

Pilotage d'algorithmes pour un diagnostic médical robuste en cardiologie

Piloting algorithms for robust medical diagnosis in cardiology

F. Portet^{1,2}

G. Carrault²

M.-O. Cordier¹

R. Quiniou¹

¹ Institut de Recherche en Informatique et Système Aléatoire (INRIA / Université de Rennes 1)

² Laboratoire de Traitement du Signal et de l'Image (INSERM / Université de Rennes 1)

IRISA, campus Beaulieu, 35000 Rennes
francois.portet@irisa.fr

Résumé

Dans un environnement clinique, les systèmes de monitoring médical sont soumis à diverses sources de bruit qui conduisent à la détection d'informations non pertinentes voire erronées, et vont empêcher un diagnostic médical fiable. Pour répondre à ce problème, nous proposons d'intégrer un pilote d'algorithmes à un système de monitoring cardiaque. Grâce à l'analyse du bruit de ligne et du contexte pathologique (état du patient), le pilote modifie en ligne la chaîne de traitement pour ne baser le diagnostic médical que sur des informations fiables (non bruitées) et strictement nécessaires. Pour valider notre approche nous avons testé le système avec des signaux pathologiques bruités typiques de situations cliniques. Les résultats de ces tests montrent l'intérêt et la faisabilité d'une telle approche.

Mots Clés

Pilotage d'algorithmes, monitoring cardiaque, diagnostic, aide à la décision, chroniques, traitement du signal biomédical.

Abstract

In clinical environment, medical monitoring systems are exposed to numerous noise sources which lead to the detection of irrelevant information and prevent a reliable medical diagnosis. To solve this problem we propose to integrate a pilot into a cardiac monitoring system. Thanks to line noise analysis and to the pathological context, the pilot can change on-line the processing chain of the monitoring application in order to base the medical diagnosis only on uncorrupted information. To validate our approach, we have tested the system with corrupted pathological signals that are typical of clinical situations. The results show the interest and the practicability of our approach.

Keywords

Piloting algorithms, cardiac monitoring, diagnosis, decision aid, chronicles.

1 Introduction

Les systèmes de monitoring médical ont pour objectif de surveiller un patient à travers des signaux physiologiques afin d'alerter le personnel hospitalier en cas de dégradation de l'état du patient et de proposer un diagnostic médical.

Cependant, en milieu clinique, les sources de bruit sont nombreuses et conduisent à la détection d'informations non pertinentes, voire erronées, qui vont perturber le diagnostic médical. Obtenir un diagnostic médical robuste au bruit est donc un objectif majeur. Les systèmes de monitoring, tels que *Guardian* [7] et *Calicot* [1], comprennent généralement deux parties distinctes : l'extraction de caractéristiques des signaux physiologiques, dédiée au traitement et à l'analyse du signal, et le raisonnement médical qui, à partir des caractéristiques et d'une base de connaissances, propose un diagnostic. L'extraction de caractéristiques est assurée par des algorithmes de traitement du signal qui reconnaissent les formes caractéristiques et les traduisent en événements étiquetés qui décrivent les signaux analysés. Ces événements permettent au module de raisonnement médical de reconnaître les motifs de certaines pathologies.

Dans ces systèmes, le contexte (le diagnostic médical en cours d'élaboration et le bruit de ligne) n'est pas mis à profit pour optimiser l'extraction de caractéristiques et obtenir ainsi un diagnostic médical robuste au bruit. Pourtant, ce contexte est utile pour adapter les algorithmes de traitement du signal [12].

Soulas *et al.* [14] ont proposé un système de monitoring cardiaque qui utilise la différence de performance de deux algorithmes de détection d'événements. Selon le type de bruit de ligne, le signal à traiter est dirigé vers l'algorithme le plus approprié. Mais ce système possède une architecture statique qui rend difficile l'intégration de nouveaux algorithmes et leur adaptation au contexte. Enfin, cette étude n'est centrée que sur la détection du QRS et non sur les autres traitements qu'un système de monitoring cardiaque doit effectuer.

Pour obtenir un diagnostic médical robuste au bruit, il se-

rait intéressant que les systèmes de monitoring puissent choisir d'utiliser uniquement les informations non contaminées par le bruit. Pour réaliser cette adaptation en ligne, nous intégrons au système de monitoring cardiaque Calicot un *pilote d'algorithmes*. Le pilotage d'algorithmes consiste à choisir, dans une base, les algorithmes à activer et à les enchaîner de façon à réaliser un traitement donné. Ce pilotage est guidé par la connaissance des contextes de bruit et de diagnostic médical du patient ainsi que par une base de règles de pilotage. De plus, le pilotage permet d'effectuer uniquement les traitements nécessaires pour effectuer le diagnostic médical. De cette manière, le diagnostic repose uniquement sur des informations fiables (non contaminées par le bruit) tout en ne consommant que les ressources nécessaires.

Cet article présente d'abord les principes du pilotage d'algorithmes en section 2. La section 3 introduit les notions d'électrocardiographie et le système de monitoring cardiaque Calicot. L'intégration du pilotage dans le système de monitoring cardiaque est détaillée en section 4. La section 5 résume les résultats obtenus sur des données réalistes. Cet article se termine par une discussion en section 6 et une courte conclusion.

2 Principes du pilotage d'algorithmes de traitement du signal

Le *pilotage d'algorithmes* consiste à choisir automatiquement, dans une base, les algorithmes à activer et à enchaîner de façon à réaliser une application donnée. Pour chaque application, le schéma de connexion entre les algorithmes est connu. En fonction du *contexte courant* de l'application (type de signaux, fréquence d'échantillonnage, etc.) le système de pilotage est capable de reparamétrer les algorithmes, de choisir d'autres algorithmes et de reconfigurer, en ligne, la chaîne de traitement de l'application, c'est-à-dire de modifier le schéma de connexion.

Le pilotage d'algorithmes s'inspire de différents travaux. Shekhar *et al.* [13] représentent une chaîne de traitement du signal par un plan d'opérations primitives. Un opérateur primitif réalise un but simple et possède des méthodes d'initialisation et de paramétrage. Une composition d'opérateurs primitifs constitue un opérateur complexe. Un traitement est représenté par une requête décomposable en sous buts. Pour chaque requête, le plan de sous buts est exécuté puis adapté par un module de contrôle de l'exécution. Cette approche, bien que posant les bases du pilotage d'un système de traitement du signal, est surtout proposée pour l'aide à la création d'une application et non pour son adaptation en ligne et en temps réel.

Karsai & Sztipanovits [6] proposent une architecture pour les programmes structurellement auto-adaptifs en ligne. Le plan du traitement à effectuer est représenté par un graphe dont les nœuds sont les opérations primitives et dont les arcs représentent les flux de signaux. Le plan est exécuté sous le contrôle d'un planificateur séparé qui est configuré par un graphe de contrôle. Un opérateur complexe

est représenté par une composition d'opérateurs primitifs sous forme de graphe. Cet *opérateur composé* contient plusieurs graphes de connexion alternatifs stockés dans les états d'une machine à états finis. Les transitions entre les états sont provoquées par des événements qui entraînent le changement de plan, c'est-à-dire le changement d'architecture de l'opérateur composé.

Nous nous inspirons de ces approches pour définir les différents concepts utilisés dans un système de pilotage d'algorithmes. Une application de traitement du signal peut être décomposée en *tâches* inter-connectées. Tout comme un opérateur composé, une tâche peut être réalisée par un ensemble d'algorithmes. Cet ensemble d'algorithmes est modifié en ligne par le pilote en fonction du contexte courant du système. Les tâches peuvent aussi être *activées* ou *désactivées* lorsque, par exemple, un signal ne fournit plus d'information ou est trop bruité pour être utilisable. Contrairement aux approches présentées, notre pilote centralise les informations puis déduit l'ensemble des modifications à effectuer sur la chaîne de traitement. De cette manière, l'ensemble des modifications à apporter à la chaîne de traitement de l'application à un temps donné sont déduites à partir du même contexte courant.

Définition 1 (Contexte) *Le contexte est l'ensemble des informations qui influent sur l'application. Ce contexte, lorsqu'il est connu, peut être utilisé pour piloter l'application.*

La connaissance du contexte courant est utile pour paramétrer les algorithmes. Par exemple le niveau de bruit de ligne permet d'adapter les seuils de filtres. Mais ce contexte peut aussi contenir des informations de haut niveau dépendantes du domaine d'application tel que le type de données transmises (texte, image, vidéo, etc.), la pathologie d'un patient sous monitoring, etc.

Définition 2 (Tâche) *Une tâche est l'exécution d'un traitement réalisé par un ou plusieurs algorithmes.*

Une tâche représente un traitement précis tel qu'une détection d'événements sur un signal. Pour réaliser ce traitement il existe souvent plusieurs algorithmes. Le pilote a pour rôle de choisir l'algorithme (ou l'ensemble d'algorithmes) le plus adéquat au contexte courant pour la tâche à réaliser.

Définition 3 (Application) *Une application est un ensemble de tâches interconnectées selon un schéma fixé pour réaliser une fonction précise.*

Les applications de traitement du signal peuvent être complexes. Nous partons du principe qu'une décomposition en tâches permet de bien structurer le fonctionnement. Par exemple, une chaîne de réception peut être décomposée en trois tâches : réception du signal, filtrage du signal, détection-classification. Le contexte courant contient des informations telles que le niveau de bruit et le codage utilisé pour la transmission. Si le niveau de bruit augmente et que le codage des données change, la tâche de filtrage

devra choisir un filtre plus sélectif et la tâche de détection-classification devra utiliser l'algorithme correspondant au codage courant.

Définition 4 (Tâche active/ tâche inactive) Une tâche est active lorsqu'elle exécute le traitement d'un signal et envoie un résultat. Une tâche est inactive lorsque pour tout signal en entrée elle ne met que ses variables internes à jour et renvoie un résultat déterministe (aucune sortie ou alarme).

Le pilote peut choisir de désactiver une tâche lorsqu'il sait qu'aucun algorithme de la tâche n'est capable d'effectuer le traitement sans provoquer un trop grand nombre d'erreurs. Cette désactivation peut avoir des effets sur les tâches en aval. C'est pourquoi la vision centralisée du pilote permet de désactiver une tâche et de modifier les autres tâches de façon à ce que l'application puisse continuer de fonctionner.

3 Application au monitoring cardiaque

Le pilotage d'algorithmes est intégré dans le système de monitoring cardiaque Calicot que nous détaillons après une brève introduction aux *arythmies cardiaques*.

3.1 Les arythmies cardiaques et l'électrocardiographie

Le cœur est composé de quatre chambres — deux oreillettes et deux ventricules — qui communiquent entre elles par un système de valves. L'oreillette de la partie droite (resp. gauche) recueille le sang des veines du corps (resp. des poumons) et le communique au ventricule droit (resp. gauche) qui l'expulse vers les poumons (resp. les artères du corps). Ce mécanisme est assuré par un stimulus électrique qui parcourt le cœur et commande la contraction de chaque chambre de manière adéquate. On appelle arythmie toute perturbation dans le parcours de ce stimulus par rapport à la normale. Le stimulus peut être mesuré en plaçant des électrodes à la surface du corps en des endroits prédéfinis. On utilise couramment en clinique l'électrocardiogramme (ECG) de surface pour obtenir une image de la propagation du stimulus à travers le cœur.

Le tracé de la Figure 1 représente les ondes et intervalles standards étudiés en électrocardiographie. L'onde P et le complexe QRS correspondent respectivement à la dépolarisation des oreillettes et des ventricules qui induit la contraction de ces chambres. L'onde T correspond à la repolarisation des ventricules.

Les arythmies peuvent être diagnostiquées à partir de la forme des ondes P et QRS et de leur écart temporel. Par exemple, un rythme *rapide* et régulier de QRS en forme d'*extrasystole ventriculaire* (commande prématurée de contraction d'un ventricule) permet le diagnostic de tachycardie *ventriculaire* alors qu'un rythme *rapide* et régulier de QRS *normaux* précédés d'*ondes P* indique une ta-

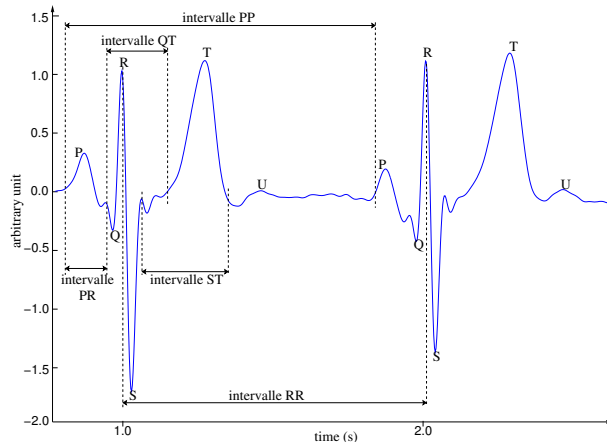


FIG. 1 – Ondes et intervalles standard d'un électrocardiogramme normal.

chycardie *supra-ventriculaire*. Le contexte du patient (médicaments, âge, pathologies, etc.) doit aussi être pris en compte pour le diagnostic.

3.2 Le système de monitoring Calicot

Calicot (Cardiac Arrhythmias Learning for Intelligent Classification of On-line Tracks) [1] est un système de reconnaissance et d'apprentissage d'arythmies cardiaques. Calicot est composé de trois principaux modules : un module d'*apprentissage de motifs d'arythmies cardiaques*, un module d'*abstraction temporelle*, et un module de *reconnaissance de chroniques* (Figure 2, partie ①).

L'apprentissage de motifs d'arythmies cardiaques (hors ligne) Les ECG sont décrits par des événements (onde P, QRS, etc.) dont la succession régulière représente un motif d'arythmie cardiaque. Un tel motif peut être vu comme *une séquence d'événements temporellement liés entre eux*, ce qui est la définition même d'une chronique [2]. L'apprentissage des motifs est réalisé par programmation logique inductive, à partir d'exemples positifs et négatifs d'ECG annotés présentant des arythmies. Les règles de reconnaissance apprises sont ensuite traduites en modèles de chroniques. Ainsi, à chaque arythmie correspond un ou plusieurs modèles de chronique.

L'abstraction temporelle (en ligne) Ce module transforme un signal électrocardiographique numérique en événements temporels étiquetés (complexe QRS normal ou anormal, onde P). L'abstraction temporelle est réalisée par des algorithmes de traitement du signal qui détectent et classent les événements de l'ECG.

La reconnaissance de chroniques (en ligne) À partir des modèles de chroniques, ce module réalise la reconnaissance des chroniques (*i.e.* des arythmies cardiaques) dans le flux d'événements générés par l'abstraction temporelle.

L'ancienne version de Calicot présente des performances de reconnaissance satisfaisantes sur les exemples d'aryth-

mies et de signaux traités [1]. Cependant, le système reste sensible aux erreurs de l'abstraction temporelle qui peuvent perturber notablement la reconnaissance des arythmies, notamment en cas de signaux bruités.

4 L'architecture du système

Le pilotage modifie la chaîne de traitement de Calicot en ligne. La section 4.1 détaille les différentes manières dont Calicot peut être piloté. En outre, le pilote doit savoir dans quels contextes une modification du système est nécessaire. Les contextes et leur analyse sont présentés en section 4.2. Enfin le fonctionnement du pilote et ses règles de pilotage sont décrits en section 4.3. L'architecture du système est représentée par la Figure 2.

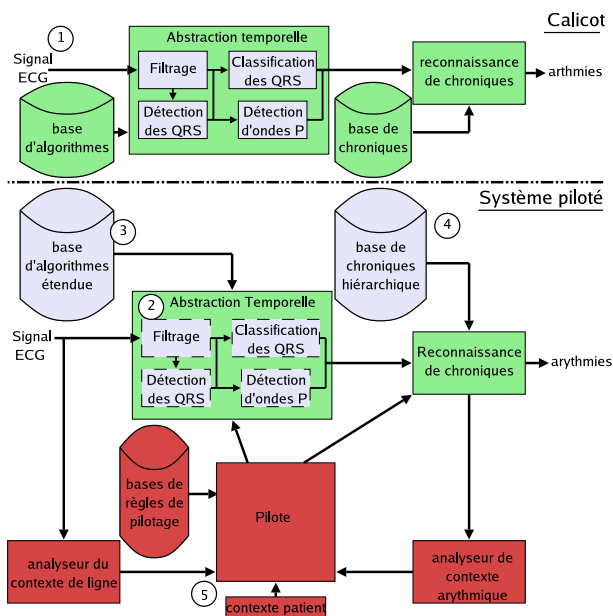


FIG. 2 – Architecture générale du système. La chaîne du haut représente le système Calicot et celle du bas représente les modules du système de pilotage.

4.1 Introduction du pilotage dans Calicot

La structure de Calicot autorise un pilotage à trois niveaux. Le pilote sélectionne le niveau de détail avec lequel l'ECG sera décrit, il active et désactive les tâches de l'abstraction temporelle et choisit les algorithmes de traitement du signal.

Le pilotage de la reconnaissance d'arythmies : Une arythmie peut être diagnostiquée selon plusieurs caractéristiques de l'ECG. Dans Calicot (partie ① de la Figure 2), toutes les caractéristiques (onde P, QRS, forme du QRS) sont extraites de manière continue et envoyées au module de reconnaissance d'arythmies. On peut remarquer cependant que, dans certains contextes, un nombre réduit de caractéristiques peut être suffisant pour reconnaître une arythmie. Par exemple, lorsque le rythme car-

diacque est rapide, on peut être en présence d'une tachycardie ventriculaire (pouvant être létale) ou d'une tachycardie supra-ventriculaire (moins dangereuse). Une reconnaissance d'arythmies basée sur les ondes P permet de les discriminer. Mais, l'analyse de la morphologie QRS, qui demande moins de ressources de calcul et qui est plus robuste, est suffisante pour discriminer les deux arythmies. Ainsi, le pilotage de la reconnaissance d'arythmies consiste à définir le niveau nécessaire d'abstraction du signal ECG et le langage d'expression des modèles de chroniques correspondants, selon les hypothèses du diagnostic courant. Pour représenter ces niveaux d'abstraction différents, une hiérarchie de modèles de chroniques (c'est-à-dire les modèles de motif d'arythmie) est apprise (selon la même méthode que dans [1]) dans les langages d'expression suivants :

- L1 inclut la date d'occurrence des QRS plus l'intervalle temporel entre les occurrences des QRS ;
- L2 ajoute à L1 la morphologie des QRS ;
- L3 ajoute à L1 la date d'occurrence des ondes P ;
- L4 ajoute à L2 la date d'occurrence des ondes P.

Les quatre niveaux de modèles de chronique *C1*, *C2*, *C3* et *C4* constituent la base de chroniques hiérarchique (cf. ④). Par exemple, si l'onde P n'est pas nécessaire alors la base de chronique *C1* ou *C2* est choisie.

Le pilotage des tâches : L'abstraction temporelle de Calicot peut être décomposée en quatre tâches principales :

- *Filtrage* : la tâche de filtrage sépare au maximum le signal ECG des bruits parasites,
- *DétectionQRS* : la tâche de détection des QRS donne leur date d'occurrence,
- *ClassificationQRS* : la tâche de classification des QRS les étiquette, et
- *DétectionOndeP* : la tâche de détection des onde P donne leur date d'occurrence.

Dans Calicot, chaque tâche est toujours activée. Mais, si une base de chronique, telle que *C2* est choisie, il est inutile d'activer *DétectionOndeP* car cette tâche n'est pas nécessaire pour la reconnaissance d'arythmies. De plus, dans plusieurs circonstances, certaines tâches ne peuvent pas être accomplies. Par exemple, si la ligne est trop bruitée pour que *DétectionOndeP* s'accomplisse sans erreur alors *DétectionOndeP* doit être désactivée. Sinon, cette tâche pénalise le système entier parce qu'elle fournit des informations erronées au module de reconnaissance de chroniques. Pour être plus efficace et baser la reconnaissance sur les informations sûres, la nouvelle architecture permet l'activation et la désactivation des tâches de l'abstraction temporelle (cf. ②) selon les besoins de la reconnaissance d'arythmies et le contexte de bruit de ligne.

Le pilotage des algorithmes de traitement du signal :

Les tâches de l'abstraction temporelle sont exécutées par des algorithmes de traitement du signal. Dans Calicot, un seul algorithme est dédié à chaque tâche. Pourtant, pour accomplir ces tâches, la littérature propose plusieurs algorithmes dont les performances varient selon le contexte. Notre étude préliminaire, décrite dans [12], a

montré que les performances d’algorithmes de détection de QRS changent en fonction du contexte courant (le bruit de ligne et la morphologie des QRS). Le pilotage d’une tâche de détection de QRS dans un contexte bruité à l’aide de ces résultats a permis de réduire le taux d’erreurs de 14,3% à 10,6% [11]. La base d’algorithmes de Calicot est étendue (cf. (3)) pour contenir plusieurs algorithmes de traitement du signal pour chaque tâche. Par exemple, la tâche *DétectionQRS* peut être effectuée par trois algorithmes classiques de la littérature :

- *pan* : le Pan et Tompkins [10] qui est un algorithme devenu standard dans la détection du QRS,
- *gritzali* : le détecteur de Gritzali [5], autre algorithme standard de détection de QRS, et
- *df2* : le détecteur de Okada modifié par Friesen *et al.* [4].

Ces trois niveaux de pilotage (reconnaissance d’arythmies, tâches, algorithmes) interagissent pour élaborer des décisions cohérentes. Par exemple, le pilotage de la reconnaissance d’arythmie ne peut pas choisir un type de modèles de chronique si les tâches nécessaires correspondantes ne peuvent pas être activées par le pilotage des tâches. Un pilotage efficace est donc entièrement dépendant de l’élaboration précise du contexte courant que nous allons décrire dans la section suivante.

4.2 Le contexte

Le contexte (cf. (5)) est composé de trois sous-contextes : le contexte de ligne, le contexte arythmique et le contexte patient. Le contexte patient (l’âge, le rythme ECG fondamental, etc.) est statique alors que le contexte de ligne et le contexte arythmique sont dynamiques et sont régulièrement actualisés par deux analyseurs :

Le contexte de ligne Un électrocardiogramme peut être parasité par différentes sources de bruit dans un environnement clinique. Dans notre étude nous considérons les trois types de bruit mutuellement exclusifs (*bw*, *ma*, et *em*) pour trois rapports signal sur bruit (5, -5, et -15 dB) des travaux de Portet *et al.* [12]. Le bruit *bw* (baseline wander) est majoritairement basse fréquence, le bruit *ma* (muscle artefact) est majoritairement haute fréquence, et le bruit *em* (electrode motion artefact) a des composantes hautes et basses fréquences. La ligne peut, bien sûr, ne présenter aucun bruit. Le contexte de ligne est donc constitué du niveau et du type de bruit présent sur la ligne à l’instant n :

$$\forall n \in \mathbf{N}, \text{ctxtline}(n) \in (\{bw, ma, em\} \times \{5, -5, -15\}) \cup \{\text{sans bruit}\} \quad (1)$$

Le détecteur de contexte de ligne est placé au début de la chaîne de traitement. Le pilote peut donc modifier la chaîne de l’abstraction temporelle avant que le signal ne soit traité par celle-ci.

Le contexte arythmique L’analyseur de contexte arythmique utilise les hypothèses du module de reconnaissance de chroniques afin d’établir une liste des arythmies les plus susceptibles d’apparaître. Cette liste constitue le contexte

arythmique et permet au pilote de faire des hypothèses sur les formes d’ondes que l’abstraction temporelle doit traiter. Les principales formes de QRS sont symbolisées par les lettres *N*, *V*, *L*, *R*, *J*, *F*, *A*, *P*, correspondant respectivement à battement Normal, extrasystole Ventriculaire, bloc de branche gauche et droit (Left et Right), etc.

4.3 Le pilote

L’architecture du pilote est représentée par la Figure 3.

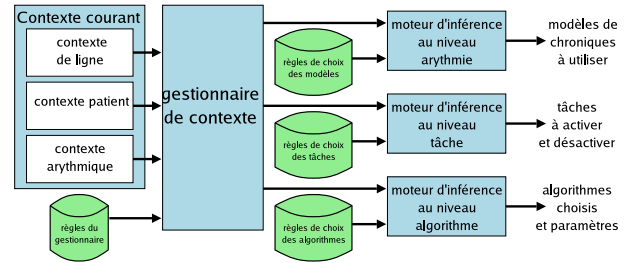


FIG. 3 – Architecture du pilote.

Le pilote est composé de trois moteurs d’inférence qui déduisent les actions à effectuer sur le système pour les trois niveaux de pilotage. À partir du contexte, le gestionnaire de contexte instancie des variables dont ont besoin les moteurs d’inférences.

Le gestionnaire de contexte : Le rôle du gestionnaire de contexte est d’instancier et d’actualiser des variables utiles pour piloter à partir des informations transmises par les analyseurs de contexte. Sa connaissance est représentée par des règles expertes sous forme de règles de production dans la *base de règles du gestionnaire*.

SI <ctxtline = sans bruit>
ALORS <OndePActivable>

Par exemple, *OndePActivable* est vrai seulement si la ligne n’est pas bruitée. De même, du contexte arythmique le gestionnaire de contexte déduit les principales formes de QRS qui seront traitées par l’abstraction temporelle. Le gestionnaire de contexte actualise la base de faits comme dans un système expert classique.

Les moteurs d’inférence : Le système est piloté à trois niveaux : au niveau de la reconnaissance des arythmies, au niveau des tâches de l’abstraction temporelle et au niveau des algorithmes de traitement du signal. Des informations transmises par le gestionnaire de contexte, les moteurs d’inférence déduisent les actions à effectuer sur le système. Leurs règles de pilotage sont définies par des experts et sont regroupées dans les règles de choix de modèles de chroniques, les règles de choix des tâches et les règles de choix d’algorithmes.

Les règles de choix des modèles de chroniques : La reconnaissance de chronique adapte le niveau d’abstraction au contexte. Par exemple, si seules la date d’occurrence des

QRS et la morphologie des QRS sont nécessaires et techniquement exécutables, le reconnaiseur de chronique doit utiliser la base de modèles de chroniques C2.

SI $\langle \text{besoinQRS} \wedge \text{besoinClassifQRS} \wedge \text{QRSActivable} \wedge \text{ClassifQRSActivable} \rangle$
ALORS $\langle \text{choisir C2} \rangle$

Les règles de choix des Tâches : Les tâches de l'abstraction temporelle sont activées selon les besoins et les contraintes techniques.

SI $\langle \text{besoinOndeP} \wedge \text{OndePActivable} \rangle$
ALORS $\langle \text{activer la tâche DétectionOndeP} \rangle$

Par exemple, pour activer *DétectionOndeP*, il est nécessaire que cette tâche soit activable et qu'elle soit requise.

Les règles de choix d'algorithmes : Les règles de choix d'algorithmes déterminent le meilleur algorithme pour réaliser les tâches de l'abstraction temporelle. Par exemple, si la tâche *DétectionQRS* est active, il est nécessaire de choisir le détecteur le plus adéquat au contexte courant. Les règles suivantes réalisent ces décisions :

SI $\langle L \wedge bw \wedge SNR \geq -5dB \rangle$
ALORS $\langle \text{choisir gritzali} \rangle$

SI $\langle (L \vee F) \wedge \text{sans bruit} \rangle$
ALORS $\langle \text{choisir gritzali} \rangle$

SI $\langle (F \vee P) \wedge bw \wedge SNR \geq 0dB \rangle$
ALORS $\langle \text{choisir gritzali} \rangle$

SI $\langle em \wedge ((N \vee A \vee P \vee R) \wedge SNR = -15dB) \rangle$
ALORS $\langle \text{choisir df2} \rangle$

SI $\langle em \wedge (SNR = -5dB \wedge P) \rangle$
ALORS $\langle \text{choisir df2} \rangle$

SI $\langle \text{defaut} \rangle$
ALORS $\langle \text{choisir pan} \rangle$

La première règle stipule que si le contexte de ligne signal détecte la présence de bruit *bw* à $-5dB$ et que le contexte médical informe qu'il a majoritairement des QRS de forme *L*, alors le détecteur *gritzali* est sélectionné. Ces règles sont actuellement dérivées de l'étude de Portet *et al.* [12]. Les règles de production se sont avérées suffisamment expressives pour l'instant mais il est clair qu'il est possible d'ajouter des préférences sur ces règles ou des méta-règles permettant de choisir dans le cas où plusieurs sont déclençables en même temps.

5 Résultats

5.1 Les algorithmes utilisés

Les algorithmes utilisés pour la détection du QRS sont ceux décrits section 4.1. Dans notre étude, les seuils des détecteurs sont choisis de manière optimale. La classification du QRS se fait uniquement par un réseau de neurones à 4 sorties donnant les types de QRS suivants : normal, bloc

de branche gauche, bloc de branche droit et extrasystole. Actuellement, la détection d'onde P est simulée et, comme l'onde P est très difficile à extraire en milieu bruité, la présence de bruit haute fréquence (*ma* et *em*) correspond à une augmentation de 50% du nombre d'erreurs (fausses alarmes et non reconnus). Les analyseurs de contexte sont aussi simulés et ne présentent aucune erreur. Enfin, le reconnaiseur de chroniques utilisé est CRS (Chronicle Recognition System) [3].

5.2 Les modèles de chroniques appris

Quatre classes de rythme ont été apprises par PLI sur 85 exemples tirés des enregistrements de la base [8] :

- pour la classe *bigéminisme* : 106, 119, 213, 223
- pour la classe *mobitzII* : 231
- pour la classe *normal* : 100, 101, 117, 121, 122
- pour la classe *pvc* : 106, 119, 213, 223

Pour chaque exemple, les ondes QRS et P sont étiquetées. Les modèles de chroniques ont été appris avec les quatre langages de description, L1 à L4, pour obtenir les modèles C1 à C4 (cf. 4.1).

5.3 La méthode de calcul de performances

Pour calculer les performances de la reconnaissance de chroniques, nous nous intéressons à trois valeurs.

- VP (Vrais Positifs) : le nombre d'arythmies reconnues qui sont bien présentes.
- FP (Faux Positifs) : le nombre d'arythmies reconnues sans être véritablement présentes ou confondues avec d'autres arythmies.
- FN (Faux Négatifs) : le nombre d'arythmies non reconnues qui sont pourtant bien présentes.

Méthode de calcul des matrices de confusion : Dans une matrice de confusion (exemple Tableau 1), les lignes représentent les reconnaissances et les colonnes les véritables éléments à reconnaître. Ainsi, pour chaque classe, les VP se trouvent dans la diagonale de la matrice de confusion, les FP sont les confusions et les FN sont les événements d'arythmie qui n'ont pas été reconnus. La colonne *inconnu* représente les confusions avec des arythmies dont les motifs n'ont pas été appris par le reconnaiseur de chronique. Ces confusions avec ces classes inconnues ne sont pas intégrées dans le calcul de performances par la suite.

	A1	A2	A3	inconnu
A1	VP	FP	FP	FP
A2	FP	VP	FP	FP
A3	FP	FP	VP	FP
Non Reconnu	FN	FN	FN	

TAB. 1 – Exemple de matrice de confusion

L'unité de calcul des performances est le battement cardiaque. Pour chaque enregistrement, chaque QRS (représentatif d'un battement cardiaque) est extrait et étiqueté par

le nom de l'arythmie à laquelle il appartient (chaque battement appartient à un rythme). Nous nous intéressons à un sous ensemble d'arythmies (cf. 5.2) et seuls les QRS référençant une arythmie comprise dans cet ensemble sont étiquetés du nom de cette arythmie. Les autres sont annotés *inconnu*.

La procédure de génération de la matrice de confusion est la suivante :

1. Pour chaque arythmie reconnue, la fenêtre temporelle allant du premier événement au dernier événement appartenant à l'arythmie est considérée. Dans cette fenêtre, tous les battements étiquetés du même nom que l'arythmie sont marqués. Comme les fenêtres de reconnaissance se chevauchent, le marquage des battements permet d'éviter d'attribuer deux fois un battement à une même arythmie. Ainsi, dans la fenêtre d'une arythmie reconnue, tous les battements étiquetés du nom de cette arythmie et non déjà marqués comptent pour des VP. S'il n'existe aucun battement du même nom que l'arythmie alors il y a eu confusion (FP).
2. Dans le cas de confusion (FP), il faut déterminer quelle colonne incrémenter, c'est à dire avec quelle arythmie il y a eu confusion. La procédure respecte les priorités suivantes :
 - si, dans la fenêtre de la chronique, il existe un battement annoté *inconnu* alors la confusion a eu lieu avec une arythmie inconnue,
 - sinon, s'il y a dans la fenêtre des battements différents de *normal*, alors la confusion a eu lieu avec l'arythmie étiquetant le premier battement différent de *normal*,
 - sinon, il y a eu confusion avec la classe *normal*.
3. Le nombre de non reconnus (FN) est le nombre de battements non marqués de chaque arythmie.

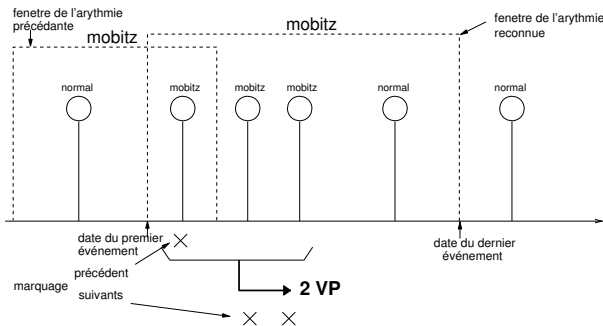


FIG. 4 – Exemple de reconnaissance.

La Figure 4 donne un exemple de calcul de performances. Une première arythmie *mobitz* a été reconnue. Sa fenêtre de reconnaissance couvrait un battement *mobitz*. La case de la ligne *mobitz* et de la colonne *mobitz* a donc été incrémentée d'un VP. Le battement a ensuite été marqué.

Une autre arythmie *mobitz* a ensuite été reconnue. Sa fenêtre couvrait quatre battements. Trois battements appartaient effectivement à la chronique mais, le premier ayant déjà été marqué, seuls deux VP ont incrémenté la case *mobitz*. Les deux battements ont ensuite été marqués et n'ont plus été attribués à une autre chronique.

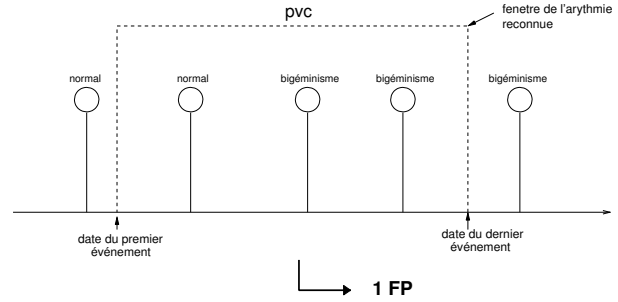


FIG. 5 – Exemple de confusion.

Un exemple de confusion est représenté par la Figure 5. Une fenêtre de chronique *pvc* recouvre trois battements qui n'appartiennent pas à la chronique reconnue, celle-ci est donc une confusion avec la première arythmie de la fenêtre *bigéminisme*. Les résultats de ces deux exemples sont donnés dans le Tableau 2.

	bigé.	mobitz	normal	pvc	inc.
bigé.	0	0	0	0	0
mobitz	0	3	0	0	0
normal	0	0	0	0	0
pvc	1	0	0	0	0
Non Reconnu	3	0	5	0	0

TAB. 2 – Calcul de la matrice de confusion avec les exemples des Figures 4 et 5

Critères pour la comparaison des résultats : À partir des valeurs de la matrice de confusion nous pouvons calculer trois critères sont calculés. Pour obtenir une mesure globale des performances, l'indice critique de succès (*ICS*) est calculé comme suit :

$$ICS = \frac{VP}{VP + FP + FN}$$

Il prend en compte les deux types d'erreur *FP* et *FN*. Un score de 0 n'indique aucune compétence tandis qu'un score de 1 montre une reconnaissance parfaite. Les critères *Pr* et *Tfa* sont aussi calculés car l'*ICS* ne distingue pas les sources d'erreur.

La probabilité de reconnaissance (*Pr*), ou sensibilité, ou encore rappel, donne la fraction d'arythmies présentes dans les enregistrements qui ont été correctement reconnues. Elle s'obtient par :

$$Pr = \frac{VP}{VP + FN}$$

Un score parfait de 1 indique que tous les événements sont reconnus. Cette mesure est sensible aux *VP* mais ignore totalement les *FP*. Le taux de fausses alarmes (*Tfa*) donne la fraction d'arythmies reconnues qui n'étaient pas vraiment présentes.

$$Tfa = \frac{FP}{VP + FP}$$

Un score parfait de 0 indique qu'il n'y a aucune confusion. Cette mesure est sensible aux *FP* mais ignore les *FN*. Le taux de fausses alarmes respecte la relation suivante avec la prédictivité positive parfois utilisée : $P+ = 1 - Tfa$

5.4 Les données de test

Pour tester notre système, 4 ECG ont été extraits de la base standard MIT-BIH [8] : les enregistrements 119 (rythme normal avec extrasystoles ventriculaires et bigéminismes), 122 (rythme normal), 214 (bloc de branche gauche et extrasystoles ventriculaires) et 231 (rythme normal intermittent avec un bloc de branche droit et mobitz de type 2). Ils représentent au total deux heures d'enregistrement. Le bruit ajouté a été choisi aléatoirement parmi les valeurs possibles du contexte de ligne (voir équation 1). Les sections de bruit ont été ajoutées sur des segments variant de 20 à 100 QRS et proviennent tous d'une base de bruit clinique réel [9].

5.5 Performances des algorithmes de traitement du signal

On constate que le pilotage des algorithmes de traitement du signal améliore les performances de l'abstraction temporelle. Le Tableau 3 présente les résultats de la tâche *DétectionQRS* sans et avec le pilotage.

<i>DétectionQRS</i>	VP	FP	FN	Ne
sans pilote	7414	710	882	1592
avec pilote	7499	716	797	1513

TAB. 3 – Résultats de la détection de QRS sans et avec pilotage

Avec le pilote, le nombre d'erreurs ($Ne = FN + FP$) est moins important. Le FN est réduit de 85 pour seulement 6 FP en plus. Le pilotage permet donc d'augmenter le nombre d'événements détectés sans augmenter les fausses alarmes ce qui contribue à améliorer la reconnaissance d'arythmies. Le classificateur de QRS présente un taux de bonnes classifications ($\frac{QRS \text{ bien classés}}{VP}$) de 55% ce qui est assez médiocre. Ce résultat n'est pas surprenant car la classification des ondes est généralement peu robuste au bruit. La désactivation de cette tâche dans les cas bruités est donc une solution pour éviter la production d'erreurs de classification.

5.6 Résultats de la reconnaissance à partir des annotations de références

Cette section présente les performances de la reconnaissance d'arythmies (Tableau 4) sur les signaux affectés de l'annotation originale, ce qui correspond à une reconnaissance dans le cadre d'une extraction de caractéristiques parfaite (détection et classification parfaites).

	bigé.	mobitz	normal	pvc	inc.	Total
bigé.	246	0	0	51	5	302
mobitz	0	833	1	0	0	834
normal	0	0	5624	0	0	5624
pvc	63	0	1	382	213	659
FN	9	5	1240	46		1300
Total	318	838	6866	479	218	

TAB. 4 – Résultats de la reconnaissance d'arythmie sur les annotations de référence avec la règle *C4*

Les résultats montrent une très bonne reconnaissance des bigéminismes avec seulement 9 battements non attribués. *pvc* et bigéminisme sont des classes difficiles à discriminer car elles sont toutes deux composées de battements normaux et d'extrasystoles ventriculaires. L'arythmie *mobitz* est très bien reconnue avec seulement 5 battements non attribués. Cependant la caractérisation du *mobitz* est basée sur l'occurrence de deux ondes P successives et peut donc être très affectée par le bruit de ligne. Pour cette classe, une caractérisation uniquement fondée sur les QRS serait plus robuste. La reconnaissance d'un rythme *normal* est assez bonne malgré les 1240 FN. La chronique d'un rythme *normal* est caractérisée par un nombre important de battements successifs normaux. Or, dans des ECG pathologiques (autre que le 122) des battements intempestifs, tels que des extrasystoles, surviennent souvent au milieu d'un rythme normal. Lorsque ces battements intempestifs sont trop fréquents, le nombre de battements normaux successifs est insuffisant pour reconnaître le rythme *normal*. Le nombre de non reconnus de la classe *normal* n'est pas gênant car dans un système de monitoring ce sont surtout les événements pathologiques qu'il est important de reconnaître. *pvc* est l'arythmie la plus difficile à reconnaître avec 46 non reconnus. Les 277 FP associés à *pvc* sont tous dus aux bigéminismes et aux trigéminismes (inconnues) de l'enregistrement 119 et 214 qui sont des arythmies contenant toutes deux des extrasystoles.

5.7 Résultats du pilotage

Cette section présente les résultats du pilotage des tâches guidé uniquement par le contexte de bruit. L'analyse de la courbe de l'indice critique de succès (Figure 6) montre la supériorité du système piloté par rapport au système non piloté sur des signaux très bruités. Le pilote apporte une meilleure probabilité de reconnaissance (Figure 7) et un

taux de fausses alarmes (Figure 8) plus faible.

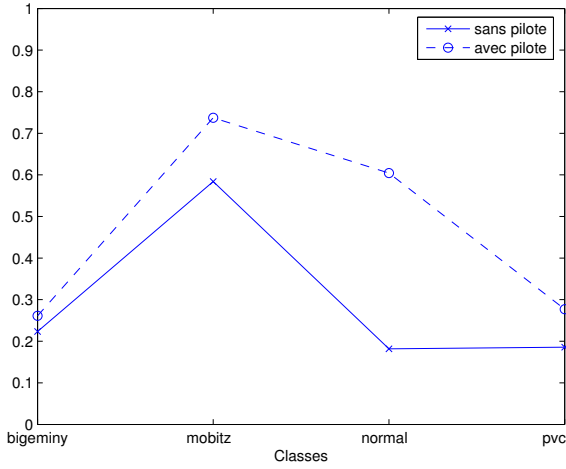


FIG. 6 – Indice Critique de Succès

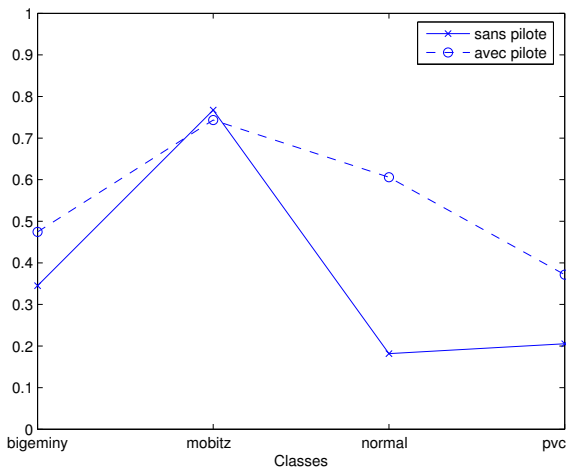


FIG. 7 – Probabilité de reconnaissance

Le Tableau 5 présente les résultats d’une reconnaissance d’arythmies sans pilotage. En comparant ces résultats aux performances de reconnaissance sur les annotations (Tableau 4), on constate une chute importante du nombre de reconnaissances et une augmentation du nombre de confusions pour bigéminisme et mobitz. Le système n’est plus en mesure de donner un diagnostic médical pour ces signaux bruités. Cette baisse de performance est due d’une part, à l’abstraction temporelle avec une classification peu robuste et d’autre part, au langage d’expression trop riche des modèles de chroniques donc trop exigeant dans cette situation.

Les résultats obtenus avec le pilotage (Tableau 6) améliorent nettement la probabilité de reconnaissance. Les résultats sont plus proches du cas de référence (Tableau 4). Ceci est dû à l’amélioration apportée par le pilotage des

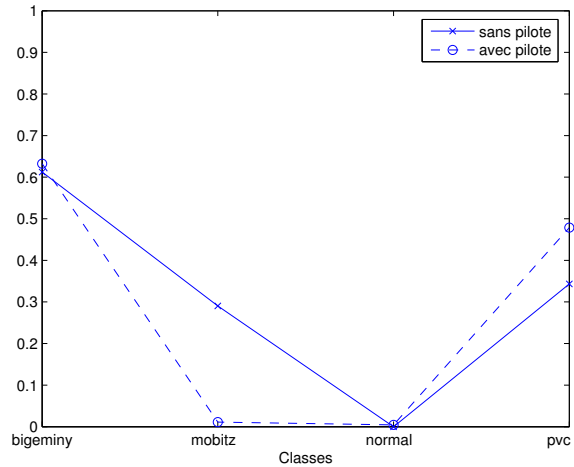


FIG. 8 – Taux de fausses alarmes

	bigé.	mobitz	normal	pvc	inc.	Total
bigé.	88	1	116	22	3	230
mobitz	1	643	249	13	4	910
normal	0	0	1247	0	0	1247
pvc	28	0	18	88	48	182
NR	167	195	5617	340		6319
Total	284	839	7247	463	55	

TAB. 5 – Résultats de la reconnaissance sans le pilotage sur les signaux bruités avec le modèle *C4*

tâches de l’abstraction temporelle et au pilotage de la reconnaissance de chroniques qui permet d’utiliser des langages d’expression moins riches lorsque l’abstraction temporelle ne peut fournir tous les éléments nécessaires. Cette hausse de reconnaissance s’accompagne d’une hausse du nombre de confusions pour la classe *pvc* ainsi que très légèrement pour la classe bigéminisme. Pourtant, au total le pilote génère moins d’erreurs avec 379 FP et 3324 FN que la reconnaissance sans pilote avec 448 FP et 6319 FN. Le nombre de fausses alarmes est donc légèrement diminué pour un gain en diagnostic médical très significatif.

	bigé.	mobitz	normal	pvc	inc.	Total
bigé.	121	0	160	48	5	334
mobitz	0	623	7	0	0	630
normal	11	7	4158	0	7	4183
pvc	60	0	86	159	83	388
NR	134	215	2706	269		3324
Total	326	845	7117	476	95	

TAB. 6 – Résultats de la reconnaissance avec le pilotage sur les signaux bruités et les modèles *C1*, *C2* et *C4*

6 Discussion

Les résultats montrent bien que dans un environnement clinique hostile et en absence de pilote, les sources d'erreurs provoquent un nombre de fausses alarmes important et un diagnostic médical défaillant. Cette baisse de performance est provoquée par le manque de robustesse des algorithmes de traitement du signal et l'impossibilité du diagnostic médical à s'adapter au contexte. L'introduction du pilotage dans le système Calicot permet de pallier à ces deux problèmes. Plutôt que d'optimiser les algorithmes de traitement du signal, nous montrons que l'utilisation adéquate des algorithmes selon les contextes de bruit diminue le nombre d'erreurs engendrées. Pour l'instant, seule la tâche de détection de QRS en bénéficie mais le pilotage des autres tâches pourrait en bénéficier avec une base d'algorithmes plus importante. De plus, le pilote adapte le diagnostic médical à la résolution courante de l'abstraction temporelle augmentant ainsi grandement le nombre de reconnaissances d'arythmies. Le nombre de fausses alarmes reste important mais dans un contexte aussi bruité (mouvement des patients, décollement d'électrodes, etc.) il faut surtout assurer une bonne reconnaissance des pathologies. dans les cas où le signal est extrêmement bruité (impossible à analyser), il serait peut-être raisonnable de désactiver la fonction diagnostic momentanément, car la production de fausses alarmes est alors très importante et sans réelle signification. De plus, plusieurs chroniques sont fréquemment instanciées simultanément augmentant de fait le nombre de fausses alarmes. Pour pallier ce problème, le système de monitoring devrait intégrer un filtrage des alarmes. Par exemple, si la classe `pvc` est instanciée en même temps que la classe `bigeminisme` alors seule cette dernière devrait être considérée car plus prioritaire.

7 Conclusion

Nous proposons un pilotage du système de monitoring Calicot. Ce pilotage choisit des algorithmes dans une base pour modifier en ligne la chaîne de traitement du signal. Il est guidé par l'analyse du contexte courant (bruit de ligne et diagnostic médical). L'architecture proposée permet de désactiver les tâches non fiables et d'adapter la reconnaissance d'arythmies à la précision courante du système. Les résultats montrent que le pilotage permet d'assurer un diagnostic médical sur des électrocardiogrammes bruités. Dans cet article, le pilotage est surtout guidé par le contexte de bruit. Il est aussi possible de l'utiliser pour sélectionner les tâches à activer en fonction des arythmies en cours de reconnaissance. Par exemple, si deux arythmies doivent être discriminées, le pilote peut éviter d'activer des tâches coûteuses en ressources si elles ne sont pas nécessaires à cet effet. Ainsi, le pilotage permettrait d'assurer une consommation de ressource minimale. La méthode de pilotage présentée peut s'appliquer aux nombreuses utilisations du monitoring médical telles que les unités de soins intensifs ou l'anesthésie ou dans d'autres domaines tels que la surveillance de processus industriels.

Références

- [1] G. Carrault, M.-O. Cordier, R. Quiniou, and F. Wang. Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms. *Artificial Intelligence in Medicine*, 28 :231–263, 2003.
- [2] M.-O. Cordier and C. Dousson. Alarm driven monitoring based on chronicles. In *Safeprocess'2000*, pages 286–291, Budapest, 2000.
- [3] C. Dousson. Alarm driven supervision for telecommunication networks II- on-line chronicle recognition. *Annales des Télécommunications*, 9/10 :501–508, 1996.
- [4] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle. A comparison of the noise sensitivity of nine QRS detection algorithms. *IEEE Transactions on Biomedical Engineering*, 37 :85–98, 1990.
- [5] F. Gritzali. Towards a generalized scheme for QRS detection in ECG waveforms. *Signal Processing*, 15 :183–192, 1988.
- [6] G. Karsai and J. Sztipanovits. A model-based approach to self-adaptive software. *IEEE Intelligent systems*, 14 :46–53, 1999.
- [7] J.E. Larsson and B. Hayes-Roth. Guardian : An intelligent autonomous agent for medical monitoring and diagnosis. *IEEE Intelligent Systems*, 13 :58–64, 1998.
- [8] R. Mark and G. Moody. MIT-BIH arrhythmia data base directory. Massachusetts Institute of Technology, 1988.
- [9] G.B. Moody, W.E. Muldrow, and R.G. Mark. A noise stress test for arrhythmia detectors. *Computers in Cardiology*, 11 :381–384, 1984.
- [10] J. Pan and W. J. Tompkins. A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3) :230–236, 1985.
- [11] F. Portet and G. Carrault. Piloting real-time QRS detection algorithms in variable contexts. In *3rd European Medical and Biological Engineering Conference (EMBE'05)*, Prague, 2005.
- [12] F. Portet, A. I. Hernández, and G. Carrault. Evaluation of real-time QRS detection algorithms in variable contexts. *Medical & Biological Engineering & Computing*, 43(3) :381–387, 2005.
- [13] C. Shekhar, S. Moisan, and M. Thonnat. Towards an intelligent problem-solving environment for signal processing. *Mathematics and Computers in Simulation*, 36 :347–359, 1994.
- [14] T. Soulas, G. Le Certen, J.P. Le Pichon, and G. Carrault. Algorithm switching in real time monitoring. In *Symposium on Electronics and Telecommunications (ETC)*, pages 145–149, Timisoara, 1998.