

# Pilotage en ligne d'algorithmes de traitement du signal guidé par le contexte courant

François Portet, Guy Carrault, Marie-Odile Cordier, René Quiniou

► **To cite this version:**

François Portet, Guy Carrault, Marie-Odile Cordier, René Quiniou. Pilotage en ligne d'algorithmes de traitement du signal guidé par le contexte courant. 7e Rencontres Jeunes Chercheurs en Intelligence Artificielle (RJCIA 2005), May 2005, Nice/France, 2005. <inria-00001097>

**HAL Id: inria-00001097**

**<https://hal.inria.fr/inria-00001097>**

Submitted on 6 Feb 2006

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

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

# Pilotage en ligne d'algorithmes de traitement du signal guidé par le contexte courant

François Portet<sup>1,2</sup>, Guy Carrault<sup>2</sup>, Marie-Odile Cordier<sup>1</sup>, René Quiniou<sup>1</sup>

<sup>1</sup> Institut de Recherche en Informatique et Systèmes Aléatoires  
INRIA / Université de Rennes 1, Campus de Beaulieu, 35042 Rennes  
{francois.portet,cordier,quiniou}@irisa.fr

<sup>2</sup> Laboratoire de Traitement du Signal et de l'Image, Unité INSERM 642  
Université de Rennes 1, Campus de Beaulieu, 35042 Rennes  
guy.carrault@univ-rennes1.fr

## Résumé :

Le travail présenté fait partie d'un projet dédié au développement d'un système de monitoring des arythmies cardiaques. Dans ce type d'application, la réduction du nombre d'erreurs de détection et de diagnostic médical dues au bruit de ligne est un objectif majeur. Pour atteindre cet objectif, nous proposons de *piloter* les algorithmes de traitement du signal par le *contexte courant* constitué par l'analyse du bruit de ligne et des arythmies en cours de reconnaissance. Grâce à la connaissance de ce contexte, le pilote choisit et paramètre les algorithmes pour augmenter les performances du système de monitoring. Le contexte permet aussi de détecter les cas défavorables dans lesquels certaines *tâches* de détection d'événements commettent trop d'erreurs. Le pilote choisit alors de baser la reconnaissance d'arythmies sur les événements correctement détectables. De cette façon, la caractérisation de l'arythmie s'appuie uniquement sur les informations les plus fiables. Les décisions prises par le pilote dépendent du contexte courant et reposent sur une expertise du domaine du traitement du signal et du domaine médical. Les premiers résultats sur le pilotage de la détection d'événements montrent la faisabilité et l'intérêt d'une telle approche.

**Mots-clés** : pilotage d'algorithmes, monitoring cardiaque, surveillance de systèmes dynamiques, reconnaissance de chroniques, diagnostic, applications de l'I.A.

## 1 Introduction

Les systèmes de monitoring médical, tels que *Guardian* (Larsson & Hayes-Roth, 1998) et *Calicot* (Carrault *et al.*, 2003), comprennent généralement deux parties distinctes : une partie abstraction temporelle dédiée à l'acquisition, au traitement et à l'analyse du signal, et une partie diagnostic médical qui infère un diagnostic à partir des informations transmises par l'abstraction temporelle et d'une base de connaissance.

Dans ces systèmes, l'état courant du diagnostic médical n'est pas mis à profit pour optimiser l'abstraction temporelle. Pourtant, l'analyse de la ligne et le diagnostic médical courant sont des informations utiles pour adapter les algorithmes de traitement du signal qui constituent l'abstraction temporelle (Portet *et al.*, 2005). Soulas *et al.* (Soulas *et al.*, 1998) proposent 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. De plus, le diagnostic médical n'est pas pris en compte pour la sélection.

Pour modifier dynamiquement l'abstraction temporelle, nous proposons une approche de type *pilotage d'algorithmes* dont l'architecture permet d'adapter en ligne une chaîne de traitement selon le *contexte courant* de son utilisation. Le pilotage d'algorithmes s'inspire de différents travaux de la littérature.

Shekhar *et al.* (Shekhar *et al.*, 1994) introduisent le pilotage de programmes qui représente 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 d'ajustement. 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 de l'architecture 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. De plus, leur méthode implique une séparation nette entre les connaissances purement traitement du signal et celles du domaine médicale et ne permet pas leur fusion pour prendre une décision. Karsai & Sztipanovits (Karsai & Sztipanovits, 1999) 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 piloter le système de monitoring Calicot. Dans notre étude, l'abstraction temporelle est 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 des informations courantes du système, rassemblées sous le nom de contexte courant. Les tâches peuvent aussi être désactivées en fonction de la possibilité technique de leur réalisation, ce qui nécessite une adaptation du niveau de détail avec lequel le diagnostic médical est effectué. Contrairement aux approches présentées, notre pilote centralise les informations puis déduit l'ensemble des modifications à effectuer sur la chaîne de traitement.

Après une introduction aux arythmies cardiaques et au système de monitoring Calicot en section 2, les motivations de notre étude sont explicitées en section 3. Puis, la

nouvelle architecture et les bases de connaissance sont décrites en section 4. Le moteur d'inférences du pilote avec ses règles de pilotage sont ensuite détaillés en section 5. Enfin, les résultats du pilotage d'une tâche de détection sont présentés en section 6.

## 2 Présentation du contexte de l'étude

Nous développons un système de surveillance de patients qui présentent des *arythmies cardiaques*. C'est une évolution du système Calicot que nous détaillons après une brève introduction aux arythmies cardiaques.

### 2.1 Les arythmies cardiaques et l'électrocardiographie

Avant d'introduire les arythmies cardiaques, il est nécessaire de rappeler quelques généralités sur le cœur. 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.

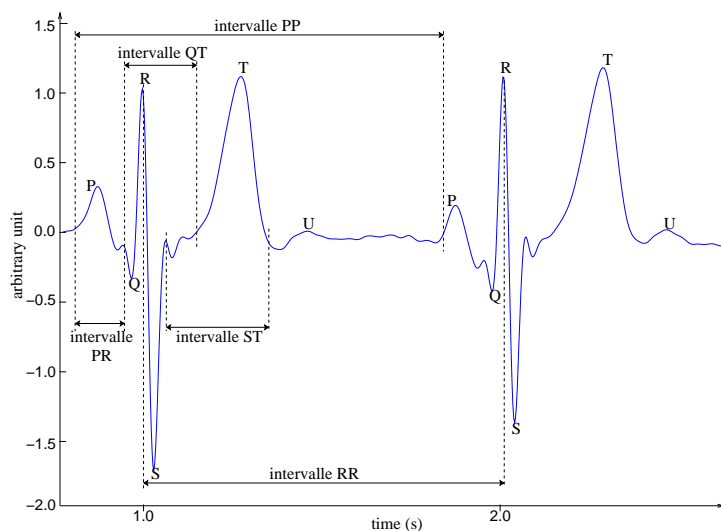


FIG. 1 – Ondes et intervalles standard d'un électrocardiogramme normal.

Le tracé de la figure 1 représente les ondes et intervalles standards étudiés en électrocardiographie. Les ondes P et 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 tachycardie *supra-ventriculaire*. Le contexte du patient (médicaments, âge, pathologies, etc.) doit aussi être pris en compte pour le diagnostic.

## 2.2 Le système de monitoring Calicot

*Calicot* (Cardiac Arrhythmias Learning for Intelligent Classification of On-line Tracks) (Carrault *et al.*, 2003) est un système de reconnaissance et d'apprentissage d'arythmies cardiaques. Sa principale caractéristique est d'associer trois domaines distincts : l'apprentissage symbolique, le traitement du signal et la reconnaissance de chroniques. Les arythmies sont vues comme *des événements temporellement liés entre eux*, ce qui est la définition même d'une chronique (Cordier & Dousson, 2000). Succinctement, Calicot est composé de trois principaux modules : un module d'*apprentissage des arythmies cardiaques*, un module d'*abstraction temporelle*, et un module de *reconnaissance de chroniques* (voir figure 2).

### *L'apprentissage des arythmies cardiaques (hors ligne)*

L'apprentissage des arythmies cardiaques est réalisé par programmation logique inductive, à partir d'exemples positifs et négatifs d'ECG d'arythmies annotés. L'apprentissage produit des règles de reconnaissance qui sont traduites en modèles de chroniques. Ainsi, à chaque arythmie correspond au moins un modèle 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 composée d'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. C'est cette partie qui réalise le *diagnostic médical*.

Calicot présente des performances satisfaisantes sur les exemples d'arythmies et de signaux traités (Carrault *et al.*, 2003). Cependant, le système reste sensible aux erreurs de l'abstraction temporelle qui peuvent perturber la reconnaissance des arythmies.

### 3 Introduction du pilotage dans Calicot

Nous partons du principe que les informations de l'analyse du signal et de la reconnaissance d'arythmies en cours permettent l'amélioration de la chaîne de traitement. Ceci nous a amené à introduire un pilote pour effectuer les modifications de Calicot en fonction de ces informations. La structure de Calicot autorise un pilotage à trois niveaux : au niveau de l'activation des tâches, au niveau de la sélection de la précision de la reconnaissance d'arythmies et au niveau de l'adaptation des algorithmes de traitement du signal.

#### *Le pilotage des tâches*

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

*Filtering* : la tâche de filtrage sépare au maximum le signal ECG des bruits parasites,

*QRSDetection* : la tâche de détection des QRS donne leur date d'occurrence,

*QRSClassification* : la tâche de classification des QRS les étiquette, et

*PWaveDetection* : la tâche de détection des onde P donne leur date d'occurrence.

Dans Calicot, chaque tâche est constamment activée. Or, selon les cas, certaines tâches ne peuvent pas être accomplies correctement. Par exemple, lorsque la ligne est trop bruitée, la détection de l'onde P est vouée à l'échec et entraîne beaucoup d'erreurs. Dans ce contexte, cette tâche est pénalisante car elle fournit de fausses informations à la reconnaissance de chroniques. Pour fonder le diagnostic sur des informations fiables, une amélioration consiste à activer et désactiver les tâches en fonction du contexte.

#### *Le pilotage de la reconnaissance d'arythmies*

Le pilotage des tâches implique un pilotage de la reconnaissance d'arythmies. En effet, même si certaines tâches ne peuvent être activées, le diagnostic médical doit pourtant être établi. Par exemple, lorsque l'on se trouve en présence d'un rythme cardiaque rapide avec des QRS rapprochés, on peut être en présence de deux arythmies : une tachycardie ventriculaire (pouvant dégénérer en fibrillation mortelle) et une tachycardie supra-ventriculaire (moins dangereuse). Lorsque l'abstraction temporelle est très précise (toutes les tâches actives), on étudie la présence ou non d'ondes P pour les discriminer. Mais, lorsque les ondes P ne sont pas techniquement détectables, le système peut utiliser la forme des QRS pour distinguer les deux arythmies. Pour représenter ces différentes stratégies de diagnostic, plusieurs jeux de modèles de chroniques (*i.e.* modèles d'arythmies) sont appris en fonction du niveau de détail de description de l'ECG. Le pilote a pour rôle de sélectionner les modèles de chronique correspondant au niveau de détail de l'abstraction temporelle (tâches actives), autrement dit, en fonction du contexte courant.

#### *Le pilotage des algorithmes de traitement du signal*

Dans l'abstraction temporelle, chaque tâche est réalisée par un ou plusieurs algorithmes de traitement du signal inter-connectés. Or, dans la littérature, pour réaliser une tâche, il existe plusieurs algorithmes dont les performances varient en fonction du

contexte d'utilisation. Portet *et al.* (Portet *et al.*, 2005) ont montré que les performances de différents détecteurs de QRS varient en fonction du contexte courant. Ainsi, le pilotage de l'abstraction temporelle choisit les algorithmes et leurs paramètres les plus adaptés aux tâches à accomplir en fonction du contexte courant.

Un pilotage efficace est donc entièrement dépendant de la détection précise du contexte courant que nous allons décrire dans la section suivante.

## 4 Le système de pilotage

L'architecture de notre système intègre des détecteurs de contexte et un pilote. Les performances du pilotage sont dépendantes de la détection du contexte. Nous introduisons tout d'abord notre représentation du contexte courant puis l'architecture générale, les bases de connaissance et l'architecture du pilote.

### 4.1 Le contexte courant et sa détection

Le contexte courant est composé de trois sous-contextes : le *contexte de ligne*, le *contexte arythmique*, et le *contexte patient*. Le contexte patient (âge, rythme ECG de base, etc.) est statique alors que le contexte de ligne et le contexte arythmique sont dynamiques et remis à jour régulièrement par deux détecteurs.

#### *Le contexte de ligne*

Le contexte de ligne est constitué du niveau et du type de bruit présent sur la ligne à l'instant  $n$ . Nous reprenons de Portet *et al.* (Portet *et al.*, 2005) les trois types de bruit mutuellement exclusifs ( $bw$ ,  $ma$ , et  $em$ ) pour trois rapports signal sur bruit (5, -5, et -15 dB). 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 prend donc une valeur parmi dix :

$$\forall n \in \mathbf{N}, \text{ctxline}(n) \in (\{bw, ma, em\} \times \{5, -5, -15\}) \cup \{no\_noise\}$$

Le détecteur de contexte de ligne est basé sur un détecteur classique de formes. Comme il est placé au début de la chaîne, il communique rapidement au pilote le contexte de ligne courant. Le pilote peut alors modifier l'abstraction temporelle avant que l'ECG ne soit traité.

#### *Le contexte arythmique*

Le contexte arythmique est constitué de la liste des arythmies en cours de reconnaissance. Le détecteur de contexte arythmique utilise les hypothèses du module de reconnaissance de chroniques afin d'établir une liste des arythmies les plus susceptibles d'apparaître. Cela 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 ici

par les lettres N, V, L, R, J, F, A, P, correspondant respectivement à battement Normal, extrasystole Ventriculaire, bloc de branche gauche (Left), etc.

## 4.2 La nouvelle architecture

L'architecture de notre système complète celle de Calicot. Le pilotage et les éléments nécessaires à son fonctionnement s'insèrent dans la chaîne de traitement. La figure 2 illustre les connexions entre les composants de notre système.

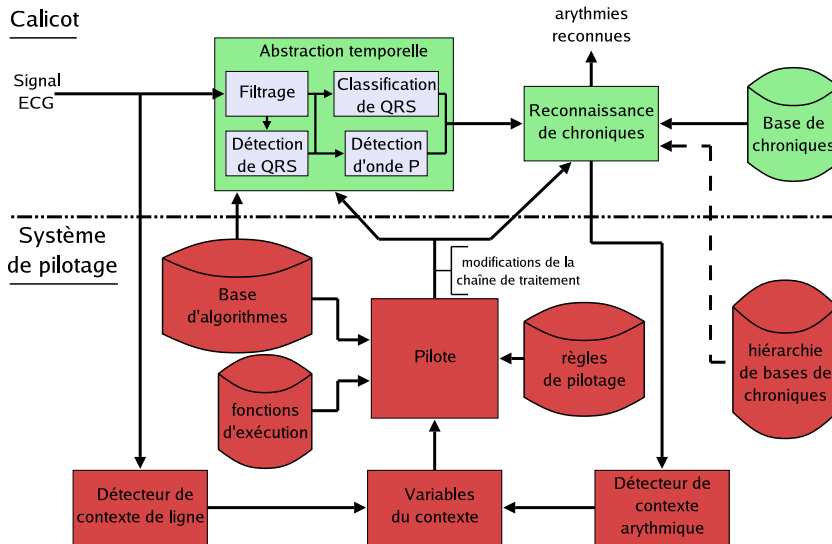


FIG. 2 – Architecture générale du système. La chaîne du haut représente le système Calicot et celle du bas représente les modules du système de pilotage.

Les deux détecteurs de contexte alimentent les variables du contexte. Le pilote effectue les modifications de la chaîne de traitement grâce aux variables du contexte et aux bases de règles de pilotage, d'algorithmes et de fonctions d'exécution. La base d'algorithmes contient un ensemble d'algorithmes de traitement du signal qui peuvent effectuer les tâches de l'abstraction temporelle. La base de fonctions d'exécution contient un ensemble de fonctions qui effectuent les modifications décidées par le pilote.

## 4.3 Les bases de connaissances

Le pilote intervient sur le système grâce à quatre bases de connaissances : une base d'algorithmes, une base de fonctions d'exécution, une base de chroniques et une base de règles de pilotage. Cette dernière est présentée avec le moteur d'inférences du pilote en section 5.



### La base d'algorithmes

La base d'algorithmes contient tous les algorithmes utilisés dans Calicot pour la réalisation des tâches de l'abstraction temporelle : *Filtering*, *QRSDetection*, *QRSClassification* et *PWaveDetection*. Dans cette étude, nous nous focalisons sur la tâche *QRS-Detection* pour laquelle trois algorithmes sont pilotés :

- pan* : le Pan et Tompkins (Pan & Tompkins, 1985) qui est un algorithme devenu standard dans la détection du QRS,
- gritzali* : le détecteur de Gritzali (Gritzali, 1988), autre algorithme standard de détection de QRS, et
- df2* : le détecteur de Okada modifié par Friesen *et al.* (Friesen *et al.*, 1990).

### La base de fonctions d'exécution

La base de fonctions d'exécution permet de réaliser la modification de la chaîne de traitement. Elles sont de trois types :

- start(task)* : la tâche *task* est activée, elle génère des événements qu'elle envoie à la reconnaissance d'arythmies,
- stop(task)* : la tâche *task* est désactivée, elle n'envoie plus d'événements,
- set(task, parameter)* : la tâche *task* est reparamétrée.

Les actions *start* et *stop* raniment ou mettent en veille la tâche afin de garder ses valeurs internes. Lorsqu'une tâche est en veille, elle n'effectue aucun traitement. Chaque tâche contient une méthode *set* capable de reparamétrer sa chaîne de traitement en fonction des paramètres passés.

### La base de chroniques

Lorsqu'une tâche de l'abstraction temporelle est désactivée, l'ECG est analysé de manière moins précise. Les niveaux de détail de description de l'ECG sont représentés par une hiérarchie de langages de description :

- L1 inclut les dates d'occurrence des QRS et l'écart temporel entre QRS,
- L2 ajoute à L1 les types (morphologies) de QRS,
- L3 ajoute à L1 les dates d'occurrence des ondes P, et
- L4 ajoute à L2 les dates d'occurrence des ondes P.

Les chroniques ont été apprises avec des exemples décrits par les quatre langages pour générer une hiérarchie de modèles de chroniques *C1*, *C2*, *C3* et *C4*. Ainsi, pour chaque combinaison de tâches actives, il existe une base de chroniques au niveau de détail correspondant. Par exemple, si seule la tâche *QRSDetection* est active alors il faut choisir la base de chroniques *C1*.

## 4.4 Le pilote

La structure du pilote est représentée par la figure 3.

Le pilote contient un *moteur d'inférences*, qui déduit les actions à effectuer sur la chaîne de traitement en fonction des variables du contexte, de la base de règles et aussi de son *contexte interne*. Le contexte interne du pilote contient les informations sur

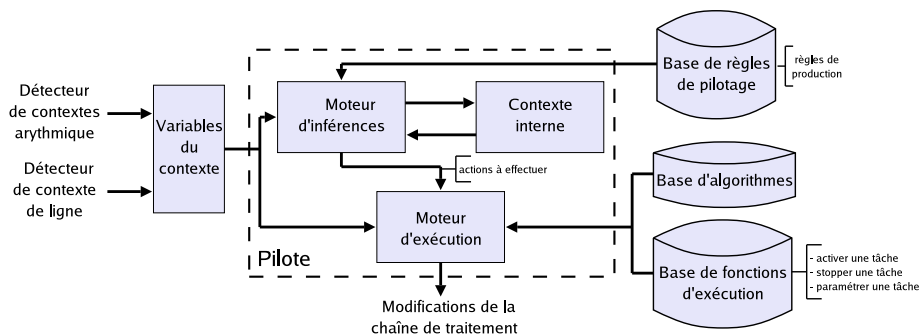


FIG. 3 – Architecture du pilote.

les tâches actives et les dernières modifications effectuées. En effet, l'abstraction temporelle de Calicot est vue comme la réalisation de plusieurs tâches interdépendantes. L'activation de ces tâches est dépendante du contexte et des autres tâches en cours. Par exemple, la tâche *QRSClassification* peut être activée uniquement si la tâche *QRSdetection* est active en même temps. C'est pourquoi, il faut mémoriser dans un contexte interne les dernières modifications effectuées sur le système.

Une fois les actions à effectuer déduites, le *moteur d'exécution* est chargé de la modification de la chaîne de traitement. Le moteur utilise des fonctions d'exécution qui permettent une gestion souple de chaque tâche. En effet, certaines tâches possèdent des variables internes à mettre à jour continuellement. Par exemple, la tâche *QRSdetection* possède une variable correspondant à la date courante. Si cette tâche est activée et désactivée sans précaution les données internes pourraient être perdues.

## 5 Le moteur d'inférences

Le moteur d'inférences est le cœur du pilote. Il déduit les actions à effectuer sur le système par un raisonnement qui utilise des informations du domaine du traitement du signal (tel que le contexte de ligne) et des informations du domaine médical (tel que le contexte arythmique). Sa connaissance est constituée du contexte courant et d'un ensemble de règles de pilotage représentées sous forme de règles de production.

**SI**

$$\langle (A \wedge B) \vee (C \wedge D) \rangle$$

**ALORS**

$\langle \text{faire action } i \rangle$

### 5.1 Le raisonnement du pilote

Le raisonnement du moteur d'inférences imite le raisonnement d'un expert ayant des connaissances médicales et techniques (traitement du signal). Le point de vue architec-

tural du raisonnement est illustré par la figure 4.

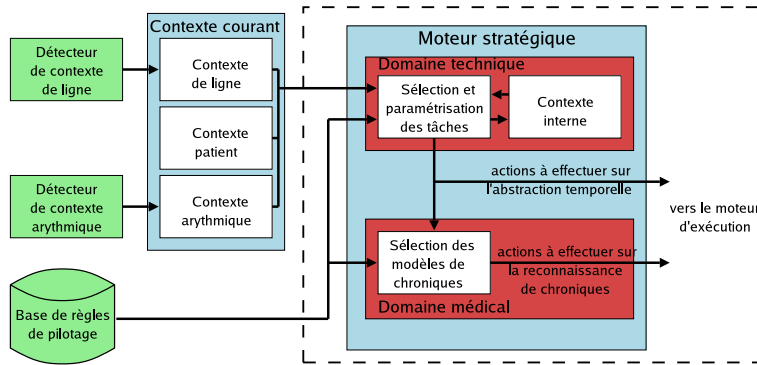


FIG. 4 – Le moteur d’inférences du pilote.

Dans le domaine technique, l’expert se sert du contexte courant mis à jour et de ses règles de pilotage pour déduire les tâches techniquement possibles et les meilleurs algorithmes pour les réaliser. Il infère les actions à effectuer pour modifier l’abstraction temporelle. Ensuite, il choisit la base de modèles de chroniques la plus adaptée en fonction des tâches possibles, autrement dit, en fonction de la précision de l’abstraction temporelle. Enfin, l’ensemble des actions à effectuer (changement de modèles, changement d’algorithmes, etc.) est transmis au moteur d’exécution. Par exemple, pour distinguer une tachycardie ventriculaire et une tachycardie supra-ventriculaire, la démarche intuitive est de choisir une stratégie médicale basée sur l’onde P. Si les ondes P ne sont pas techniquement détectables (trop de bruit), la reconnaissance sera erronée. Mais, un expert possédant des connaissances techniques pourra choisir une stratégie basée uniquement sur le QRS. Si une tâche est techniquement possible, le raisonnement technique décidera quels algorithmes et quels paramètres choisir grâce aux variables du contexte. Par exemple, si la tâche de détection de QRS est activée alors que la ligne présente du bruit *em*, le détecteur *df2* sera choisi plutôt que le détecteur *pan*.

## 5.2 Les règles de pilotage

Les règles de pilotage sont principalement définies par un expert et sont regroupées en règles techniques de fonctionnement, en règles de choix de tâches, en règles de choix de modèles de chroniques, et en règles de pilotage des algorithmes de traitement du signal. Nous détaillons ces classes de règles avec des exemples.

### Règles techniques de fonctionnement

Les règles techniques de fonctionnement mettent à jour des variables utiles pour la suite du pilotage. Par exemple, la règle suivante permet d’initialiser une variable pour désactiver la tâche de détection d’ondes P.

**SI**

$\langle \text{ctxligne} \neq \text{no\_noise} \rangle$

**ALORS**

$\langle \text{tooMuchNoiseForPWave} \rangle$

#### Règles d'activation des tâches

Les tâches sont activées principalement selon le contexte de ligne, par exemple pour activer *PWaveDetection* il est nécessaire d'avoir une ligne non bruitée.

**SI**

$\langle \neg \text{Active}(\text{PWaveDetection}) \wedge \neg \text{tooMuchNoiseForPWave} \rangle$

**ALORS**

$\langle \text{start}(\text{PWaveDetection}) \rangle$

S'il est techniquement possible d'obtenir la tâche de détection d'ondes P, alors la tâche est activée mais seulement si elle n'était pas déjà active. Après cette activation, le prédicat *Active(PWaveDetection)* devient vrai. Ce type de règle existe pour chaque tâche.

#### Règles de choix du niveau de détail de la reconnaissance de chroniques

Les modèles de chroniques doivent être adaptées à la précision de l'abstraction temporelle. Par exemple, si les tâches *QRSDetection*, *QRSClassification* sont actives et *PWaveDetection* est inactive, alors le reconnaiseur de chroniques doit utiliser la base de chroniques C2.

**SI**

$\langle \text{Active}(\text{QRSDetection}) \wedge \text{Active}(\text{QRSClassification}) \wedge \neg \text{Active}(\text{PWaveDetection}) \rangle$

**ALORS**

$\langle \text{utiliser la base de chroniques C2} \rangle$

#### Règles de pilotage d'algorithmes de traitement du signal

Après le choix des tâches, les règles de pilotage d'algorithmes permettent de les paramétrer. Par exemple, si la tâche *Active(QRSDetection)* est active il faut choisir le détecteur le plus approprié :

Pour la première règle, si le contexte de ligne dit qu'il y a présence de bruit bw à -5 dB et que le contexte médical informe qu'il a majoritairement des QRS de forme L, alors le détecteur *gritzali* est sélectionné. Ces règles, plus riches que les précédentes, sont dérivées de Portet *et al.* (Portet *et al.*, 2005).

**SI**

$\langle L \wedge bw \wedge SNR \geq -5dB \rangle$

**ALORS**

$\langle set(QRSDetection, changeQRS(gritzali, param(gritzali, ctxtligne))) \rangle$

**SI**

$\langle (LVF) \wedge no\_noise \rangle$

**ALORS**

$\langle set(QRSDetection, changeQRS(gritzali, param(gritzali, ctxtligne))) \rangle$

**SI**

$\langle (FVP) \wedge bw \wedge SNR \geq 0dB \rangle$

**ALORS**

$\langle set(QRSDetection, changeQRS(gritzali, param(gritzali, ctxtligne))) \rangle$

**SI**

$\langle em \wedge ((NVA \vee PVR) \wedge SNR = -15dB) \rangle$

**ALORS**

$\langle set(QRSDetection, changeQRS(df2, param(df2, ctxtligne))) \rangle$

**SI**

$\langle em \wedge (SNR = -5dB \wedge P) \rangle$

**ALORS**

$\langle set(QRSDetection, changeQRS(df2, param(df2, ctxtligne))) \rangle$

**SI**

$\langle default \rangle$

**ALORS**

$\langle set(QRSDetection, changeQRS(pan, param(pan, ctxtligne))) \rangle$

## 6 Résultats du pilotage de la tâche de détection du QRS

Le pilotage d'algorithmes a été testé sur cinq ECG générés à partir de la base standard MIT-BIH arrhythmia database<sup>1</sup>. Ces ECG ont été bruités avec du bruit clinique réel de type *bw*, *ma* et *em* et le pilotage s'est effectué avec des détecteurs parfaits de contexte. Les ECG ont une durée de 20 à 30 minutes et représentent au total 2 heures d'enregistrement. Ils sont composés des contextes suivants :

- ECG\_1 : 1200 QRS dont 300 de forme L sans bruit, 300 de forme V avec *bw* à -5dB, 300 de forme P avec *bw* à -5dB et 300 de forme V avec *bw* à -5dB
- ECG\_2 : 1800 QRS de forme L dont 300 bruitées *ma* à -5dB, 300 sans bruit, 600

---

<sup>1</sup><http://ecg.mit.edu/>

- bruités bw à -5dB et 600 sans bruit
- ECG\_3 : 1200 QRS de forme N dont 300 sans bruit, 300 bruités em à -15dB, 300 sans bruit et 300 bruités em à -15dB
  - ECG\_4 : 1200 QRS de forme R dont 300 sans bruit, 300 bruités em à -15dB, 300 sans bruit et 300 bruités em à -15dB
  - ECG\_5 : 1800 QRS de forme P dont 600 sans bruit, 600 bruités em à 5dB et, 600 bruitées em à -5dB

Chaque ECG est composé de 3 à 4 contextes différents de façon à évaluer non seulement, les performances des détecteurs dans ces contextes, mais aussi, le comportement du pilotage au moment des transitions entre contextes. Pour chacun de ces contextes, le pilote décide, avec les règles de pilotage, quel algorithme est le plus adapté. Pour chaque test, le nombre de faux négatifs  $FN$  (QRS non détectés) et de faux positifs  $FP$  (fausses alarmes) sont calculés pour obtenir le nombre d'erreurs,  $Ne = FP + FN$ . Le taux d'erreurs est  $Te = \frac{Ne}{N_{QRS}}$  où  $N_{QRS}$  représente le nombre total de QRS présents. Le seuil des détecteurs est choisi de manière optimale dans le sens où  $Ne$  est minimum. Les détecteurs sont ainsi utilisés au maximum de leur performance.

TAB. 1 – résultats du pilotage de la tâche de détection du QRS

ECG	1	2	3	4	5	total	
	$Ne$	$Ne$	$Ne$	$Ne$	$Ne$	$Ne$	$Te(\%)$
<i>pan</i>	<b>20</b>	<b>*91</b>	*240	*312	*367	<b>1030</b>	<b>14,3</b>
<i>gritzali</i>	<b>20</b>	*160	388	360	<b>*295</b>	1223	17
<i>df2</i>	307	278	<b>*174</b>	<b>*160</b>	*302	1221	17
<i>pilote</i>	20	88	185	167	304	764	10,6

\* algorithme utilisé par le pilote

Le tableau 1 montre que le meilleur algorithme (en gras) est différent pour chaque ECG. Le pilotage obtient toujours un score très proche du détecteur optimal car il tire parti, dans chaque contexte, des performances du meilleur algorithme. Dans le cas des signaux ECG\_3, ECG\_4 et ECG\_5, les résultats du pilote sont un peu moins bons que l'utilisation simple de *df2*. Cette différence est due aux transitions de contextes. En effet, pour être traité en temps réel, le signal est découpé en buffer. Or, même si un buffer contient deux contextes différents (transitions entre contextes), il sera traité par un seul détecteur. Par exemple dans ECG\_3, la transition entre le premier contexte sans bruit et le deuxième contexte avec *em* à -15dB est contenu dans un buffer et le choix du pilote a été de choisir *pan* alors que *df2* aurait été moins pénalisant. Cependant, le pilotage permet de diminuer de façon conséquente le nombre d'erreurs total générées par rapport à l'utilisation simple du meilleur détecteur en moyenne (ici *pan*). Ces résultats montrent l'intérêt d'utiliser un pilotage intelligent des algorithmes de détection de QRS en utilisant une fusion d'informations du domaine du traitement du signal (niveau de bruit) et du domaine médical (forme des QRS). Ceci ouvrent des perspectives encourageantes pour le pilotage des autres tâches de l'abstraction temporelle.

## 7 Conclusion et perspectives

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 la détection du contexte courant que nous définissons comme étant principalement constitué de l'analyse du bruit de ligne et du diagnostic médical. L'architecture proposée permet aussi de n'activer que les tâches fiables et d'adapter la reconnaissance d'arythmies à la précision courante du système. Les résultats montrent que le taux d'erreurs moyen de 14.3% obtenu avec un seul algorithme de détection de QRS passe à 10.6% avec le pilotage. Dans cette article, la précision de l'abstraction temporelle est guidée par les possibilités techniques de leur réalisation. Une autre partie de notre travail consiste à guider le pilotage de l'abstraction temporelle par la reconnaissance d'arythmies. En effet, si nous nous trouvons dans un contexte où il est possible de discriminer deux arythmies sans utiliser la détection d'ondes P, qui est très coûteuse en temps et peu robuste, alors le pilote n'activera pas cette tâche. Ce pilotage à deux niveaux, c'est-à-dire un pilotage qui prend en compte les besoins d'abstraction de la reconnaissance d'arythmies et la faisabilité des tâches de l'abstraction temporelle, fait partie de nos prochains travaux.

## Références

- CARRAULT G., CORDIER M., QUINIOU R. & WANG F. (2003). Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms. *Artificial Intelligence in Medicine*, **28**, 231–263.
- CORDIER M. & DOUSSON C. (2000). Alarm driven monitoring based on chronicles. In *Safe-process'2000*, p. 286–291.
- FRIESEN G. M., JANNETT T. C., JADALLAH M. A., YATES S. L., QUINT S. R. & NAGLE H. T. (1990). A comparison of the noise sensitivity of nine QRS detection algorithms. *IEEE Transactions on Biomedical Engineering*, **37**, 85–98.
- GRITZALI F. (1988). Towards a generalized scheme for QRS detection in ECG waveforms. *Signal Processing*, **15**, 183–192.
- KARSAI G. & SZTIPANOVITS J. (1999). A model-based approach to self-adaptive software. *IEEE Intelligent systems*, **14**, 46–53.
- LARSSON J. & HAYES-ROTH B. (1998). Guardian : An intelligent autonomous agent for medical monitoring and diagnosis. *IEEE Intelligent Systems*, **13**, 58–64.
- PAN J. & TOMPKINS W. J. (1985). A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, **BME-32(3)**, 230–236.
- PORTET F., HERNÁNDEZ A. I. & CARRAULT G. (2005). Evaluation of real-time QRS detection algorithms in variable contexts. *Medical & Biological Engineering & Computing*, **43(3)**, 381–387.
- SHEKHAR C., MOISAN S. & THONNAT M. (1994). Towards an intelligent problem-solving environment for signal processing. *Mathematics and Computers in Simulation*, **36**, 347–359.
- SOULAS T., CERTEN G. L., PICHON J. L. & CARRAULT G. (1998). Algorithm switching in real time monitoring. In *Symposium on Electronics and Telecommunications (ETC)*, p. 145–149, Timisoara.