas de l'histamine, ce temps est d'environ 10 minutes, dans le cas du SLS, la réponse est évaluée après 4 h, 8 h, ...

Le protocole proposé ne permet en aucun cas d'évaluer la dynamique de la réponse, on ne connaît que le point de départ, avant le stimulus, et le point d'arrivée, quand la réponse a atteint sa situation de régime.

## 1.3 Propriétés thermiques du corps humain

Le corps humain est un milieu très hétérogène, chaque partie possédant des caractéristiques thermiques, des sources de chaleur d'origine métabolique ainsi que des mécanismes de transport de la chaleur qui lui sont propres. La température interne est contrôlée de façon très stricte, cette régulation est assurée notamment par modulation du flux de chaleur à travers l'épiderme. Par ailleurs, la température de la peau est susceptible d'être modifiée par une série de facteurs, certains étant d'origine interne ou métabolique, tandis que d'autres sont liés aux paramètres d'environnement comme la

température de l'air, le taux d'humidité, ... Typiquement, la gamme de températures normalement observables au niveau de l'épiderme s'étend de 29 à 33 °C.

Le calcul de cette température de surface implique de connaître à la fois le champ thermique régnant à l'intérieur de l'organisme, ainsi que les paramètres d'environnement. Si ces derniers sont aisément accessibles, il n'en est pas de même des paramètres internes. La température absolue n'est donc pas un paramètre fiable, car trop susceptible d'être modifiée par des facteurs internes. Par contre, si nous effectuons des mesures à haute résolution autour d'un point d'équilibre, tout ce dont nous avons besoin est d'un modèle linéaire (ou linéarisé) de la relation entre le flux thermique et la température, sans avoir à se soucier de sources internes distribuées, ...

La bibliographie à la fin de cette thèse contient un certain nombre de références d'articles concernant des études de la modélisation des flux thermiques et des répartitions de température dans le corps humain.

## 1.4 Description des mesures

Nous avons obtenu une série de courbes dont les caractéristiques temporelles sont les suivantes :

- fréquence d'échantillonnage 4 Hz;
- 1024 points au total, soit une durée d'acquisition de 17 minutes;
- ces courbes constituent des réponses thermiques à une grandeur d'excitation qui est un flux de chaleur. Cette grandeur varie par tout en rien, réalisant ainsi une série périodique de créneaux;
- ces courbes correspondent à différents sites de mesures, nous les distinguerons en parlant de leur dose de modification.

## 1.5 Utilisation des résultats

Pour exploiter les différentes courbes, il faut disposer d'une modèle entrée-sortie entre la grandeur d'excitation et la réponse mesurée. Étant donné les caractéristiques physiques du système, nous avons posé comme prérequis que le modèle obtenu devait répondre à un certain nombre de conditions :

- le modèle doit être stable, et la réponse de régime rester finie. En effet, à l'équilibre, la chaleur apportée dans le système doit être égale à la chaleur dissipée vers le milieu ambiant, dont on suppose que sa capacité thermique est infiniment grande par rapport à celle du système;

– dans un modèle thermique, il n'y a pas de résonance d'énergie entre des modes de stockage d'énergie de type capacitif et inductif : cela implique que les pôles doivent être réels.
– un système continu est stable si ses pôles sont à partie réelle négative. La relation entre les constantes de temps analogiques $t_c$ et les pôles $z_d$ d'un système de période d'échantillonnage $T_s$ sont

$$t_c = \frac{-T_s}{\log z_d} \tag{1.1}$$

Pour arriver à ce résultat, considérons la suite $x(t)$, correspondant à une exponentielle décroissante, de constante de temps $t_c$, évaluée en des multiples de la période d'échantillonnage $T_s$ : $t = 0, T_s, 2T_s, \ldots$:

$$x(t) = 1, \exp\frac{-T_s}{t_c}, \exp\frac{-2T_s}{t_c}, \ldots \tag{1.2}$$

dont la transformée en $\mathcal{Z}$ peut s'écrire

$$X(z) = 1 + \exp\frac{-T_s}{t_c}z^{-1} + \exp\frac{-2T_s}{t_c}z^{-2} + \ldots \tag{1.3}$$

si la constante de temps analogique est positive, cette série converge

$$X(z) = \frac{1}{1 - exp\frac{-T_s}{t_c}\ z^{-1}} \tag{1.4}$$

le pôle digital $z_d$ est la valeur qui annule le dénominateur, soit

$$z_d = exp\frac{-T_s}{t_c}, \tag{1.5}$$

ce qui mène bien à la relation annoncée. Par conséquent, le partie réelle des pôles digitaux doit être positive.

## 1.6 Présentation des méthodes utilisées

Nous avons utilisé successivement plusieurs approches pour essayer de trouver un modèle de la relation entrée-sortie répondant à ces critères :

– identification boîte noire;
– identification paramétrique; en établissant une correspondance entre un modèle discret et les paramètres fournis comme solution du problème de minimisation de l'erreur de prédiction;

– identification basée sur la résolution de l'équation de propagation de la chaleur entre la source et la mesure;
– identification cepstrale, consistant à utiliser une transformation particulière où l'image de la convolution est une addition;
– méthode de Prony générique, consistant à approximer les signaux mesurés par une série d'exponentielles dépendant d'un jeu commun de paramètres.

Avant de présenter séparément chacune d'elle, nous allons brosser rapidement un tableau des différents comportements observés.

Il est apparu en utilisant les méthodes classiques que :

1. les modèles travaillant en prédiction (ARX) donnaient quasi-systématiquement des pôles instables. En effet, dans ces modèles, il n'y a aucune pénalité si le modèle est instable, puisque la prédiction d'ordre 1 reste bornée;
2. les modèles travaillant en simulation (OE) sont extrêmement sensibles aux conditions initiales. Dans l'implémentation MatLab, cela résulte du fait que les conditions initiales ne font pas partie des paramètres à optimiser mais résultent d'une approche extrêmement sensible au bruit;
3. par ail leur, une classification par modélisation ne nous semblait guère parlante : qu'est-ce qui est physiquement différent lorsque deux modèles donnent des coefficients distincts ? Nous sommes passé par de l'identification paramétrique, mais un certain nombre de résultats correspondaient à des modèles qui n'étaient pas physiquement réalisables;
4. en outre, nous ne savions pas comment faire la part des choses, en comparant des résultats différents, entre les variations dues aux paramètres physiques et les variations engendrées par la méthode de calcul (choix du nombre de coefficients, du délai entrée-sortie, ...)

Nous sommes passés par un approche cepstrale, afin de vérifier l'absence de chemins de propagation multiples. Nous avons alors essayé, en travaillant sur des segments du signal, d'obtenir une réponse de régime, mais sans succès.

Si nous avions appliqué la méthode de Prony classique, nous aurions dû rechercher des modèles par tronçons, correspondant aux zones ou la commande est constante. Il n'y aurait pas eu de cohérence globale. Nous avons ensuite mis au point une extension de la méthode de Prony, en constatant que les matrices de régression des différents segments pouvaient être combinées et donner lieu à une recherche globale. Nous avons ainsi une méthode qui donne un modèle global, calculé sur toute la durée du signal. Ce modèle fournit également les conditions initiales, ce qui est utile étant donné que notre signal évolue systématiquement dans une zone transitoires. En même temps, ce modèle permet de prédire le comportement asymptotique ou de régime.

Cette approche nous semble plus intéressante physiquement, puisque l'on raisonne en terme d'amplitude d'onde et de constante de temps. De plus, en faisant varier les

fenêtres sur lesquelles nous avons calculé les modèles, nous avons obtenu une série de réalisations. Si on regarde les diverses courbes générées, on constate que la variation entre les différentes réalisations est nettement plus petite que la variation due à l'agent irritant, ce qui est rassurant quant à la possibilité d'obtenir une classification par rapport à la dose d'agent irritant.

En étudiant les matrices de régression pour les différentes expériences, il est apparu que le comportement quasi-intégrateur avait pour effet de rendre cette matrice mal conditionnée : en effet, les colonnes formées avec y(t) et y(t-1) sont très peu différentes. Il en résulte des difficultés lors de l'inversion de la matrice de régression, et une susceptibilité accrue au bruit. C'est probablement ce qui explique la variabilité des coefficients obtenus par les méthodes classiques. Dans l'approche de Prony généralisée, la détermination des amplitudes des exponentielles se base sur une matrice de régression ou les colonnes sont nettement différentes et donc mieux conditionnées.

En outre, la méthode de Prony requiert que les fonctions analysantes (servant à former la matrice de régression) soient linéairement indépendantes, ce qui est nettement moins restrictif que d'autres méthodes d'analyse comme la DFT ou les fonctions analysantes doivent être non-corrélées. On peut donc imaginer de développer des modèles où les signaux analysants contiendraient l'une ou l'autre composante plus exotique, comme une dérive linéaire, une parabole, ...

Chapitre 2

# Classification par modélisation

## 2.1 Contexte

Nous avons essayé diverses approches pour classifier les jeux de courbes. Avant d'exposer ces méthodes, illustrons d'abord un exemple-type de courbes obtenues. La figure 2.1 contient trois courbes, translatées pour démarrer du même endroit. Elles correspondent, de bas à haut, à des modifications respectives de 0 % (site de référence), 0.5 % et 2 %.

## 2.2 Modélisation de type boîte noire

Cette méthode consiste à ne se préoccuper que de la relation mathématique liant la sortie à l'entrée. Dans cette approche, plusieurs types de modèles permettent de trouver le « meilleur » jeu de paramètres : après avoir choisi une structure de départ, un algorithme détermine, pour cette structure, comment approcher au mieux la relation entrée-sortie observée. Les différents modèles correspondent à différentes hypothèses au niveau de la modélisation, le point le plus important étant l'influence du bruit.

Les inconvénients de cette approche sont d'une part qu'il est difficile de donner un sens physique aux coefficients obtenus, et d'autre part que les solutions obtenues ne satisfaisaient pas toujours aux contraintes imposées de stabilité et de pôles digitaux réels et positifs.

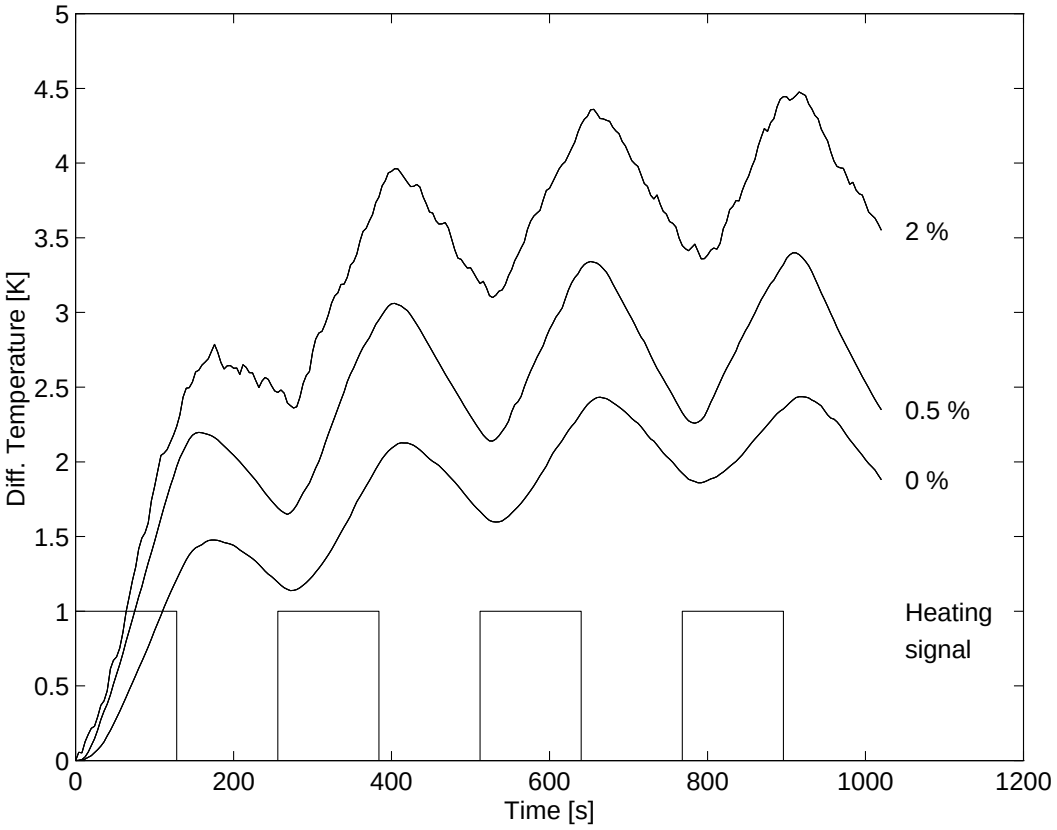



FIG. 2.1: réponse thermique en fonction de la dose de produit irritant

## 2.3 Modélisation par cellules R–C

Nous avons utilisé l'analogie[1] entre les systèmes thermiques et électriques pour représenter la propagation thermique par la mise en cascade de deux cellules de type R–C. La première cellule modélise l'élément de chauffe, à savoir une source de flux thermique $Q$, une capacité $C_t$ et une conductance $k_t$ vers le milieu extérieur, la température au point d'injection valant $T_t$.

La seconde cellule modélise la propagation à travers le milieu, à savoir une conductance $k_c$ de couplage ainsi qu'une conductance $k$ et une capacité $C$ vers le milieu extérieur.

Le lecteur remarquera que le capteur n'apparaît pas dans ce schéma. En fait, nous avons fait l'hypothèse que sa dynamique thermique était faible par rapport à celle du

1. Analogie Thermique – Électrique

| | Thermique | | | Électrique | | |
|---|---|---|---|---|---|---|
| Variable d'état (potentiel) : | température | T | [K] | tension | V | [V] |
| Flux : | flux de chaleur | Q | [J/s] | courant | I | [C/s] |
| Lien flux – potentiel : | conductance | k | [J/s.K] | conductance | g | [C/s.V] |
| Stockage d'énergie : | capacité | C | [J/K] | capacité | C | [C/V] |

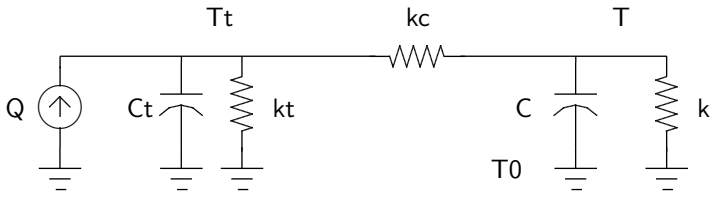



FIG. 2.2: modélisation du transfert thermique

phénomène étudié, et considéré que sa température interne pouvait être directement assimilée à celle du point de mesure tandis que ses paramètres thermiques sont englobés dans les valeurs $k$ et $C$.

Une correspondance entre les paramètres extraits d'une identification de type « boîte noire » a été établie, complétée par la détermination des paramètres de la sonde pour suppléer à la différence entre le nombre de paramètres nécessaires et la dimension du modèle identifié.

D'une manière générale, nous avons pu conclure après identification, pour cette méthode ainsi que la précédente :

1. au niveau de l'identification, un des pôles du dénominateur est systématiquement très proche de 1. Au niveau d'un modèle ARX, il y a donc un risque d'obtenir un modèle instable, ce qui ne correspond pas aux prérequis que nous avons pu poser sur base du modèle physique;
2. un certain nombre de courbes n'ont pu être exploitées, parce que générant des pôles numériques à partie réelle négative. D'une manière générale, nous avons rejeté tout modèle menant à des constantes physiques négatives;
3. Le lien entre les paramètres physiques et la dose de modification s'est avéré être peu marqué. Certains paramètres ont un comportement assez erratique, en particulier celui fonction du couplage thermique entre la le milieu et la thermistance. Seule la conductance thermique du milieu a montré une dépendance marquée par rapport à la dose.

## 2.4 Modélisation par l'équation de la diffusion

Par analyse du modèle thermique et résolution de l'équation de propagation de la chaleur, nous avons établi l'allure de la réponse à un échelon de chauffe [2]. Le modèle fait intervenir un certain nombre de constantes physiques, ainsi que trois paramètres représentant le transfert de chaleur : un coefficient de convection, une conductivité et une capacité.

L'onde de température générée par une source de flux thermique dont l'amplitude est constante peut s'écrire :

$$\theta(z,t) = \frac{\dot{q}_0}{\sqrt{\frac{2h_0k}{L}}} \left[ \begin{array}{l} + e^{-\sqrt{\frac{h_0}{k\,L}}} \; erfc\left(\frac{z}{2\sqrt{D\,t}} - \sqrt{\frac{h_0}{k\,L} D\,t}\right) \\ - e^{+\sqrt{\frac{h_0}{k\,L}}} \; erfc\left(\frac{z}{2\sqrt{D\,t}} + \sqrt{\frac{h_0}{k\,L} D\,t}\right) \end{array} \right] \tag{2.1}$$

La réponse à un créneau s'obtient par superposition de réponses indicielles décalées dans le temps.

Les variables utilisées dans cette équation sont :

$\dot{q}_0$ l'amplitude de la source de flux thermique;
$h_0$ le coefficient de convection;
$k$ la conduction thermique dans le milieu;
$L$ la taille de la couche de convection;
$D = \frac{\rho c}{k}$ la constante de diffusion;
$z$ la distance par rapport à la source;
$t$ le temps;
$erfc$ est la fonction d'erreur complémentaire.

| Dose | $h_0$ [W/m$^2$ K] | k [W/m K] | $\rho$c [MJ/m$^3$K] |
|---|---|---|---|
| 0 % | 3.08 | 0.44 | 3.57 |
| 0.5 % | 2.28 | 0.41 | 2.34 |
| 2 % | 1.64 | 0.27 | 1.91 |

TAB. 2.1: Modèle non-linéaire

Globalement, les résultats ont été encourageants, montrant une dépendance plus ou moins marquée entre les paramètres et la dose. Par exemple, le tableau 2.1 donne les paramètres correspondants aux courbes de la figure 2.1, on constate une évolution monotone de chacun d'eux avec la dose, la différence entre les paramètres pour la situation de référence (0 %) et pour une faible dose (0.5 %) étant peu marquée.

Il n'en reste pas moins que l'interprétation de ces modèles n'est pas très simple. Par ailleurs, il est également apparu que, en répétant une mesure, il y avait une bonne similarité entre les réponses, mais avec des différences assez marquées entre les paramètres obtenus, ces différences pouvant être de l'ordre de 5 %.

Chapitre 3

# Classification par transformée cepstrale

## 3.1 Introduction

La transformée cepstrale fut d'abord développée comme technique d'analyse de données séismographiques [78], permettant de mettre en évidence les effets de trajets multiples. En 1975, Oppenheim [82] a dérivé la formule du cepstre complexe dans le cadre de la théorie des systèmes homomorphiques.

Depuis, de nombreuses applications en ont été développées, nous pouvons citer le traitement et la reconnaissance d'image, la compression de signaux, la dé-réverbération d'enregistrements acoustiques, ...

Le terme « cepstre » est un anagramme du mot spectre, parce que les premiers auteurs ayant utilisé cet outil effectuaient dans le domaine fréquentiel des traitements d'habitude appliqués au domaine temporel et vice-versa. Ce terme traduit ainsi cette sorte d'inversion entre les deux domaines.

## 3.2 Définition

À l'origine, le cepstre était obtenu en prenant la transformée inverse du logarithme de la transformée de Fourier discrète d'un signal. Le problème de cette approche était de disposer d'une version *continue* de la phase, dont la DFT ne calcule que la valeur principale.

Le cepstre complexe d'un signal est défini comme le logarithme de sa transformée en $\mathcal{Z}$ :

$$\hat{X}(z) = \log X(z) \tag{3.1}$$

Le qualificatif de complexe provient de ce qu'interviennent dans les calculs les pôles et les zéros (digitaux) complexes. Néanmoins pour un signal réel ces termes vont par paires conjuguées et le résultat final est réel.

La propriété fondamentale de cette transformation est la suivante :

$$y(t) = h(t) * x(t) \Leftrightarrow Y(z) = H(z).X(z) \Leftrightarrow \hat{Y}(z) = \hat{H}(z) + \hat{X}(z) \tag{3.2}$$

Pour des signaux *stables* et *à minimum de phase*, lorsque la transformée en $\mathcal{Z}$ peut s'écrire comme le quotient de deux polynômes, ceux-ci sont factorisables en un produit de monômes, et chacun de ces termes peut être développé en série de Laurent :

$$X(z) = \frac{K_0 \prod_{i=1}^{n_b}(1 - b_i z^{-1})}{\prod_{i=1}^{n_a}(1 - a_i z^{-1})} \tag{3.3}$$

$$\hat{X}(z) = \log K_0 + \sum_{i=1}^{n_b} \log(1 - b_i z^{-1}) - \sum_{i=1}^{n_a} \log(1 - a_i z^{-1}) \tag{3.4}$$

Pour la classe de signaux considérés, les pôles et les zéros sont de module inférieur à l'unité.

Utilisant le développement

$$\log(1 - \alpha z^{-1}) = -\sum_{n=1}^{\infty} \frac{\alpha^n}{n} z^{-n}, \quad |z| > |\alpha|, \tag{3.5}$$

il vient finalement

$$\hat{x}(n) = \begin{cases} \log |K_0| & n = 0 \\ \frac{1}{n}(\sum_{i=1}^{n_a} a_i^n - \sum_{i=1}^{n_b} b_i^n) & n > 0 \\ 0 & n < 0. \end{cases} \tag{3.6}$$

Un délai pur ne peut pas se modéliser de cette façon car il ne possède pas de développement en série convergent. On peut cependant en tenir compte en décalant de manière adéquate les données.

Le cepstre d'une séquence quelconque peut se calculer numériquement par une formule récursive, évitant de cette façon les problèmes inhérents de la transformée de Fourier.

## 3.3 Traitement cepstral

Pour classifier nos signaux, nous leur avons appliqué une opération que nous avons appelée « liftrage », consistant à :

1. séparer le signal de sortie en tronçons durant lesquels le signal d'entrée est constant;
2. calculer le cepstre de chacun de ces segments;
3. regrouper les segments correspondant à une même valeur du signal d'entrée, puis effectuer une moyenne des points correspondant au même endroit du cycle;
4. calculer la transformée inverse de cette moyenne et resynthétiser un signal continu qui est une approximation de la réponse de régime du système.

Mathématiquement, l'opération de liftrage consiste à obtenir un nouveau cepstre $\hat{x}_{av}(n)$ par une opération de moyenne sur les différents cepstres partiels, puis à repasser en temporel par la transformation inverse

$$lx(x) = \mathcal{Z}^{-1}\left[\exp\left(\mathcal{Z}\left(\hat{x}_{av}(n)\right)\right)\right] \tag{3.7}$$

Illustrons l'ensemble de ces opérations sur un signal qui ressemble à la sortie de notre système, mais est obtenu par une simulation pure. Le signal d'entrée est un signal en créneau constitué de quatre cycles élémentaires et comporte en tout 256 points. Le signal de sortie est obtenu par un filtre dont le numérateur est unitaire et dont le dénominateur comporte deux pôles, respectivement en 0.9 et 0.99. Ce dernier a été choisi pour simuler un système quasi-intégrateur. En effet, les différentes courbes obtenues montrent à chaque fois une réponse très lente à la composante moyenne du signal d'entrée. Par commodité, les différents graphes ont été regroupés sur une même page.

Les signaux d'entrée et de sortie, ainsi que le cepstre de la sortie, sont illustrés dans les figures 3.1 et 3.2. On remarquera en particulier que le cepstre est la superposition d'une composante ressemblant à une exponentielle décroissante ainsi que d'impulsions qui apparaissent aux instants de variation du signal d'entrée. Ce cepstre illustre la possibilité de détecter des trajets multiples : en effet, le signal en créneau peut être considéré comme étant la somme d'échos d'un signal rectangulaire ne comportant qu'un seul cycle.

La soustraction des cepstres de la sortie et de l'entrée donne la figure 3.3, il s'agit de la transformée de la réponse impulsionnelle du filtre. Nous avons travaillé ici de façon globale, c.-à-d. en considérant la *totalité* des signaux.

En pratique, il n'a pas été possible de calculer le cepstre de cette façon globale sur nos résultats expérimentaux. Nous avons constaté que la présencede bruit dans les premiers instants du signal peut perturber le calcul et rendre l'algorithme divergent.

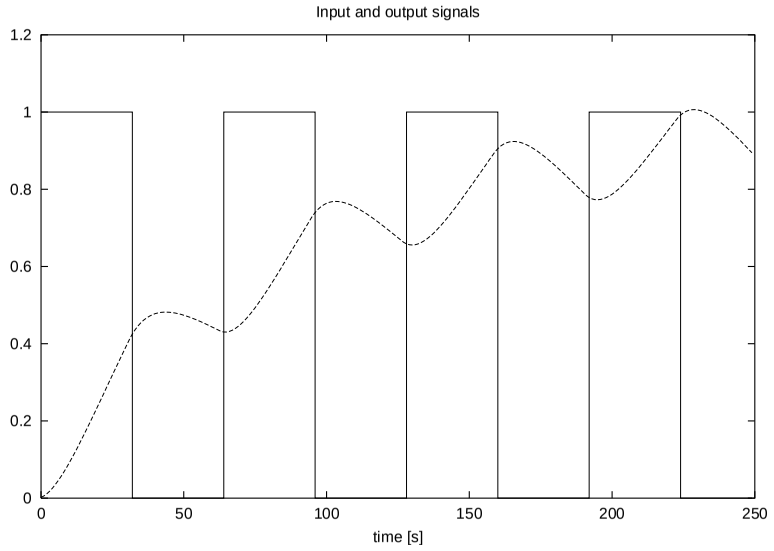



FIG. 3.1: signaux d'entrée et de sortie

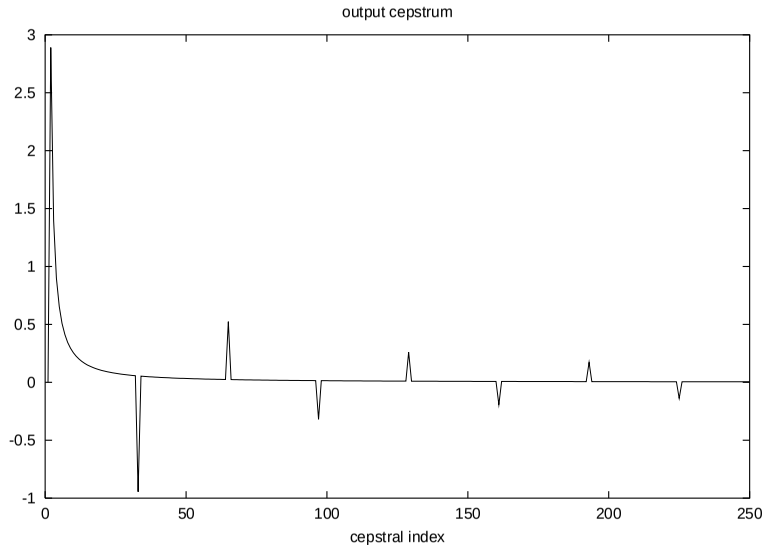



FIG. 3.2: cepstre du signal de sortie

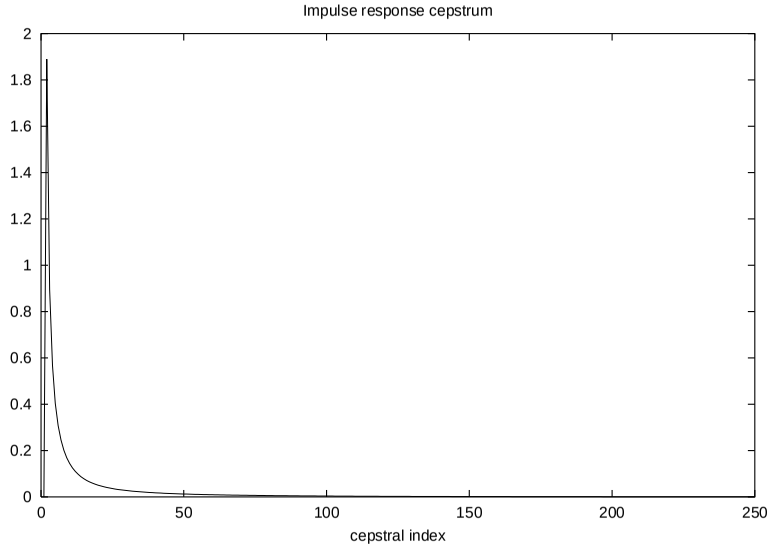



FIG. 3.3: cepstre de la réponse impulsionnelle

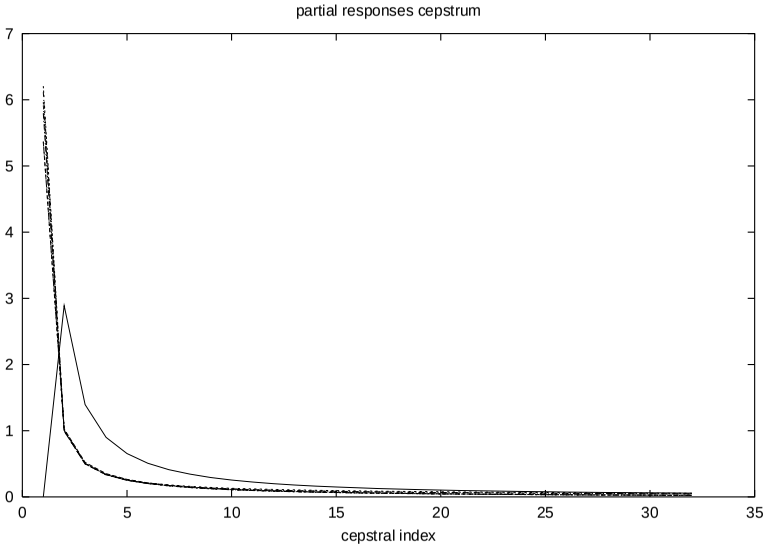



FIG. 3.4: cepstres partiels

Néanmoins, il est quand même possible de travailler par voie cepstrale à condition de travailler par segments : considérons en effet la figure 3.4. Elle contient les cepstres de *segments* de la sortie, correspondant aux instants où le signal d'entrée est constant. Nous avons donc 8 courbes comportant 32 points. Nous constatons qu'il y a en fait une convergence assez rapide entre ces courbes. En effet, nous ne distinguons que deux traces, l'une démarrant à l'origine, l'autre étant constituée de la superposition de courbes quasiment confondues.

Nous pouvons considérer que le signal de sortie, par exemple entre les instants $T_{per}$ et $2T_{per}$, résulte de la valeur de la convolution du signal d'entrée avec la réponse impulsionnelle, considérée entre les mêmes instants. Mais nous pouvons aussi considérer qu'il s'agit de la sortie du système, calculée à partir d'un nouvel instant initial de valeur nulle, en ayant injecté dans le filtre des valeurs initiales choisies de façon à ce que le début du signal simulé se raccorde correctement à la fin du signal de la période précédente. En fait, nous avons transformé un décalage temporel (ce qui ne peut pas s'exprimer en terme de cepstre) en un problème de détermination de valeurs initiales.

Si nous prenons la différence entre deux courbes de la figure 3.4 et en calculons la transformée cepstrale inverse, nous obtenons à chaque fois un signal dont la largeur est de deux points, soit le degré de notre système ou encore le nombre de conditions initiales nécessaires pour générer n'importe quel segment.

L'opération de moyenne citée plus haut revient donc à chercher des conditions initiales qui permettent de prédire au mieux l'ensemble de nos courbes. En toute rigueur, il serait plus logique de déterminer la valeur de régime des conditions initiales, ce qui permettrait de déterminer le cycle de régime sans avoir à attendre de l'avoir atteint.

## 3.4 Résultats

Cette opération de liftrage a donné quelques résultats expliqués dans [3]. La figure 3.5 illustre cette opération : elle contient les cepstres partiels de la courbe « 2 % » pour la première partie du cycle, correspondant à l'échelon de chauffe. Ce tableau contient 3 cepstres partiels (courbes continues), et leur moyenne sous forme de trait pointillé.

On remarquera que la convergence des courbes est très rapide, elles se rejoignent après quelques échantillons. Après avoir pris la moyenne des 4 traces, nous obtenons par synthèse inverse une courbe liftrée. La figure 3.6 montre le résultat de ce liftrage. Néanmoins, on constatera que l'on n'obtient pas strictement des conditions de régime.

Le tableau 3.1 donne des résultats numériques pour deux jeux d'expériences. Il permet de comparer des résultats obtenus via un modèle ARX sur le signal de départ ainsi que par un modèle OE appliqué au signal liftré. Ces deux modèles sont légèrement différents au niveau de la modélisation du bruit, mais ceci a peu d'importance car la majorité de l'énergie est à basse fréquence. Les prérequis de la section 1.5 ont été satisfaits pour les modèles obtenus sur signaux liftrés.

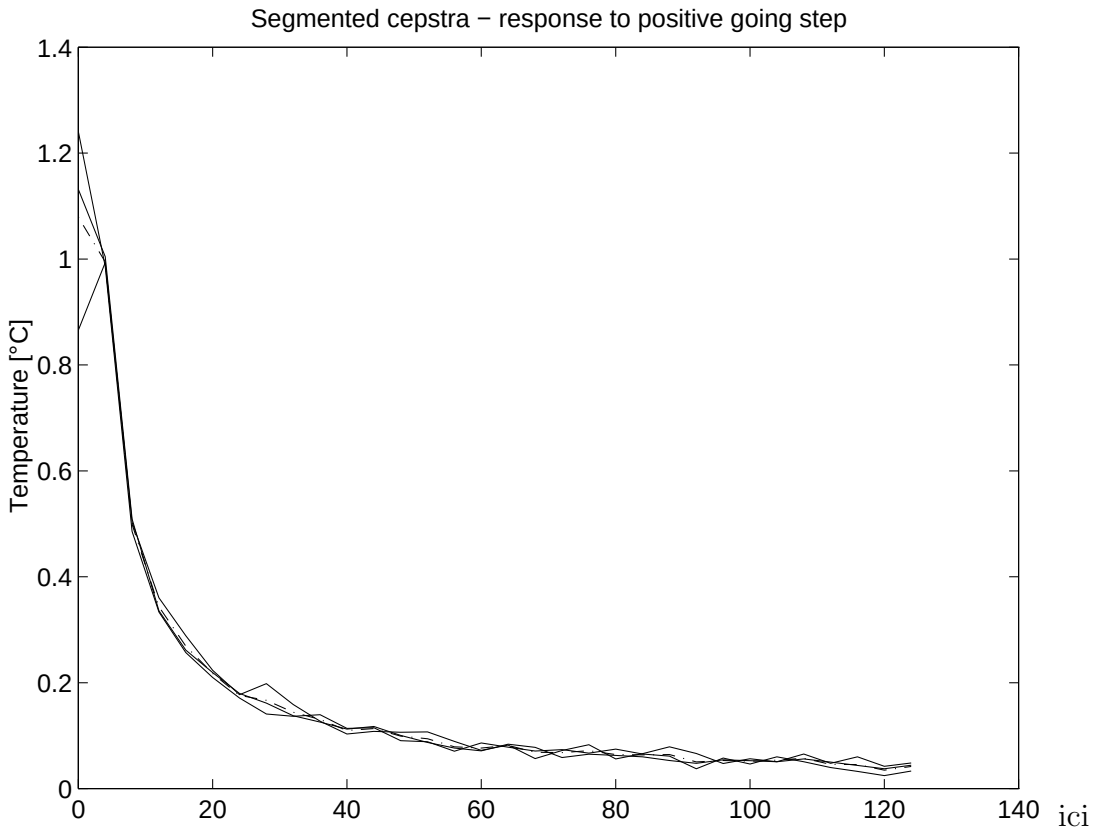



FIG. 3.5: cepstres partiels

La structure du modèle ne contient que deux pôles. Nous ne présentons pas les coefficients des polynômes, qui ne reflètent pas réellement l'ordonnancement que l'on observe sur la figure 2.1, mais plutôt des grandeurs qui en découlent, à savoir :

$T_c, \sigma T_c$ la plus petite constante de temps et sa variance;

$DC, \sigma DC$ le gain statique qui représente l'amplitude finale du signal généré par un échelon de chauffe. La variance a été obtenue à partir de résultats publiés dans

| Subject 1, raw signal | | | | | | Subject 1, "liftered" signals | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Dose | Tc [s] | $\sigma$Tc [s] | DC [W/K] | $\sigma$DC [W/K] | $\sigma$Si [K] | Dose | Tc [s] | $\sigma$Tc [s] | DC [W/K] | $\sigma$DC [W/K] | $\sigma$Si [K] |
| 0 % | 10.6 | 1.7 | 17.4 | 3.8e-3 | 0.14 | 0 % | 26.4 | .38 | 26.1 | 4.4e-6 | 0.02 |
| 0.5 % | 27.2 | 4.5 | 24.9 | 2.6e-3 | 0.16 | 0.5 % | 18.5 | .93 | 52 | 16e-6 | 0.10 |
| 2 % | 29 | 2.5 | 25.5 | 1.5e-3 | 0.16 | 2 % | 20.1 | .14 | 39.8 | 4.1e-6 | 0.03 |
| Subject 2, raw signal | | | | | | Subject 2, "liftered" signals | | | | | |
| Dose | Tc [s] | $\sigma$Tc [s] | DC [W/K] | $\sigma$DC [W/K] | $\sigma$Si [K] | Dose | Tc [s] | $\sigma$Tc [s] | DC [W/K] | $\sigma$DC [W/K] | $\sigma$Si [K] |
| 0 % | 33.2 | 4.5 | 15.2 | 1.1e-3 | 0.04 | 0 % | 26.0 | .12 | 24.8 | 2.7e-6 | 0.01 |
| 0.5 % | 22.6 | 2.7 | 20.4 | 1.4e-3 | 0.09 | 0.5 % | 18.5 | .27 | 34.2 | 6.8e-6 | 0.03 |
| 2 % | 4.2 | .62 | 33.3 | 4.0e-3 | 0.31 | 2 % | 20.4 | .34 | 43.9 | 12e-6 | 0.09 |

TAB. 3.1: Comparaison identification ARX – signaux liftrés

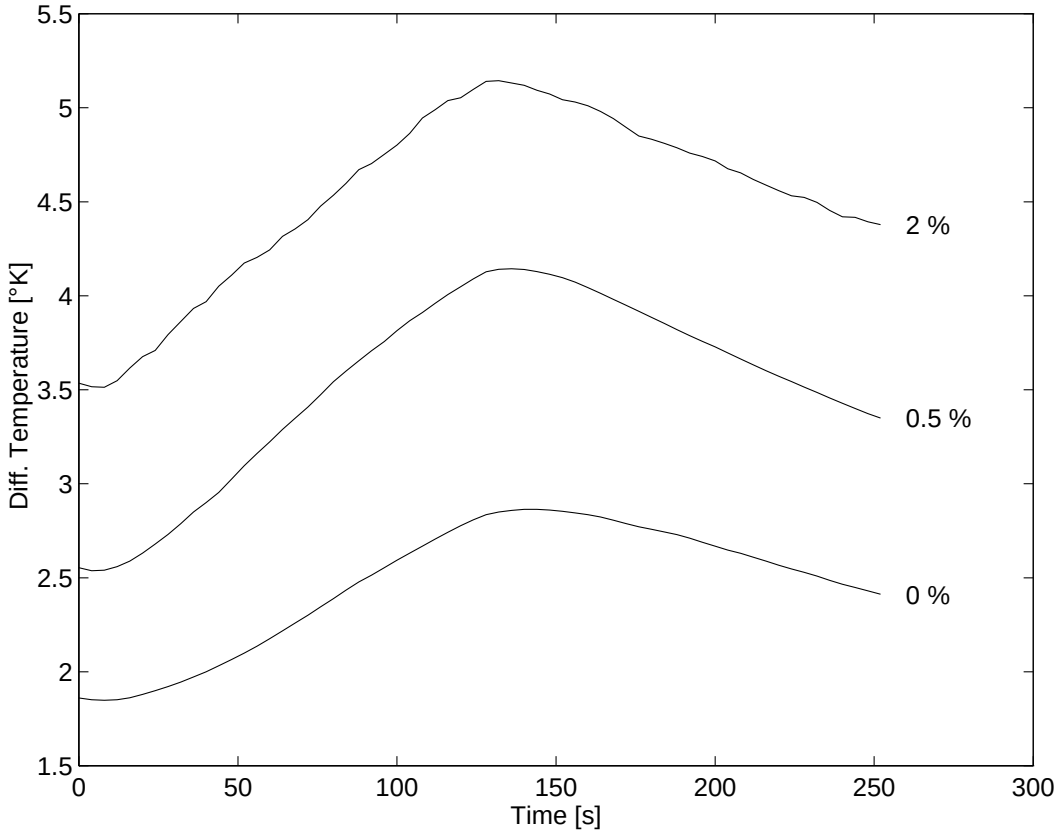



FIG. 3.6: courbes finales après liftrage

[80], à ne pas confondre avec le gain statique défini dans [20]

$\sigma Si$ est la variance de l'erreur de *simulation*, c.-à-d. la puissance du bruit obtenue en comparant les valeurs mesurées et simulées de la sortie.

Dans le processus de liftrage, les points du premier segment ont en général dû être rejetés parce que le cepstre obtenu était divergent, ce qui indique que le signal n'est pas à minimum de phase. Une explication possible est que le caractéristique de type diffusion est la plus marquée au début de la courbe, avant de céder le pas à un comportement de type exponentiel.

## 3.5 Discussion

Les modèles obtenus sur base des signaux liftrés présentent de meilleures performances que ceux provenant des signaux non-traités, bien que le nombre de points disponibles soit 4 fois plus petit, ce qui aurait pu augmenter la variance d'un facteur 2. Ce processus a également permis de réduire les artefacts. De plus, nous pouvons constater que les constantes de temps sont beaucoup plus proches chez les deux sujets pour un même niveau de modification.

Ceci tend à nous conforter dans l'idée que nous avons robustifié le processus d'identification : en effet, aucune des valeurs obtenues ne semble problématique, ce qui n'est pas le cas avec les signaux bruts. La différence de variance en ce qui concerne le gain DC peut s'expliquer par la réduction du nombre de points comme développé dans [80] et par le fait que, suite aux différences entre les structures des modèles, nous n'avons pas utilisé les mêmes opérateurs pour arriver au résultat.

## 3.6 Classification par coefficients cepstraux

Le premier coefficient cepstral tel que défini dans [20] peut également être utilisé : il n'a pas d'interprétation physique précise, mais reflète la « distance » entre les courbes. Le tableau 3.2 contient ce coefficient pour les deux jeux de courbes.

| Dose | Subject 1 | Subject 2 |
|---|---|---|
| 0 % | 0.54 | 0.62 |
| 0.5 % | 1.06 | 0.94 |
| 2 % | 1.15 | 1.26 |

TAB. 3.2: Distance cepstrale des courbes

Entre les deux sujets, les paramètres sont assez semblables pour un même degré de modification, tandis que les variations liées à la dose sont plus importantes, en particulier entre les niveaux 0 % et 0.5 %.

## 3.7 Discussion

Cette approche donne des paramètres plus robustes que ceux obtenus par une voie plus classique, car leur variance est systématiquement réduite. Ceci résulte d'une identification qui se focalise sur les segments où le modèle colle le mieux au signal, accompagnée d'une réduction du bruit et des artefacts, sans avoir à effectuer de pré-traitement sur le signal.

La plupart des paramètres que nous avons montrés semblent présenter une corrélation avec la dose de modification, même si certains d'entre eux semblent moins discriminants à ce propos.

Nous ne sommes pas arrivé à obtenir un moyen de calculer rigoureusement la valeur limite des conditions initiales. Nous estimons cependant que c'est une voie de recherche qu'il serait intéressant de poursuivre. Dans une identification de type "Output Error", un des problèmes cruciaux est de déterminer les conditions initiales : dans l'implémentation MatLab, on se contente de prédire au mieux la partie initiale du signal, cette approche est en fait particulièrement sensible à la présence de bruit dans cette partie.

Dans l'approche cepstrale, l'influence des conditions initiales se manifeste sur une partie importante du cepstre partiel, ce qui devrait mener à une identification plus robuste, d'autant plus que l'on doit répéter cette opération pour chaque segment : disposant de plusieurs jeux de conditions initiales, il sera possible de déterminer leur évolution, voire de repérer des valeurs plus bruitées, dont on pourrait ne pas tenir compte du tout.

Une dernière méthode de classification, basée sur une extension de la méthode de Prony, a été étudiée. Étant donné qu'elle s'est avérée la plus adéquate dans le cadre de cette problématique de classification, elle fera l'objet des deux chapitres suivants; nous exposerons d'abord les développements théoriques qui y mènent, et ensuite nous présenterons des résultats obtenus.

Chapitre 4

# Établissement de la méthode de Prony générique

Dans ce chapitre, nous commencerons par donner quelques rappels de notions fondamentales au niveau identification des processus, puis nous expliquerons une méthode que nous avons mise au point et appliquée pour obtenir un modèle de nos courbes, la méthode de Prony générique.

## 4.1 Introduction

D'une manière générale, la méthodologie d'*identification des processus* procède de la manière suivante : soit un système, excité par un signal d'entrée mesuré $\mathbf{u}(t)$, produisant une sortie observée $\mathbf{y}(t)$. Pour traiter numériquement ces données, il est nécessaire d'échantillonner ces deux grandeurs. L'intervalle entre deux mesures étant en général constant, on a $t = k\,\Delta T$, $k$ appartenant à $\mathcal{N}$, et la dimension temporelle (analogique) est généralement gommée.

On écrira donc $\mathbf{y}$ pour $\mathbf{y}(n\,\Delta T)$. Les vecteurs et matrices considérés par la suite auront autant de lignes que d'échantillons observés. Remarquons enfin que $y$ et $u$ sont des mesures, et peuvent à ce titre être entachés d'erreur, et ne pas représenter exactement les signaux réellement présents dans le système.

Le but de l'identification est de modéliser la relation entre ces deux grandeurs, ce qui implique de choisir une structure de modèle et de trouver, parmi les différentes réalisations possibles au niveau de cette structure, celle pour laquelle la sortie prédite s'approche le mieux de la sortie observée.

## 4.2 Identification des processus

### 4.2.1 Cadre de référence

D'une manière générale, la relation entrée-sortie d'un système linéaire, invariant dans le temps s'écrit

$$\mathbf{y}(t) = G(q)\,\mathbf{u}(t) + H(q)\,\mathbf{e}(t) \tag{4.1}$$

dans laquelle

$G(q)$ est la réponse impulsionnelle du système. Elle peut être exprimée sous forme d'une liste (finie) de coefficients (filtre FIR), ou sous forme d'un rapport de polynômes en $q$ (filtre IIR);

$\mathbf{e}(t)$ est une séquence de variables aléatoires, caractérisée statistiquement par ses moments d'ordre 1 et 2, qui représente le « bruit » présent dans le système;

$H(q)$ est un polynôme monique, modélisant la fonction de transfert du bruit. Celui-ci peut en effet ne pas suivre le même chemin que le signal principal.

### 4.2.2 Modélisation

Une procédure d'identification consiste à :

1. Expliciter (4.1) sous la forme d'une relation entre $y$ et $u$ du type

$$\mathbf{y} = \mathbf{A}(\theta_\mathbf{1})\mathbf{c} + \epsilon \tag{4.2}$$

dans laquelle:

**A** est une matrice de régression, construite soit sur base de fonctions de **y** et **u**, soit dépendant d'un vecteur de paramètres $\theta_\mathbf{1}$;

**c** est un vecteur de coefficients, tels que **y** s'exprime comme une combinaison linéaire des colonnes de **A**;

$\epsilon$ est un vecteur d'erreur, indiquant l'écart entre la valeur mesurée et la valeur calculée par le modèle. Ce vecteur représente la contribution de l'ensemble des sources d'erreur : bruit, erreur de modélisation, ... ;

$\theta = \begin{pmatrix} \theta_\mathbf{1} \\ \mathbf{c} \end{pmatrix}$ est un vecteur contenant tous les paramètres à déterminer.

L'équation (4.2) recouvre en fait trois problèmes différents :

(a) $\theta_\mathbf{1}$ est nul, $\mathbf{A}$ ne dépend que de $\mathbf{y}$ et $\mathbf{u}$, et $\mathbf{c} \equiv \theta$. Il s'agit alors d'un problème de régression linéaire.

(b) $\mathbf{c}$ est un vecteur contenant soit des valeurs connues d'avance, soit identiquement la constante 1 et $\mathbf{A}$ est fonction de $\theta$. Ce problème porte le nom de régression non-linéaire.

(c) $\mathbf{A}$ dépend de $\theta_\mathbf{1}$ et $\mathbf{c}$ est inconnu, ce qui constitue le cas le plus général. Dans ce cas, nous verrons qu'il est possible de séparer le problème en deux problèmes indépendants.

2. choisir la structure de $\mathbf{A}$, c.-à-d. les fonctions de base pour générer ses colonnes et la dimension de $\theta$.
   Le choix de la structure dépend de l'utilisation que l'on veut en faire : s'agit-il de prédire, de modéliser, d'identifier (mesurer indirectement) des paramètres, de concevoir un régulateur, ... ?
3. trouver la valeur des éléments de $\theta_\mathbf{1}$ et de $\mathbf{c}$, qui minimisent l'« erreur » du modèle.
4. réitérer la procédure, jusqu'à obtenir un bon compromis entre la valeur de l'erreur et la longueur du modèle. Pour cela, on définit une fonction de coût, qui pondère la valeur de l'erreur par une fonction croissante de la longueur.

### 4.2.3 Critère de performance

L'équation (4.2) contient deux jeux de paramètres à trouver, l'un étant non-linéaire ($\theta_1$) et l'autre linéaire ($\mathbf{c}$). Le critère de recherche est en général la minimisation de la norme de l'erreur

$$\epsilon_F(t|\theta) = L(q)\epsilon(t|\theta) \tag{4.3}$$

$$V_N(\theta,\mathbf{Z}^N) = \frac{1}{N}\sum_{t=1}^{N} l(\epsilon_F(t|\theta)) \tag{4.4}$$

dans lesquelles

$L(q)$ est un filtre linéaire stable, permettant d'effectuer un traitement spécifique sur le bruit;

$\mathbf{Z}^N$ est une matrice contenant les $N$ échantillons de $y(t)$ et $u(t)$;

$l(\cdot)$ est une fonction scalaire à valeur positive. Pour les développements ultérieurs, nous demanderons également qu'elle soit différentiable. Un choix classique est $l(x) = \frac{1}{2}x$ et $L(q) = 1$; ce qui donne :

$$e = \frac{1}{2}\left\|\epsilon\right\|_2 = \frac{1}{2}\,\epsilon^T \cdot \epsilon. \tag{4.5}$$

Les paramètres $\theta_1$ et $\mathbf{c}$ peuvent être liés, néanmoins une procédure de recherche explorera indépendamment et successivement un jeu, puis l'autre. Pour effectuer cette recherche, il est nécessaire de disposer d'une expression de la dérivée du critère de performance par rapport aux paramètres. Par la suite, cela permettra de calculer la matrice de covariance, et d'en tirer les variances individuelles.

### 4.2.4 Détermination des paramètres linéaires

La détermination de $\mathbf{c}$, à $\mathbf{A}$ fixe, est assez aisée :

$$\epsilon = \mathbf{y} - \mathbf{A}(\theta)\mathbf{c} \tag{4.6}$$

À ce stade, deux options s'offrent à nous :

1. considérer que seul $\mathbf{y}$ est bruité, cela mène à la solution classique découverte par Gauss des moindres carrés : considérant l'équation (4.5), le « meilleur » modèle est celui qui assure un minimum de ce critère d'erreur, ou, ce qui est équivalent, qui annule la dérivée (gradient) de ce critère par rapport au vecteur de paramètres. Cette valeur est donnée par

$$\mathbf{c} = (\mathbf{A}^T \cdot \mathbf{A})^{-1}\mathbf{A}^T \cdot \mathbf{y} \tag{4.7}$$

   L'avantage de cette formulation est de fournir une relation explicite entre les différents éléments. Mais, au cas où la matrice $\mathbf{A}$ serait elle-même bruitée, ou mal conditionnée, $c$ peut être affectée d'un biais. Cela peut se produire dans les modèles auto-regressifs, où $\mathbf{A}$ contient plusieurs versions décalées de $\mathbf{y}$. L'approche suivante permet de prendre en compte ces difficultés :
2. Dans la méthode des moindres carrés totaux (Total Least Squares) [76], l'équation est exprimée sous la forme

$$\big(\mathbf{A}\,;\quad \mathbf{y}\big) \cdot \big(\mathbf{c}^T\,;\quad -1\big)^T \approx 0 \tag{4.8}$$

   La résolution de ce problème se base sur la décomposition en valeurs singulières de la matrice $\big(\mathbf{A}\,;\quad \mathbf{y}\big)$, cette matrice composite ayant $n+1$ colonnes. Appelant $\sigma_{n+1}$ la plus petite valeur singulière, on aboutit à l'expression analytique

$$\hat{\mathbf{c}} = \big(\mathbf{A}^T \cdot \mathbf{A} - \sigma_{n+1}^2\,\mathbf{I}\big)^{-1}\mathbf{A}^T \cdot \mathbf{y} \tag{4.9}$$

   où $\mathbf{I}$ est la matrice unité de taille $n$. La formulation est un peu plus complexe au cas ou $\sigma_{n+1}^2$ serait nul.
   L'idée est de rechercher une base de l'espace généré par les colonnes de $\big(\mathbf{A}\,;\quad \mathbf{y}\big)$ et de considérer que la colonne liée à la valeur singulière la plus petite ne contient que du bruit. La solution $\mathbf{c}$ est celle qui donne la meilleure approximation de $\mathbf{y}$ par l'espace généré par les colonnes restantes.

## 4.3 Modélisation par somme d'exponentielles

Il est de nombreuses branches de l'art de l'ingénieur où l'on souhaite modéliser la sortie d'un système comme une somme d'exponentielles : en spectroscopie, en physique nucléaire, en chimie, ... Pour des raisons de stabilité, on se limite à des exponentielles décroissantes. Telle que présentée classiquement, la méthode de Prony suppose que les données ne comprennent ni décalage ni dérive. Nous entendons par décalage une erreur constante et par dérive une erreur linéairement proportionnelle au temps.

Les textes sur l'identification suggèrent comme pré-traitement des données de rendre le signal de moyenne nulle et de ne considérer que l'écart par rapport à une droite de régression. Étant donné qu'une exponentielle contient un terme linéaire, d'autant plus important que la décroissance de celle-ci est très lente par rapport à la fenêtre d'observation, nous préférons considérer que la détermination de ces deux grandeurs fait partie intrinsèque du problème.

Le point de départ est de supposer que la sortie peut se modéliser comme la somme de $M$ exponentielles :

$$\mathbf{y}_i = c_1\, e^{\theta_1 x_i} + c_2\, e^{\theta_2 x_i} + \cdots + c_M\, e^{\theta_M x_i} + c_{M+1} + c_{M+2}\, x_i \tag{4.10}$$

$$x_i = i\, T_s\,; \quad i = 0,\, \ldots,\, N-1 \tag{4.11}$$

où

$c_1,\, \ldots,\, c_M$ représentent l'amplitude de chaque exponentielle;
$\theta_1,\, \ldots,\, \theta_M$ conditionnent la vitesse de décroissance;
$c_{M+1}$ est l'erreur constante ou le décalage par rapport à l'origine;
$c_{M+2}$ est un terme de dérive linéaire dans le temps.

Par comparaison avec (4.2), nous avons l'identification

$$\big[\mathbf{A}\big]_{M+1,\, j} = 1 \tag{4.12}$$

$$\big[\mathbf{A}\big]_{M+2,\, j} = j\, T_s \tag{4.13}$$

$$\big[\mathbf{A}\big]_{k,\, ,j} = e^{\theta_k\; j\; T_s} \quad 1 \leq k \leq M. \tag{4.14}$$

La matrice $\mathbf{A}$ est non-linéaire par rapport à $\theta_{\mathbf{1}}$, nous sommes donc dans le cas général, impliquant la détermination de paramètres linéaires et non-linéaires.

### 4.3.1 Méthode non-linéaire

La résolution passe donc par les étapes suivantes :

1. calcul de $\mathbf{c}$ par (4.7);
2. calcul de $\mathbf{A}$ par minimisation de (4.2).

Pour la seconde étape, calculons la valeur du gradient de l'erreur

$$\hat{y} = \mathbf{A}(\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}^T \cdot \mathbf{y}) \tag{4.15}$$

Nous voyons que, de par l'existence d'une relation linéaire entre $\mathbf{A}$ et $\mathbf{c}$, il est possible d'obtenir une expression dans laquelle $\mathbf{c}$ n'apparaît plus. Pour différentier cette expression, nous utiliserons le résultat suivant, dans lequel $\mathbf{M}$ représente une matrice carrée

$$\frac{\partial \mathbf{M}^{-1}}{\partial x} = -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial x}\mathbf{M}^{-1} \tag{4.16}$$

Nous avons donc

$$\begin{aligned}\frac{\partial \hat{y}}{\partial \theta_i} = &\frac{\partial \mathbf{A}}{\partial \theta_i}\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}^T \cdot \mathbf{y} \\ &+ \mathbf{A}\left[-\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\left[\frac{\partial \mathbf{A}^T}{\partial \theta_\mathbf{i}}\mathbf{A} + \mathbf{A}^T\frac{\partial \mathbf{A}}{\partial \theta_\mathbf{i}}\right]\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}^T \cdot \mathbf{y}\right. \\ &\left. + \left[\mathbf{A}^T\mathbf{A}\right]^{-1}\frac{\partial \mathbf{A}^T}{\partial \theta_\mathbf{i}}\mathbf{y}\right].\end{aligned} \tag{4.17}$$

Le second terme peut se simplifier en

$$\mathbf{A}\left[\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\left[\frac{\partial \mathbf{A}^T}{\partial \theta_\mathbf{i}}\underbrace{(\mathbf{y} - \mathbf{A}\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}^T \cdot \mathbf{y})}_{\epsilon} - \mathbf{A}^T\frac{\partial \mathbf{A}}{\partial \theta_\mathbf{i}}\underbrace{\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}^T \cdot \mathbf{y}}_{\mathbf{c}}\right]\right] \tag{4.18}$$

ce qui mène au résultat

$$\frac{\partial \hat{y}}{\partial \theta_i} = \left[I - \mathbf{A}\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\right]\frac{\partial \mathbf{A}}{\partial \theta_i}\mathbf{c} + \mathbf{A}\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\frac{\partial \mathbf{A}^T}{\partial \theta_\mathbf{i}}\epsilon. \tag{4.19}$$

Cette méthode n'est guère simple et, comme pour toute méthode itérative, nécessite de prendre un certain nombre de précautions au niveau du point de départ pour en assurer la convergence. Elle peut par contrer s'avérer utile pour raffiner une solution si l'on a des doutes quant à sa validité.

### 4.3.2 Méthode de Prony

Le problème de base peut en fait être décomposé en deux problèmes linéaires. Prony a montré que l'on peut trouver une relation auto-régressive entre les valeurs de $y(t)$, $y(t-1)$, $\cdots$, $y(t-M)$. Pour cela, commençons par poser

$$z_m = e^{\theta_m T_s} \tag{4.20}$$

ce qui permet d'écrire

$$\mathbf{y}_i \;\; = c_1 \; z_1^i + c_2 \; z_2^i + \cdots + c_M \; z_M^i + c_{M+1} + c_{M+2} \; i \, T_s \tag{4.21}$$

Une relation auto-régressive implique qu'il existe un jeu de coefficients $\alpha_k$, avec $\alpha_0 = 1$, tels que

$$\sum_{k=0}^{M} \alpha_k \; y_{n-k} \simeq 0. \tag{4.22}$$

Récrivons cela en développant l'expression de $\mathbf{y}$, et en faisant usage de l'équation (4.20) :

$$\sum_{k=0}^{M} \alpha_k \; \big( \sum_{m=1}^{M} c_m \; z_m^{n-k} \;\; + c_{M+1} + c_{M+2} \; (n-k) \; T_s \big) = \qquad 0 \tag{4.23}$$

permutant les deux sommations, il vient

$$\sum_{m=1}^{M} c_m \; z_m^{n-M} \; \big( \sum_{k=0}^{M} \alpha_k \; z_m^{M-k} \big) + \sum_{k=0}^{M} \alpha_k \; \big( c_{M+1} + c_{M+2} \; (n-k) \; T_s \big) = \qquad 0 \tag{4.24}$$

Nous pouvons satisfaire cette équation si

- chacune des sommes $\sum_{k=0}^{M} \alpha_k \; z_m^{M-k}$ est nulle. De façon générique, nous avons une équation de degré $M$ qui peut s'écrire

$$F(z) = z^M + \sum_{k=1}^{M} \alpha_k \; z^{M-k} = \prod_{m=1}^{M} (z - z_m) \tag{4.25}$$

  admettant $M$ racines qui sont les $z_m$ recherchés.
- nous identifions deux termes supplémentaires :
  - $c_{M+1} \;\; \sum_{k=0}^{M} \alpha_k$ (terme constant)
  - $c_{M+2} \;\; \sum_{k=0}^{M} \alpha_k$ (dérive proportionnelle au temps)

La procédure complète comprend donc les étapes suivantes :

1. chercher les paramètres $\alpha_k$ par

$$\big( z^{-1} \, \mathbf{y} \quad \ldots \quad z^{-k} \, \mathbf{y} \quad 1 \quad n \big) \; \big( \alpha_1 \quad \ldots \quad \alpha_k \quad x_0 \quad x_1 \big)^T = \qquad -\mathbf{y} \tag{4.26}$$

   D'une façon générale, ce système d'équations sera surdéterminé et résolu par une approche de type moindres carrés ou moindres carrés totaux.
2. chercher les racines $z_m$ de (4.25).

3. chercher les paramètres linéaires par

$$\begin{pmatrix} z_1^n & \dots & z_M^n & 1 & n \end{pmatrix} \begin{pmatrix} c_1 & \dots & c_M & c_{M+1} & c_{M+2} \end{pmatrix}^T = \qquad \mathbf{y} \tag{4.27}$$

Ici encore, le système d'équation sera surdéterminé.

Ouvrons ici une parenthèse en ce qui concerne l'identification du terme de dérive. Si une des constantes de temps est grande par rapport à l'intervalle d'observation, l'exponentielle présentera essentiellement sa partie linéaire. Il est à craindre qu'il y ait une dépendance linéaire marquée entre deux colonnes des matrices de régression, ce qui peut entraîner une susceptibilité accrue au bruit.

Considérant que les termes de décalage et de dérive ne font pas partie du signal, la transformée en $\mathcal{Z}$ de (4.10) résultera en

$$\sum_{i=1}^{M} \frac{c_i}{1 + z_i z^{-1}} \tag{4.28}$$

Partant de la relation entre la transformée en $\mathcal{Z}$ et le spectre, il est possible d'obtenir le spectre d'un signal en

1. calculant la transformée en $\mathcal{Z}$ via la méthode de Prony
2. calculant la valeur de de $X(z)$ en $z = e^{j\,\omega\,Ts}$

Il est donc possible de décrire $\mathbf{y}$ tel qu'exprimé dans (4.10) par une relation purement auto-régressive entre $y(t)$, $y(t-1)$, $\dots$, $y(t-k)$ dont les racines permettront de trouver les termes $a_i$. Dans une seconde étape, ayant généré la matrice $\mathbf{A}$, nous aurons un second problème linéaire à résoudre pour trouver les termes $c_i$.

### 4.3.3 Comparaison

D'un point de vue numérique, la seconde solution est infiniment plus simple que la première, puisqu'elle se réduit à deux problèmes de régression linéaire. Ces deux méthodes sont-elles similaires du point de vue des résultats obtenus? Non. Les différences majeures entre elles sont :

- De par la construction de la matrice $\mathbf{A}$, la méthode de Prony présente une susceptibilité au bruit contenu dans $\mathbf{y}$. Si ce bruit contient des corrélations à court terme, cela influencera les termes $\alpha_i$ si l'on ne prend pas la précaution d'utiliser la méthode des moindres carrés totaux pour la première étape.
- Dans la méthode non-linéaire, la construction de la matrice de régression se fait par recherche progressive, et cette matrice n'est pas supposée contenir du bruit.

## 4.4 Extension de la méthode de Prony

### 4.4.1 Comment la nommer ?

Le terme « généralisé » apparaît chez plusieurs auteurs et recouvre une série d'améliorations diverses. Par exemple, l'introduction de contraintes supplémentaires au niveau de la matrice de régression permet à celle-ci de devenir circulante. Au niveau des fréquences pour un modèle sinusoïdal, on peut imposer que les solutions doivent être multiples l'une de l'autre pour représenter des harmoniques d'un signal de base.

Étant donné que nous avons enlevé une restriction inhérente à la méthode de départ, nous proposons de nommer cette approche « Méthode de Prony générique », puisque la solution est dans ce cas plus générale. Voyons à présent plus en détail cette extension.

### 4.4.2 Identification de réponse indicielle

Telle que présentée classiquement, la méthode de Prony permet d'identifier UN jeu d'exponentielles, caractérisé par ses coefficients $c_i$ et $\theta_i$. Nous allons montrer qu'il s'agit en fait de la réponse indicielle à l'application d'un échelon en entrée du système. D'une manière tout à fait générale, le fonctionnement d'un système linéaire peut être décrit par

$$\dot{x} = a\,x + b\,u \tag{4.29}$$
$$y = c\,x + d\,u + e \tag{4.30}$$

En Laplace, nous avons

$$s\,X(s) - X_0 = A\,X(s) + B\,U(s) \tag{4.31}$$
$$Y(s) = C\,X(s) + D\,U(s) + E(s) \tag{4.32}$$

que nous pouvons résoudre comme

$$(s\,I - A)X(s) = B\,U(s) + X_0 \tag{4.33}$$
$$Y(s) = (C(s\,I - A)^{-1}B + D)\,U(s) + C(s\,I - A)^{-1}X_0 + E(s) \tag{4.34}$$

Le déterminant de $(s\,I - A)$ est un polynôme en $s$ dont les racines sont les pôles du système. Ces pôles sont communs aux fonctions de transfert de $U(s)$ et de $X_0$. À chacun de ces pôles sera associée une exponentielle via la transformée inverse de Laplace. Si le système est stable, chacune des exponentielles convergera vers zéro à l'équilibre.

La réponse indicielle de ce système, obtenue en considérant $U(s) = K/s$, sera du type

$$y(t) = (\sum_{i=1}^{M} c_i \ \exp(\theta_i t) + c_{M+1}) \ K + \sum_{i=1}^{M} d_i \ \exp(\theta_i t) \tag{4.35}$$

$$y(t) = \sum_{i=1}^{M} (c_i \ K + d_i) \ \exp(\theta_i t) + c_{M+1} K \tag{4.36}$$

Étudions plus en détails les différents termes :

- les termes $c_i$ résultent de la transformation en fractions simples du facteur de $U(s)$ dans (4.34). Le terme $c_{M+1}$ provient de l'introduction du facteur $1/s$;
- les $\theta_i$ sont associés aux pôles du système;
- les $d_i$ proviennent de la décomposition en fractions simples du facteur de $X_0$, multiplié par la valeur correspondante de $X_0$.

Par comparaison de 4.10 et de 4.36, nous constatons que les coefficients $c_i$ des exponentielles identifiés par la méthode de Prony résultent d'une combinaison linéaire de la commande et des conditions initiales. Une séparation n'est possible que si :

- il n'y a pas de commande à proprement parler dans le système. Cela implique que nous observions l'évolution du système sous l'effet de ses conditions initiales, et que l'état de régime soit nul.
- le système démarre d'une situation d'équilibre correspondant à la disparition de toutes les conditions initiales, c.-à-d. que nous avons attendu, après l'enclenchement des dispositifs, que le système atteigne un état de régime nul. De plus, la seule commande permise est l'échelon.

Le cas général, correspondant à l'existence d'une commande plus complexe *et/ou* de conditions initiales non-nulles, ne rentre pas dans ces deux catégories. Nous allons à présent montrer qu'il est néanmoins possible, et sous quelles conditions, de séparer l'influence de la commande de celle des conditions initiales.

### 4.4.3 Signal constant par morceaux

Soit le signal défini par

$$\begin{aligned} u(t)_{0 \leq t < \tau_1} &= u_0 \\ u(t)_{\tau_1 \leq t < \tau_2} &= u_1 \\ &\dots \end{aligned} \tag{4.37}$$

La transformée de Laplace de ce signal vaut

$$U(s) = \frac{u_0}{s} + \frac{u_1 - u_0}{s}\exp(-\tau_1) + \frac{u_2 - u_1}{s}\exp(-\tau_2) + \dots \tag{4.38}$$

La réponse sera constituée d'une combinaison linéaire des réponses indicielles :

$$\begin{aligned}
y(t)_{0\le t<\tau_1} &= (\sum_{i=1}^{M} c_i \ \exp(\theta_i t) + c_{M+1}) \ u_0 + \sum_{i=1}^{M} d_i \ \exp(\theta_i t) \\
y(t)_{\tau_1\le t<\tau_2} &= (\sum_{i=1}^{M} c_i \ \exp(\theta_i t) + c_{M+1}) \ u_0 + \sum_{i=1}^{M} d_i \ \exp(\theta_i t) \\
&\quad + (\sum_{i=1}^{M} c_i \ \exp(\theta_i (t-\tau_1)) + c_{M+1})(u_1 - u_0) \\
y(t)_{\tau_2\le t<\tau_3} &= (\sum_{i=1}^{M} c_i \ \exp(\theta_i t) + c_{M+1}) \ u_0 + \sum_{i=1}^{M} d_i \ \exp(\theta_i t) \\
&\quad + (\sum_{i=1}^{M} c_i \ \exp(\theta_i (t-\tau_1)) + c_{M+1})(u_1 - u_0) \\
&\quad + (\sum_{i=1}^{M} c_i \ \exp(\theta_i (t-\tau_2)) + c_{M+1})(u_2 - u_1) \\
&\dots
\end{aligned} \tag{4.39}$$

Si nous découpons ce signal en tronçons où l'amplitude de la commande reste constante, nous pourrions appliquer la méthode de Prony individuellement sur chacun d'eux. Dans ce cas, nous obtiendrions un jeu $c(k,i)$ ainsi qu'un jeu $theta_(k,i)$ pour le segment d'indice $k$.

Si le système étudié est invariant, nous devrions en théorie obtenir le même jeu de paramètres $theta_(k,i)$ puisque les pôles du système n'ont pas changé, tandis que les coefficients $c(k,i)$ varient sous l'effet de la disparition progressive des conditions initiales. L'inconvénient de cette approche est que, de par la présence de bruit, les $\theta_(k,i)$ pourraient varier d'un segment à l'autre : comment caractériser globalement le système ? La relation entre les données et les solutions étant non-linéaire, il est impossible d'appliquer des méthodes de réduction du bruit par filtrage linéaire, comme par exemple en prenant la moyenne des solutions.

Nous avons en fait besoin de travailler sur la totalité du signal, et d'obtenir un jeux de coefficients globaux. Nous allons à présent montrer comment arriver à obtenir ces coefficients globaux.

Sous forme générique, la réponse durant le segment n°S vaudra

$$y(t)_S = \sum_{i=1}^{M} \Big( c_i \big[ \sum_{j=1}^{S} (\exp(-\theta_i \tau_j)(u_j - u_{j-1}) + u_0 \big] + d_i \Big) \exp(\theta_i t) + c_{M+1} u_S \tag{4.40}$$

En outre, le jeu de coefficients $c_i$ répond à une condition supplémentaire. Pour le montrer, prenons la transformée en $\mathcal{Z}$ de la relation entrée-sortie, en l'absence de conditions initiales :

$$Y(z) = \frac{N(z)}{D(z)} U(z) = \sum_{i=1}^{N} \frac{c_i}{1 - p_i z^{-1}} \tag{4.41}$$

ou

$N(z)$ est le numérateur de la fonction de transfert, que nous allons écrire sous la forme d'un gain $G_0$ que multiplie un polynôme dont le premier coefficient est unitaire;

$D(z)$ est le dénominateur de la fonction de transfert, il s'agit d'un polynôme monique;

$U(z)$ est la commande appliquée au système;

$N$ est le nombre total de pôles;

$p_i$ représente l'ensemble des pôles, appartenant soit à la fonction de transfert, soit à la commande;

$c_i$ provient du développement en fractions simples de $Y(z)$, et est associé au pôle de même indice.

En fait, nous avons utilisé l'équivalence suivante :

$$y(t) = c_i \ \exp(\theta_i \, t) \Leftrightarrow Y(z) = \frac{c_i}{1 - \exp(\theta_i) \ z^{-1}} \tag{4.42}$$

Deux cas sont à prendre en considération : les simulations *analogiques* et *digitales*. Par la suite, nous établirons une différence entre ces deux cas via une variable $n_k$, qui représente le délai implicite entre $y(t)$ et $u(t)$.

En l'absence de conditions initiales, la première valeur de $y(t)$ doit être nulle dans le cas analogique et doit valoir $G_0 u_0$ dans le cas digital. Cette valeur est donnée par le terme indépendant de $z$ dans l'équation 4.41, ce qui correspond à $\sum_{i=1}^{N} c_i$ dans le terme de droite. Voyons comment satisfaire cette contrainte.

Dans le cas analogique, nous devons en fait retarder la commande d'une unité ou, ce qui est équivalent, prendre $N(z) = z^{-1} + b_1 z^{-2} + \ldots$ Cette condition implique donc que la somme $\sum_{i=1}^{N} c_i$ doit être nulle, et que $n_k = 1$.

Dans le cas digital, cette somme doit valoir $G_0 u_0$, ce qui implique d'identifier séparément ce terme $G_0$. Il n'y a pas de délai implicite : $n_k = 0$.

Nous pouvons écrire cette condition comme suit :

$$c_{M+1} = (nk-1)\,G_0 - \sum_{i=1}^{M} c_i \tag{4.43}$$

### 4.4.4 Algorithme de Prony générique

Nous allons à présent montrer comment évaluer globalement les termes $\theta_i$, $c_i$ et $d_i$, ce qui n'était pas possible avec l'approche classique. Considérant le segment n°S du signal, c.-à-d. une zone où $u$ est constant, nous pouvons écrire

$$\sum_{k=0}^{M} \alpha_k\, y_{S,n-k} = \sum_{k=0}^{M} \alpha_k \Bigg( \sum_{m=1}^{M} \Big[ c_m \big( \sum_{j=1}^{S} \exp\left(-\theta_m\, \tau_j\right) (u_j - u_{j-1}) + u_0 \big) + d_m \Big]\, z_m^{n-k} + c_{M+1} u_S + d_{M+1} + d_{M+2}\,(n-k)\,T_s \Bigg) \simeq 0 \tag{4.44}$$

que nous pouvons factoriser en

$$\sum_{m=1}^{M} \left[ c_m \big( \sum_{j=1}^{S} \exp\left(\theta_m\, \tau_j\right) (u_j - u_{j-1}) + u_0 \big) + d_m \right]\, z_m^{n-M} \left( \sum_{k=0}^{M} \alpha_k z_m^{M-k} \right) + \sum_{k=0}^{M} \alpha_k \left( c_{M+1} u_S + d_{M+1} + d_{M+2}\,(n-k)\,T_s \right) \tag{4.45}$$

Analysons plus en détail ce que recouvre cette équation, à la lumière de l'équation 4.24. Dans le premier terme de la somme, le contenu entre crochet varie de segment en segment, mais la somme des pôles est un terme constant si nous supposons le système invariant.

Par les mêmes arguments que ceux évoqués lors de la discussion sur la méthode de Prony « simple », nous voyons que cette somme est annulable si :

- les $z_m$ sont les racines de (4.25);
- nous identifions, outre un terme constant et un autre proportionnel au temps, la contribution de $u$.

Pour le calcul des pôles, nous pouvons donc combiner les matrices de régression associées aux segments individuels pour en arriver à seule matrice de portée globale. De cette façon, nous obtiendrons un seul jeu de pôles, correspondant à une optimisation *globale* au niveau de la *totalité* du signal.

En ce qui concerne la seconde matrice de régression, nous allons la construire avec des colonnes qui dépendent du vecteur d'entrée pour trouver les coefficients $c_i$ et d'autres qui sont constantes pour trouver les $d_i$. Pour que ce système d'équation (surdéterminé) soit résoluble, il convient que les différentes colonnes de la matrice de régression sont linéairement indépendantes.

Ceci nous donne la condition de séparabilité entre les $c_i$ et les $d_i$ : *Le signal u doit donc contenir au moins une transition*, pour que son influence soit distinguable de celle d'un signal constant.

Le signal le plus élémentaire contient donc une commande possédant deux niveaux distincts, chaque niveau durant $M+1$ point pour disposer au minimum d'une équation de type 4.44 par segment.

La procédure de Prony générique consiste donc à

1. chercher les paramètres $\alpha_k$ par (l'indice de $y$ et de $u$ fait référence au numéro du segment)

$$\begin{pmatrix} z^{-1}\mathbf{y_0} & \dots & z^{-M}\mathbf{y_0} & \mathbf{u_0} & \mathbf{1} & \mathbf{n} \\ z^{-1}\mathbf{y_1} & \dots & z^{-M}\mathbf{y_1} & \mathbf{u_1} & \mathbf{1} & \mathbf{n} \\ \vdots & & & & & \\ z^{-1}\mathbf{y_S} & \dots & z^{-M}\mathbf{y_S} & \mathbf{u_S} & \mathbf{1} & \mathbf{n} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_M \\ x_0 \\ x_1 \\ x_2 \end{pmatrix} = - \begin{pmatrix} \mathbf{y_0} \\ \mathbf{y_1} \\ \vdots \\ \mathbf{y_S} \end{pmatrix} \tag{4.46}$$

   Les différentes matrices sont construites comme suit :

   $\mathbf{y_s}$,$z^{-m}\mathbf{y_s}$ sont des vecteurs contenant les échantillons associés au segment $s$. Chaque vecteur comprend $M$ points de moins que la longueur du segment considéré;

   $\mathbf{u_s}$ contient la valeur constante de la commande sur le segment considéré;

   $\mathbf{1}$ est une colonne contenant identiquement la valeur « 1 », elle sert à identifier le décalage;

   $\mathbf{n}$ est un vecteur dont chaque élément représente l'indice temporel. Au sein d'un même segment, deux éléments consécutifs diffèrent d'une unité tandis qu'en passant d'un segment au suivant, la différence vaut $M$.

   Cette première étape fournit d'une part les termes $\alpha_i$ de la relation auto-régressive ainsi que des termes $x_i$. Le terme $x_0$ représente la pondération de $\mathbf{u}$, il s'agit en fait du gain $G_0$ dont nous avons besoin dans le cas digital. Les grandeurs $x_1$ et $x_2$ ne sont pas exploitables telles quelles parce que cette matrice de régression est construite sur base de versions décalées de $\mathbf{y}$ et non pas des fonctions de base.
2. chercher les racines $z_m$ de

$$z^M + \sum_{k=1}^{M} \alpha_k \; z^{M-k} = 0 \tag{4.47}$$

3. Rechercher les paramètres linéaires. Pour calculer la matrice de régression associée, récrivons $\mathbf{y}$ sous forme discrète : $t$ devient $k\,T_s$; $\tau_i = \delta_i\,T_s$. Réutilisant (4.20), il vient

$$\begin{aligned}\mathbf{y}_k = \sum_{i=1}^{M} \big(c_i\big[\sum_{j=1}^{S} z_i^{k-\delta_j}(u_j - u_{j-1}) + u_0\ z_i^k\big] + d_i\ z_i^k\big) \\ + c_{M+1}\ u_k\end{aligned} \tag{4.48}$$

Le terme $z_i^{k-\delta_j}$ n'est à prendre en considération que si $k - \delta_j \geq 0$.
La matrice de régression peut donc s'écrire

$$\begin{pmatrix}\sum_{j=1}^{S} z_m^{n-\tau_j}(u_j - u_{j-1}) + u_0\, z_m^n & u_n & z_m^n & 1 & n\end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_M \\ c_{M+1} \\ d_1 \\ \vdots \\ d_M \\ d_{M+1} \\ d_{M+2}\end{pmatrix} = \mathbf{y} \tag{4.49}$$

Sous l'influence du bruit, la condition (4.43) ne sera satisfaite que si nous l'intégrons directement dans la matrice de régression. La forme définitive sera donc

$$\begin{pmatrix}\sum_{j=1}^{S} z_m^{n-\delta_j}(u_j - u_{j-1}) + u_0\, z_m^n - u_n & z_m^n & 1 & n\end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_M \\ d_1 \\ \vdots \\ d_M \\ d_{M+1} \\ d_{M+2}\end{pmatrix} = \mathbf{y}' \tag{4.50}$$

avec

$$\mathbf{y}' = \mathbf{y} + (n_k - 1)\, G_0\, \mathbf{u} \tag{4.51}$$

Dans le membre de gauche de l'équation 4.50, le bruit résulte des erreurs sur les $z_i$. À droite, $\mathbf{y}$ contient du bruit de par la modélisation, avec une contribution éventuelle apportée via le terme correctif.

À première vue, la complexité numérique semble très élevée, puisque chaque élément associé à un des termes $c_i$ requiert de calculer $S$ fois un terme $z_m^{n-\delta_j}$. En pratique, lors du calcul de l'élément d'indices $(n,\ m)$, la valeur $z_m^n$ pourra être réutilisée pour tout élément $(n',\ m)$ tel que $n + \delta_i = n'$. Il s'agit en fait d'une matrice circulante.

### 4.4.5 Réponse indicielle et gain statique

Ayant obtenu les termes $c_i$ et $d_i$ après avoir observé la réponse à un signal $u$ contenant une transition, nous pouvons calculer la valeur du gain statique pour le système. Celui-ci étant soumis à un échelon d'amplitude constante, les seuls constituants subsistant lorsque $t \to \infty$ sont $c_{M+1}$ ainsi que les termes de correction. Si nous considérons uniquement une correction constante, le gain statique vaut donc $K_s = c_{M+1} - d_{M+1}$.

#### Signal périodique : calcul de la forme d'onde en régime

Le contexte qui a nécessité le développement de cette méthode est la prédiction de la forme d'onde en situation d'équilibre, dans un système où les constantes de temps sont trop grandes par rapport à une durée acceptable de prise de mesures.

Nous allons à présent montrer comment calculer la forme d'onde en régime à partir de la méthode de Prony générique. Considérons pour cela un signal d'entrée périodique, dont la première phase dure $P_1$ échantillons, et la seconde $P_2$ :

$$\begin{aligned} u(n) &= u_0 \qquad \text{mod } (n, P1 + P2) < P1 \\ u(n) &= u_1 \qquad \text{mod } (n, P1 + P2) \geq P1 \end{aligned} \tag{4.52}$$

À partir d'ici, il y deux façons d'aborder le problème. Reprenons l'équation 4.40, et considérons un segment $S$ commençant en $\tau_S$, suffisamment loin de l'origine pour que les conditions initiales soient éteintes. Voyons comment va se comporter la somme $\sum_{j=1}^{S} \exp(\theta_i(t-\tau_j))$. Le dernier terme vaudra $u_j - u_{j-1}$, le second $\exp(\theta_i(\tau_j - \tau_{j-1}))(-u_j + u_{j-1})$, et ainsi de suite. Cette série tend vers zéro et sa somme possède une expression analytique simple pour un signal d'entrée périodique. Par rapport à l'expression de départ, nous avons simplement associé à chacun des termes $c_i$ une commande « de régime », ce qui implique de devoir recalculer une matrice de régression pour prédire la forme d'onde en régime.

Pour éviter ce calcul supplémentaire, nous avons essayé de résoudre le problème suivant : est-il possible de trouver des conditions initiales qui placent directement le système dans sa situation de régime ?

La réponse est affirmative, et l'avantage par rapport à la situation précédente est que, de par le calcul des coefficients de Prony, nous avons déjà calculé la matrice de régression adéquate.

Considérons pour cela l'expression de $y(t)$ telle qu'elle apparaît dans le groupe d'équations 4.39. Nous serons dans une situation périodique si le premier échantillon de la troisième période, commençant en $t = \tau_2$, est identique au premier échantillon de la première période, qui démarre à $t = 0$.

La valeur initiale du troisième cycle est donnée par

$$\begin{aligned} y(P1+P2) = &\sum_{i=1}^{M}(c_i\ u_0 + d_i)z_i^{P1+P2} + c_{M+1}u_1 \\ &+ \sum_{i=1}^{M} c_i\ z_i^{P2}\ (u_1 - u_0) \\ &+ \sum_{i=1}^{M} c_i\ (u_0 - u_1) + c_{M+1}\ (u_0 - u_1) \end{aligned} \tag{4.53}$$

tandis que nous avons, pour le premier cycle

$$y(0) = \sum_{i=1}^{M}(c_i\ u_0 + d_i) + c_{M+1}\ u_0 \tag{4.54}$$

Le régime sera atteint pour des conditions initiales assurant l'égalité de ces expressions, ce qui nous mène à l'égalité, après simplifications :

$$\sum_{i=1}^{M}\left[c_i\big(u_0\,(z_i^{P1+P2} - z_i^{P2}) + u_1(z_i^{P2} - 1)\big) + d_i\,z_i^{P1+P2}\right] = \sum_{i=1}^{M} d_i \tag{4.55}$$

La façon naturelle de satisfaire la condition de périodicité est d'assurer cette égalité membre-à-membre au niveau des termes associés aux mêmes $z_i$, c.-à-d. de rechercher une relation explicite entre $c_i$ et $d_i$. Il vient

$$d_i = \frac{c_i\left[u_0 z_i^{P2}\,(z_i^{P1} - 1) + u_1\,(z_i^{P2} - 1)\right]}{1 - z_i^{P1+P2}} \tag{4.56}$$

Cette expression n'est valable qu'en association avec une matrice de régression calculée à partir de $t = 0$.

Un avantage de cette approche est qu'il est possible, après avoir identifié les coefficients de Prony sur un cycle quelconque du signal, de déterminer s'il s'agit ou non de conditions de régimes : en effet, si le jeu de coefficients liés aux exponentielles transitoires obtenu après identification est identique à celui calculé par l'équation 4.56, cela implique que le cycle relevé est bien un cycle limite.

### 4.4.6 Pôles complexes

Il se peut que les pôles aillent par paires complexes conjuguées. Cela indique que la réponse indicielle comporte un dépassement de la valeur de régime, suivi d'une série

d'oscillations, amorties ou non. Les développements précédents ont été effectués sans présupposer que les pôles étaient réels ou complexes, les résultats sont d'application dans les deux cas. En particulier, la formule (4.56) appliquée à une paire de pôles complexes conjugués, associés à des coefficients formant eux-mêmes une paire de complexes conjugués, donnera une réponse du même type. Une régression de type (4.50) donnera un résultat réel. Numériquement, il pourrait apparaître une partie imaginaire résultant d'erreur d'arrondi, elle ne sera donc pas prise en considération.

### 4.4.7 Implémentation : `pronysq.m` et `pronyeq.m`

Les résultats développés précédemment ont été implémentés via deux fonctions MatLab :

`pronysq.m` calcule le modèle de Prony pour un système excité par un signal constant par morceaux;

`pronyeq.m` calcule les coefficients de Prony lorsque le système est à l'équilibre.

Voyons à présent l'utilisation de ces deux fonctions.

```
function [B, Z, G, Lam, yk, LB] = pronysq(z, n, ts, method, corr)
```

Les arguments d'entrées sont

$z$ : il s'agit d'une matrice, dont la première colonne est la sortie ($y$) et la seconde l'entrée ($u$) du système, par analogie avec les autres routines d'identification.

$n = (M\, nk)$ : $M$ est l'ordre (le nombre de pôles) du modèle, $nk$ un délai permettant de travailler avec des simulations analogiques et numériques.
$nk$ prend donc la valeur 1 pour un système analogique, et zéro pour un système numérique. Par défaut, le système est supposé analogique.

$ts$ est le temps d'échantillonnage, il intervient pour le calcul des constantes de temps numériques.

*method* indique comment résoudre les systèmes d'équations surdéterminés : soit par moindre carrés, le comportement par défaut, soit en utilisant la valeur $'TLS'$ pour une résolution par moindres carrés totaux.

*corr* indique l'ordre du terme correctif : $'lindrift'$ pour une dérive proportionnelle aux temps, $'sqdrift'$ pour une dérive quadratique. Par défaut, seul un terme constant est pris en considération.

Les arguments de sortie sont :

$B$ est un vecteur contenant les $c_i$ et $d_i$. Il est ordonné comme suit :

1. $M + 1$ composants associés à la commande;
2. $M$ composants associés aux conditions initiales;
3. un terme de correction constant;
4. éventuellement, les termes de correction linéaire et quadratique.

$Z$ est un vecteur contenant les $M$ pôles du système, dans le même ordre que les composants de $B$.

$G$ est le gain du système.

$Lam$ permet de calculer les exposants des exponentielles donnant l'expression temporelle sous la forme

$$\mathbf{y}(t) = \sum_{i=1}^{M} c_i \ \exp(\lambda t) + \dots \tag{4.57}$$

Ces valeurs découlent de

$$\lambda = \log(Z)/T_s \tag{4.58}$$

Les constantes de temps peuvent se calculer directement dans MatLab comme $-1./Lam$

$yk$ contient les valeurs prédites pour $y$;

$LB$ est la matrice de régression telle que $yk = LB \cdot B$;

Le programme repère automatiquement les transitions dans le vecteur $u$; la période ne doit pas nécessairement être constante. La seule contrainte est que chaque segment doit contenir au minimum $M + 1$ points, sans quoi il n'est pas possible d'écrire d'équation de récurrence pour ce segment.

La première matrice de régression contient $M - 2$ lignes par segment; la seconde contient le même nombre de lignes que $y$. Le calcul des racines se fait via la procédure `roots()`; aucun traitement supplémentaire n'étant effectué (réflexion à l'intérieur du cercle unité des pôles instables, ...)

```
function [B0, LB0] = pronyeq(B, Z, per, U, ts, corr);
```

Arguments d'entrée :

$B$ coefficients linéaires d'un cycle quelconque du signal. Seule la partie liée à la commande (indices 1 à $M$) et à la constante de correction est prise en considération;

$Z$ pôles du signal;

*per* : une valeur scalaire indique la durée totale d'un cycle; un vecteur contient les longueurs de chacune des phase;

$U$ est un vecteur contenant la valeur de la commande durant chacune des phases, exprimée soit sous la forme $(u_0, u_1)$, soit en tant que $\mathbf{u}$;

$ts$ est le temps d'échantillonnage, et n'est utilisé que s'il est nécessaire de reconstruire une matrice de régression;

*corr* joue le même rôle que dans la fonction `pronysq`.

En sortie, $B0$ est un vecteur permettant de calculer le cycle limite d'un signal, par $ylim = LB0 \cdot B0$; $LB$ étant la matrice de régression renvoyée par `pronysq` si la première période du signal est complète. Les coefficients liés aux conditions initiales sont donnés par (4.56).

Si nécessaire, cette routine renvoie également la matrice de régression démarrant en $t = 0$. Cela peut s'avérer utile si, lors de la phase d'identification, la première période du signal n'est pas utilisable telle quelle, nous songeons notamment à

- l'introduction d'un délai entre **y** et **u**, ce qui a pour effet d'ajouter des points fictifs au début de la commande;
- lorsque l'acquisition commence par une phase initiale sans commande appliquée.

### 4.4.8 Illustration : `pronysq`

Pour valider toute cette démarche, nous allons identifier une courbe et montrer les coefficients obtenus par la méthode de Prony générique. Reprenons l'exemple apparaissant dans les figures 3.1 à 3.4, page 158. Les deux pôles étant situés en 0.9 et 0.99, la fonction de transfert du système est

$$H(z) = \frac{1}{1 - 1.89z^{-1} + 0.891z^{-2}} \tag{4.59}$$

$$= \frac{-10}{1 - 0.9z^{-1}} + \frac{-11}{1 - 0.99z^{-1}} \tag{4.60}$$

En outre, nous avons choisi une valeur de gain de 0.8.

Pour obtenir la réponse indicielle, nous devons multiplier cette fonction de transfert par $N(z) \cdot (1/(1 - z^{-1}))$ et décomposer le tout en fractions simples. $N(z)$ est un numérateur choisi pour obtenir l'équivalent d'une simulation analogique ou digitale comme expliqué dans les commentaires de l'équation 4.41, tandis que le second terme est la transformée en $\mathcal{Z}$ d'un échelon.

La transformée en $\mathcal{Z}$ de la réponse indicielle se calcule comme

$$\left(\frac{-10}{1 - 0.9z^{-1}} + \frac{-11}{1 - 0.99z^{-1}}\right)\frac{N(z)}{1 - z^{-1}} = \frac{a}{1 - 0.9z^{-1}} + \frac{b}{1 - 0.99z^{-1}} + \frac{c}{1 - z^{-1}} \tag{4.61}$$

Après développement et comparaison des termes de même ordre en $z^{-1}$, nous obtenons, sous forme matricielle :

$$\begin{pmatrix} 1 & 1 & 1 \\ -1.99 & -1.9 & -1.89 \\ 0.99 & 0.9 & 0.891 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} N(0) \\ N(1) \\ N(2) \end{pmatrix} \tag{4.62}$$

Remarquons tout d'abord que cette matrice est mal conditionnée : son déterminant est égal à 9 $10^{-5}$ et le rapport de la plus grande valeur singulière à la plus petite avoisine les 10 000. Cela indique qu'il faut être prudent au niveau de l'inversion de cette matrice.

Commençons par le cas analogique. Si nous décidons de partir de conditions initiales nulles, cela implique que la première valeur soit également nulle, d'où nous déduisons que la valeur de $N$ est (0, 0.8, 0) : la commande équivalente est un échelon retardé d'une unité, soit $z^{-1}\ U(z)$.

Les trois coefficients $a$, $b$, et $c$ valent dans ce cas 80, -880, et 800, leur somme étant bien nulle. Nous avons procédé à la simulation suivante :

```
%% fix the system poles
A=poly([.9 .99]);
%% input signal : square wave, 4 cycles of 32 '1' and 32 '0'
x=[ones(1,32) zeros(1,32)]';x=[x;x;x;x];
%% generate the output signal
y= 3 + filter(.8, A, [0; x(1:255)]);
%% Compute the Prony approximation
[B,Z,Lam,yk,LB, G0]=pronysq([y x], [2 1],1,'TLS');
```

Dans l'appel de la fonction `pronysq.m`, nous avons demandé un modèle d'ordre 2 et un délai unitaire. La réponse $B$ est donnée dans le tableau 4.1.

| coefficient | rôle |
|---|---|
| 80 | signal carré, filtré au premier ordre par le pôle en 0.9 |
| -880 | signal carré, filtré au premier ordre par le pôle en 0.99 |
| 800 | commande |
| 0 | exponentielle liée au pôle en 0.9 (condition initiale) |
| 0 | exponentielle liée au pôle en 0.99 (condition initiale) |
| 3 | terme constant |

TAB. 4.1: simulation simple, coefficients renvoyés par la méthode `pronysq`

Observons que nous avons bien retrouvé nos trois termes $a$, $b$, et $c$ ainsi que le terme de décalage. Notre algorithme n'est donc pas pris en défaut.

Introduisons à présent des conditions initales et du bruit :

```
%% fix the system poles
A=poly([.9 .99]);
%% input signal : square wave, 4 cycles of 32 '1' and 32 '0'
x=[ones(1,32) zeros(1,32)]';x=[x;x;x;x];
%% generate the output signal
y= 3 + filter(.8, A, [0; x(1:255)],[36 18]);
```

```
%% try different noise levels
indn=1e-5;
%% Compute the Prony approximations
for indi=1:1000,
  %% generate random noise
  n0 = randn(size(y));
  %% normalize it
  n0 = n0 - mean(n0); n=n0*std(y)/std(n0);
 [Bt(:,indi),Zt(:,indi),Gt(indi)]=
        pronysq([y+n0*indn(indi) x], [2 1],1,'TLS');
  Bet(:,indi) = (Bt(:,indi)-B)./B*100;
  Zet(:,indi) = (Zt(:,indi)-Z)./Z*100;
  Get(  indi) = (Gt(  indi)-G)./G*100;
  [Bl(:,indi),Zl(:,indi),Gl(indi)]=
        pronysq([y+n0*indn(indi) x], [2 1],1,'LS');
  Bel(:,indi) = (Bl(:,indi)-B)./B*100;
  Zel(:,indi) = (Zl(:,indi)-Z)./Z*100;
  Gel(  indi) = (Gl(  indi)-G)./G*100;
end
```

On peut effectuer deux types de simulations : soit on étudie l'effet de l'amplitude du bruit, soit on répète la simulation pour en chercher les propriétés statistiques.

Dans le premier cas, si on place dans un graphe doublement logarithmique la valeur de *indn* qui conditionne la puissance du bruit et l'erreur relative, on obtient une série de droites à 45 °. En ce qui concerne les deux méthodes, la solution "TLS" est légèrement plus précise que la solution "LS", mais décroche à partir d'un niveau de bruit de $1\ 10^{-3}$, tandis que la solution "LS" décroche seulement une décade plus loin, c.-à-d. à partir d'un niveau de $1\ 10^{-2}$.

Dans le second cas (Monte-Carlo), nous avons généré 1000 réalisations de bruit, de déviation standard valant $1\ 10^{-5}$ la déviation standard de la sortie. Voici les résultats pour cette simulation, complétés par l'erreur relative, exprimée en pourcentage de la valeur vraie.

Commençons par le cas "TLS" :

| TLS | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $Z_i$ | $\mu$ | $\sigma$ | $c_i$ | $\mu$ | $\sigma$ | $d_i$ | $\mu$ | $\sigma$ |
| .9 | $7.0\ 10^{-7}$ | $6.0\ 10^{-4}$ | 80 | $1.5\ 10^{-4}$ | $1.6\ 10^{-2}$ | -500 | $8.0\ 10^{-5}$ | $7.1\ 10^{-3}$ |
| .99 | $-6.4\ 10^{-6}$ | $6.0\ 10^{-4}$ | -880 | $-5.4\ 10^{-4}$ | $5.3\ 10^{-2}$ | 500 | $-4.8\ 10^{-4}$ | $4.7\ 10^{-2}$ |
| 1 | | | 800 | $-6.1\ 10^{-4}$ | $6.0\ 10^{-2}$ | 3 | $8.9\ 10^{-2}$ | $8.8\ 10^{+0}$ |

Voyons à présent le cas “LS” :

| LS | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $Z_i$ | $\mu$ | $\sigma$ | $c_i$ | $\mu$ | $\sigma$ | $d_i$ | $\mu$ | $\sigma$ |
| .9 | $6.4\ 10^{-6}$ | $6.0\ 10^{-4}$ | 80 | $3.0\ 10^{-4}$ | $1.6\ 10^{-2}$ | -500 | $1.4\ 10^{-4}$ | $7.1\ 10^{-3}$ |
| .99 | $-1.2\ 10^{-5}$ | $6.0\ 10^{-4}$ | -880 | $-1.0\ 10^{-3}$ | $5.3\ 10^{-2}$ | 500 | $-9.0\ 10^{-4}$ | $4.7\ 10^{-2}$ |
| | | | 800 | $-1.2\ 10^{-3}$ | $6.0\ 10^{-2}$ | 3 | $1.7\ 10^{-1}$ | $8.8\ 10^{+0}$ |

En ce qui concerne le gain, nous avons :

| | TLS | LS |
|---|---|---|
| $\mu$ | $1.9\ 10^{-4}$ | $-3.3\ 10^{-6}$ |
| $\sigma$ | $5.6\ 10^{-3}$ | $5.6\ 10^{-3}$ |

Nous pouvons effectuer les observations suivantes :

1. les pôles sont les paramètres les moins affectés par le bruit;
2. en ce qui concerne les coefficients des exponentielles, ceux associés au pôle le plus petit sont moins affectés par le bruit que ceux associés au pôle proche de 1;
3. le terme constant est celui dont la sensibilité au bruit est la plus grande;
4. la méthode “TLS” présente un biais légèrement inférieur à la méthode “LS”; tandis que les écarts-types sont identiques;
5. pour le gain statique, les paramètres statistiques sont du même ordre de grandeur que ceux associés aux termes $c_i$.

Dans le cas digital, la matrice du système est donnée par l'équation 4.62, et le polynôme $N(z)$ vaut à présent $(0.8, 0, 0)$, ce qui donne les coefficients -72, -871.2, et 800, dont la somme est bien égale à 0.8. Au niveau de la simulation, les seules différences par rapport au cas précédent sont d'une part que la commande n'est pas retardée d'une unité, et que la variable $nk$ dans l'appel à la fonction `pronysq` vaut à présent 0.

Résultats pour le cas “TLS” :

| TLS | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $Z_i$ | $\mu$ | $\sigma$ | $c_i$ | $\mu$ | $\sigma$ | $d_i$ | $\mu$ | $\sigma$ |
| .9 | $9.0\ 10^{-7}$ | $6.0\ 10^{-4}$ | 72 | $1.9\ 10^{-4}$ | $1.6\ 10^{-2}$ | -500 | $9.7\ 10^{-5}$ | $7.2\ 10^{-3}$ |
| .99 | $-7.6\ 10^{-6}$ | $6.0\ 10^{-4}$ | -871.2 | $-6.5\ 10^{-4}$ | $5.3\ 10^{-2}$ | 500 | $-5.6\ 10^{-4}$ | $4.5\ 10^{-2}$ |
| 1 | | | 800 | $-7.2\ 10^{-4}$ | $5.9\ 10^{-2}$ | 3 | $1.0\ 10^{-1}$ | $8.5\ 10^{+0}$ |

Résultats pour le cas “LS” :

| LS | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $Z_i$ | $\mu$ | $\sigma$ | $c_i$ | $\mu$ | $\sigma$ | $d_i$ | $\mu$ | $\sigma$ |
| .9 | $6.5\ 10^{-6}$ | $6.0\ 10^{-4}$ | 72 | $3.2\ 10^{-4}$ | $1.6\ 10^{-2}$ | -500 | $1.4\ 10^{-4}$ | $7.2\ 10^{-3}$ |
| .99 | $-1.2\ 10^{-5}$ | $6.0\ 10^{-4}$ | -871.2 | $-1.0\ 10^{-3}$ | $5.3\ 10^{-2}$ | 500 | $-8.6\ 10^{-4}$ | $4.5\ 10^{-2}$ |
| | | | 800 | $-1.1\ 10^{-3}$ | $5.9\ 10^{-2}$ | 3 | $1.6\ 10^{-1}$ | $8.5\ 10^{+0}$ |

Résultats pour le gain :

| | TLS | LS |
|---|---|---|
| $\mu$ | $2.0\ 10^{-4}$ | $4.4\ 10^{-6}$ |
| $\sigma$ | $5.6\ 10^{-3}$ | $5.6\ 10^{-3}$ |

Ces résultats n'appellent aucune remarque particulière, ils sont extrêmement semblables à ceux obtenus via la simulation analogique.

### 4.4.9 Illustration : `pronyeq`

Pour valider le calcul de cycle de régime via la routine `pronyeq`, nous allons procéder assez simplement. Le point de départ est similaire à l'exemple précédent, à ceci près que le pôle le plus grand sera de 0.94, pour assurer une décroissance plus rapide de l'exponentielle associée. Le signal de commande prendra les valeurs 1.5 et 0.5, pour montrer que l'approche est générique. Nous allons calculer un jeu de paramètres $B0$ pour la totalité du signal, à partir duquel nous calculerons le jeu $B'$ correspondant au cycle de régime. Pour valider, nous calculerons 3 autres jeux de paramètres, $B1$ à $B3$, démarrant respectivement à la seconde, à troisième et à la quatrième période du signal.

La simulation est la suivante :

```
%% fix the system poles
A=poly([.9 .94]);
%% input signal : square wave, 4 cycles of 32 '1' and 32 '0'
x= .5 + [ones(1,32) zeros(1,32)]';x=[x;x;x;x];
%% generate the output signal
yd= 3 + filter(.8, A, x);
%% Compute the Prony approximations
[B0,Z0]=pronysq([yd x], [2 0],1,'TLS');
B1=pronysq([yd(1+64:256) x(1+64:256)], [2 0],1,'TLS');
B2=pronysq([yd(1+128:256) x(1+128:256)], [2 0],1,'TLS');
B3=pronysq([yd(1+192:256) x(1+192:256)], [2 0],1,'TLS');
%% Compute equivalent coefficients to steady-state conditions
[B, LBe]=pronyeq(B0,Z0, 64, [1.5 .5]);
```

Les résultats de ces deux approches sont :

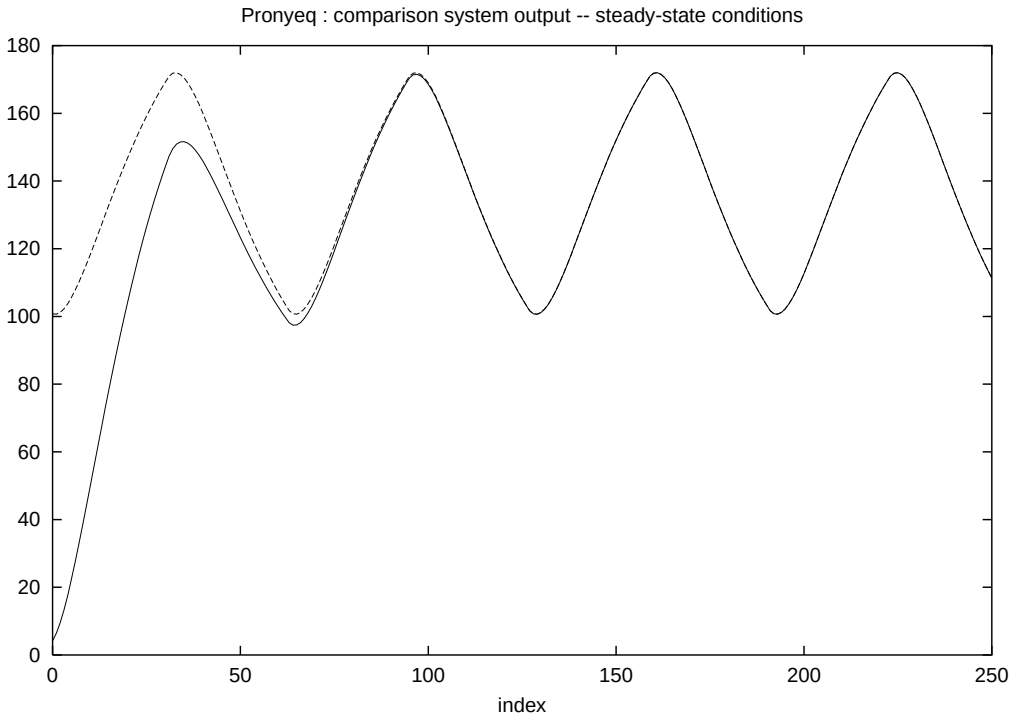



FIG. 4.1: Comparaison de la sortie du système avec le cycle de régime

| B0 | B1 | B2 | B3 | Be |
|---|---|---|---|---|
| 162 | 162 | 162 | 162 | 162 |
| -294.5 | -294.5 | -294.5 | -294.5 | -294.5 |
| 133.3 | 133.3 | 133.3 | 133.3 | 133.3 |
| 0.0 | -86.3 | -86.4 | -86.4 | -86.4 |
| 0.0 | 179.5 | 182.9 | 190.0 | 190.0 |
| 3.0 | 3.0 | 3.0 | 3.0 | 3.0 |

TAB. 4.2: évolution par cycle des $c_i$ et $d_i$

La figure 4.1 illustre d'une part la sortie du système et d'autre part le cycle de régime, répété quatre fois. La première courbe est celle démarrant à proximité de l'origine : on observe bien la convergence de cette courbe vers le cycle de régime, ce qui correspond à la disparition des conditions initiales.

Comme on peut le constater, les deux approches, à savoir Prony sur le dernier cycle et le calcul des coefficients du cycle de régime, donnent des résultats identiques. Au niveau de la quatrième et de la cinquième ligne du tableau, nous constatons une convergence des coefficients, qui finissent par donner un résultat identique à ceux contenus dans la dernière colonne ($Be$), correspondant au calcul des conditions initiales dans la situation de régime, démarrant avec une valeur du temps réinitialisée à zéro.

Pour terminer cette illustration, voici les trois premiers et les trois derniers points de la dernière période ainsi que du cycle de régime :

| index | dernier cycle | cycle limite |
|---|---|---|
| 0 | 100.82 | 100.82 |
| 1 | 100.69 | 100.69 |
| 2 | 101.19 | 101.19 |
| 61 | 105.31 | 105.31 |
| 62 | 103.47 | 103.47 |
| 63 | 101.70 | 101.70 |

### 4.4.10 Commentaires

1. La démarche de Prony nous semble beaucoup plus proche de la modélisation physique. La recherche d'une dimensionnalité concerne surtout le nombre de pôles du système. Dans l'approche de type « Identification des processus », la dimensionnalité concerne les degrés des polynômes au numérateur et au dénominateur de la fonction de transfert (discrète) du système. Il s'agit d'une logique différente, car, une fois fixé le degré du dénominateur, que représente physiquement tel ou tel choix de degré du numérateur ?
2. L'algorithme de Prony est direct. Par comparaison, une recherche d'un modèle « Output Error » (OE) consiste à
   - choisir un « bon » point de départ, c.-à-d. un modèle stable;
   - minimiser l'erreur de prédiction par une recherche de type « steepest descent », avec des risques de non-convergence.

   En contrepartie, aucune modélisation du bruit n'est possible par la méthode de Prony.
3. Les fonctions de base de l'algorithme de Prony sont des exponentielles, éventuellement complexes (sinusoïdes amorties). Contrairement à d'autres méthodes exigeant que les fonctions de base soient *non-corrélées*, la seule restriction est ici

que ces fonctions soient *linéairement indépendantes*. C'est ainsi que la présence de signaux dont la périodicité n'est pas exactement un multiple de la fréquence de base peut générer des erreurs importantes sur la transformée de Fourier discrète d'un signal, à cause de la discrétisation des fréquences des signaux analysants. Dans le cas de la méthode de Prony, il n'y a pas de discrétisation[1] des valeurs que peuvent prendre les paramètres.

4. La méthode de Prony permet également, sous une forme légèrement différente, de modéliser un signal sinusoïdal. À partir des pôles identifiés, il est possible de déterminer précisément la fréquence de base. On parle dans ce cas de super-résolution car la résolution effective n'est pas liée à la fréquence d'échantillonnage comme dans les approches basées sur la transformée de Fourier.
5. Un certain nombre de méthodes d'analyse associent $n$ valeurs à un signal d'entrée de $n$ valeurs : TFD, analyse par ondelettes, ... Dans le cas de l'algorithme de Prony, il y a une *réduction* de la quantité d'informations nécessaire pour représenter un signal : il suffit de $2M + 2$ paramètres pour représenter la majeure partie du signal.
6. Un certain nombre de méthodes d'Identification demandent la disparition des conditions initiales et l'absence de dérive dans le signal. Pour cela, il est de coutume d'utiliser la fonction `detrend()` de MatLab, qui enlève la valeur moyenne et éventuellement une dérive linéaire. En outre, comme expliqué précédemment, le traitement des conditions initiales est réellement un point faible de la méthode OE de MatLab. Dans le cas de la méthode de Prony, ces paramètres font intégralement partie du problème, et sont traités de façon adéquate.
7. La solution de Prony permet également de calculer la transformée en $\mathcal{Z}$ du signal, ainsi que son spectre. Ce calcul peut se faire à n'importe quelle fréquence, sans restrictions liées à la discrétisation des fréquences des signaux analysants.
8. Nous avons montré comment obtenir le gain statique ainsi que le cycle limite d'un système soumis à une commande périodique. À notre connaissance, cela n'est pas possible directement via les méthodes d'identification classique. Pour résoudre ce problème, il faudrait simuler la réponse du système durant un temps suffisamment long pour voir apparaître le cycle limite.

## 4.5 Discussion

D'un point de vue conceptuel, nous pouvons comparer cette méthode générique avec la détection synchrone en instrumentation. Le but de celle-ci est de séparer l'effet d'une source de celui du bruit basse fréquence lié à des problèmes de décalage ou de dérive lente. Pour cela, le signal analysant est déplacé dans l'espace spectral dans une zone où ces perturbations sont fortement atténuées.

1. autre que celle liée à la représentation par un nombre fini de bits

Il en est de même dans cette méthode de Prony générique : plutôt que de mesurer une réponse impulsionnelle et de ne pouvoir faire la différence entre ce qui est lié à l'entrée et ce qui est constant, nous modulons la commande. À partir de ce moment, une séparation claire est possible entre son effet et celui de corrections constantes, voire dépendant de la grandeur temporelle.

## 4.6 Résumé

Nous avons exposé une méthode, originale, appelée méthode Prony générique, permettant d'identifier les pôles d'un système, ainsi que la contribution de la commande et des conditions initiales. Cette méthode se distingue de la méthode de Prony « simple », qui ne s'applique qu'à un signal évoluant sous l'effet de ses conditions initiales ou à une réponse indicielle en l'absence de conditions initiales.

Contrairement à d'autres méthodes d'identification, Il n'y a pas de discrétisation de la valeur des solutions, et l'algorithme n'est pas itératif.

Nous avons également montré comment obtenir le spectre et le cycle limite d'un système excité par un signal d'entrée périodique. L'intérêt de cette approche est de donner de façon tout à fait rigoureuse la valeur de régime de la réponse du système, alors que l'identification a eu lieu sur la partie transitoire de la réponse. L'avantage de cette approche est une réduction de la durée de la mesure, puisqu'il n'est plus nécessaire d'attendre d'arriver à une situation de régime.

Nous allons à présent montrer un exemple d'application concret, où nous avons modélisé les réponses thermiques obtenues au cours des expériences décrites au début de cette partie.

Chapitre 5

# Prony générique : application à la modélisation des propriétés de transmission thermique

## 5.1 Introduction

Rappelons brièvement le contexte : nous avons obtenu un certain nombre de réponses thermiques obtenues sur des sites caractérisés par leur degré de modification par rapport à une situation de référence. Le but de ces expérimentations est d'essayer d'établir, à partir de la réponse thermique, un classement qui permette de retrouver ce degré de modification.

Nous avons passé en revue un certain nombre de méthodes que nous avons utilisées dans ce but, ces méthodes n'ayant guère donné de résultats probants, en particulier parce que les modèles obtenus par identification pouvaient être inexploitables parce que ne répondant pas à certains critères établis sur base d'une réflexion sur le modèle physique de la transmission de la chaleur.

Ayant repris récemment nos investigations, nous avons cherché à modéliser nos courbes d'une façon plus « parlante », c.-à-d. en nous intéressant non plus aux coefficients du modèle, mais au cycle de régime. Étant donné l'opacité de la relation entre les amplitudes relevées et les jeux de coefficients obtenus, nous avons ainsi cherché à obtenir une différenciation plus simple entre les différentes réponses thermiques.

## 5.2 Traitement numérique

Après avoir exploré les pistes présentées dans le chapitre 2 pour tenter de classifier les courbes relevées, nous avons essayé la méthode de Prony générique, dont nous venons

d'expliquer le fonctionnement. Il est apparu qu'un certain nombre de courbes qui étaient totalement inexploitables via les méthodes utilisées précédemment donnaient lieu à des modèles acceptables du point de vue des propriétés des pôles.

Par ailleurs, nous avons choisi le cycle de régime comme caractérisation de la réponse thermique à la dose de modification. Nous avons en effet montré que la méthode de Prony générique permet d'obtenir la valeur de régime de la réponse du système.

Rappelons que cette méthode générique ne permet de faire la part des choses entre la réponse indicielle et la réponse à une excitation que si le signal d'entrée présente au moins une transition. Cette condition est satisfaite dans nos expérimentations, puisque le signal d'entrée est un signal en créneau comportant au minimum deux cycles.

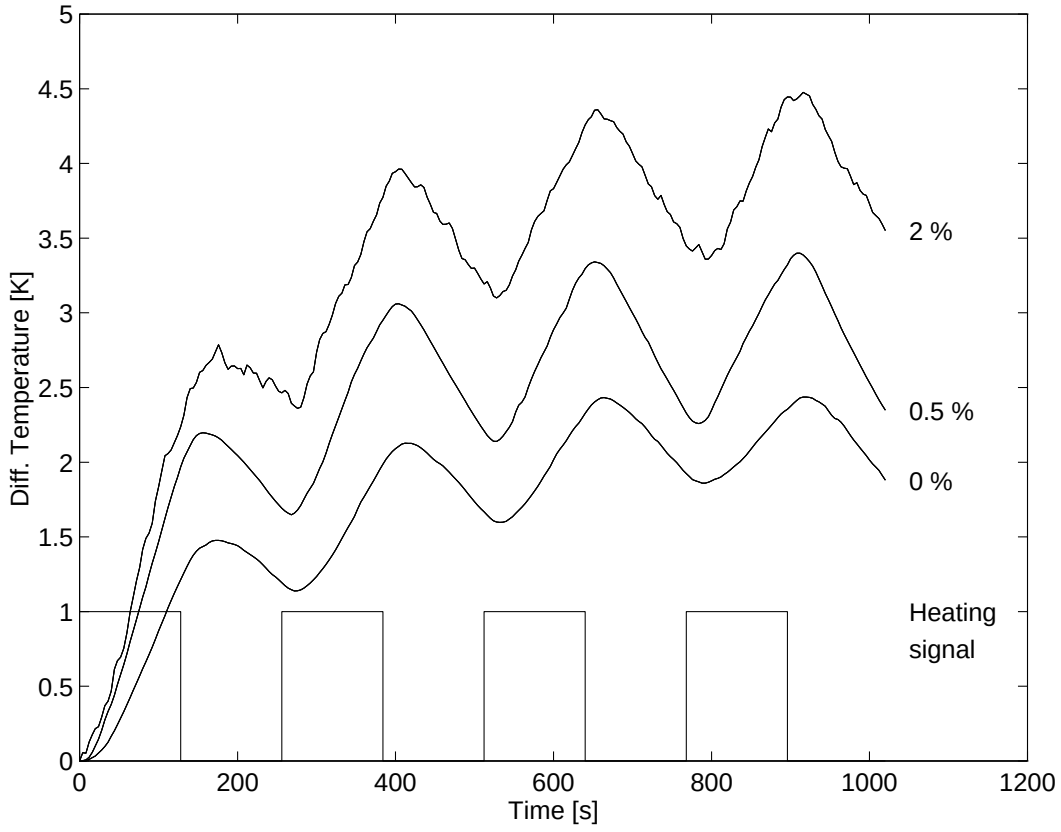



FIG. 5.1: réponse thermique en fonction de la dose de modification

Au niveau des traitements numériques, nous avons effectué les recherches en prenant systématiquement des modèles à 1, 2, et 3 pôles. Il est apparu en comparant les variances de l'erreur que le modèle à deux pôles décrit le mieux notre système, le modèle à trois pôles n'étant pas toujours physiquement réalisable. Nous avons à chaque fois pris un modèle avec correction de la valeur constante, même si dans la plupart des cas cette valeur était très petite.

Dans chaque situation, nous avons également étudié le délai. Il est de tradition en identification d'essayer plusieurs valeurs du délai entrée-sortie, et de conserver celui qui donne lieu à l'erreur la plus basse. Dans le cas de nos courbes, il est apparu que cette approche est à proscrire. En effet, les courbes simulées correspondant à un modèle obtenu en décalant artificiellement la sortie par rapport à l'entrée contenaient le plus souvent des variations brusques à l'endroit des transitions de l'entrée. De plus, comment caractériser une constante de temps en présence d'un délai pur ?

Nous avons donc restreint le délai à ne prendre que les valeurs "0" et "1", ce qui correspond à des modèles digitaux ou analogiques, la seule différence entre eux résidant dans la somme des coefficients linéaires liés à la sortie. Pour les étapes ultérieures, nous avons conservé les paramètres du modèle donnant lieu à l'erreur la plus faible.

Pour clarifier les idées, reprenons les réponses observées chez le sujet n°2. La figure 5.1, qui est déjà apparue page 152 illustre les trois courbes associées respectivement aux doses 0 %, 0.5 % et 2 %.

Illustrons tout d'abord les courbes fournies par la méthode de Prony. Nous avons repris dans les figures 5.2 et 5.3 d'une part la température, et d'autre part la courbe prédite par le modèle de Prony, respectivement pour la dose 0 % et 0.5 %. Nous constatons que le modèle « colle » correctement à la mesure, et plus particulièrement vers la fin.

Nous avons étudié un certain nombre de modèles, en faisant varier le nombre de points pris en considération. Nous avons d'abord pris la totalité des points, puis ignoré la première demi-période, puis la première période, et ainsi de suite jusqu'à ne plus prendre en considération que la dernière période.

Nous avons répété ces calculs avec les méthodes "LS" et "TLS" telles que détaillées dans le chapitre précédent. Nous avons ensuite généré un graphe reprenant toutes les courbes limites. Il est apparu que le dernier cas (une seule période) donnait lieu à un cycle nettement différent des autres : nous avons donc pris comme limite un signal ne contenant qu'un cycle et demi.

## 5.3 Résultats : sujet n°2

Les figures 5.4 et 5.5 illustrent les résultats de ces investigations. Nous avons 3 jeux de 6 courbes, ainsi qu'une courbe supplémentaire qui représente la moyenne d'un jeu. En fait, chacune des 6 courbes contient une estimée du cycle de régime. Nous pouvons tout d'abord observer que, pour un niveau de modification donné, ces estimées forment un groupe relativement compact, sauf pour celles associées à la dose 2 % avec la méthode "LS". Dans ce cas, on constate que les courbes sont parallèles, la dispersion est induite par une évaluation du terme constant qui semble plus problématique.

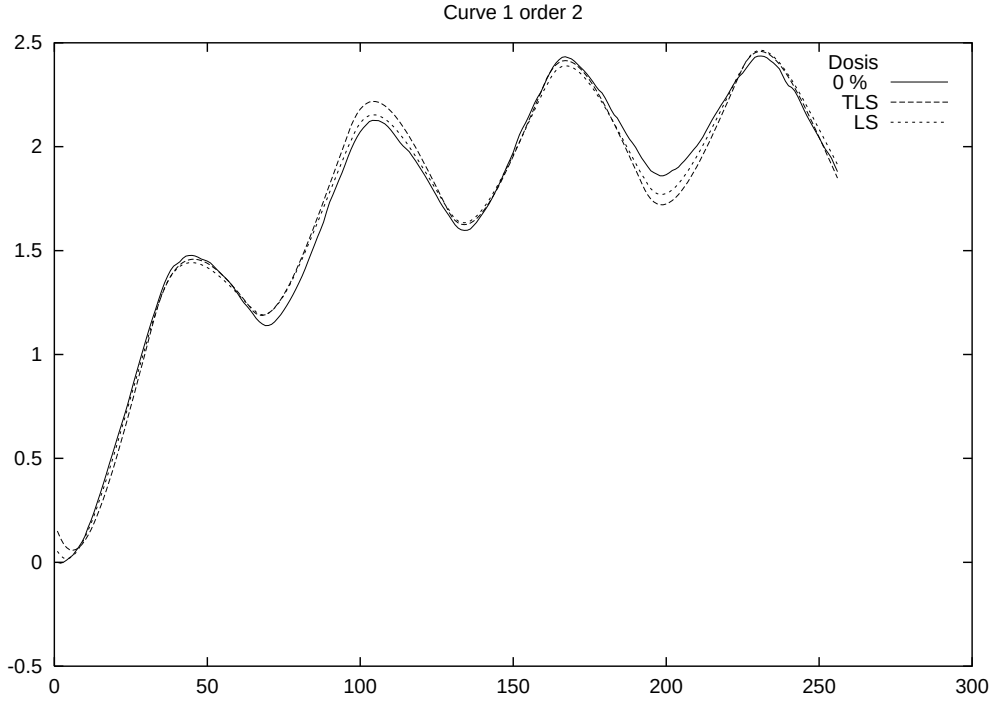



FIG. 5.2: Comparaison température – courbes simulées

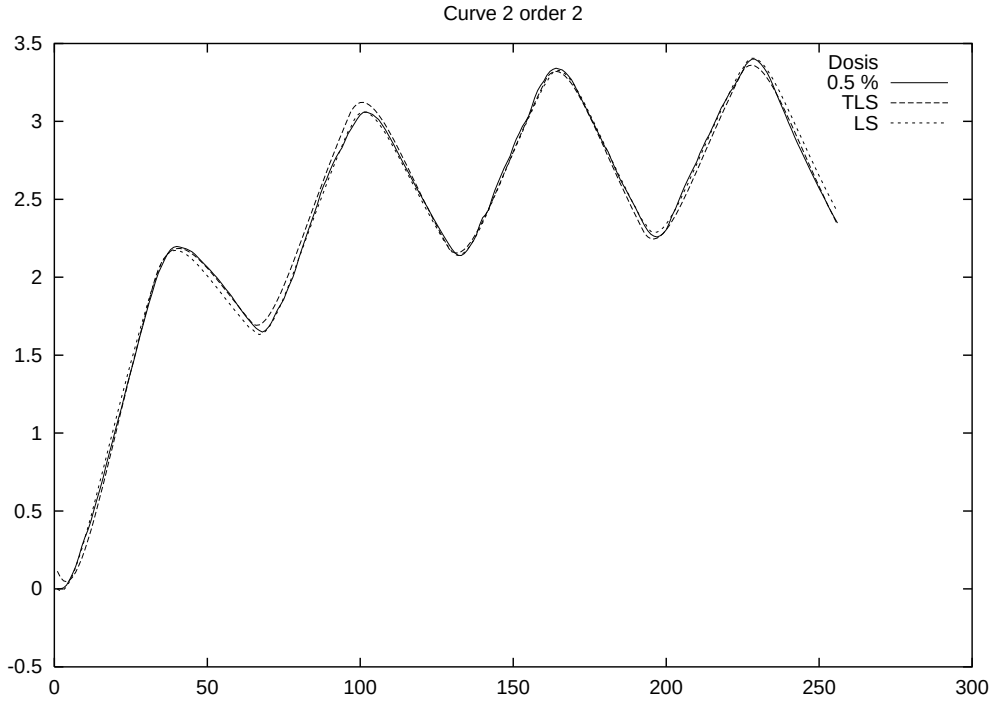



FIG. 5.3: Comparaison température – courbes simulées

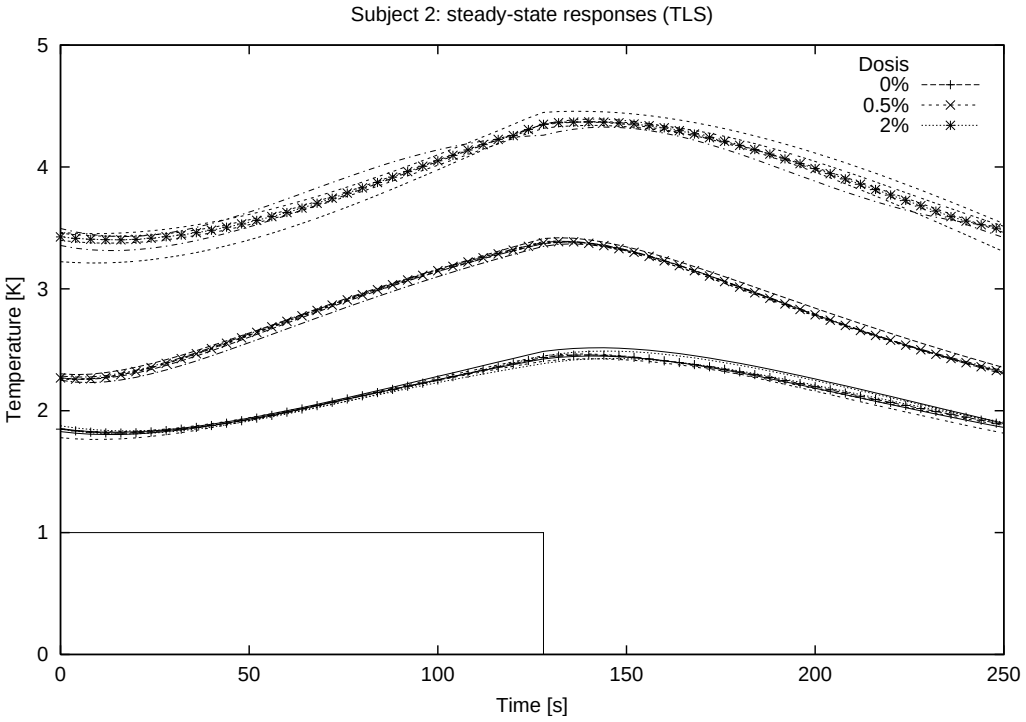



FIG. 5.4: cycle limite en fonction de la dose de modification, méthode TLS

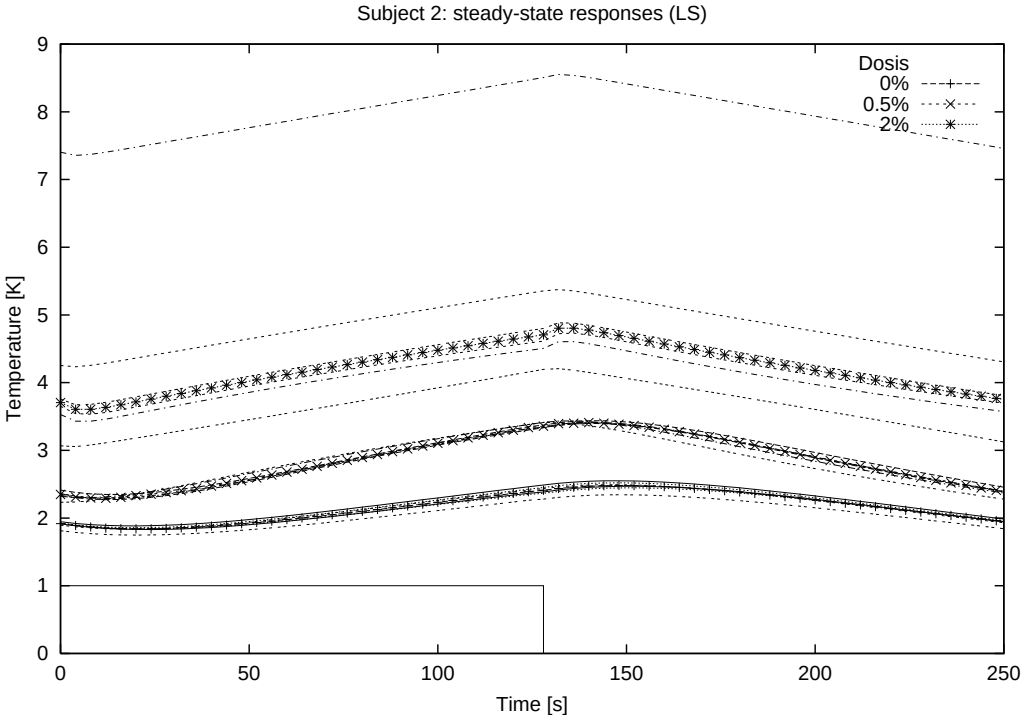



FIG. 5.5: cycle limite en fonction de la dose de modification, méthode LS

Nous avons alors appliqué une nouvelle fois la méthode de Prony sur la *moyenne* de chaque jeu de courbes, les résultats en sont présentés dans le tableau 5.1. Remarquons que dans ce cas, l'approche est tout à fait rigoureuse: nous effectuons une moyenne entre plusieurs estimées d'une même quantité, ce qui aura pour effet de d'atténuer l'influence du bruit.

| Dose | $\sigma_G$ | $\sigma_c$ | $K_s$ | $\dot{y}$ | PTP | TC 1 | TC 2 |
|---|---|---|---|---|---|---|---|
| 0 % | 0.065 | $5.6\ 10^{-5}$ | 2.18 | 7.4 mK/s | 0.68 K | 44 s | 190 s |
| 0.5 % | 0.040 | $3.4\ 10^{-5}$ | 2.83 | 11 mK/s | 1.11 K | 25 s | 148 s |
| 2 % | 0.060 | 0.002 | 3.91 | 11.4 mK/s | 1.06 | 44 s | 260 s |

TAB. 5.1: paramètres du cycle de régime moyen en fonction de la dose

Les variables de ce tableau sont définies comme suit :

$\sigma_G$ écart-type de l'erreur de simulation globale;

$\sigma_c$ écart-type de l'erreur au niveau de la moyenne des cycles limite;

$Ks$ élévation de température sous l'effet de la composante moyenne du signal d'entrée;

$\dot{y}$ pente maximale de la courbe, exprimée en K/s;

$PTP$ amplitude crête-à-crête du cycle limite;

$TC1$, $TC2$ constantes de temps du modèle.

Dans ce tableau, nous pouvons constater :

1. par rapport à la méthode cepstrale, nous constatons que nous avons effectivement obtenu des cycles de régime;
2. la réduction de l'erreur est très spectaculaire : elle est d'environ 1100 dans les deux premières expériences et de 30 dans le cas de la troisième. Si on observe les courbes brutes, on remarquera que cette dernière est nettement plus bruitée, par un signal qui est probablement lié au cycle respiratoire.
3. le gain statique est le paramètre qui varie le plus significativement avec la dose. Il reflète probablement une augmentation de la conductivité thermique.
4. au niveau de la pente et de l'amplitude crête-à-crête, étant donné que les courbes présentent des allures très semblables, ces deux paramètres sont probablement équivalents. On constate une augmentation d'environ 50 % en passant de la courbe de référence aux courbes correspondant à une modification.
5. au niveau des constantes de temps, le comportement est un peu plus erratique. En particulier, beaucoup de modèles pour la quatrième courbe contenaient des pôles complexes conjugués, ce qui avait pour effet de produire en sortie une oscillation tendant à s'aligner sur le bruit présent. Dans le cycle limite moyen, ce bruit a disparu et les pôles sont réels.

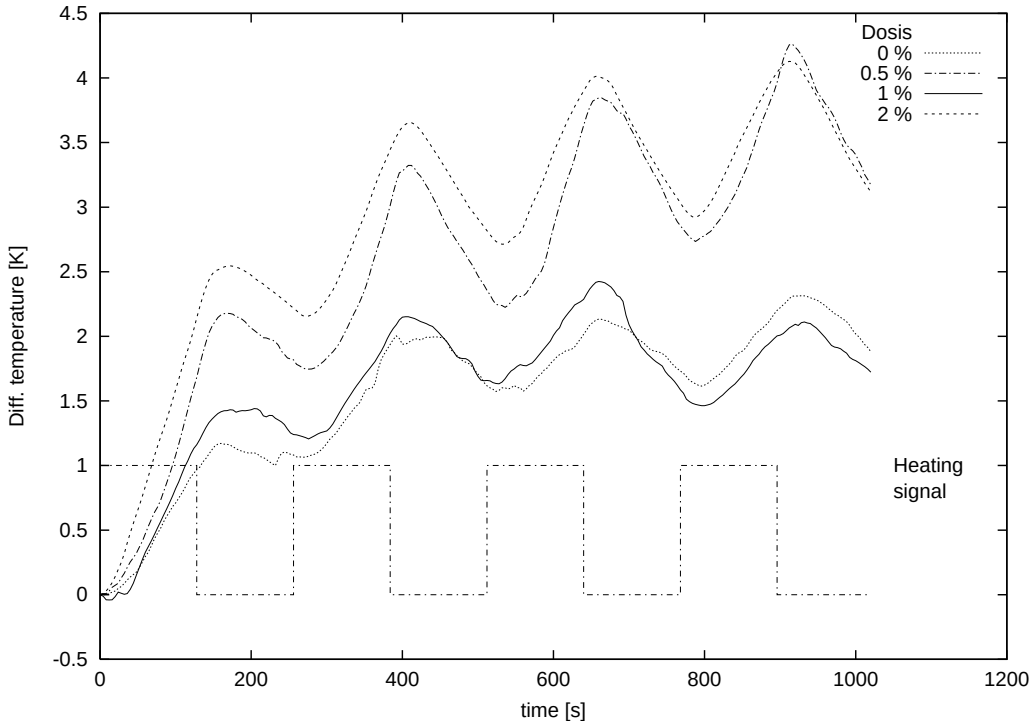



FIG. 5.6: réponse thermique en fonction de la dose de modification

## 5.4 Résultats : sujet n°1

Commençons par quelques observations sur les courbes. Celle liée à la dose de 1 % est particulièrement bruitée et problématique : en effet, après environ 10 minutes, elle se met à décroître et le dernier cycle est plus bas que le cycle précédent. Nous ne l'avons donc pas prise en considération pour les analyses ultérieures.

Nous avons constaté que la première courbe était relativement bruitée. Au niveau des cycles de régime, celui identifié sur la totalité et celui obtenu en retirant la première demi-période étaient assez différents des autres, nous avons donc pris le parti de ne pas prendre en considération la première période. Nous avons repris l'analyse appliquée précédemment et obtenu cette fois quatre jeux de courbes, desquels nous avons extrait la moyenne. Les figures 5.7 et 5.8 illustrent ces courbes.

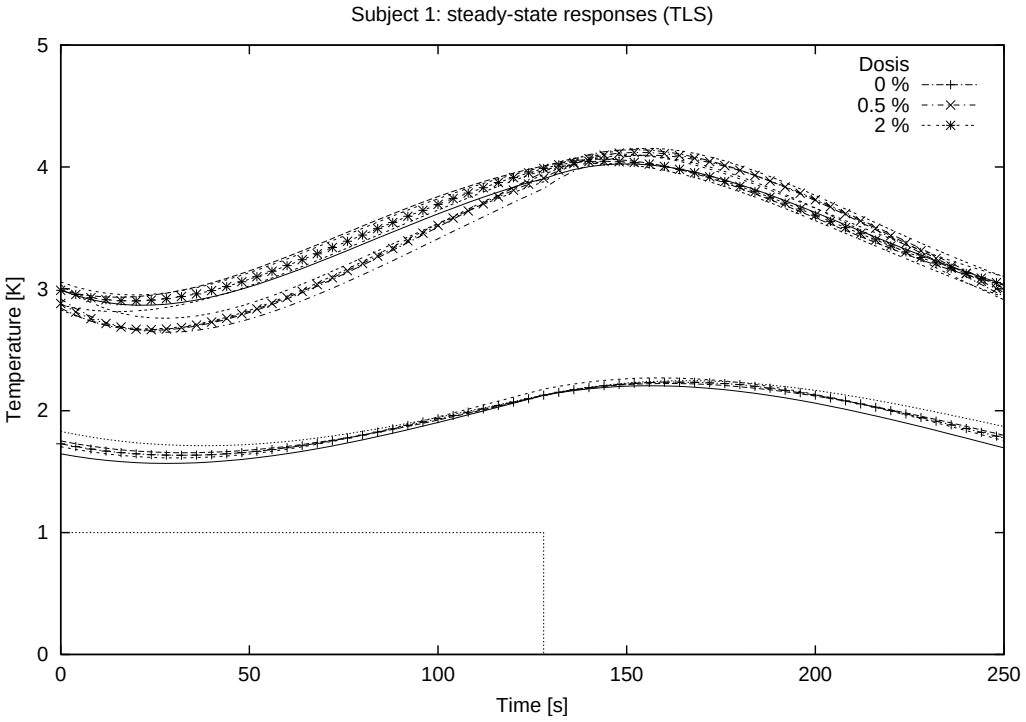



FIG. 5.7: cycle limite en fonction de la dose de modification, méthode TLS

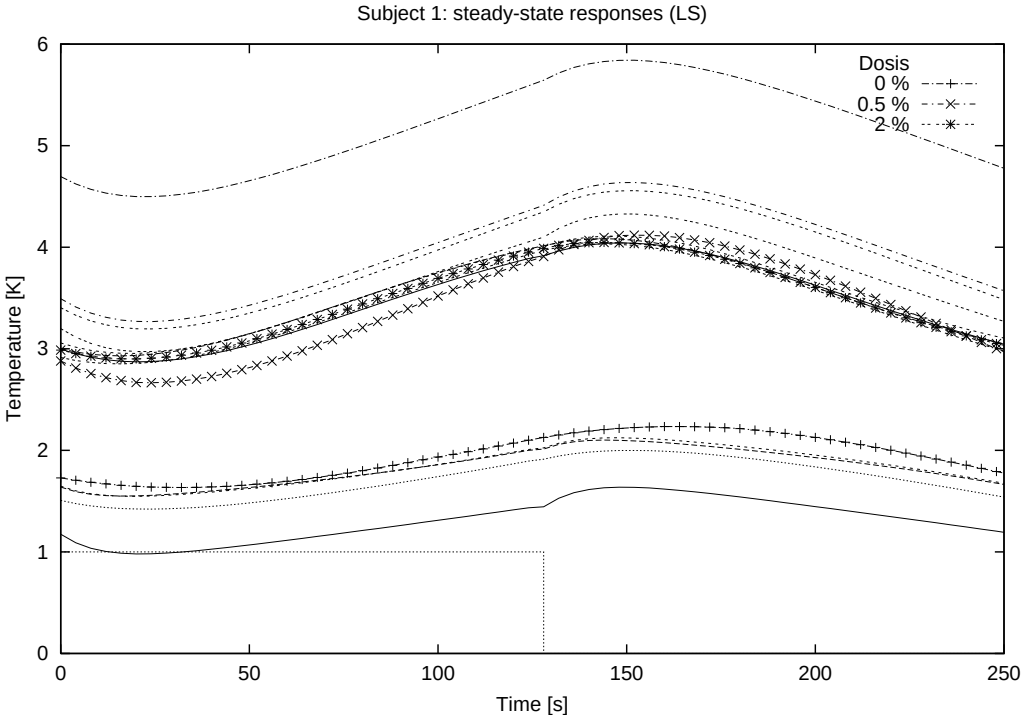



FIG. 5.8: cycle limite en fonction de la dose de modification, méthode LS

Au niveau de la méthode “LS”, les courbes liées aux doses 0.5 % et 2 % s'organisent comme suit : on trouve tout en haut la courbe correspondant à 3 périodes pour la dose 0.5 %, puis viennent en descendant, dans l'ordre, celles correspondant à 2.5, 2, et enfin 1.5 périodes. On trouve ensuite la courbe correspondant à 3 périodes pour la courbe 2 %, puis celles correspondant à des durées plus courtes, dans le même ordre que précédemment.

Au niveau des courbes moyennes, il n'y a pas vraiment de distinction marquée entre les courbes liées aux doses 0.5 % et 2 %, ce qui est logique étant donné que les courbes brutes sont elles aussi fort semblables.

Après application de la méthode de Prony au cycle de régime moyen, nous avons obtenu les résultats repris dans le tableau 5.2.

| Dose | $\sigma_G$ | $\sigma_c$ | $K_s$ | $\dot{y}$ | PTP | TC 1 | TC 2 |
|---|---|---|---|---|---|---|---|
| 0 % | 0.153 | $59\ 10^{-5}$ | 1.95 | 15.3 mK/s | 0.6 K | 205s | |
| 0.5 % | 0.108 | $1.92\ 10^{-3}$ | 3.40 | 35.3 mK/s | 1.43 K | 84 s | |
| 2 % | 0.078 | $92\ 10^{-5}$ | 3.47 | 25.6 mK/s | 1.16 K | 105 s | |

Tab. 5.2: paramètres du cycle de régime moyen en fonction de la dose

Dans cet exemple, toutes les courbes moyennes ont généré des pôles complexes conjugués. Nous avons donné à titre indicatif l'inverse de la partie réelle du nombre $\lambda$, le facteur multiplicatif du temps dans l'argument des exponentielles. Au niveau de l'interprétation, nous observons une différence assez marquée entre la situation de référence et celle correspondant à une modification. En ce qui concerne la réduction de l'écart-type de l'erreur, le gain s'échelonne de 56 pour le deuxième cas à 260 pour le premier.

## 5.5 Perspectives

Au vu des résultats obtenus précédemment, nous pensons qu'une description purement quantitative des courbes, quelle que soit la sophistication du modèle sous-jacent, reste tributaire du bruit présent dans les mesures et, à ce titre, ne sera qu'un reflet partiel de la réalité observée.

La description du phénomène thermique par son cycle de régime nous semble être un complément indispensable, permettant de mettre en évidence des comportements difficilement quantifiables. Par exemple, nous pensons qu'un indice de qualité peut être attribué en observant les différentes réalisations du cycle de régime : une dispersion faible sera l'indice d'une courbe peu bruitée, dont les paramètres n'ont pas varié dans le temps.

Nous proposons la liste suivante pour décrire globalement les propriétés des cycles de régime :

$Ks$ gain statique : il s'agit de la température que le système atteindrait sous l'effet de la composante moyenne du signal d'entrée, ce paramètre donne une idée du couplage thermique entre la source et le récepteur.

$PTP$ amplitude crête-à-crête : elle est conditionnée par la réponse au signal variationnel et sera d'autant plus faible que la bande passante est limitée;

$\dot{y}$ variation maximale du signal : cette valeur s'obtient dans la zone où la variation du signal est linéaire, c.-à-d. pour des valeurs du temps petites, lorsque le développement en série des exponentielles peut se limiter au terme proportionnel au temps;

$TC1$, $TC2$ valeur des constantes de temps, ce qui donne une idée de la dynamique du système.

## 5.6 Conclusions

Cette partie ne contient guère de résultats, et ce n'est pas faute d'avoir accumulé les expérimentations et tentatives de modélisation. Nous attribuons cela à la difficulté de maîtriser les conditions de transfert thermique entre le capteur et le site de mesure.

Dans le domaine électrique, la différence de conductivité entre un élément conducteur ou de valeur de résistance usuelle (inférieure à 100 MΩ) et l'isolant qui l'entoure est telle que l'on peut considérer que « tout » le courant passe par la résistance, l'écart par rapport à la réalité étant difficile à mettre en évidence.

À l'opposé, dans le domaine thermique, il y a d'office une résistance de contact s'opposant au transfert thermique entre l'objet étudié et le capteur de température. Certaines expérimentations nous ont même montré que des réponses thermiques pouvaient résulter de phénomènes de pression!

Le message que nous voulons faire passer est qu'il est essentiel de maîtriser ces phénomènes de transfert, et nous pensons aux points suivants :

- contrôle de la position relative du capteur par rapport à l'objet étudié
- contrôle de la pression exercée
- contrôle de l'état de surface du capteur et de l'objet
- contrôle des conditions de couplage thermique vers l'ambiance
- etc...

Nous pensons notamment qu'il convient de revoir complètement la méthode de fixation du capteur au niveau de l'épiderme. Une alternative potentiellement intéressante serait de fixer le capteur sur un support coulissant verticalement, appliqué avec une pression bien définie en lestant de manière adéquate ce support.

Au niveau de la méthode de Prony, nous pensons que la méthode générique est intéressante à plus d'un point de vue. Dans la méthode classique, l'on considère en fait la réponse indicielle du système, supposé provenir d'un état initial d'équilibre, et passant à un nouvel état d'équilibre sous l'effet de l'application d'un changement du signal d'entrée.

Toute réponse à une perturbation affectant la réponse sera indiscernable de la réponse indicielle, nous songeons ici aux effets de dérive. Si nous répétons plusieurs fois l'expérimentation, nous ne ferons qu'accumuler plusieurs réalisations d'une même réponse indicielle, sans ajouter la possibilité d'ignorer la dérive.

Dans la méthode générique, nous avons un signal d'entrée contenant plus d'information : de ce fait, il devient possible de séparer l'influence du bruit et du signal d'entrée, car ils sont linéairement indépendants. Nous pouvons ainsi obtenir, par la répétition d'un nombre réduit de cycles, une meilleure estimée des caractéristiques du système, en un temps plus court.

Nous avons également montré comment obtenir plusieurs estimées du cycle de régime et que l'application d'une opération de moyenne sur ces estimées permet d'obtenir une courbe qui se modélise plus précisément que le signal initial.

Nous pensons donc que cette méthode présente un intérêt dans le cas de systèmes à commande par tout ou rien, où le temps d'établissement est trop long pour se permettre d'enregistrer une mise en régime.

*Cinquième partie*

---

# ***Conclusions***

Voici venu le moment de conclure, en combinant une vue rétrospective et une vue prospective.

Commençons par l'aspect rétrospectif. Les résultats qui nous semblent les plus importants dans cette recherche sont :

- au niveau instrumental, le développement d'une électronique d'acquisition à haute performances et à faible consommation électrique. Couplé à un environnement logiciel spécifique, ce système fonctionne dans un cadre temps réel strict et distribué.
- en ce qui concerne les algorithmes d'Identification, nous retiendrons surtout la méthode de Prony générique qui permet, à partir d'un signal de commande en créneau, de séparer l'influence des conditions initiales de celle de la commande, en obtenant une solution globale, c.-à-d. qui représente la solution optimale au niveau de la totalité du signal étudié. Nous avons également montré comment obtenir le cycle de régime en procédant à identification sur une partie transitoire du signal
- pour ce qui est de l'évaluation cardio-respiratoire, nous avons balisé les grandes étapes de la méthodologie permettant d'obtenir une série de renseignements sur l'état respiratoire d'une personne à partir d'un équipement de mesure simple et peu contraignant. L'application qui nous semble pouvoir en bénéficier directement est la polysomnographie, mais il n'est pas exclu que d'autres applications soient envisageables, où l'intérêt réside dans ce monitoring permanent.

Passons à l'aspect rétrospectif. Que reste-t-il à faire avec ces résultats ?

- nous avons cité les microcentrales de mesure à microPC. Une telle microcentrale, couplée à une version industrielle miniaturisée de l'interface d'acquisition permettra d'arriver à un des buts fixés au départ, à savoir la réalisation d'équipements autonomes utilisables en mode ambulatoires. Ils seraient autonomes car possédant un système d'enregistrement et de traitement numériques, et utilisables en mode ambulatoires grâce à l'attention apportée aux aspects de consommation électrique
- la méthode de Prony générique nous semble, comme son nom l'indique, tout à fait générique comme méthode d'Identification. En outre, nous trouvons que les

coefficients fournis, à savoir des constantes de temps et des amplitudes d'ondes, sont plus proches du système physique étudié.

Il reste à en étudier plus en détail les propriétés, comme notamment la sensibilité au bruits additifs.

Les applications qui nous semblent les plus prometteuses sont celles ou l'on doit obtenir le cycle de régime d'une réponse, alors que le temps requis pour arriver à ce régime est prohibitif.

– Pour le monitoring cardio-respiratoire, il reste d'une part à acquérir plus d'expérience sur les signatures des différents paramètres et phénomènes physiologiques d'intérêt, ce qui nécessite de réaliser des campagnes de mesure plus étoffées en terme de nombre de cas différents, mais également en terme de paramètres acquis : pour mieux connaître les corrélations entre les variations de pression et les principaux paramètres physiologiques, il nous semble nécessaire de disposer d'autres grandeurs, comme le rythme cardiaque par ECG, le rythme respiratoire, ... Ces campagnes permettront également d'optimiser les méthodes de mesure de façon à offrir la meilleure résistance possible aux bruits et aux artefacts, sans pour cela nécessiter une complexité au niveau du traitement algorithmique incompatible avec un traitement rapide.

Annexe A

# Algorithmes

## Gestionnaire de bas niveau

Voici, à titre d'exemple, comment s'effectue le transit des données entre l'ADC et le SVF.

### Routine `read_adc` :

Cette routine sert à transférer les données accumulées dans les tampons liés à l'ADC aux programmes en mode utilisateur.

- vérification des paramètres, renvoi d'un code d'erreur en cas d'incohérence;
- si l'ADC est déjà en cours d'utilisation, renvoi d'un code d'erreur, sinon le réserver.
- s'il s'agit de la première mesure, regarder si l'ADC doit démarrer immédiatement. Si c'est le cas, le programmer en conséquence.
- vérifier s'il y a des tampons disponibles, sinon renvoyer un code d'erreur;
- vérifier s'il y a une source potentielle d'interruption activée (sinon il n'y aura jamais de données reçues !)
- tant qu'il n'y a pas d'erreurs :
  - tant que le tampon actuel est plein :
    - transférer les données au programme utilisateur;
    - mettre à jour le nombre de données transférées;
    - marquer le tampon actuel comme vide, passer au suivant;
    - si le nombre total d'échantillons transférés est égal à une valeur programmée lors de la phase d'initialisation, arrêter l'ADC et renvoyer ce total; il s'agit d'une sécurité pour s'assurer que toute acquisition finira par se terminer;

        - si le nombre de données demandé par le programme utilisateur est atteint, fin de la routine, en laissant l'ADC actif;
    - attendre, pendant un délai fixé, que des données arrivent.
    - si ce délai est dépassé, ou parce qu'un signal[1] a été reçu, arrêter l'ADC et renvoyer un message d'erreur;
- imprimer une série d'informations sur les registres de la carte;
- réinitialisation;
- renvoyer un code d'erreur.

Les données ont été générées, soit par la routine adc_interrupt décrite au paragraphe suivant, soit automatiquement par transfert DMA[2] en provenance de la carte Lab-PC.

### Routine `adc_interrupt` :

Cette routine est appelée chaque fois que la carte Lab-PC émet une demande d'interruption ET que un des drapeaux de statut lié à l'ADC est actif.

- en cas d'erreur (OVERRUN ou OVERFLOW)
    - afficher l'état de la carte;
    - arrêter l'ADC;
    - désactiver les sources d'interruption;
    - réveiller la routine adc_select en attente d'exception. Si un programme en mode utilisateur a effectué un appel à la routine `select()`, cet appel se terminera. Le programme pourra alors savoir sur quel descripteur de fichier s'est produit le problème.
- en cas de transfert DMA (une interruption est générée pour signaler qu'un bloc de taille pré-programmée a été complètement transféré) :
    - vérifier que le mode DMA était actif, sinon le traiter comme une erreur de type IO_ERROR : en effet, il n'y a pas de raisons pour que cette interruption soit générée si la carte n'a pas été au préalable programmée dans ce mode;
    - mise à jour des compteurs internes;
    - passer au tampon suivant et vérifier qu'il est vide. Si ce n'est pas le cas, le traiter comme une erreur (BUFFER_JAM);
    - informer la carte Lab-PC que sa demande a été traitée, pour qu'elle mette fin à sa demande d'interruption;

1. il s'agit de signaux aux sens Unix, c.-à-d. d'un mécanisme permettant de faire réagir de façon asynchrone un programme utilisateur. La plupart de ces signaux correspondent à une situation d'erreur et le programme qui les reçoit termine son exécution.

2. Direct Memory Access, il s'agit d'un mode de transfert entre un périphérique et la mémoire centrale sans intervention du CPU.

  - réactiver les routines (du gestionnaire) en attente de données. En l'occurrence, il peut s'agir de adc_read() ou adc_select().
- sinon, en cas de donnée disponible (mode normal)
  - tant qu'il y a des données dans le FIFO
    - lire une donnée, la placer dans le tampon
    - si le tampon est rempli, vérifier que le suivant est vide. Si ce n'est pas le cas, le traiter comme une erreur (BUFFER_JAM); sinon passer à ce tampon et réactiver les routines en attente de données;
    - relire le statut de l'ADC. En cas d'erreur (OVERRUN ou OVERFLOW), effectuer le même traitement qu'au début de la routine
- sinon, signaler que la routine a été appelée sans cause d'interruption détectée, et afficher l'état de la carte.

## Fonctionnement du gestionnaire

Cette partie est nettement plus spécialisée et contient l'implémentation des algorithmes exposés plus haut, ainsi que des structures de données associées.

Les paragraphes sont organisés comme suit : on trouvera tout d'abord un extrait du code source, en langage C. Directement après ce code source vient une explication des étapes principales.

daq_interrupt.c

```
/*
 * The Interrupt Service Routine. Note that interrupt lines can be
 * shared between multiple LabPC cards
 */

#if defined __RT__
static void daq_rtinterrupt( void )
#else
static void daq_interrupt( int irq, void *dev_id,
                           struct pt_regs *registerp )
#endif
{
        LabPC_t *board = (LabPC_t *)dev_id;
        int     done, i, status, dio_status;

        i = 0;
        for (i = MAXINTERRUPTLOOP; i > 0; i--) {
                /*
                 * Small changes : removed some elses. Now this
                 * routine will service more than one source if
                 * simultaneous requests.
                 */
                status = readport( board->regs.stat ); done = 0;
```

```
                if (status & CNTINT) {
                        dac_interrupt( board->dac, NUM_DACS );
                        done = 1;
                }
                /* As the adc recovery may be done by dio_interrupt */
                if (status & (ADC_OVERRUN | ADC_OVERFLOW )) {
                        printcons("problem, status 0x%x, calling err_handle\n",
                                        status );

                        board->adc->err_handle( board->adc,
                                                status &
                                                (ADC_OVERRUN | ADC_OVERFLOW));
                    status &= ~(DMATC | DAVAIL); /* Just in case */
                    printcons( "err_handle called, status %#x\n",
                               status );
                    dprintregs( board );
                    done = 1;
                }
                if ((board->regs.cmd3.value & DIOINTEN) &&
                    ((dio_status = inb( board->dio->status_port ))
                         & DIO_INTERRUPT)) {
                        board->dig_isr( board->dio, dio_status );
                    done = 1;
                }

                if (status & DMATC) {
                    adc_DMATC_interrupt( board->adc, status );
                    done = 1;
                } else if ((status & DAVAIL) && !board->adc->dmaen) {
                    adc_interrupt( board->adc, status );
                    done = 1;
                }
                if (0 == done)
                /* Exit if nothing done after scanning all sources */
                  break;
        }
        if (i < 1) {
        printcons("ISR looped %d times: status = 0x%x - terminating\n",
                   MAXINTERRUPTLOOP, status );
        }
}
```

daq_interrupt.c

Les grandes étapes sont :

1. en ligne 23, lecture d'un registre contenant le statut de la carte. Les bits à "1" correspondent aux dispositifs activés.
2. en ligne 24, on teste si le DAC est prêt à émettre une donnée, et dans ce cas on appelle sa routine de gestion.
3. en ligne 29, s'il y a eu une erreur au niveau de l'ADC, une routine de correction, destinée à réinitialiser l'acquisition, est activée.

4. en ligne 41, le statut associé au DIO est testé. À titre de vérification, le statut du DIO lui-même (circuit de type i8255) est interrogé. Une routine est appelée via un pointeur de fonction, cette routine pouvant être la routine standard ou une routine spécialisée. Nous avons rajouté cette option afin de pouvoir considérer le DIO comme formant un tout, c.-à-d. une interface de communication vers le convertisseur sigma-delta.
5. en ligne 49, on regarde si un transfert par DMA vient de se terminer pour l'ADC. Dans ce cas, la routine activant le transfert suivant est exécutée.
6. en ligne 51, si une donnée est disponible pour l'ADC, et s'il n'utilise pas le transfert par DMA, la routine de lecture est activée.
7. une variable `done` permet de vérifier qu'au moins un événement lié à la carte a pu être traité. Si ce n'est pas le cas, on sort de la boucle principale.
8. tout cela est encapsulé dans une boucle : si, suite à l'appel d'une des fonctions mentionnées, une nouvelle interruption se produit, elle sera traitée immédiatement, et non plus par le mécanisme de gestion des interruptions contenu dans le noyau. Néanmoins, afin de ne pas monopoliser le processeur durant un temps trop long, le nombre de passages est limité.

La logique de balayage est la suivante :

- la priorité la plus élevée est donnée au DAC. La raison principale est qu'il ne contient qu'un simple mécanisme de double tampon entre le bus de l'ordinateur et le circuit proprement dit. Comme c'est la première routine appelée, sa latence est minimale. Ainsi, il est possible de synchroniser un dispositif externe sur le DAC, et de réaliser simultanément une émission et une acquisition de données sans qu'il y ait d'incertitude sur les instants d'échantillonnage ni de perte d'échantillons.
- les erreurs de l'ADC sont ensuite traitées. En effet, nous avons implémenté la possibilité de corriger une erreur sur l'ADC dans la routine gérant le DIO.
- vient ensuite le DIO, qui n'a aucune mémoire et ne peut lire qu'une donnée à la fois.
- l'ADC contenant un FIFO de 16 éléments est interrogé en dernier. En effet, si aucune suite n'est donnée à une demande de lecture, le résultat suivant sera acquis et stocké correctement. Nous avons modifié la routine initiale pour que la totalité du FIFO soit lue au cours d'un passage dans la routine d'interruption.

En ce qui concerne les routines d'interruption, le contenu est chaque fois plus ou moins similaire. En effet, réaliser une acquisition ou une émission revient à lire les registres de statut de la carte, et à effectuer des transferts entre la mémoire du PC et certains registres de la carte. Avant d'illustrer le code, explicitons la structure des données, telle que définie dans le fichier common.h :

common.h

```
typedef struct DmaBuffer_t {
    struct DmaBuffer_t *next;    /* Circular linked list      */
```

```
    short  * const buffer;      /* The data buffer      */
    short  *bufp;               /* Next data to xfer in buffer  */
#if 0
    short   *fillp;             /* Next data to xfer by ISR */
#endif
    short * const lastp;        /* Last place in the buffer */
    const unsigned int  size;   /* Buffer size (in words)   */
    unsigned int    count;      /* Number of words in buffer    */
    volatile int    inuse;      /* Is the buffer busy ?     */
} DmaBuffer_t;

/* I guess it is faster to unconditionally compute the diff, and
   add it to the buffer rather than testing if it is !=0 before
   doing addition */

extern inline void adjust_bufp(DmaBuffer_t *ptr)
{
    if (ptr->size < ptr->count) {
        printk("LabPC+ adjust_bufp : size is < count !!!  \n");
        ptr->bufp = ptr->buffer;
    } else
        ptr->bufp = ptr->buffer + (ptr->size - ptr->count);
}

/* This structure may be appended to a block of data to track
   missed interrupts */

typedef struct Guard_t {
    struct timeval  timestamp;  /* Last data acquisition time   */
    long long   jiffies;
    unsigned int    missed;     /* To keep missed count     */
} Guard_t;

typedef struct PortBuffer_t {
    DmaBuffer_t *buflist;       /* List of dma buffers      */
    DmaBuffer_t *head;          /* Buffer in list for xfer  */
    DmaBuffer_t *tail;          /* Buffer in list for emptying  */
    unsigned int    bufsize;    /* Size of buffers      */
    unsigned int    nbuf;       /* Number of buffers        */
    volatile unsigned int nused;    /* Number of used buffers   */
    unsigned long   residue;    /* Words left to xfer       */
    unsigned long   count;      /* Words left to user xfer  */
    unsigned long   total;      /* Starting number of samples   */
    const unsigned int imask;   /* Enabling interrupt source    */
    const unsigned int smask;   /* Ready for read/write ?   */
    unsigned int    emask;      /* Enabled status source    */
#if defined SYNC_USE_WAKEUPS
    volatile unsigned int wakeups;/* Number of waiting processes */
#endif
    struct wait_queue *wait;    /* Processes sleeping on port   */
    struct wait_queue *exwait;  /* Exception handlers       */
    struct Counter_t  *counter; /* hooked counter, sanity check */
```

```
    int        needtime;          /* May we put a timestamp ? */
    struct semaphore sem;         /* Device contention avoidance  */
    void         *priv;           /* Generic hook for suppl. data */
} PortBuffer_t;

/* Struct for accessing the LabPC registers via the I/O ports */
typedef struct DaqPort_t {
    int value;                    /* Saved value of this register */
    int address;                  /* Port address of this register*/
} DaqPort_t;
```

common.h

**Structure** `DmaBuffer_t`

Comme son nom l'indique, il s'agit d'une structure permettant la gestion d'un tampon utilisable via des accès de type DMA. Des accès séquentiels sont bien entendu aussi possibles.

Par rapport à l'implémentation originale, nous avons apporté deux modifications :

1. les structures de gestion des tampons forment à présent une liste chaînée, contrairement à l'implémentation initiale où l'on avait un tableau contenant un certain nombre d'exemplaires de la structure `DmaBuffer_t`. Le balayage consistait à vérifier que l'on n'était pas au bout du tableau, et à revenir au début dans le cas échéant. L'avantage de notre approche est d'une part que le passage d'un tampon au suivant requiert systématiquement le même nombre d'opérations, et d'autre part que l'on peut ajouter dynamiquement des tampons, par modification de l'enchaînement des éléments.
2. nous avons introduit un pointeur, `lastp`, pointant vers le dernier élément utilisable. Pour tester si un buffer est rempli, nous avons remplacé le code
   ```
   if (bufp >= buffer + size)
   ```
   qui nécessite une addition et une comparaison, par
   ```
   if (bufp >= lastp)
   ```
   qui ne demande qu'une seule comparaison.

Les éléments de cette structure sont :

**next** est un pointeur vers la structure `DmaBuff_t` suivante.

**buffer** est un pointeur vers le tampon contenant les données.

**bufp** est un pointeur vers l'élément en cours d'utilisation, qui est un entier court (16 bits)

**fillp** est un pointeur similaire, que nous avons remplacé au profit de l'élément suivant:

**lastp** est un pointeur vers le dernier élément utilisable dans le tampon.

**size** est le nombre d'éléments que peut contenir le tampon

**count** est le nombre d'éléments à transférer via ce tampon, qui peut être plus petit que `size`

**inuse** est un drapeau permettant de savoir si le tampon est actuellement en cours d'utilisation.

Pourquoi avons nous supprimé `fillp`? Il faut savoir que dans le noyau, la mémoire est allouée via des blocs de taille prédéfinie : 32 octets, 64 octets, ... Avec `fillp`, la structure aurait nécessité 36 octets, et en aurait occupé 64. Au vu des contraintes d'économie de mémoire, cette opération nous a permis de ramener la taille à 32 octets.

#### Fonction `adjust_bufp()`

Du fait que nous avons supprimé le pointeur `filp`, il est devenu nécessaire de réinitialiser correctement bufp après avoir rempli un tampon. Tel est le rôle de cette fonction : après un appel, `bufp` pointe vers le premier emplacement utilisable, ce qui dépend du nombre d'éléments effectivement utilisés.

#### Structure `Guard_t`

Nous avons crée cette structure permettant d'ajouter à un tampon une série d'informations pour pouvoir vérifier le bon déroulement d'une acquisition. Les éléments en sont :

**timestamp** qui contient le nombre de secondes écoulées depuis minuit, au moment de l'acquisition de la dernière valeur, afin de situer précisément l'instant d'acquisition;

**jiffies** qui permet d'échantillonner une variable du noyau du même nom. Cette variable est mise à jour au moins 100 fois par seconde (plus avec RT-Linux). Cela permet de calculer la différence de temps entre deux tampons successifs avec une résolution bien supérieure à la seconde;

**missed** est le nombre total d'éléments perdus que certaines routines d'interruptions peuvent mettre à jour lorsqu'elles détectent ce genre de problème.

#### Structure `PortBuffer_t`

Cette structure assez complexe contient toute l'information sur un « port », à savoir un des dispositifs d'entrée/sortie. Dans le cas de transferts bidirectionnels, il est nécessaire d'utiliser deux structures, une pour chaque sens. Cette structure permet de gérer les tampons liés à un transfert, que ce soit de la carte Lab-PC vers la mémoire du device driver ou de cette mémoire vers les programmes utilisateurs à travers le SVF.

Cette structure contient les éléments suivants :

**buflist** est un pointeur vers les structures de contrôle des tampons d'acquisition;

**head, tail** : ces tampons sont utilisés deux fois : une application y écrit des données, une autre les relit ultérieurement.Par exemple, pour l'ADC, la routine d'interruption écrit ce qui a été lu dans le FIFO ou qui a été transféré par DMA, et la fonction faisant partie du SVF lit ce qui est disponible et le renvoie vers le programme final. Le principe est d'écrire via `head` et de lire via `tail`. Ces pointeurs ne peuvent pas se rattraper, ce qui indiquerait une saturation des tampons (tous les tampons sont utilisés) ou un délai de réponse trop grand (tous les tampons sont vides);

**bufsize** est la taille d'un tampon;

**nbuf** est le nombre de tampons en cours d'utilisation;

**residue, count, total** sont les nombres de données respectivement restant à écrire, restant à lire, et totaux à transférer;

**imask** est un masque indiquant quelles sont les sources pouvant générer une interruption

**smask** est un masque indiquant le statut du transfert;

**emask** est un masque indiquant quelles sont les sources d'interruption activées;

**wait** est un pointeur vers une queue d'attente, permettant à un programme d'utilisateur d'être réactivé lorsque des données sont disponibles;

**exwait** permet à un programme utilisateur d'être réactivé en cas d'exception (erreur d'entrée/sortie)

**counter** est un pointeur vers une des structures de gestion des compteurs, pour l'utiliser conjointement avec ce port. Le nombre d'éléments transférés, déduit par logiciel, doit théoriquement être égal à la valeur du compteur, qui sera câblé de façon à être activé à chaque demande d'interruption. Si ce n'est pas le cas, il est possible, par différence entre ce nombre et la valeur du compteur, d'en déduire combien d'éléments ont été perdus.

**needtime** est un drapeau indiquant s'il est nécessaire ou non de mettre à jour le champ `timestamp` décrit plus haut;

**sem** est un sémaphore, permettant de sérialiser les accès à un port : par essence, un seul programme utilisateur peut accéder à un port;

**priv** est un pointeur pour une structure de données supplémentaire, ce qui permet de prendre en compte les particularités de chaque type de port.

#### Structure `DaqPort_t`

Cette structure permet de résoudre un problème avec certains registres de la carte Lab-PC, qui ne fonctionnent qu'en écriture. Elle sert à conserver en mémoire la dernière valeur écrite, qui est également la valeur présente dans ces registres.

## Protection des zones critiques

Certaines parties de code manipulant des registres de la carte peuvent provoquer instantanément la génération d'une interruption, ce qui aurait pour effet de rentrer prématurément dans la routine associée. Le code original se prémunissait contre cette éventualité comme suit :

```
cli();
Start_ADC();
sti();
```

où `cli()` et `sti()` sont deux fonctions du noyau permettant respectivement d'inhiber et d'autoriser les demandes d'interruption. Le problème de cette approche est que les interruptions sont systématiquement réactivées, quelle que soit la situation antérieure.

Nous avons opté pour l'approche suivante :

```
{
  unsigned long flags;

  save_flags(flags); cli();
  Start_ADC();
  restore_flags(flags);
}
```

Dans ce cas, les fonctions `save_flags()` et `restore_flags()` permettent de sauvegarder dans une variable et de restaurer à partir d'une variable le registre d'état du processeur. Dans cette approche, les drapeaux d'état sont préservés, et la partie critique du code est protégée de façon adéquate.

## Routine d'interruption

Voyons à présent le corps de la routine d'interruption que nous avons spécialement développée pour gérer le sigma-delta.

ext_interrupt.c

```
void ext_interrupt ( Dio_t *dio, int status)
{
  __label__             EXT_ISR_exit;
  volatile int          loop;
  unsigned short        low;
  Ext_t                 *ext  = dio->board->ext;
  DaqPort_t             *port = NULL;
  PortBuffer_t          *dio_sync = NULL, *ext_sync = NULL;

  /*
   * This is the core acquisition routine. It is a very
   * simplified and customised version of dio_interrupt.
```

```
 * Amongst the specificities, we may notice it never
 * suspends interrupts, and it is responsible to select
 * next Sigma-Delta channel to read.
 */

loop  = status & dio->interrupt_mask;

if (0 == loop) {
  /* Test if we get an interrupt for port B */
  if (status & DIO_INTRB) {
    printcons( "%s", "IRQB received, disabling it\n" );
    port  = dio->port + EXT_IO;
    port->value &= ~((1 << EXT_SCLK) | (DIO_INTRB));
    /* Disable interrupt generation */
    ISR_writeport( dio->mode_port, EXT_SCLK << 1 );
    /* Write a '0' to this source */
    ISR_writeport( dio->mode_port, DIO_INTRB << 1 );
    return;
  }

  printcons( "IRQ for no good reason, status 0x%x\n", status );
  dio_spuriousIRQ( dio, status, loop );
  goto EXT_ISR_exit;
}

dio_sync  = dio->sync + 2*EXT_IN + DIO_INSYNC;

ext_sync  = ext->sync; /* + ext->chanidx; */

if (0 == (ext->chanstat & CS_STATUS_ACTIVE)) {
  /* This is a glitch -- just purge buffer for now */
  port = dio->port + EXT_IN;

  if (status & dio_sync->emask) {
    low = inb( port->address );
  }
  return;
}

/* Show we're in the interrupt routine */
port  = dio->port + EXT_OUT;
ISR_setbits( *port, EXT_PERMEAS );
/* setbits( port, EXT_PERMEAS ); */


if (loop & dio_sync->imask) {
  /* Read Sigma-Delta value */
  port = dio->port + EXT_IN;

  if (status & dio_sync->emask) {
    low = inb( port->address );
    ext_nread++;
```

```
        port = dio->port + EXT_OUT;
        ISR_clearbits( *port, EXT_PERMEAS );

#if defined __i386__
        asm volatile ("xorl %%eax,%%eax\n\t"
                      "inb %%dx,%%al\n\t"
                      "movl %%eax,%0\n\t"
                      "andl %%ebx,%%eax\n\t"
                      "jnz 2f\n\t"
                      "movl $3,%%ecx\n\t"
                      ".align 4\n"
                      "1:\t"
                      /* "xorl %%eax,%%eax\n\t"
                             not necessary, as eax IS 0! */
                      "inb %%dx,%%al\n\t"
                      "movl %%eax,%0\n\t"
                      "andl %%ebx,%%eax\n\t"
                      "jnz 2f\n\t"
                      "decl %%ecx\n\t"
                      "testl %%ecx,%%ecx\n\t"
                      "jne 1b\n"
                      "2:\t"
                      :"=m" (status), "=a" (loop)
                      :"d" (dio->status_port), "b" (dio->interrupt_mask)
                      :"%ecx","cc","memory");
#else
        {
          register int i = 4;
          do {
                  /*
                   * If first reply is negative, ask three time, to be really
                   * sure ;-) Without this test, it could return too early,
                   * and we have to wait upper layer or next interrupt before
                   * going active again.
                   */

            if (0 != (loop = (status = inb( dio->status_port )) &
                      dio->interrupt_mask))
              break;
          } while (--i > 0);
        }
#endif
        ISR_setbits( *port, EXT_PERMEAS );
        port = dio->port + EXT_IN;

        if (0 == (status & dio_sync->emask)) {
          /* Failed to read second value */
          printcons( "%s", "Unable to read second part\n" );
          /* If it doesn't come up right, we wait a bit
             and read anyway */
          udelay(100);
```

```
        }
#if defined __i386__
        /*
         * as gcc doesn't pretend to recognise those operations
         * are dealing with bytes and avoid the shifting and the
         * oring, we do "The Right Thing(tm) in asm
         */
        asm volatile ("inb %%dx,%%al\n\t"
                      "movb %%al,%%ah\n\t"
                      "movb %1,%%al\n\t"
                      :"=a" (*(ext_sync->head->bufp++))
                      : "m" (low), "d" (port->address) : "memory");
#else
        high = inb( port->address );
        *(ext_sync->head->bufp++) =         low |  (high << 8);

#endif

        ext_nread++;

        if (ext->nbits > 16) {
#if defined __i386__
          asm volatile ("xorl %%eax,%%eax\n\t"
                        "inb %%dx,%%al\n\t"
                        "movl %%eax,%0\n\t"
                        "andl %%ebx,%%eax\n\t"
                        "jnz 2f\n\t"
                        "movl $3,%%ecx\n\t"
                        ".align 4\n"
                        "1:\t"
                        "inb %%dx,%%al\n\t"
                        "movl %%eax,%0\n\t"
                        "andl %%ebx,%%eax\n\t"
                        "jnz 2f\n\t"
                        "decl %%ecx\n\t"
                        "testl %%ecx,%%ecx\n\t"
                        "jne 1b\n"
                        "2:\t"
                        :"=m" (status), "=a" (loop)
                        :"d" (dio->status_port), "b" (dio->interrupt_mask)
                        :"%ecx","cc","memory");
#else
          {
            register int i = 4;
            do {
                  /*
                   *  If first reply is negative, ask three
                   *  time, to be really sure ;-) Without this
                   *  test, it could return too early, and we
                   *  have to wait upper layer or next
                   *  interrupt before going active again.
                   */
```

```
            if (0 != (loop = (status = inb( dio->status_port )) &
                      dio->interrupt_mask))
              break;
          } while (--i > 0);
        }
#endif
        if (0 == (status & dio_sync->emask)) {
          /* Failed to read third value */
          printk(KERN_ERR "Lab-PB+ %s:%s unable to read third part\n",
                 __FILE__,__FUNCTION__ );
        }
        *(ext_sync->head->bufp++) = inb( port->address );
        ext_nread++;
        /* Do you know about 32 bits Sigma-Delta ? May be one day
           ... */
      } /* ext->nbits > 16 */

  /* Now check the buffer, wakeup sleeping processes, ... */

      if (ext_sync->head->bufp >= (short *)ext_sync->head->lastp) {
        DmaBuffer_t *head_saved = ext_sync->head; /* keep a backup */

                /* Put ASAP a timestamp at the tail of this buffer */
        if (ext_sync->needtime) {
          ((Guard_t *)ext_sync->head->lastp)->jiffies =
#if defined __RT__
            rt_get_time()
#else
            jiffies
#endif
            do_gettimeofday( &((Guard_t *)ext_sync->head->lastp)
                             ->timestamp );
        }
        ext_sync->head->inuse = 1;
        ext_sync->nused++;
        ext_nbuf++;
        low = ext_sync->residue;
        if (ext->nbits > 16)
          ext_sync->residue -= ext_sync->head->count >> 1;
        else
          ext_sync->residue -= ext_sync->head->count;

                /* Get the next data transfer buffer */
        ext_sync->head = ext_sync->head->next;
                /* setup next transfer */
                /* As residue is unsigned, testing if >0 makes no sense */
        if (ext_sync->residue)  {
          /* Check the back buffer has been emptied */
          if (ext_sync->head->inuse != 0) {
            printcons( "%s", "IRQ: no buffer available\n" );
            dprintregs( dio->board );
```

```
    wake_up( &(ext_sync->exwait) );
    /*
     *  In a multi-channel acquisition scheme,
     *  isn't it problematic to go further ?
     */
    if (ext->scan) {
      printcons( "%s", "Scanning mode, giving up\n" );
      goto EXT_ISR_exit;
    }
    if ((++((Guard_t *)ext_sync->head->lastp)->missed) > 2 ) {
      printcons( "%s", "Too many failures, giving up\n" );
      goto EXT_ISR_exit;
    }
    /* Revert to preceeding situation. */
    ext_sync->head = head_saved;
    ext_sync->nused--;
    ext_sync->residue = low;
    ((Guard_t *)ext_sync->head->lastp)->missed++;
    if (ext->nbits > 16)
      /* go four bytes (two shorts) backward */
      ext_sync->head->bufp -= 2;
    else
      /* go two bytes backward */
      ext_sync->head->bufp--;
  } else {

    /* Reset bufp for preceeding buffer */
    adjust_bufp( head_saved );

    /* Setup next buffer */
    ((Guard_t *)ext_sync->head->lastp)->missed = 0;
    /* Not yet */
    ext_sync->head->count =
      (ext_sync->residue < ext_sync->head->size) ?
      ext_sync->residue : ext_sync->head->size;
    adjust_bufp( ext_sync->head );
  }
} else {
  /* Reset bufp for preceeding buffer */
  adjust_bufp( head_saved );

  /* prevent other interrupts to occur */
  dio->interrupt_mask &= ~(dio_sync->imask);
  dio_sync->emask = 0;
  writeport( dio->mode_port,
             ((dio_priv_t *)dio_sync->priv)->flipflop );
}

        /* One buffer processed. Woke up awaiting processes */
        /*
         * As the acquisition process is rather slow AND we can't
         * loose sync, we don't disable interruptions. It is the
```

```
                * responsability of the upper layers to have enough
                * buffering
                */
        if (ext_sync->wait) {
          wake_up( &(ext_sync->wait) );
        }
      }/* (ext_sync->head->bufp >= ext_sync->head->lastp) */

  /*
   * Update Sigma-Delta channel to be OK at NEXT read. CS1390 will
   * update its port at the falling edge of CS
   */
      ext->chancount++;
      if        (ext->chancount > ext->more[ ext->chanidx] ) {
        ext->chancount = 1;
        if (ext->scan ) {
          if (ext->chanidx > 0)
            ext->chanidx--;
          else
            ext->chanidx = ext->channel;
        }
      }
      /* The CS1390 must be reprogrammed after each measurement */
      ext_cs1390(ext, 0);
    } else {
      printcons( "%s", "PORTA IBFA not set ??\n" );
      /* Nothing available ?? */
      goto EXT_ISR_exit;
    }
  } else {
    printcons( "%s", "INTRA not set ??\n" );
    /* Why this interrupt ? */
    goto EXT_ISR_exit;
  }

  /* Now, we restart the adc, if needed */
  loop = atomic_read( &(ext->adc_restart.count) );
  if (loop) {
    Adc_t *adc = ext->board->adc;
    PortBuffer_t *adc_sync = adc->adc_sync;
    register int nchan = adc->channel + 1;

    loop = adc_sync->head->bufp - adc_sync->head->buffer;
    if (adc->scan) {
      /* Were did it stop ? */
#if defined __RT__
      long long jnow = rt_get_time();
      printcons( "resuming at 0x%lx%lx RT_ticks, xtime %d\n",
                 (long )(jnow >> 32), (long)jnow, xtime.tv_sec );
#else
      printcons( "resuming at %ld jiffies , xtime %d\n",
                 jiffies , xtime.tv_sec );
```

```
#endif

      printcons( "sample(s) present in this buf : %d; nchan %d\n",
                 loop, nchan );

      loop = loop % nchan;

      printcons( "re-enabling ADC, %d sample(s) lost\n",
                 nchan - loop );
      do {
                /*
                 * There is no overflow here, as we're filling one scan,
                 * and buffers have a size multiple of numchan
                 */
        *(adc_sync->head->bufp++) = ADC_OUTOFRANGE + loop;
      } while (++loop < nchan);
    } else {
#if defined __RT__
      long long jnow = rt_get_time();
      printcons( "resuming at 0x%lx%lx RT_ticks, xtime %d\n",
                 (long )(jnow >> 32), (long)jnow, xtime.tv_sec );
#else
      printcons( "resuming at %ld jiffies , xtime %d\n",
                 jiffies , xtime.tv_sec );
#endif
      printcons( "sample(s) present in this buf : %d; nchan %d\n",
                 loop, nchan );
      printcons( "%s", "re-enabling ADC, 1 sample lost\n" );

      *(adc_sync->head->bufp++) = ADC_OUTOFRANGE;

    }

    if (adc_sync->head->bufp >= (short *)adc_sync->head->lastp)
      adc_check_buffer( adc, adc_sync );

    /*
     * In channel scaning mode, looks like we must reprogram the
     * scanning sequence
     */
    if (adc->scan) {
      ISR_clearbits( adc->board->regs.cmd1, SCANEN );
      /* Channel Scanning is ON */
      if (adc->intscan) {
                                /* Also Interval Scanning is ON */
#if defined LAB_PC_PLUS
        if (adc->intvlclock == 0)
          /* Extern. Gate for Interval Scanning */
          ISR_setbits( adc->board->regs.cmd4, (EOIRCV | INTSCAN) );
        else
          ISR_setbits( adc->board->regs.cmd4, INTSCAN );
#else
```

```
        printcons( "%s", "This operation only supported by LabPC+\n" );
#endif
      } else {

        ISR_setbits( adc->board->regs.cmd1, SCANEN );
      }
    } else if (adc->intscan) {
#if defined LAB_PC_PLUS
      ISR_clearbits( adc->board->regs.cmd1, SCANEN );
      /* Single Channel Interval Acquisition Mode */
      WriteIntervalCounter( adc, adc->intvlcount );
      ISR_setbits( &(adc->board->regs.cmd4), INTSCAN );
      if (adc->intvlclock == 0)
              /* Extern. Gate for Interval Acquisition Mode */
        ISR_setbits( adc->board->regs.cmd4, (EOIRCV | INTSCAN) );
      else
        ISR_setbits( adc->board->regs.cmd4, INTSCAN );
#else
      printcons( "%s", "only supported for LabPC+\n" );
#endif
    }

    /* EXTGATA0 is lost. Now we must use software trigerring */
    adc->triggermode = ADC_SOFTTRIGGER;
    ISR_setbits( adc->board->regs.cmd2, SWTRIG );
    /* re-enable interrupts */
    if (adc->dmaen) {
      EnableDMAInterrupts( adc );
    } else {
      EnableADCInterrupts( adc );
    }
    loop = atomic_read( &(ext->adc_restart.count) );
    down( &(ext->adc_restart) );
    printcons( "setting adc_restart from %d to %d\n",
              loop, atomic_read( &(ext->adc_restart.count) ) );

  }
  /*
   * Having read the Sigma-Delta, nothing should happen for a few
   * milliseconds
   */

  /* Show we're off the interrupt routine */
  port  = dio->port + EXT_OUT;
  ISR_clearbits( *port, EXT_PERMEAS );

  return;

 EXT_ISR_exit:
  /* Something went wrong, disable interrupts */
  {
    Dio_t *dio = ext->board->dio;
```

```
    SuspendDioTransfer( dio, dio->sync + 2 * EXT_IN + DIO_INSYNC );
    up( &(dio->restart) );
  }

  port    = dio->port + EXT_OUT;
  ISR_clearbits( *port, EXT_PERMEAS );

  printk(KERN_CRIT "Lab-PB+ %s:%s ISR problem, "
         "disabling interrupts\n",__FILE__,__FUNCTION__ );
  printk(KERN_CRIT "Lab-PB+ %s:%s total %ld, count %ld, "
         "residue %ld\n",__FILE__,__FUNCTION__ ,
         ext_sync->total, ext_sync->count, ext_sync->residue);

  printregs( dio->board );
  wake_up( &(ext_sync->exwait) );
}
```

ext_interrupt.c

Voyons-en le fonctionnement section par section :

- aux lignes 20 à 36, on regarde quelle est le la source d'interruption. L'interface étant prévu pour fonctionner sur le port A, on prend les mesures suivantes si on rentre dans le gestionnaire pour une autre raison :
  - si l'interruption provient du port B, il est reprogrammé pour ne plus générer d'interruption;
  - sinon, on passe par une routine qui va afficher la totalité de l'état de la carte, à savoir les demandes d'interruption actives et l'ensemble des registres.
- aux alentours de la ligne 40, on prépare deux pointeurs vers des structures de synchronisation, un pour le DIO, un pour le port EXT.
- entre les lignes 41 à 50, on vérifie si une acquisition a démarré via le sigma-delta. Si ce n'est pas le cas, l'interruption résulte d'un parasite (“glitch”), et on lit la valeur (sans l'utiliser) afin de libérer le tampon d'entrée.
- afin de permettre de visualiser l'activité de cette routine, le code entre les lignes 51 et 55 met un accès du port B à “1” avant de lire le port A.
- une première lecture du port A a lieu en ligne 63, précédée de test pour voir si l'interruption correspond effectivement aux sources activées. Cette lecture donne les 8 bits de poids faible du résultat.
- l'interface est prévu pour générer une seconde demande de lecture dès la fin de la première pour transférer les bits de poids fort. Au niveau de la routine, nous avons pris la décision de faire du « polling », c.-à-d. de lire plusieurs fois de suite le statut. Afin de minimiser le délai de cette boucle de polling, nous avons écrit une version en langage d'assemblage pour processeur Intel386. En cas de compilation sur un autre type de plate-forme, le code équivalent en C est choisi par un mécanisme d'inclusion conditionnelle. Nous avons pris cette décision après inspection du code assembleur généré par le compilateur, qui contenait une série

de transferts en mémoire inutiles ou redondants. Cette boucle se trouve entre les lignes 69 et 104

- en ligne 105 et 106, on remet à "0" l'accès du port B pour indiquer que l'on est sorti de la boucle de polling
- entre les lignes 108 et 113, on vérifie que le port A est effectivement prêt pour un second transfert;
- entre les lignes 115 et 130, une seconde lecture est effectuée, et les deux résultats partiels concaténés en un seul, sur 16 bits. Ici encore, le code a été récrit en assembleur, le compilateur n'ayant pas détecté que seuls les 8 bits les moins significatifs des registres contenaient de l'information : le code généré comportait des opérations sur des registres de 32 bits.
- entre les lignes 133 et 181 se trouve une portion de code permettant de répéter cette opération de lecture, ce qui permet d'utiliser des convertisseurs de plus de 16 bits : dans ce cas, après un nouveau polling, le code lit un troisième octet.
- à partir de la ligne 185 se trouve le code de mise à jour des tampons. À la ligne 194, le code accède à l'horloge interne du noyau. Le drapeau `needtime` n'est actif que si les deux conditions suivantes sont réunies :
  - l'utilisateur a explicitement demandé cette option via une opération de type IOCTL;
  - la demande de lecture (dans ext_read) concerne un nombre de données correspondant exactement à la taille d'un tampon additionnée de la taille de la structure Guard_t. Il s'agit d'une précaution, afin d'éviter qu'un programme ayant été compilé avec une version précédente du gestionnaire, et ignorant cette possibilité, ne puisse recevoir un tampon contenant des données qu'il interpréterait comme des échantillons en provenance de la carte.
- ensuite, à la ligne 212, s'il reste des données à transférer, une portion de code vérifie que le tampon suivant n'est plus utilisé.Si ce n'est pas le cas, la routine tente d'abord de remédier au problème :
  - en 217, activation (réveil) du programme utilisateur qui gère les erreurs d'entrée/sortie;
  - en 220, si on est en mode de balayage de canal, l'acquisition en cours est arrêtée;
  - en 224, si l'on constate qu'il y a déjà eu 2 échantillons de perdus, on arrête également. Nous avons considéré qu'un échantillon perdu relève du domaine de l'accidentel mais qu'au-delà, il convient de prendre des mesures pour éviter que la situation n'empire.
  - entre 229 et 238, on revient à la situation avant l'acquisition, tout en augmentant le compteur du nombre d'éléments perdus.
- de 240 à 250 se trouve le code correspondant au déroulement normal du programme. Le pointeur `bufp` est mis à jour pour le tampon précédent et le tampon actuel, et il y a calcul du nombre d'éléments que contiendra le nouveau tampon.

- le code entre 254 à 260 est utilisé à la fin normale d'une acquisition, et reprogramme la carte pour ne pas générer de nouvelles demandes d'interruption.
- en 268 et 269, on réactive les programmes utilisateurs qui seraient en attente de données.
- de 265 à 286, on calcule sur quel canal aura lieu l'acquisition suivante, et le multiplexeur de la carte d'acquisition est reprogrammé.
- le code entre les lignes 285 et 296 est activé si l'interruption n'est pas générée par une demande de lecture sur le port A;
- le code entre les lignes 297 et 399 sert à réactiver l'ADC. La routine d'interruption liée à l'ADC active un sémaphore lorsqu'elle détecte une erreur, elle bloque également la génération de demande d'interruption. Cette portion de code teste le sémaphore, et procède comme suit :
  - les échantillons perdus sont remplacés dans le tampon par une valeur spéciale indiquant qu'ils ne doivent pas être utilisés;
  - reprogrammation complète de l'ADC, et réinitialisation des tampons;
  - reprogrammation du trigger
  - réactivation de la génération d'interruption
  - désactivation du sémaphore
- entre les lignes 400 et 405, l'accès du port B est définitivement remis à "0", jusqu'à la lecture suivante;
- de 410 à 430 se trouve le code désactivant les interruptions et affichant les registres de la carte Lab-PC pour aider à la localisation du problème.

Annexe B

# Colophon

La totalité de ce texte a été produite via LaTeX, dans le jeu de caractère standard "Computer Modern". Ce n'est pas un sytème de traitement de texte[1], mais plutôt un assistant de mise en page : on lui fournit en entrée le *contenu* du texte ainsi que des indications dans un langage descriptif (ceci est un titre de chapitre, commencer un nouveau paragraphe, ...) et il se charge d'en effectuer la mise en page.

Ce document est réalisé dans la classe "book".

Certaines parties du texte proviennent de documents intermédiaires rédigés à l'origine en WP51 (WordPerfect Corporation, maintenant repris par Corel). Nous avons trouvé un utilitaire, `wp2latex`, qui nous a permis d'extraire le texte et les équations, et de les convertir en LaTeX. À cette occasion, nous avons découvert que certaines d'entre-elles n'étaient pas correctes au niveau de l'appariement des parenthèses.

## french

Ce paquetage est distribué par l'association francophone des utilisateurs de TeX, GUTenberg. Il implémente aussi précisément que possible les règles de la typographie française.

## AMS-TeX, AMS-LaTeX

Il s'agit d'un style développé par l'American Mathematical Society, enrichissant TeX au niveau des commandes de mises en formes d'éléments mathématiques : équations, matrices, symboles, ...

1. le terme anglais "Word Processing" reflète un des problèmes majeurs de ce type de programme : ils traitent des *mots* qui, assemblés, forment un texte sans beaucoup de considérations de l'apparence *globale*, des règles de typographies régissant l'agencement des différentes parties d'une texte ainsi que de la cohérence des divers éléments.

## deriv

Ce paquetage contient plusieurs commandes pour écrire des dérivées ainsi que des dérivées partielles. En particulier, pour la dérivée de fonctions d'une seule variable, le symbole de dérivation « d » est droit, et la variable associée en italique, pour faciliter la lecture :

$$\frac{\mathrm{d}}{\mathrm{d}x}(f \cdot g) = \frac{\mathrm{d}f}{\mathrm{d}x} \cdot g + f \cdot \frac{\mathrm{d}g}{\mathrm{d}x}$$

## fncychap

Ce paquetage contient en standard quelques mises en pages de titres de chapitre.

## listings

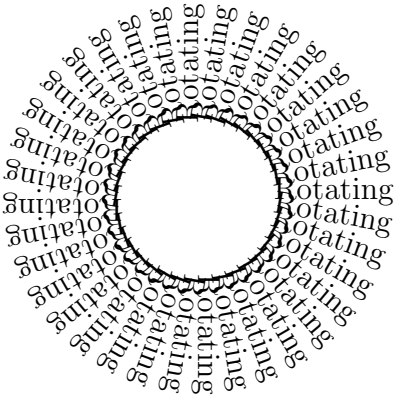
C'est ce paquetage qui a été utilisé pour réaliser les extraits de code : il fallait assurer une bonne lisibilité du texte, tout en conservant une mise en page assez similaire à ce qu'elle est dans l'éditeur, qui utilise des caractères à espacement constant.

## ispell

Ce programme de correction orthographique a utilisé le dictionnaire de l'association GUTenberg, Francais-GUTenberg-v1.0.

## rotating

Je laisse ce paquetage faire sa propre publicité : voici un exemple qui en est directement inspiré.




Ce paquetage a été utilisé pour les tableaux au format paysage.

Annexe C

# Environnement de travail

Certaines sociétés informatiques font du forcing pour déposer des brevets logiciels, ce qui aurait pour conséquences de forcer les utilisateurs à payer des royalties en plus du coût des licences et mises à jour. Nous sommes convaincus qu'il existe des alternatives valables et viables à de très nombreux logiciels commerciaux, sous forme de logiciel "Open Source".

Depuis de nombreuses années, nous n'utilisons plus que ce genre de logiciel. Pour effectuer notre recherche et rédiger ce document, nous n'avons utilisé que du logiciel "Open Source". En voici quelques exemples :

**GNU/Linux** comme système opératoire et environnement de développement.

**XEmacs** est un éditeur de texte couplé à un moteur LISP, ce qui en fait un outil très puissant : il peut notamment effectuer une analyse syntaxique de nombreux langages, et aider à naviguer à travers la structure du document. Cela s'avère extrêmement utile par exemple dans l'éditeur d'équations.

**LaTeX** , déjà décrit au chapitre précédent.

**octave** : il s'agit d'un environnement mathématique très puissant. Par commodité, son interface est très similaire à celle de MatLab. Cet outil contient en standard de très nombreuses routines de traitement numériques.

**gnuplot** est un programme de génération de graphiques, simple d'emploi et très riche en possibilités. Il a été utilisé pour toutes les courbes mathématiques.

**xfig** : ce programme de dessin vectoriel a été utilisé pour les graphiques de type schéma électrique.

**hp2xx** : il s'agit d'un traducteur de dessins au format HPGL. Il a été utilisé pour convertir les schéma de circuits provenant d'OrCAD.

Depuis nos premier contacts avec le monde Linux, nous avons assisté à des progrès considérables. Si les toutes premières versions nécessitaient des connaissances étendues d'administration système de type Unix, ce n'est plus le cas actuellement : certaines

distributions permettent d'arriver à un environnement fonctionnel quasiment sans difficultés. Il suffit de démarrer l'ordinateur depuis un CD-ROM, de choisir le type d'installation et le reste se fait automatiquement.

Il reste cependant un problème majeur au niveau de certains fabricants de matériel informatique, qui restent réticents à fournir des gestionnaires pour Linux, ou refusent carrément de diffuser les informations nécessaires au développement de tels gestionnaires. Ils craignent en effet de permettre à d'autres de copier leur matériel. Heureusement, cet état d'esprit est en train de changer progressivement.

Pour nous, l'avantage principal de cette approche "Open Source" est que chaque utilisateur est à même d'identifier un problème ou d'apporter une amélioration à un logiciel. Parfois, le simple fait d'avoir accès au code pour comprendre comment d'autres personnes ont résolu un problème ou implémenté un algorithme peut être enrichissant. Chaque utilisateur devient ainsi un acteur actif de son environnement informatique.

# Références

Les références [1] à [6] sont reprises dans l'introduction.

## Température différentielle


[10] P. N. Murgatroyd and M. Belloufi. A sensitive differential thermometer. *Measurement Science and Technology*, 1(1):9–12, 1990. doi: 10.1088/0957-0233/1/1/002. URL `http://stacks.iop.org/0957-0233/1/i=1/a=002`.

[11] M. Van Ruymbeke, H. Westerhaus, and J. Fernandez. Temperature measurements. In *XI$^{th}$ International Symposium on Earth Tides*, Helsinki, August 1989.


## Modélisation des transferts de chaleur corporels


[12] H. Arkin, M. M. Chen, and K. R. Holmes. Adaptive thermal modelling: A concept for measurement of local blood perfusion in heated tissues. *Journal of Biomechanical Engineering, Transactions of the ASME*, 108: 306–311, November 1986.

[13] H. Arkin, L. X. Xu, and K. R. Holmes. Recent developments in modelling heat transfer in blood perfused tissues. *IEEE Trans. on Biomedical Engineering*, BME-41(2): 97–107, February 1994.

[14] J. W. Baish, P. S. Ayyaswamy, and K. R. Foster. Small-scale temperature fluctuations in perfused tissue during local hyperthermia. *Journal of Biomechanical Engineering, Transactions of the ASME*, 108: 246–250, August 1986.

[15] J. W. Baish, P. S. Ayyaswamy, and K. R. Foster. Heat transport mechanisms in vascular tissues: A model comparison. *Journal of Biomechanical Engineering, Transactions of the ASME*, 108: 324–331, November 1986.

[16] O. Bertrand, J. Bohorquez, and J. Pernier. Time-frequency digital filtering based on an invertible wavelet transform: An application to evoked potentials. *IEEE Trans. on Biomedical Engineering*, BME-41(1): 77–88, January 1994.

[17] A. J. Chapman. *Heat transfer*. Collier McMillan International editions, third edition, 1974. ISBN : 0-02-321450-3.

[18] C. K. Charny and R. L. Levin. Simulation of MAPA and APA heating using a whole body thermal model. *IEEE Trans. on Biomedical Engineering*, BME-35 (5): 362–371, May 1988.

[19] Z. Dagan, S. Weinbaum, and L. M. Jiji. Parametric studies on the three-layer microcirculatory model for surface tissue energy exchange. *Journal of Biomechanical Engineering, Transactions of the ASME*, 108: 89–96, February 1986.

[20] G. Delhomme, W. H. Newman, B. Roussel, M. Jouvet, H. F. Bowman, and A. Dittmar. Thermal diffusion probe and instrument system for tissue blood flow measurements: validation in phantoms and in vivo organs. *IEEE Trans. on Biomedical Engineering*, BME-41(7): 656–662, July 1994.

[21] M. Giot. *Phénomènes de transfert. Fluides, chaleur, masse*. Ciaco, 1987. ISBN : 2-87085-072-7.

[22] R. G. Gordon, R. B. Roemer, and S. M. Horvath. A mathematical model of the human temperature regulatory system – transient cold exposure response. *IEEE Trans. on Biomedical Engineering*, BME-23(6): 434–444, November 1976.

[23] N. I. Maluf, E. L. McNutt, S. Monroe, D. L. Tanelian, and G. T. A. Kovacs. A thermal signal generator probe for the study of neural thermal transduction. *IEEE Trans. on Biomedical Engineering*, BME-41(7): 649–655, July 1994.

[24] W. Poessnecker. Heat losses which cannot be compensated for and their contribution to the temperature dependence of caloric measuring errors in dynamic differential calorimeters. *Thermochim. Acta*, 229: 97–109, 1993.

[25] D. T. Tompkins, R. Vanderby, S. A. Klein, W. A. Beckman, R. A. Steeves, and B. R. Paliwal. Effect of interseeds spacing, tissue perfusion, thermoseed temperatures and catheters in ferromagnetic hyperthermia: Results from simulations using finite element models of thermoseeds and catheters. *IEEE Trans. on Biomedical Engineering*, BME-41(10): 975–985, October 1994.

[26] D. Van Neste. *Évaluation de l'état fonctionnel de la peau par des méthodes non invasives : intérêt des mesures de capacitance, de perte transépidermique d'eau, de flux sanguin et de leur évaluation combinée pour quantifier des processus inflammatoires cutanés chez l'homme*. UCL. Faculté de Médecine. Unité de Dermatologie Professionnelle et de l'Environnement, Bruxelles, 1990. J.-M. Lachapelle, promoteur.

[27] D. Van Neste. Skin response to histamine dry skin prick test: influence of duration of the skin prick on clinical parameters and on skin blood flow monitoring. *Journal of dermatological science*, 1(6): 435–439, November 1990.

[28] D. Van Neste. In vivo evaluation of unbound water accumulation in stratum corneum. the influence of acute skin irritation induced by sodium laurylsulfate. *Dermatologica*, 181(3): 197–201, 1990.

## Études du sommeil et phénomènes respiratoires

[29] R. J. O. Davies, K. Vardi-Visy, M. Clarke, and J. R. Stradling. Identification of sleep disruption and sleep disordered breathing from the systolic blood pressure profil. *Thorax*, 48(12): 1242–1247, December 1993.

[30] D. J. Pitson, A. Sandell, R. van den Hout, and J. R. Stradling. Use of pulse transit time as a measure of inspiratory effort in patients with obstructive sleep apnoea. *European Respiratory Journal (UK)*, 8(10): 1669–1674, October 1995.

[31] Robin P. Smith, Jerôme Argod, Jean-Louis Pépin, and Patrick A. Lévy. Pulse transit time: an appraisal of potential clinical applications. *Thorax*, 54(5): 452–458, May 1999.

## Convertisseurs Sigma/Delta

[32] K. C.-H. Chao, S. Nadeem, W. L. Lee, and C. G. Sodini. A higher order topology for interpolative modulators for oversampling A/D converters. *IEEE Trans. on Circuits and Systems*, CAS-37(3): 309–318, March 1990.

[33] B. P. Del Signore, D. A. Keth, N. S. Sooch, and E. J. Swanson. A monolithic 20-b delta-sigma A/D converter. *IEEE Journal of Solid-State Circuits*, SC-25(6): 1311–1316, December 1990.

[34] V. F. Dias, G Palmisano, P. O'Leary, and F. Maloberti. Fundamental limitations of switched-capacitor sigma-delta modulators. *IEE Proceedings-G*, 139(1): 27–32, February 1992.

[35] G. Grandbois and T. Pickerell. Quantized feedback takes its place in analog-to-digital conversion. *Electronics*, 13 October 1977.

[36] K. Kalliojärvi, J. Kontro, and Y. Neuvo. Floating-point sigma-delta D/A converter. In J. Vandewalle, R. Boite, M. Moonen, and A. Oosterlinck, editors, *Theories and Applications*, number VI in Signal Processing. Elsevier Science Publishers, 1992.

[37] J. G. Kenney and L. R. Carley. Design of multibit noise-shaping data converters. *Analog Integrated Circuits and Signal Processing*, 3: 259–272, 1993. Kluwer Academic Publishers, Boston.

[38] J. L. LaMay and H. T. Bogard. How to obtain maximum practical performances from state-of-the-art delta-sigma Analog-to-Digital converters. *IEEE Trans. on Instrumentation and Measurement*, 41(6): 861–869, December 1992.

[39] T. C. Leslie and B. Singh. Sigma-delta modulators with multibit quantizing elements and single-bit feedback. *IEE Proceedings-G*, 139(3): 356–362, June 1992.

[40] B. Leung. The oversampling technique for analog to digital conversion : A tutorial overview. *Analog Integrated Circuits and Signal Processing*, 1: 65–74, 1991. Kluwer Academic Publishers, Boston.

[41] Y. Matsuya, K. Uchimura, A. Iwata, T. Kobayashi, M. Ishikawa, and T. Kaneko. A 16-bit oversampling A-to-D conversion technology using triple-integration noise shaping. *IEEE Journal of Solid-State Circuits*, SC-22(6): 921–929, December 1987.

[42] Y. Matsuya, K. Uchimura, A. Iwata, and T. Kaneko. A 17-bit oversampling D-to-A conversion technology using multistage integration noise shaping. *IEEE Journal of Solid-State Circuits*, SC-24(4): 969–975, August 1989.

[43] P. J. A. Naus, E. C. Dijkmans, E. F. Stikvoort, A. J. McKnight, D. J. Holland, and W. Bradinal. A CMOS stereo 16-bit D/A converter for digital audio. *IEEE Journal of Solid-State Circuits*, SC-22(3): 390–394, June 1987.

[44] O. J. Nys and E. Dijkstra. On configurable oversampled A/D converters. *IEEE Journal of Solid-State Circuits*, SC-28(7): 736–742, July 1993.

[45] T. Okamto, Y. Maruyama, and A. Yukawa. A stable high-order delta-sigma modulator with an FIR spectrum distributor. *IEEE Journal of Solid-State Circuits*, SC-28(1): 730–734, July 1993.

[46] S. Park. Multistage decimation filter design technique for high-resolution sigma-delta A/D converters. *IEEE Trans. on Instrumentation and Measurement*, 41(6): 868–873, December 1992.

[47] D. B. Ribner. A comparison of modulators networks for high-order oversampled sigma-delta Analog-to-Digital converters. *IEEE Trans. on Circuits and Systems*, CAS-38(2): 145–159, February 1991.

[48] R. Schreier. On the use of chaos to reduce idle-channel tone in Delta-Sigma modulators. *IEEE Trans. on Circuits and System I: Fundamental Theory and Applications*, 41(8): 539–547, August 1994.

[49] D. Seitzer, G. Pretzl, and N. A. Hamdy. *Electronic Analog-to-Digital Converters.* Wiley Interscience, New York, 1983.

[50] R. J. van de Plassche. A sigma-delta modulator as an A/D converter. *IEEE Trans. on Circuits and Systems*, CAS-25(7): 510–514, July 1978.

[51] R. J. van de Plassche. A/D - D/A converters. In *Eurochip course on Advanced Analog Digital Design*, ESAT, Leuven, October-November 1992.

[52] R. J. van de Plassche and R. E. J. van der Grift. A five-digit analog-digital converter. *IEEE Journal of Solid-State Circuits*, SC-12(6): 656–661, December 1977.

[53] P. W. Wong. Fully sigma-delta modulation encoded FIR filters. *IEEE Trans. on Signal Processing*, 40(6): 1605–1610, June 1992.

[54] J. Yu, M. B. Sandler, and R. E. Hawken. Adaptative quantisation for one-bit sigma-delta modulation. *IEE Proceedings-G*, 139: 39–44, February 1992.

## Unix – Linux

[55] R. Baruch and C. Schroeter. Writing character device driver for Linux. Disponible sur le site du « Linux Lab Project » , http://www.llp.fu-berlin.de/, 1994.
[56] E. Dumas R. Card and F. Mével. *Programmation Linux 2.0. API système et fonctionnement du noyau.* Eyrolles, Paris, 1998. ISBN 2-212-09018-8.
[57] R. W. Stevens. *UNIX network programming.* Prenctice Hall, Englewoods Cliffs, NJ, 1990. ISBN 0-13-949876-1.

## TEX– LATEX

[58] M. Goosens and S. Rahtz. *The LaTeXWEB Companion, Integrating TeX, HTML and XML.* Addison-Wesley series on tools and techniques for computer typesetting. Addison-Wesley, Reading, Massachusetts, first edition, May 1999. ISBN 0–201–43311–7.
[59] M.-P. Knuth. FAQ LaTeX version 2.24. Posted on the UseNet Newsgroup fr.comp.tex, December 2000.
[60] H. Kopka and P. W. Daly. *A guide to LaTeX 2ε, document preparation for beginners and advanced users.* Addison-Wesley, second edition, 1995. ISBN 0–201–42777–X.
[61] T. Lachand-Robert. *La maîtrise de TeXet LaTeX.* Masson, Paris, 1995. ISBN 2–225–84832–7.

## Ondelettes

[62] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Trans. on Information Theory*, 38: 713–718, March 1992.
[63] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. on Information Theory*, 36: 961–1005, September 1990.
[64] R. DeVore and B. J. Lucier. Wavelets. *Acta Numerica*, 1: 1–56, 1992.
[65] David Donoho, Xiaoming Huo, and Thomas P.-Y. Yu. *Preview of WaveLab .800*, 1999. URL `http://www-stat.stanford.edu/%7Ewavelab/`.
[66] Amara Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2): 50–61, summer 1995.
[67] A. Haar. Zur Theorie der Orthogonalen Funcktionensysteme. *Mathematics Annal.*, 69: 331–371, 1910.
[68] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analysis. *SIAM Review*, 36(3): 377–412, September 1994.
[69] D. T. L. Lee and A. Yamamoto. Wavelet analysis: Theory and applications. *Hewlett-Packard Journal*, pages 44–54, December 1994.

[70] S. G. Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11 (7): 674–693, July 1989.
[71] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Mag.*, 9: 14–38, April 1991.
[72] G. Strang. Wavelets and dilatations equations : a brief introduction. *SIAM Review*, 31(4): 614–627, December 1989.
[73] G. Strang. Wavelets. *American Scientist*, 82: 250–255, April 1994.
[74] W.-K. Yoon and M. J. Devaney. Power measurement using the wavelet transform. *IEEE Trans. on Instrumentation and Measurement*, 47(5): 1205–1210, October 1998.

## Identification

[75] Lennaert Ljung. *System identification: theory for the user*. Prenctice-Hall Information and System Science Series. Prenctice-Hall, Englewood Cliffs, New Jersey, 1987.
[76] Van Huffel S. and Vandewalle J. *Frontiers in Applied Mathematics. The total least-square problem, Computational aspects and analysis*. SIAM, 1991.

## Cepstre et transformée cepstrale

[77] W. D. Blair, J. E. Conte, and T. R. Rice. An instructive example of homomorphic signal processing. *IEEE Trans. on Education*, 38(3): 211–216, August 1995.
[78] B. P. Bogert, M. J. R. Healy, and J. W. Tukey. The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autovariance, cross-cepstrum and saphe cracking. In *Proceedings of the Symposium of Time Series Analysis*, pages 209–242, 1963.
[79] W. L. Cameron. Precise expression relating the Fourier Transform of a continuous signal to the Fast Fourier Transform of signal samples. *IEEE Trans. on Signal Processing*, 43(12): 2811–2821, December 1995.
[80] P. Carrette, G. Bastin, Y. Y. Genin, and M. Gevers. Discarding data may help in system identification. *IEEE Trans. on Signal Processing*, 44(9): 2300–2310, September 1996.
[81] W.-J. Kang, J.-R. Shiu, C.-K. Cheng, J.-S. Lai, H.-W. Tsao, and T.-S. Kuo. The application of cepstral coefficients and maximum likelihood method in EMG pattern recognition. *IEEE Trans. on Biomedical Engineering*, BME-42(8): 777–785, August 1995.
[82] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. Prenctice-Hall, Englewood Cliffs, 1975.

[83] P. W. Smith and N. Nandhakumar. An improved power cepstrum based stereo correspondance method for textured scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(3): 338–348, March 1996.

## Méthode de Prony

[84] Y. U. A. Bykovskii, K. V. Kolosov, V. V. Zuev, A. D. Kiryukhin, and S. I. Rasmagin. Determination of the recombination parameters of a semiconducting material by the Prony method. *Technical Physics.*, 44(4): 399–403, April 1999.

[85] S. Chakrabarti, K. R. Demarest, and E. K. Miller. An extended frequency domain Prony's method for transfer function parameter estimation. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 6(4): 269–281, nov 1993.

[86] S. W. Chen. A two-stage discrimination of cardiac arrhythmias using a total least squares-based Prony modeling algorithm. *IEEE Trans. on Biomedical Engineering*, BME-47(10): 1317–1327, October 2000.

[87] G. N. Dovzhenko, D. L. Dovzhenko, and V. B. Smirnov. Analysis of multiexponential curves by the Prony method. *Optics and Spectroscopy*, 86(1): 15–18, January 1999.

[88] V. C. Fernicola, L. Rosso, R. Galleano, T. Sun, Z. Y. Zhang, and K. T. V. Grattan. Investigations on exponential lifetime measurements for fluorescence thermometry. *Review of Scientific Instruments*, 71(1): 2938–2943, July 2000.

[89] V. Garoosi and B. H. Jansen. Development and evaluation of the piecewise Prony method for evoked potential analysis. *IEEE Trans. on Biomedical Engineering*, BME-47(12): 1549–1554, December 2000.

[90] M. Hansson, T. Gansler, and G. Salomonsson. Estimation of single event-related potentials utilizing the Prony method. *IEEE Trans. on Biomedical Engineering*, 43(10): 973–981, October 1996.

[91] P. Koehl. Linear prediction spectral analysis of NMR data. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 34(3–4): 257–299, May 1999.

[92] L. N. Konovalov and I. D. Merkulenko. Determining the parameters of signals from one sampling of observations from the output of an equidistant antenna array. *Radioelectronics and Communication Sytems*, 30(3): 49–55, 1987.

[93] T. Lobos and J. Rezmer. Real-time determination of power system frequency. *IEEE Trans. on Instrumentation and Measurement*, 46(4): 877–881, August 1997.

[94] R. Marz. Equal approximations of exponential functions. *Messen Steuern Regeln mit Automatisierungspraxis*, 15(1): 24–26, 72.

[95] M. R. Osborne and G. K. Smyth. A modified Prony algorithm for exponential function fitting. *SIAM Journal on Scientific Computing*, 16(1): 119–138, January 1995.

[96] H. P. Sava and J. T. E. McDonnell. Modified forward backward overdetermined

Prony's method and its application in modelling heart sounds. *IEE Proceedings Vision Image and Signal Processing*, 142(6): 375–380, December 1995.

[97] S. Schreurs, J. P. Francois, P. Adriaensens, and J. Gelan. Qualitative and quantitative analysis of solid state free induction decay (H-1 NMR) curves using a combination of the methods of Gardner and Prony: Isotactic polypropylene as a case study. *Journal of Physical Chemistry B*, 103(9): 1393–1401, March 1999.

[98] G. K. Smyth. Employing symmetry constraints for improved frequency estimation of eigenanalysis methods. *Technometrics*, 42(3): 277–289, aug 2000.

[99] T. Sun, Z. Y. Zhang, K. T. V. Grattan, and A. W. Palmer. Quasidistributed fluorescence-based optical fiber temperature sensor system. *Review of Scientific Instruments*, 69(1): 146–151, January 1998.

[100] V. Viti, C. Petrucci, and P. Barone. Prony methods in NMR spectroscopy. *International Journal of Imaging Systems and Technology*, 8(6): 565–571, 1997.

[101] J. Z. Yang and C. W. Liu. A precise calculation of power system frequency and phasor. *IEEE Trans. on Power Delivery*, 15(2): 494–499, April 2000.