

## Le tapis de course

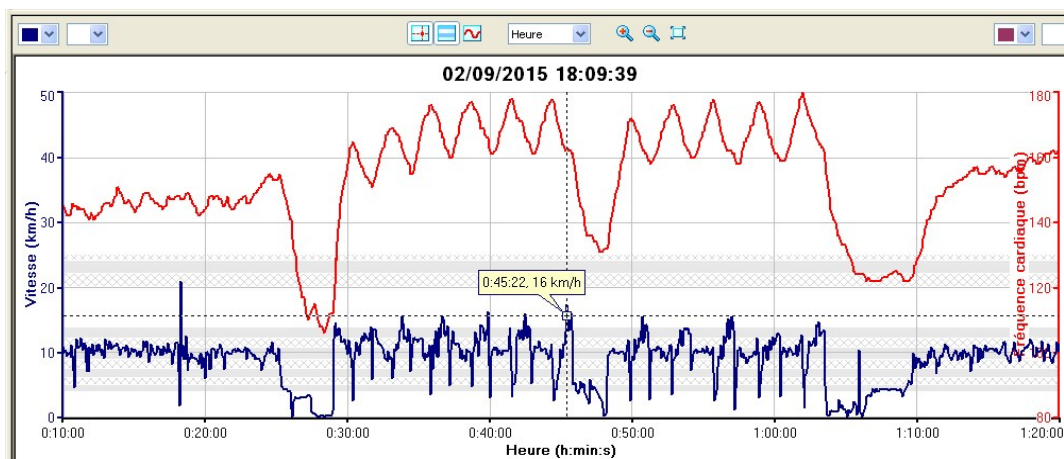
Les algorithmes sont à rendre sous la forme d'un fichier Libre Office. Vous serez attentif à la mise en page et n'utiliserez qu'une seule page par algorithme. L'ensemble de votre réalisation est à rendre sous la forme d'un fichier Zip à votre nom.

### 1. Mise en situation

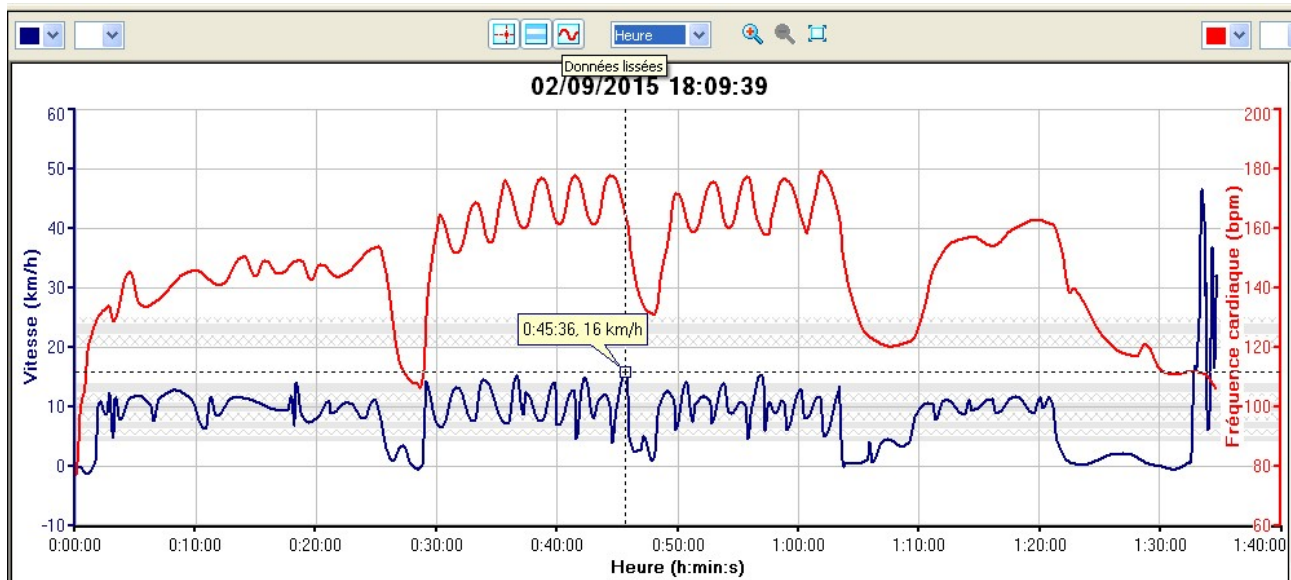
Afin d'optimiser leur entraînement, les sportifs ont besoin d'analyser les courbes **vitesse de course** et de **fréquence cardiaque** associées. Le système tapis de course dispose d'un calculateur chargé d'effectuer ces relevés : En plus du rythme cardiaque et de la vitesse de course d'autres paramètres comme la distance totale parcourue, et les calories consommées sont déduits par intégration. Ces différents paramètres sont affichés sur un écran graphique. En fin de séance, le sportif peut visualiser les valeurs enregistrées sous forme de courbes et les enregistrer sur sa clé USB.



L'acquisition d'une grandeur physique à intervalles de temps réguliers demande souvent la suppression de valeurs parasites. Un filtrage numérique de l'information est alors nécessaire. L'étude proposée ici développe plusieurs méthodes pour réaliser la courbe lissée (exemple sur une séance de fractionnée : courbe de vitesse en bleu, fréquence cardiaque en rouge).



Mêmes courbes obtenues après lissage :



## 2. Nuage de points

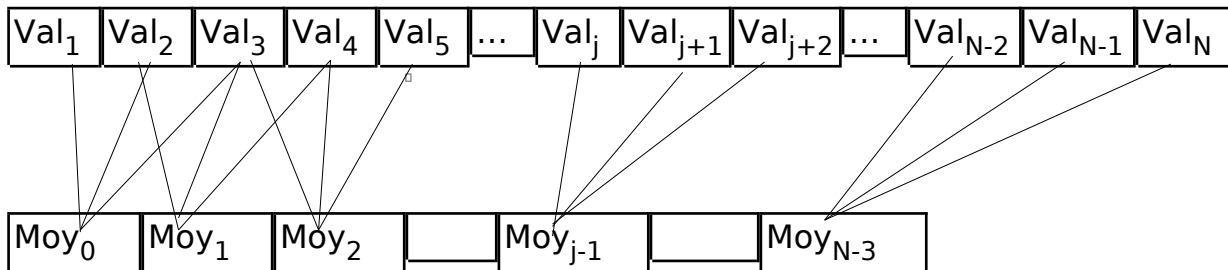
Cette méthode consiste à mettre en place à la source de la chaîne d'acquisition, un filtrage des points erronés. Pour chaque valeur de mesure, on réalise successivement **NB\_RELEVE** dans un intervalle de temps rapproché afin de constituer un ensemble de points autrement appelé « nuage de points ». De cet ensemble, la plus petite et la plus grande valeur sont retirées (ce sont sûrement des parasites) puis on effectue la moyenne des valeurs restantes.

- Réalisez un algorithme qui permet de saisir un nombre de valeurs fixé par la constante **NB\_RELEVE**, les stocke dans le tableau **nuagePoints** et calcule la moyenne. On utilisera la méthode de filtrage énoncée ci-dessus. La valeur moyenne est affichée ensuite. Les valeurs sont issues directement d'un convertisseur analogique/numérique et sont donc de type entier non signé sur 12 bits (vous mettrez donc en œuvre les mécanismes nécessaires pour que l'utilisateur ne puisse pas saisir des valeurs ne correspondant pas à ce critère).
- Simulez le fonctionnement de l'algorithme pour **NB\_RELEVE** égal à 5.
- Codez cet algorithme en langage C sous **NetBeans** dans un projet nommé **TestNuagePoints**.

### 3. Moyenne glissante

Les courbes en bleu et rouge sur la deuxième figure représentent les courbes lissées, pour les obtenir, on peut faire une moyenne de quelques valeurs autour d'un point puis, ainsi de suite pour chacun des points. Ici, on parle de moyenne glissante.

Cette méthode consiste par exemple à associer aux trois premières valeurs leur moyenne, puis, de refaire ce calcul pour tous les triplets successifs de la suite initiale selon le schéma ci-dessous :



- Sachant que le nombre d'acquisitions est fixé par une constante **NB\_ACQUIS** et que les grandeurs stockées sont de type réel, indiquez la déclaration du tableau **valeursBrutes** à mettre dans le lexique des variables.
- Quelle dimension doit posséder le tableau **valeursMoyennes** servant à recevoir le résultat du calcul de la moyenne mobile ? Justifiez votre choix.
- À partir de la description du calcul de la moyenne mobile, réalisez un algorithme permettant la saisie des données Val<sub>1</sub> à Val<sub>N</sub> (utiliser les constantes appropriées) dans le tableau **valeursBrutes** et calculant leur moyenne mobile dans le tableau **valeursMoyennes**.
- Réalisez l'algorithme de la fonction **AfficherTableau**, elle prend en paramètre d'entrée le tableau à afficher et le nombre d'éléments qu'il contient et réalise l'affichage d'un tableau de réels.
- Codez dans un projet nommé **MoyenneGlissante** les deux algorithmes présentés ci-dessus, vous afficherez le tableau des mesures brutes et le tableau des valeurs moyennes dans ce programme.

La déclaration de la fonction se fera de la manière suivante :

```
void AfficherTableau(float Donnees[], int nbElements) ;
```

L'appel de la fonction dans le programme principal se fera de la manière suivante :

```
AfficherTableau(valeursBrutes, NB_ACQUIS) ;
```

# Annexes

Pour ceux qui ne l'auraient pas encore fait :

1. Liens pour installer NetBeans, prendre la version **All** :

<https://netbeans.org/downloads/>

2. Attention NetBeans fonctionne avec le JDK Java 8.2 :

<https://www.oracle.com/technetwork/es/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>

3. Pour Windows, il est peut être nécessaire, au préalable d'installer mingw :

<https://mingw-w64.org/doku.php>