Functional data analysis applied to neurology

Clément Bonvoisin, Pierre Ludmann

30 juin 2014

Résumé

Il s'agit de segmenter des signaux de marche, dans le cadre d'une collaboration du CMLA (ENS Cachan) et Cognac-G (Paris V).

On propose donc ici des algorithmes pour détecter des ruptures. Cela permet en aval aux médecins de mieux étudier les différents régimes de marche. Un algorithme efficace et rapide semble encore manquer ; la marche étant souvent abordée sur de longues durées [OJBS12].

Si la détection d'un unique changement trouve des implémentations reconnues, on a cherché à généraliser à de multiples ruptures. Aussi on a changé le paramètre de décision : on exige un nombre précis de résultats plutôt qu'un seuil de détection.

Malgré de fortes hypothèses de travail, on obtient des résultats statisfaisants sur des signaux réels et synthétiques bien que des améliorations restes possibles.

Son utilisation doit faire place à un apprentissage sur les segments de régime obtenus.

Remerciements

Nous tenions à remercier nos encadrants juniors Emile Contal pour avoir trop souvent fait passer ses stagiaires devant son travail et sa thèse et Laurent Oudre pour son aiguillage de connaisseur, nos encadrants seniors Eva Wesfreid et Nicolas Vayatis qui sont à l'origine du sujet de stage.

Table des matières

Chapitre 1

Introduction au problème

1.1 Motivations

La motivation initiale de ce stage provient de la médecine, et plus particulièrement de la neurologie. Le projet, piloté par le groupe Cognac-G, vise à analyser en détail des signaux physiologiques, issus d'une expérience très simple. Le protocole expérimental se décline comme suit :

On place sur le patient un ensemble de capteurs :

— sur la tête

— sur la ceinture, dans le dos

— sur chaque pied

Ces capteurs sont des centrales inertielles, qui permettent une mesure de l'accélération et de la vitesse angulaire du patient.

On lance alors l'acquisition :

— pendant quelques secondes, le patient est à l'arrêt

— puis, il commence à marcher sur une dizaine de mètres

— effectue un demi-tour

— et fait une marche retour.

Il s'arrête, et on peut alors arrêter l'acquisition.

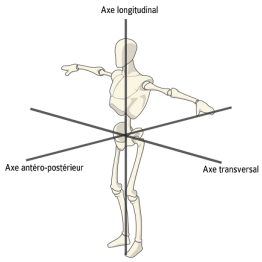


FIGURE 1.1 – Le repère (antéro-postérieur ; médio-latéral ; vertical)

On replace alors les signaux obtenus dans un repère adapté au corps humain, formé de trois axes : l'axe antéro-postérieur, l'axe transversal (aussi appelé médio-latéral), et l'axe longitudinal (aussi appelé axe vertical).

On obtient alors des signaux physiologiques ayant 6 composantes distinctes pour chaque

capteur, à une fréquence d'acquisition dépendante de la centrale inertielle utilisée ; actuellement, le groupe de travail Cognac-G dispose d'instruments permettant un échantillonnage à 100Hz.

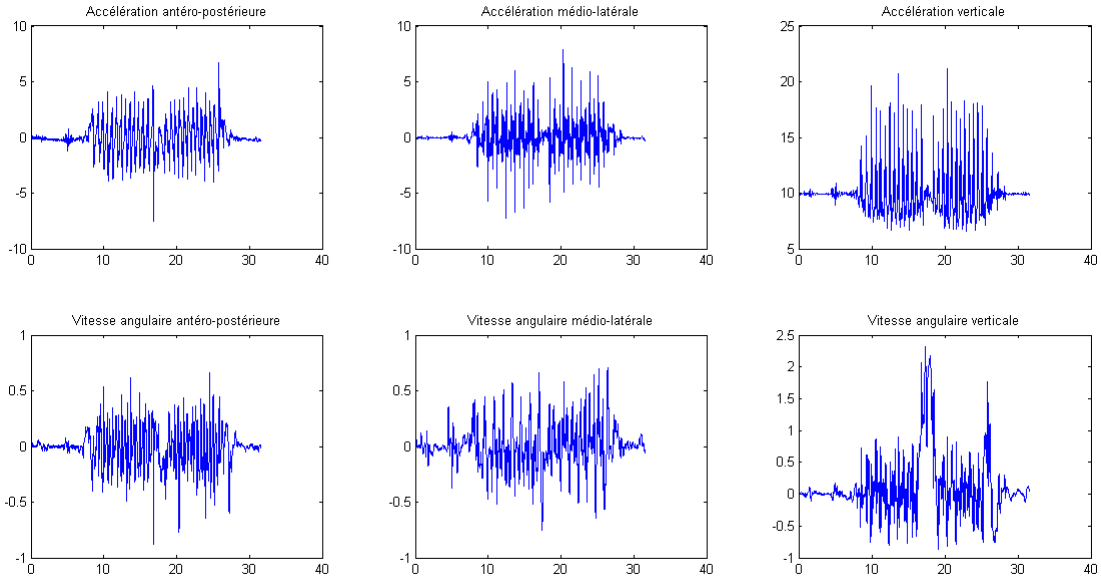


FIGURE 1.2 – Un exemple de signal de marche (enregistré à la ceinture)

Sur ces signaux apparaissent de manière claire les différentes phases de l'expérience :

- Dans un premier temps, le patient est à l'arrêt, les 6 signaux sont quasiment constants (on n'observe que du bruit)
- Dans une seconde phase, le patient commence à marcher : on peut observer une phase transitoire entre l'arrêt et la marche dite de croisière
- La troisième phase de l'expérience correspond à la marche de croisière : le patient effectue une dizaine de mètres
- On observe ensuite le demi-tour (particulièrement sur les composantes verticales du signal), qui dure environ 1 seconde
- Puis, on a une nouvelle phase de marche, retour cette fois-ci
- Finalement, le patient s'arrête : on a à nouveau une phase transitoire, puis l'arrêt total du patient (où il ne reste plus que du bruit)

Partant de ceci, on peut donc constater que les signaux acquis par les centrales inertielles peuvent être segmentés, qu'on peut isoler les différentes phases de l'expérience. Sur un seul signal, cela peut être fait de manière manuelle ; néanmoins, pour un neurologue enregistrant de manière régulière ce type d'expérience, il est concevable de désirer des algorithmes robustes permettant de traiter de manière automatique le problème de la détection des points de changement, décomposant ainsi le signal en sous-signaux correspondant à chacune des phases de l'expérience, afin de pouvoir les analyser séparément.

1.2 Formalisme mathématique

Afin de pouvoir traiter mathématiquement le problème, il nous faut tout d'abord poser des définitions claires sur l'objet du problème : il s'agit donc de définir ce qu'est une rupture, au sens mathématique.

La littérature propose une approche statistique du problème : on considère les signaux comme la réalisation d'une suite (finie) de variables aléatoires.

$$(x_i)_{i\in[\![1\,;N]\!]} = (X_i(\omega))_{i\in[\![1\,;N]\!]} \tag{1.1}$$

Ce formalisme, qui peut paraître quelque peu abstrait en première approche, permet d'exprimer de manière simple la notion de rupture dans un signal.

Sur les signaux précédents, on constate, par exemple, une différence nette d'écart-type entre la phase à l'arrêt et la phase de marche ; de même, sur la vitesse angulaire verticale, on constate un changement de moyenne entre les phases de marche et le demi-tour. Il paraît donc naturel de considérer la distribution statistique des différents points d'un signal multivarié pour formaliser le concept de rupture.

On comprend alors la définition donnée par la littérature d'un point de rupture à un instant t_0 :

$$\begin{array}{l} \forall i \in [\![1\,;t_0-1]\!], X_i \sim p_0 \\ \forall i \in [\![t_0\,;N]\!], X_i \sim p_1 \end{array} \tag{1.2}$$

On généralise de manière évidente au cas de R ruptures aux points $(t_r)_{r\in[\![0\,;n-1]\!]}$:

$$\begin{array}{l} \forall r \in [\![0\,;R-1]\!], \\ \forall i \in [\![t_{r-1}\,;t_r-1]\!], X_i \sim p_r \\ (\forall i \in [\![t_r\,;t_{r+1}-1]\!], X_i \sim p_{r+1}) \end{array} \tag{1.3}$$

où l'on a posé : $t_{-1} = 1$ et $t_R = N + 1$.

Ce problème étant formalisé, il s'agit maintenant de trouver des méthodes pour détecter ces points de rupture.

Chapitre 2

Algorithme CUSUM : résolution du cas d'une seule rupture

2.1 Principe de l'algorithme CUSUM

L'algorithme CUSUM fut proposé en 1954 par E.S. Page [Pag54] ; une methode "en ligne" utilisant un seuil de détection. Les signaux étant ici déjà réalisés, on lui préfère son adaptation "hors ligne" [BN93].

Dans le formalisme précédent, on conçoit que les méthodes de détection de rupture se fondent sur des outils statistiques. L'idée de base de l'algorithme CUSUM hors ligne est la suivante : comparer l'hypothèse qu'il existe une rupture dans le signal considéré à l'hypothèse qu'il n'y en a pas. L'outil utilisé pour cette comparaison est la notion de vraisemblance. Pour simplifier, on fera maintenant l'hypothèse d'indépendance des variables aléatoires $(X_i)_{i\in[\![1\,;N]\!]}$.

La fonction de vraisemblance quantifie la vraisemblance d'une hypothèse étant donnée une observation. Ici, l'observation faite est le signal, qui est la réalisation d'un nombre fini de variables aléatoires. On cherche à comparer l'hypothèse H_t qu'à l'instant t, il y a une rupture, à l'hypothèse H_0 qu'il n'y en a pas dans le signal. Les vraisemblances de ces hypothèses sont :

$$\begin{gathered}\forall t \in [\![1\,;N]\!], l(H_t) = \prod_{i=1}^{t-1} p_0(x_i) \prod_{i=t}^{N} p_1(x_i) \\ l(H_0) = \prod_{i=1}^{N} \tilde{p}(x_i)\end{gathered} \tag{2.1}$$

Introduisons alors le rapport de vraisemblance des hypothèses :

$$\forall t \in [\![1\,;N]\!], \Lambda_t = \frac{l(H_t)}{l(H_0)} = \frac{\prod_{i=1}^{t-1} p_0(x_i) \prod_{i=t}^{N} p_1(x_i)}{\prod_{i=1]}^{N} \tilde{p}(x_i)} \tag{2.2}$$

Pour simplifier, on fera ici l'hypothèse que nos lois sont issues d'une famille de lois indexée par un paramètre θ qui vit dans un espace donné.

On a ainsi :

$$p_0 = p_{\theta_0} \ ; \ p_1 = p_{\theta_1} \ ; \ \tilde{p} = p_{\tilde{\theta}}$$

On ne connaît pas, a priori, les paramètres $\tilde{\theta}$, θ_0 et θ_1. Pour les estimer, on va donc utiliser l'estimation du maximum de vraisemblance. Il en découle alors la formule suivante, à la base

de l'algorithme CUSUM, qui donne le logarithme du rapport de vraisemblance (*log-likelihood ratio*, en anglais) :

$$\forall t \in [\![1\,;N]\!], L_t = \ln\left[\frac{\sup_{\theta_0}\left\{\prod_{i=1}^{k-1} p_{\theta_0}(x_i)\right\}\cdot\sup_{\theta_0}\left\{\prod_{i=k}^{N} p_{\theta_0}(x_i)\right\}}{\sup_{\tilde{\theta}}\left\{\prod_{i=i}^{N} p_{\tilde{\theta}}(x_i)\right\}}\right] \tag{2.3}$$

Finalement, l'instant où il est le plus probable de trouver une rupture est celui qui maximise L_t. Ainsi, l'estimateur du maximum de vraisemblance (*maximum likelihood estimator*) par CUSUM du point de rupture d'un signal est :

$$\hat{t_0} = \arg\max_{1\leq t\leq N} L_t \tag{2.4}$$

2.2 Simplification dans le cas de distributions gaussiennes

Dans le cas général, la formule 2.3 est peu exploitable en raison de la présence de bornes supérieures. Afin de simplifier le problème des estimateurs au maximum de vraisemblance , on peut se proposer de faire une hypothèse sur la famille de lois $(p_\theta)_{\theta\in\mathbb{R}^d}$.

On suppose donc, à partir de maintenant, que chacune des variables aléatoires qui composent le signal suit une loi gaussienne, *i.e.* :

$$p_{\mu,\sigma}(y) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right]$$

Les estimateurs aux maximas de vraisemblance de la loi normale sont alors connus [Was04]. Ceci nous permet alors de simplifier l'expression de L_t car les bornes supérieures sont atteintes par ces estimateurs. Notons que l'hypothèse du gaussien ramène le problème du changement de loi à trois cas : changement de moyenne, changement d'écart-type, et changement de moyenne et d'écart-type.

Dans le cas d'un signal multi-varié, on peut généraliser aisément tout ce qui a été dit précédemment en faisant l'hypothèse d'indépendance des différentes composantes du signal entre elles. On calcule alors les *log-likelihood ratios* des différentes composantes du signal indépendamment avant de les additionner, et de chercher le maximum de cette résultante. Tout du long, en vertu de cette remarque, on ne théorisera que le cas univarié.

Bien que nous n'allons pas immédiatement traîter les signaux physiologiques vus précédemment – ceux-ci comportent plusieurs ruptures, l'algorithme CUSUM ne permettant d'en détecter qu'une seule – il est de bon ton de joindre des exemples imagés à ce flot de formules. Les signaux utilisés dans cette partie sont des signaux synthétiques, obtenus grâce à MATLAB.

Les formules sont données directement, les calculs étant directs et sans difficulté.

2.2.1 Changement en moyenne

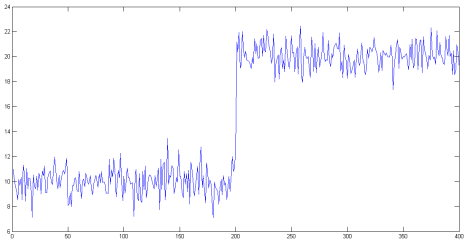
Faisant l'hypothèse d'un écart-type constant sur l'ensemble du signal, le paramètre θ est ici uniquement la moyenne μ.

2.3 se réécrit donc après calculs :

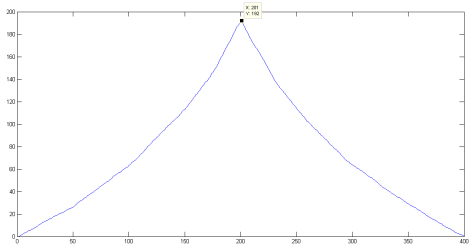
$$\forall t \in [\![1\,;N]\!], L_t = \frac{1}{2\sigma^2}\left[(t-1)\hat{\mu_0}^2 + (N-t+1)\hat{\mu_1}^2 - N\hat{\hat{\mu}}^2\right] \tag{2.5}$$

σ étant donc fixé au préalable. Et $\hat{\hat{\mu}}$, $\hat{\mu}_0$ et $\hat{\mu}_1$ sont les estimateurs $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$ du signal respectivement de 1 à N, de 1 à $t-1$ et de t à N.

On se place dans le cas d'un changement en moyenne de $\mu_1 = 10$ à $\mu_2 = 20$, avec $t_0 = 201$. On utilise cette fois l'équation 2.5 pour obtenir la figure 2.1b.



(a) Pour $t \leq 200$, $X_t \sim \mathcal{N}(10,1)$; pour $t > 200$, $X_t \sim \mathcal{N}(20,1)$



(b) *Log-likelihood ratio* du signal précédent

FIGURE 2.1 – Exemple de détection d'une rupture par la moyenne

On obtient donc ici l'estimateur du point de rupture par CUSUM, avec un score de 192 :

$$\hat{t_0} = 201; \hat{t_0} - t_0 = 0$$

De même que dans le cas précédent, on observe la robustesse de l'estimateur par CUSUM du point de rupture de ce signal. Pour quantifier cette robustesse, on peut s'appuyer sur la déviation quadratique moyenne de cet estimateur (RMSD), en répétant 5.000 fois l'expérience précédente, on trouve : $RMSD = 0$.

2.2.2 Changement en écart-type

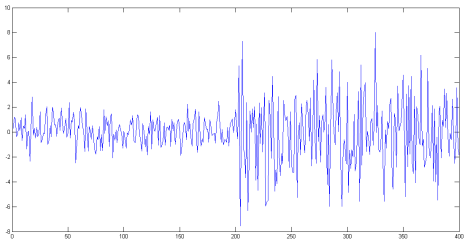
Faisant cette fois l'hypothèse d'une moyenne constante sur l'ensemble du signal, le paramètre θ est ici uniquement l'écart-type σ.

2.3 se réécrit alors :

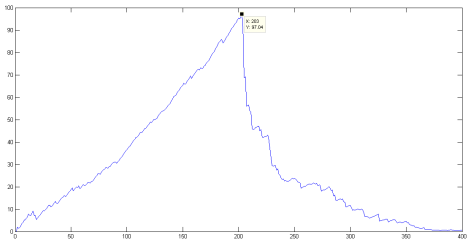
$$\forall t \in [\![1\,;N]\!], L_t = N\ln(\hat{\hat{\sigma}}) - (t-1)\ln(\hat{\sigma_0}) - (N-t+1)\ln(\hat{\sigma_1}) \tag{2.6}$$

μ étant préalablement fixé. Et $\hat{\hat{\sigma}}$, $\hat{\sigma}_0$ et $\hat{\sigma}_1$ sont les estimateurs $\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}$ du signal respectivement de 1 à N, de 1 à $t-1$ et de t à N.

On se place dans le cas d'un changement en écart-type de $\sigma_0 = 1$ à $\sigma_1 = 3$, à l'instant $t_0 = 201$.



(a) Pour $t \leq 200$, $X_t \sim \mathcal{N}(0,1)$; pour $t > 200$, $X_t \sim \mathcal{N}(0,3)$



(b) *Log-likelihood ratio* du signal précédent

FIGURE 2.2 – Exemple de détection d'une rupture par l'écart-type

L'équation 2.6 permet alors d'obtenir la figure 2.2b

Ce comportement est le comportement typique de la courbe d'un *log-likelihood ratio* sur un signal gaussien comportant une rupture. On observe ici que le maximum de la courbe est à l'instant 203, avec un *log-likelihood ratio* $L_{203} = 97,04$. On a donc l'estimateur par CUSUM du point de rupture :

$$\hat{t_0} = 203; \hat{t_0} - t_0 = 2;$$

Ainsi, l'estimateur de points de rupture par CUSUM semble robuste dans le cas d'un changement par écart-type. Pour 5.000 signaux différents suivant les caractéristiques précédentes, on trouve : $RMSD = 2,605$.

2.2.3 Changement en moyenne et en écart-type

Dans ce dernier cas, on doit estimer à la fois la moyenne et l'écart-type des différentes portions du signal. On obtient un résultat quasiment identique à 2.6 :

$$\forall t \in [\![1\,;N]\!], L_t = N\ln(\hat{\tilde{\sigma}}) - (t-1)\ln(\hat{\sigma}_0) - (N-t+1)\ln(\hat{\sigma}_1) \tag{2.7}$$

La seule différence intervient dans l'estimateur de l'écart-type : au lieu d'utiliser une moyenne fixée sur l'ensemble du signal (ce qui peut se traduire légitimement par son estimateur sur la globalité du signal), on utilise ici les estimateurs de la moyenne sur les différentes portions du signal :

$\hat{\tilde{\mu}}$, $\hat{\mu}_0$ et $\hat{\mu}_1$ sont les estimateurs $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$ du signal respectivement de 1 à N, de 1 à $t-1$ et de t à N. Et $\hat{\tilde{\sigma}}$, $\hat{\sigma_0}$ et $\hat{\sigma}_1$ sont les estimateurs $\hat{\sigma} = \sqrt{\frac{1}{n}\left[\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2\right]}$ du signal respectivement de 1 à N, de 1 à $t-1$ et de t à N.

Chapitre 3

Cas de plusieurs ruptures : implémentation dichotomique

Nous allons à présent nous intéresser au problème que l'on cherche à résoudre, à savoir celui d'un signal avec plusieurs ruptures. Voyons ici une première approche, qui semble particulièrement intuitive, à savoir l'approche dichotomique.

3.1 Principe théorique

Nous allons ici reprendre le principe de l'algorithme CUSUM tel qu'il est décrit dans le chapitre 2, et l'adapter au cas d'un nombre R de ruptures.

Le principe de la dichotomie est simple : à chaque étape de la dichotomie, on compare l'hypothèse qu'il y a une rupture dans le signal à l'hypothèse qu'il n'y en a pas. Il nous faudra ensuite comparer ces ruptures pour en choisir une. Voyons cela plus en détail.

À la première étape de la dichotomie, on applique l'algorithme CUSUM classique à notre signal tout entier. Notons qu'on ne cherche pas à valider l'hypothèse qu'il y a une rupture dans le signal, mais qu'on cherche bien à trouver l'instant où cette hypothèse est la plus vraisemblable : on conçoit donc que même si l'hypothèse est fausse (puisqu'il y a plusieurs ruptures dans le signal), son maximum de vraisemblance constituera un bon estimateur pour l'une des ruptures du signal. On a ainsi détecté une rupture, $\hat{t_0}$.

On peut alors séparer le signal en deux sous-signaux : $(X_t)_{t\in[1,\hat{t_0}-1]}$ et $(X_t)_{t\in[\hat{t_0},N]}$. Sur chacun de ces sous-signaux, on peut alors à nouveau effectuer l'algorithme CUSUM, et détecter ainsi deux nouveaux maximas de vraisemblance, que l'on notera $t_{1,1}$ et $t_{1,2}$, associés aux *log-likelihood ratios* $L^1_{t_{1,1}}$ et $L^2_{t_{1,2}}$. On peut alors comparer ces deux *log-likelihood ratios* et en déduire la rupture la plus vraisemblable : si $L^1_{t_{1,1}} > L^2_{t_{1,2}}$, on considère $\hat{t_1} = t_{1,1}$; sinon, on considère $\hat{t_1} = t_{1,2}$.

Considérons qu'on est dans le premier cas ici (le second se traite de manière analogue). On peut alors à nouveau découper en deux sous-signaux : $(X_t)_{t\in[1,\hat{t_1}-1]}$ et $(X_t)_{t\in[\hat{t_1},\hat{t_0}-1]}$. On peut à nouveau déterminer les maximas de vraisemblance de chaque côté, et obtenir ainsi deux instants $t_{2,1}$ et $t_{2,2}$, associés aux *log-likelihood ratios* $L^2_{t_{2,1}}$ et $L^2_{t_{2,2}}$. On compare alors

entre eux les *log-likelihood ratios* $L^2_{t_{2,1}}$, $L^2_{t_{2,2}}$ et $L^1_{t_{1,2}}$. Le plus élevé de ces trois, disons $L^1_{t_{1,2}}$ par exemple, permet de sélectionner la rupture suivante : ce sera ici $\hat{t_2} = t_{1,2}$.

On redécoupe alors le signal à partir de la dernière rupture observée : ici, on découpe $(X_t)_{t\in[\hat{t_0},N]}$ en $(X_t)_{t\in[\hat{t_0},\hat{t_2}-1]}$ et $(X_t)_{t\in[\hat{t_2},N]}$, et on réapplique l'algorithme CUSUM de chaque côté, en comparant à chaque fois tous les maximas qui n'ont pas été sélectionnés. On répète cette opération jusqu'à avoir le nombre R de ruptures voulu.

3.2 Implémentation algorithmique

L'implémentation dichotomique du CUSUM consiste alors à calculer un arbre de la façon suivante :

— Maintenir un ensemble de ruptures choisis initialement vide (les noeuds intérieurs).
— Maintenir un ensemble de ruptures éligibles qui contient initialement le résultat du CUSUM sur le signal entier (la frontière, les feuilles).
— En extraire la rupture t_0 qui a le plus grand ratio de vraissemblance.
— Calculer les deux ruptures – qui sont alors éligibles – en applicant le CUSUM entre la rupture choisie précédant chronologiquement t_0 et t_0 et entre t_0 et la rupture choisie suivant chronologiquement t_0.
— Recommencer jusqu'à obtenir le nombre désiré de ruptures.

Procéder par dichotomie permet d'aller très rapidement dans le traitement : la complexité d'un CUSUM est linéaire en la taille du signal à traiter, on se retrouve dans le pire des cas avec une complexité en $\mathcal{O}(NR)$ avec N la taille du signal et R le nombre de ruptures.

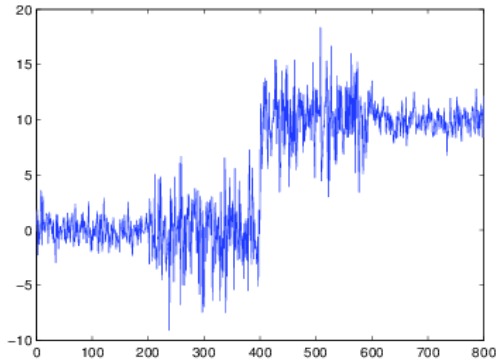
Mais remarquons que ce n'est pas nécessairement la première approche dichotomique qu'on pourrait envisager. Une version plus naïve serait de calculer un arbre de dichotomie complet de taille le nombre de ruptures directement – en retirant les moins bonnes si surplus. C'est en tout cas plus évident que d'aller faire éclore la meilleure de feuille de l'arbre de travail à chaque tour de boucle.

Sauf que l'arbre complet tombe dans un grave écueuil : plusieurs de ses sous-arbres sera vite confiné à calculer des ruptures là où elles sont minimes. Par exemple si la racine correspond à la rupture de début de marche tout le sous-arbre de son fils gauche va pointer sur des ruptures dans le régime immobile.

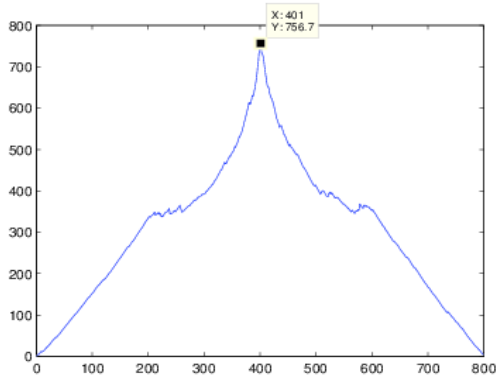
On pourrait être tenté de remplacer la demande de taille par la demande de profondeur. Et on choisirait les R meilleurs ruptures parmi les 2^R de l'arbre complet. Mais le souci vient alors de cette possibilité, réelle, de choisir une rupture sans choisir ses parents, *i.e.* choisir une rupture sans dire que la part de signal qui a permis de la choisir commence et finisse à de bons instants. Car si on ne choisit pas ses ruptures parentes c'est bien qu'on considère ce début et cette fin comme n'étant pas significatives.

Observons en détails la manière de procéder sur un exemple (signal **??**) Puisque le changement est autant en écart-type qu'en moyenne, on utilise la formule 2.7 pour le CUSUM.

Et on l'applique dès le début à tout le signal comme en 3.1b. Ceci nous donne $\hat{t}_0 = 401$ avec $L_{401} = 756,7$. On a notre première rupture, on utilise donc le CUSUM de part et d'autre de $\hat{t}_0$, donnant 3.2. C'est $\hat{t}_1 = 601$ qui a l'avantage avec $L_{601} = 88,98$ sur la rupture proposée par le CUSUM entre 1 et 400 – $L_{202} = 87,72$. Le CUSUM va alors chercher des ruptures entre 401 et 600 et entre 601 et 800 mais cela donne lieu à de très faibles vraisemblances (cf 3.3). Ainsi la troisième et dernière rupture choisie est $\hat{t}_2 = 202$ qui avait était proposé au tour précédent.

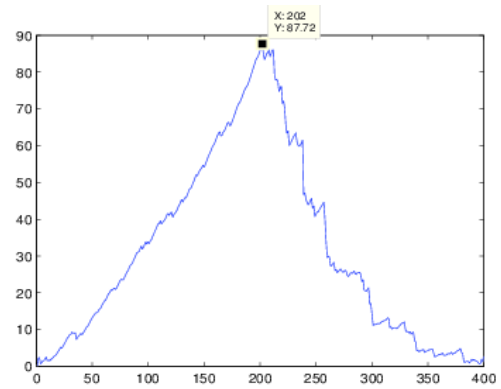


(a) De 1 à 200 : $\mathcal{N}(0,1)$; de 201 à 400 : $\mathcal{N}(0,3)$; de 401 à 600 : $\mathcal{N}(10,3)$; de 601 à 800 : $\mathcal{N}(10,1)$

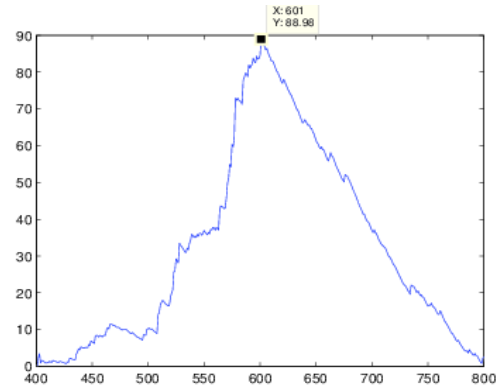


(b) Résultat du CUSUM sur tout le signal

FIGURE 3.1 – Au début de l'algorithme dichotomique : le signal et un premier CUSUM

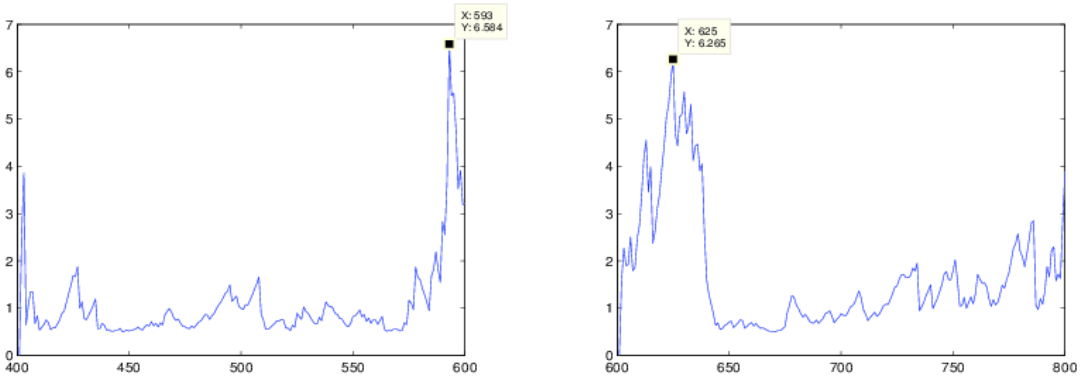


(a) Résultat du CUSUM de 1 à 400



(b) Résultat du CUSUM de 401 à 800

FIGURE 3.2 – Les nouveaux résultats de CUSUM après le premier choix



(a) Résultat du CUSUM de 401 à 600 (b) Résultat du CUSUM de 601 à 800

FIGURE 3.3 – Les nouveaux résultats de CUSUM après le deuxième choix

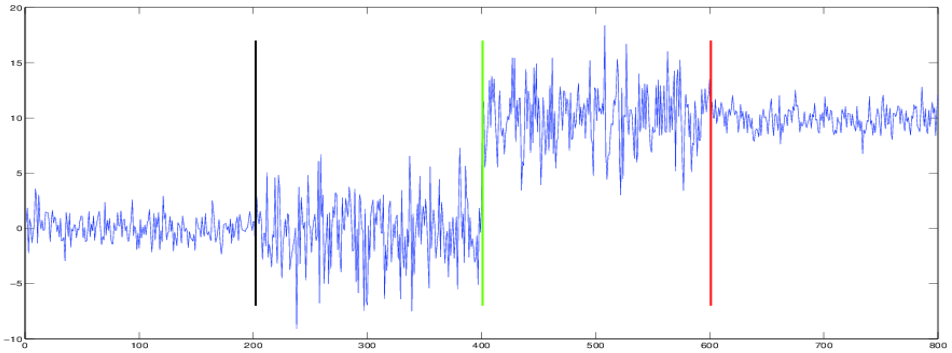


FIGURE 3.4 – Signal segmenté par la méthode dichotomique ; en vert : rupture 1 ; en rouge : rupture 2 ; en noir : rupture 3

Chapitre 4

Cas de plusieurs ruptures : implémentation par fenêtres

Il a été vu dans la partie précédente que l'implémentation dichotomique du CUSUM débordait du cadre théorique : en effet, on quantifie à chaque itération la vraisemblance de l'hypothèse d'une unique rupture dans le signal, alors que l'on désire en détecter plusieurs. De plus, on cherche, dans cette approche, à comparer entre elles des ruptures sur des signaux qui n'ont a priori aucun rapport les uns les autres (les tailles des signaux que l'on compare ne sont pas toujours les mêmes, par exemple). Pour pallier ce défaut, nous proposons ici une approche par fenêtres.

4.1 Principe théorique

L'algorithme CUSUM, tel qu'il est présenté dans le chapitre 2, ne permet de détecter qu'une seule rupture dans un signal donné. Nous traitons ici le cas de signaux ayant un nombre R de ruptures, avec $R \geq 1$. Si l'on désire se servir de l'algorithme CUSUM, pour rester dans le cadre théorique, il nous faudra nous assurer que le signal qu'on observe ne possède qu'une seule rupture.

Pour cela, en gardant les notations du chapitre 1, supposons que l'on dispose de l'écart minimal entre deux ruptures :

$$p = \min_{-1 \leq r \leq R-1} (t_{r+1} - t_r)$$

Ainsi, pour $t \in [p+1, N-p]$, on est assuré d'avoir au plus une rupture dans l'intervalle $[t-p, t+p]$. On peut donc appliquer l'algorithme CUSUM sur des fenêtres centrées en t et de taille $2p+1$. Le principe de l'implémentation par fenêtres est alors d'appliquer l'algorithme CUSUM sur chacun des intervalles $[t-p, t+p]$ pour $t \in [p+1, N-p]$, et de sélectionner ensuite les R meilleures ruptures.

Pour réduire le problème, simplifions quelque peu l'algorithme CUSUM : on considère ici uniquement l'hypothèse d'une rupture à la date t. On a ainsi, pour $t \in [t-p, t+p]$, deux hypothèses :

$$H_0^t : \forall i \in [t-p, t+p], X_i \sim p_{\tilde{\theta}}$$

$$H_1^t : \begin{cases} \forall i \in [t-p, t-1], X_i \sim p_{\theta_0} \\ \forall i \in [t, t+p], X_i \sim p_{\theta_1} \end{cases}$$

Il s'agit alors, comme c'est fait dans l'algorithme CUSUM pour une rupture, de comparer ces deux hypothèses, ceci pour tout t :

$$L_t = \ln \left[\frac{\prod_{i=t-p}^{t-1} p_{\hat{\theta_0}}(x_i) \prod_{i=t}^{t+p} p_{\hat{\theta_1}}(x_i)}{\prod_{i=t-p}^{t+p} p_{\hat{\theta}}(x_i)} \right] \tag{4.1}$$

Cette formule se simplifie de la même manière que ce qui a été fait dans le chapitre 2, selon les différents types de changement : en moyenne, en écart-type, ou les deux en même temps. On cherche alors à sélectionner les R instants où il est le plus vraisemblable de trouver une rupture, c'est-à-dire les R plus grands maximas locaux de la suite $(L_t)_{t\in[p+1,N-p]}$.

4.2 Implémentation algorithmique

Le principe théorique de la détection de ruptures par fenêtres éclaire sur la manière d'implémenter l'algorithme. Pour cela, il nous faut :

- Calculer les scores L_t selon la formule 4.3 après simplification (selon qu'on cherche un changement en moyenne, en écart-type, ou les deux)
- Déterminer $\hat{t_0} = \arg\max_{t\in[p+1,N-p]} L_t$
- Considérant $\hat{t_0}$ comme une rupture, on sait qu'il n'y a pas d'autre rupture dans la fenêtre $[\hat{t_0} - p, \hat{t_0} + p]$ par hypothèse ; on peut donc retirer de la suite (L_t) les valeurs $(L_t)_{t\in[\hat{t_0}-p,\hat{t_0}+p]}$
- On recommence alors sur la sous-suite ainsi obtenue, jusqu'à avoir n ruptures

Un tel algorithme a une complexité en $\mathcal{O}(NR)$. Observons en détails la manière de procéder sur un exemple (signal 4.1)

On peut ici assurer un intervalle entre deux ruptures de 200 points (dans la pratique, il faut estimer cet intervalle). Pour pouvoir observer ce qui se passe avant la première rupture et après la dernière, on prendra ici des fenêtres de 150 points de chaque côté. On travaille de plus avec des ruptures en moyenne et en écart-type. On obtient alors la suite (L_t) montrée sur la figure 4.2.

On demande alors à trouver 3 ruptures à partir de cette figure. On commence par détecter le maximum à $\hat{t_0} = 403$, où $L_{\hat{t_0}} = 184,2725$. On exclut alors les valeurs de L_t pour $t \in [253, 553]$. On cherche alors le nouveau maximum : on trouve $\hat{t_1} = 198$ pour $L_{\hat{t_1}} = 105,1670$. On recommence alors la procédure d'exclusion des points dans le voisinage de cette rupture, et on trouve le dernier maximum : $\hat{t_2} = 601$ pour $L_{\hat{t_2}} = 77,0389$. La segmentation obtenue est montrée sur la figure 5.1.

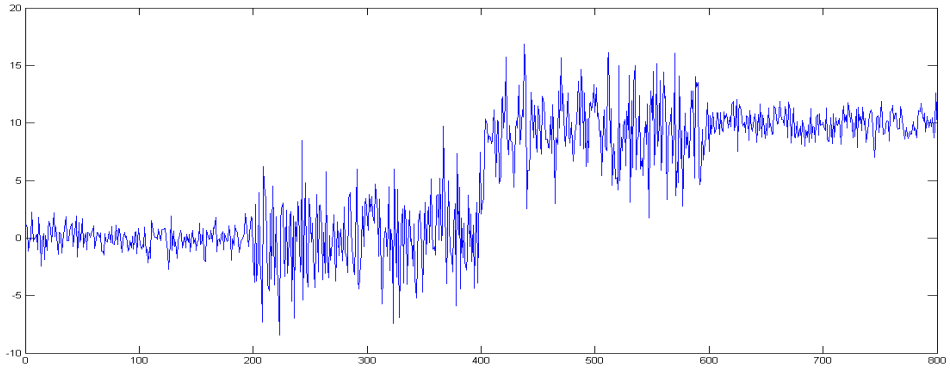


FIGURE 4.1 – De 1 à 200 : $\mathcal{N}(0,1)$; de 201 à 400 : $\mathcal{N}(0,3)$; de 401 à 600 : $\mathcal{N}(10,3)$; de 601 à 800 : $\mathcal{N}(10,1)$

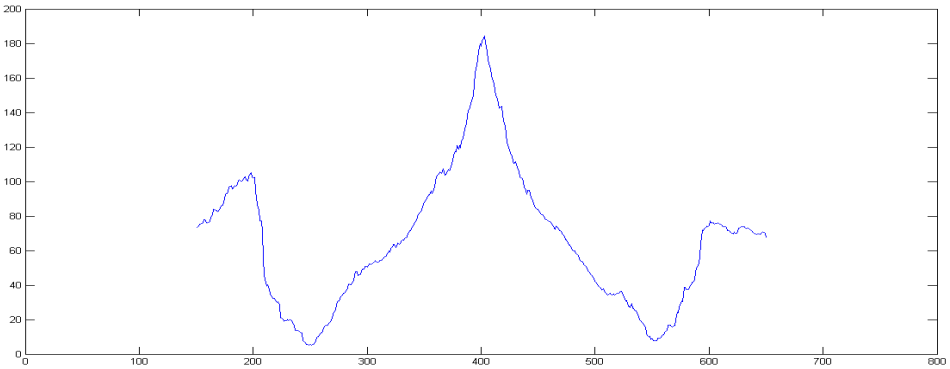


FIGURE 4.2 – Suite $(L_t)_{t\in[151,650]}$

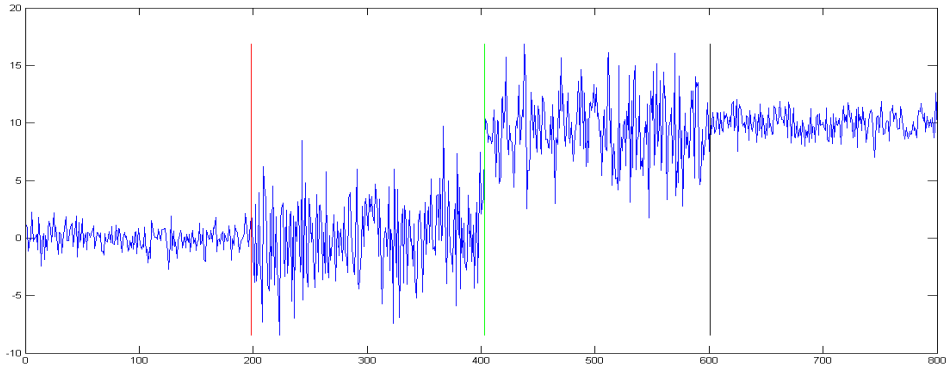


FIGURE 4.3 – Signal segmenté par la méthode par fenêtres ; en vert : rupture 1 ; en rouge : rupture 2 ; en noir : rupture 3

Chapitre 5

Évaluation des performances

5.1 Sur des signaux synthétiques

Ces signaux sont générés en jouant sur 3 critères :
— le nombre de ruptures
— l'espace moyen entre les ruptures
— l'espace minimal entre mes ruptures

Les valeurs de l'espace entre ruptures sont tirés uniformément sur l'intervalle ouvert de $\mathbb{R}^{+*}$ qui s'étend de l'espace minimal au double de l'espace moyen moins l'espace minimal. Les paramètres d'une distribution sont tout deux – moyenne et écart-type – tiré uniformément sur $]0\,;1[$. On utilise donc le CUSUM avec la formule 2.7. La fenêtre est évidemment de rayon l'espace minimal.

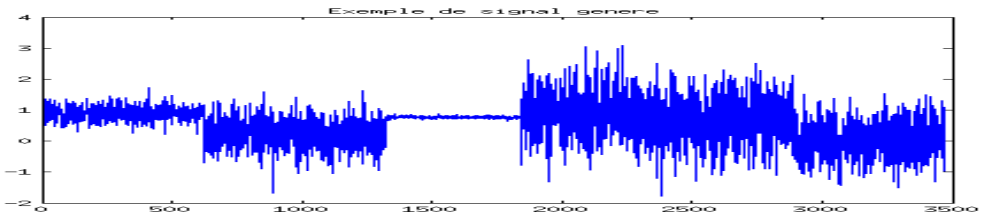


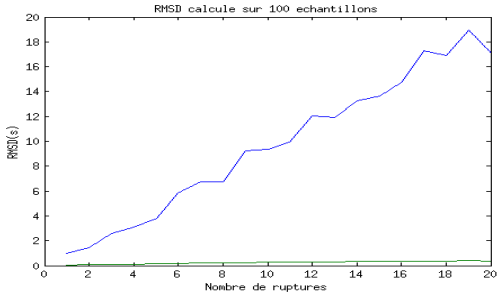
FIGURE 5.1 – Exemple de signal généré

On génère ainsi 100 signaux unidimensionnels pour un triplet d'argument donné afin de calculer dessus le RMSD – la déviation quadratique moyenne – mais aussi le taux d'erreur moyen. Le taux d'erreur étant le nombre de ruptures calculées à une distance supérieure de 10% de l'espace minimal d'une rupture réelle sur le nombre totale de ruptures (réelles ou calculées, il y en a autant) et les ruptures redondantes.

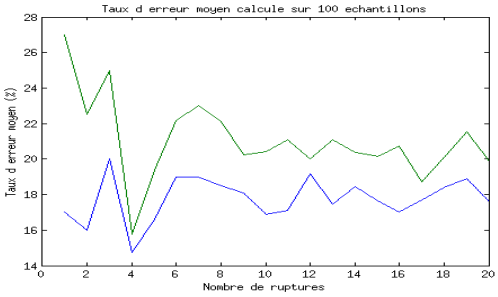
5.1.1 Selon le nombre de ruptures

On fixe l'espace minimal à 100 réalisations, en estimant que l'on cherche à débusquer des régimes physiologiques de l'ordre de la seconde et que les capteurs travaillent à 100Hz.

L'espace moyen est reglé à 500 points (3 à 4000 points par acquisitions présentant environ 6 ruptures). Et on fait varier le nombre de points de ruptures de 1 à 20, ce qui va largement

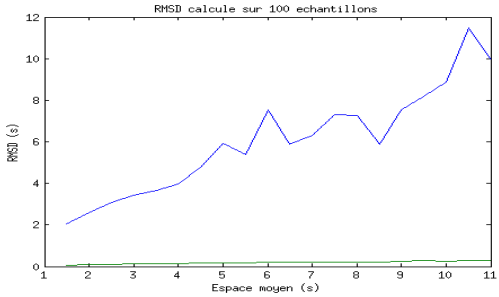


(a) RMSD en fonction du nombre de ruptures

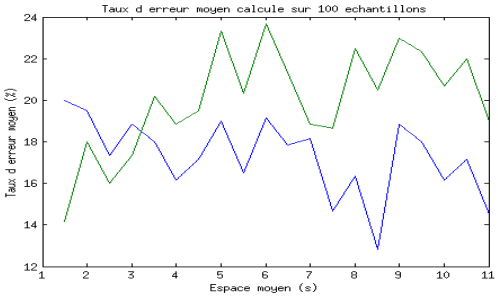


(b) Taux d'erreur moyen en fonction du nombre de ruptures

FIGURE 5.2 – Scores calculés sur 100 signaux présentant des ruptures à une distance minimale de 100 points (1s) et une distance moyenne de 500 points (5s) ; en bleu pour l'implémentation dichotomique, en vert pour l'implémentation par fenêtre

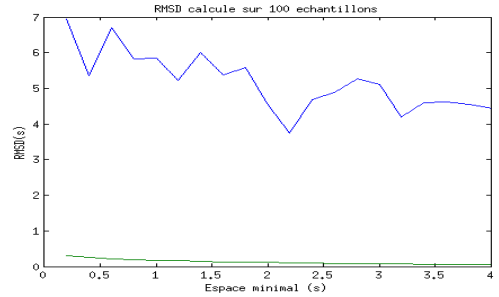


(a) RMSD en fonction de l'espace moyen entre les ruptures

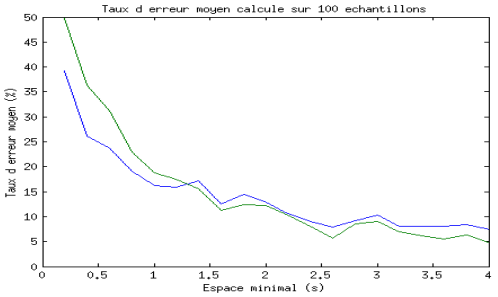


(b) Taux d'erreur moyen en fonction de l'espace moyen entre les ruptures

FIGURE 5.3 – Scores calculés sur 100 signaux présentant 6 ruptures à une distance minimale de 100 points (1s) ; en bleu pour l'implémentation dichotomique, en vert pour l'implémentation par fenêtre



(a) RMSD en fonction de l'espace minimal entre les ruptures



(b) Taux d'erreur moyen en fonction de l'espace minimal entre les ruptures

FIGURE 5.4 – Scores calculés sur 100 signaux présentant 6 ruptures à une distance moyenne de 500 points (5s) ; en bleu pour l'implémentation dichotomique, en vert pour l'implémentation par fenêtre

plus loin que la demande réelle.

On peut s'attendre à ce que l'implémentation la plus incapacitée par l'augmentation du nombre de ruptures soit l'approche dichotomique car on s'éloigne alors de plus en plus du cadre théorique. Tandis que l'approche par fenêtre ne fait pas la différence : il ne devrait pas y avoir de changement notable dans sa précision.

Les résultats sont observables sur la figure 5.2.

5.1.2 Selon l'espace moyen entre les ruptures

On fixe le nombre de ruptures à 6 et on reste avec un espace minimal de 100. L'espace moyen va varier de 150 à 1100 par pas de 50. Les résultats sont observables sur la figure 5.3.

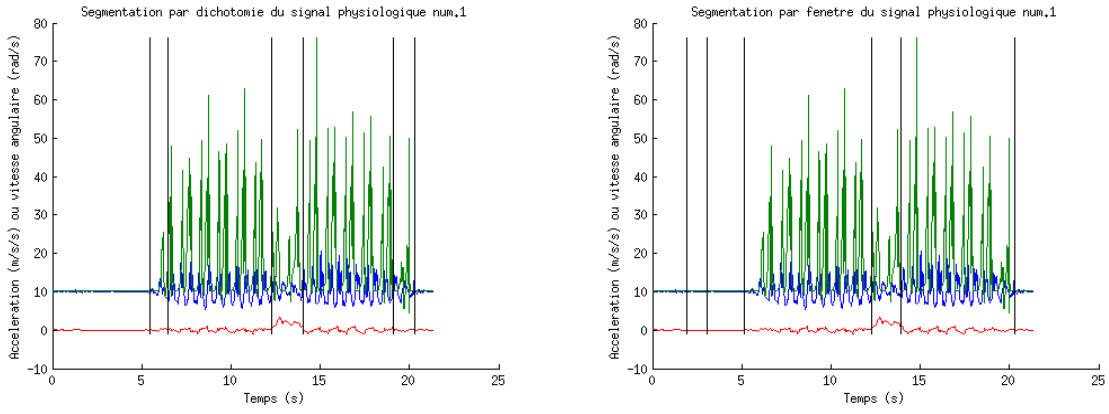
5.1.3 Selon l'espace minimal entre les ruptures

Ici les paramètres sont fixés à 6 ruptures et un espace moyen de 500 points. L'espace minimal va varier de 20 à 400 par pas de 20 ; et la fenêtre prend le même rayon que l'espace minimal – donc évolue. Les résultats sont observables sur la figure 5.7.

5.2 Sur des signaux physiologiques

Voici les résultats de segmentation des deux algorithmes comparés sur trois signaux physiologiques différents. Les signaux physiologiques ne sont pas directement les 12 composantes enregistrées à la ceinture et au pied droit. Ils ont été transformé avant d'être fournir aux algorithmes de segmentation qui ont été présenté. Ainsi les programmes reçoivent :

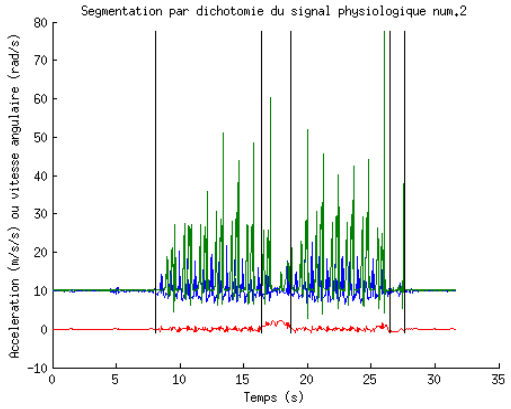
— la norme (euclidienne) de l'accélération enregistrée à la ceinture
— la norme de l'accélération enregistrée au pied droit
— la rotation propre enregistrée à la ceinture *i.e.* la vitesse angulaire suivant autour de l'axe vertical (orienté vers le bas)



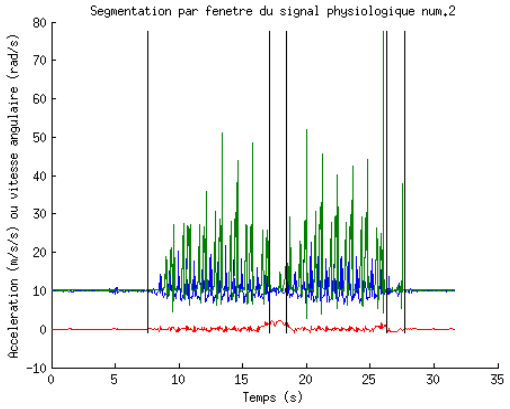
(a) Segmentation par dichotomie

(b) Segmentation par fenêtre de rayon 100 (1s)

FIGURE 5.5 – 6 ruptures sur un premier signal physiologique : en bleu l'accélétation enregistrée à la ceinture ; en vert l'accélération enregistrée au pied droit ; en rouge la rotation propre enregistrée à la ceinture

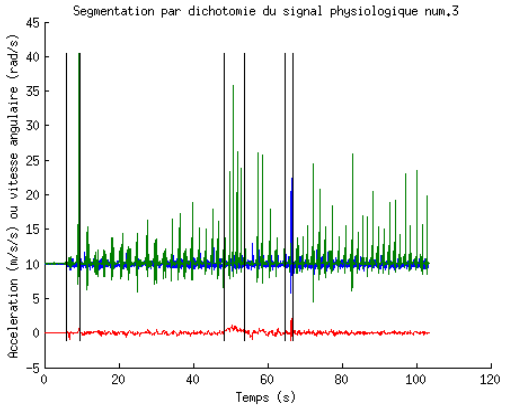


(a) Segmentation par dichotomie

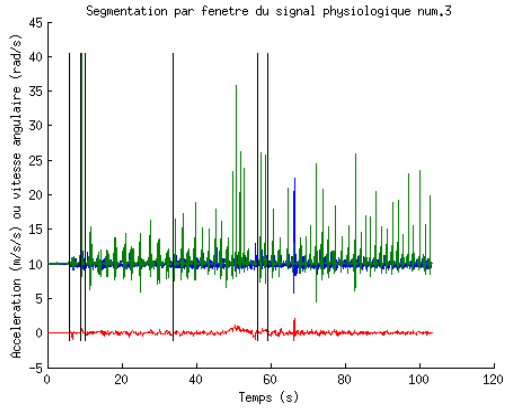


(b) Segmentation par fenêtre de rayon 100 (1s)

FIGURE 5.6 – 5 ruptures sur un deuxième signal physiologique : en bleu l'accélétation enregistrée à la ceinture ; en vert l'accélération enregistrée au pied droit ; en rouge la rotation propre enregistrée à la ceinture



(a) Segmentation par dichotomie



(b) Segmentation par fenêtre de rayon 100 (1s)

FIGURE 5.7 – 6 ruptures sur un deuxième signal physiologique : en bleu l'accélétation enregistrée à la ceinture ; en vert l'accélération enregistrée au pied droit ; en rouge la rotation propre enregistrée à la ceinture

Chapitre 6

Conclusion

Le problème posé initialement est assez intuitif : à partir de signaux physiologiques, on doit être capable de retrouver algorithmiquement les différentes phases de l'expérience (un aller-retour en marchant). On a vu que le formalisme mathématique qui y était associé était essentiellement statistique, et que les outils utilisés pour résoudre le problème l'étaient également. En particulier, on remarquera que le problème n'est pas résolu de manière exacte, mais approchée. On s'étonnera aussi de la difficulté pour résoudre un problème a priori très intuitif, car très visuel : on peut clairement segmenter à vue de nez les différents signaux observés dans ce rapport !

La partie précédente montre qu'en pratique, l'approche par fenêtres semble donner de meilleurs résultats que l'approche par dichotomie. En effet, si l'on semble avoir un taux d'erreurs plus élevé dans la plupart des cas avec l'approche par fenêtres qu'avec l'approche dichotomique, il importe de voir que le seuil de 10% fixé ici est avant tout arbitraire, et correspond à une qualité que l'on peut espérer sur les signaux physiologiques. Il est plus pertinent d'observer la déviation quadratique moyenne (ou RMSD) pour comparer les deux algorithmes.

En effet, le RMSD est une mesure de l'écart quadratique moyen entre une valeur réelle et son estimateur : il mesure, ici, l'écart quadratique moyen entre la rupture réelle et celle détectée par l'algorithme. On constate alors que les ruptures détectées par l'implémentation par fenêtres sont en moyenne plus proches des ruptures réelles que celles détectées par l'implémentation dichotomique. Ceci peut provenir du fait que l'implémentation dichotomique donne nécessairement des scores moins distingués, car on y quantifie la vraisemblance d'une hypothèse qui est nécessairement fausse (étant donné qu'on cherche de toute façon à détecter plusieurs ruptures).

Bibliographie

[BN93] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes : Theory and Applications*. 1993.

[OJBS12] Laurent Oudre, J. Jakubowicz, P. Bianchi, and C. Simon. Classification of periodic activities using wasserstein distance. *IEEE Transactions on Biomedical Engineering*, 2012.

[Pag54] E. S. Page. Continuous inspection scheme. *Biometrika*, 1954.

[Was04] Larry Wasserman. *All of Statistics : A concise course in statistical inference*. 2004.