

ÉCOLE CENTRALE NANTES

Option DATASIM

Projet : **Détection automatique de figures de skateboard**

**Rendu final du projet.**

Une image contenant personne

Description générée automatiquement

*Elèves :*

*Julien Douxami ;*

*Pierre Libault*

**Sommaire**

[1. Description du projet 3](#_Toc99462228)

[1.1. Contexte général du projet 3](#_Toc99462229)

[1.2. Objectifs du projet 4](#_Toc99462230)

[1.3. Description de la démarche suivie 4](#_Toc99462231)

[2. Etat de l’art 5](#_Toc99462232)

[3. Travaux réalisés 8](#_Toc99462233)

[3.1. Architecture générale de l’algorithme 8](#_Toc99462234)

[3.2. Détection d’événements. 8](#_Toc99462235)

[3.3. Pré-traitement des données 9](#_Toc99462236)

[3.4. Evaluation de la performance 12](#_Toc99462237)

[3.5. Début d’algorithme de classification 12](#_Toc99462238)

[4. Travaux à venir 13](#_Toc99462239)

1. Description du projet
   1. Contexte général du projet

Le Skateboard est une discipline sportive très en vogue ces dernières années, avec près de 20 millions de pratiquants dans le monde[[1]](#footnote-1). Cette pratique recouvre différentes disciplines sportives qui sont apparues aux JO de Tokyo en 2021, qui ont toutes en commun la réalisation de figures acrobatiques, notées par un jury. L’officialisation de ce sport dans des compétitions de renom comme les Jeux Olympiques entraine une recherche de performance, qui peut être assistée par l’exploitation de données sportives spécifiques au skateboard.

C’est dans ce contexte général que nous souhaitons **développer un ensemble d’algorithmes de traitement de signal permettant de détecter et de caractériser les différentes figures réalisées par des pratiquants**, afin qu’ils puissent s’améliorer en visualisant leurs entrainements. Ces algorithmes se basent sur l’exploitation de données issues d’une centrale inertielle (Capteur Movuino) présenté ci-dessous :

Une image contenant texte

Description générée automatiquementUne image contenant personne

Description générée automatiquement

A gauche le Movuino placé dans le boitier et à droite le boitier fixé sur le skate

Ce capteur récolte les données d’accélération et de gyroscope, ce qui peut être suffisant pour détecter de multiples figures. Cependant, il est important de noter que certaines figures ne peuvent être identifiées uniquement à partir de données issues du skateboard puisque que les mouvements du pratiquant (un tour sur soi-même par exemple) peuvent modifier la figure. Dans le cadre de ce projet, nous nous intéressons principalement à certaines figures identifiables avec les données issues du skate uniquement.

Ce capteur a déjà été utilisé par Pierre lors de sa césure, ce qui facilite son utilisation et l’extraction des données. Cependant, certaines problématiques associées à l’extraction des données vont faire leur apparition au fur et à mesure des travaux réalisés.

* 1. Objectifs du projet

Comme mentionné précédemment, l’objectif général est de mettre en place un pipeline d’algorithmes qui permettent de détecter et de classer les différentes figures réalisées par un pratiquant de Skateboard.

Cet objectif général est découpé en plusieurs sous-objectifs intermédiaires :

* Détecter le nombre de figures réalisées sur une plage d’enregistrement ;
* Collecter suffisamment de données labellisées ;
* Mettre en place un protocole d’évaluation de la qualité des algorithmes ;
* Classer les différentes figures dans certaines classes ;
* Détecter si une figure est réussie ou non ;
* Déterminer des données pertinentes d’évaluation de la qualité d’une figure :
  + Hauteur ;
  + Vitesse initiale vs vitesse finale ;
  + Etc…

Les objectifs mentionnés ci-dessus sont les objectifs initiaux de ce projet, qui sont naturellement amenés à évoluer en fonction de la difficulté de certaines tâches et de l’apparition de nouvelles difficultés techniques à résoudre.

* 1. Description de la démarche suivie

Afin de mener à bien ce projet, nous avons dans un premier temps réalisé un état de l’art, sur les différentes approches qui permettent de détecter et de classer des mouvements. De plus en plus, les données deviennent un réel enjeu qui permettent d’améliorer des performances sportives, ce qui explique la quantité d’articles scientifiques traitant de ce sujet. Ainsi, nous n’avons pas eu de difficultés à répertorier une dizaine d’articles traitant de la reconnaissance de mouvement, et plus particulièrement de classification de figures (soit dans le Skateboard directement ou dans d’autres activités sportives similaires comme le Snow, surf…).

L’analyse de cet état de l’art et les discussions avec M. Le Carpentier (encadrant de ce projet) nous a poussé à adopter une démarche segmentée, avec deux taches principales à réaliser. Dans un premier temps, nous nous sommes concentrés sur la détection et le pré-traitement de données d’évènements dans un signal d’acquisition. Cela nous permet ainsi d’isoler des évènements, ce qui facilite ainsi la labellisation des données et la mise en place d’un algorithme de classification. Une fois cette première étape passée, nous souhaitons mettre en place un algorithme de classification puis de déterminer une méthode permettant d’évaluer la performance de cet algorithme.

1. Etat de l’art

Comme mentionné précédemment, la data devient un réel enjeu de performance dans le sport. En particulier, de plus en plus de scientifiques s’intéressent aux sports impliquant des figures comme le skateboard ou encore le snowboard, le ski… L’objectif est d’accompagner le sportif dans son entrainement en fournissant des métriques qualitatives sur les figures réalisées. Ainsi, il existe de multiples articles récents traitant du sujet de reconnaissance de figures. L’état de l’art réalisé regroupe certains articles jugés pertinents et qui nous ont fourni des pistes de réflexion pour mener à bien notre projet.

Dans l’article [1], les auteurs segmentent leur algorithme en deux parties :

* Une partie sur la détection d’événements. Cela permet d’identifier et d’isoler les figures dans un signal d’une grande durée, ce qui assure une réduction de la quantité de la donnée que l’on souhaite classer ;
* Un algorithme de classification qui détecte le type de figure réalisée.

Dans leurs algorithmes, les scientifiques exploitent l’information sur l’atterrissage après une figure, ce qui est une donnée à laquelle nous n’avons pas accès. La classification est ensuite assurée en étudiant 6 types de figures différentes identifiées par de multiples features statistiques sur le signal (moyenne, variance, kurtosis, skewness…). Au total, 343 signaux sont à disposition pour l’entrainement et l’évaluation de leurs algorithmes. La matrice de confusion obtenue est la suivante :

Une image contenant table

Description générée automatiquement

Figure 1 : Matrice de confusion obtenue dans [1].

**La méthode exploitée dans cet article assure ainsi une bonne performance de la classification mais nous ne possédons pas suffisamment de données pour envisager une classification exploitant autant de features statistiques. Cependant, segmenter l’algorithme en deux parties distinctes semble être une bonne idée pour évaluer la performance de détection et de classification de manière indépendante.**

Dans l’article [8], les auteurs exploitent cette fois-ci un réseau de neurones artificiel qui est entrainé avec 459 échantillons de données (80% sont utilisées pour l’entrainement et 20% pour la validation). L’architecture du réseau est schématisée ci-dessous :

Une image contenant texte, horloge

Description générée automatiquement

Figure 2 : Schéma du réseau de neurones exploité dans [8].

Pour assurer la performance d’un réseau de neurones artificiel, chaque paramètre du réseau doit être optimisé. Cette optimisation est assurée dans la phase d’entrainement, qui est d’autant plus performante qu’il y a de données disponibles. Notre quantité de données est bien trop faible pour espérer exploiter un tel algorithme. Cependant, le capteur utilisé par les auteurs de l’article fournit les mêmes types de donnée que le capteur que nous utilisons. L’article étudie en parallèle la performance de leur algorithme en n’exploitant qu’un seul axe sur les trois disponibles et l’axe de rotation Z semble être suffisant pour classer les différents types de figure. Cela peut s’expliquer par l’extraction de multiples features et leur interprétation par le réseau de neurones. Bien que nous ne pouvons pas exploiter un tel algorithme, cette méthode nous permet également d’envisager l’étude uniquement de certains axes qui dépendent de la figure étudiée lors de la classification nous donne une piste de réflexion pour étudier seulement certains axes en fonction de la figure étudiée.

D’autres articles comme [4] exploitent des données GPS pour améliorer la précision et le suivi du sportif. Il est intéressant de savoir qu’il est possible de prendre en compte ces données mais le capteur utilisé actuellement pour le projet ne permet pas de récupérer de telles données.

Enfin, l’article [3] recense de multiples études sur la reconnaissance de mouvements mais la plupart de ces méthodes exploitent des algorithmes qui nécessitent une phase d’entrainement ainsi qu’une phase de validation, qui ne sont possible que lorsque les données sont suffisamment abondantes. Ainsi, certains scientifiques comme dans l’article [7] exploitent des CWT ou des SVM qui semblent très performants mais restent inenvisageables dans le cadre de ce projet.

**Conclusion sur l’état de l’art**

Dans cet état de l’art, nous nous sommes concentrés sur les études qui se basent sur l’exploitation du même type de donnée :

* Accélération selon les trois axes de rotation ;
* Données sur le gyroscope.

Nous avons remarqué qu’il est fréquent d’exploiter des algorithmes de Machine Learning ou d’intelligence artificielle et que ces algorithmes fournissent de très bons résultats, que ce soit pour la détection d’événements ou bien pour la classification des figures. Cependant, ces algorithmes nécessitent une phase d’entrainement importante qui exploite de multiples données labellisées.

Or, ces données ne sont pas en libre-accès et nous souhaitons exploiter les données fournies par notre capteur. Ainsi, nous devons labelliser nous même ces données, ce qui est une tâche chronophage et nous ne pouvons pas regrouper suffisamment de données pour mettre en place ces algorithmes d’apprentissage. Cependant, ces différents articles nous ont fourni de bonnes pistes de réflexion sur l’architecture général de notre algorithme ainsi que sur certaines bonnes pratiques à adopter.

Ces articles seront d’autant plus pertinents dans le futur lorsque nous aurons été en mesure de collecter et de labelliser suffisamment de données.

# Bibliographie

[1] Benjamin & Kautz, Thomas & Schuldhaus, Dominik & Eskofier, Bjoern. (2015). IMU-based Trick Classification in Skateboarding.

[2] Emily E Cust, Alice J Sweeting, Kevin Ball & Sam Robertson (2019) Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance, Journal of Sports Sciences, 37:5, 568-600, DOI: 10.1080/02640414.2018.1521769

[3] Cust EE, Sweeting AJ, Ball K, Robertson S. Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance. J Sports Sci. 2019 Mar;37(5):568-600. doi: 10.1080/02640414.2018.1521769. Epub 2018 Oct 11. PMID: 30307362.

[4] T. Holleczek, J. Schoch, B. Arnrich and G. Tröster, "Recognizing turns and other snowboarding activities with a gyroscope," International Symposium on Wearable Computers (ISWC) 2010, 2010, pp. 1-8, doi: 10.1109/ISWC.2010.5665871.

[5] Anlauff, Jan & Weitnauer, Erik & Lenhardt, Alexander & Schirmer, Stefanie & Zehe, Sebastian & Tonekaboni, Keywan. (2010). A method for outdoor skateboarding video games. 40-44. 10.1145/1971630.1971642.

[6] Harding, Jason & Mackintosh, Colin & Hahn, Allan & James, Daniel & Estivalet, Margaret. (2009). Classification of Aerial Acrobatics in Elite Half-Pipe Snowboarding Using Body Mounted Inertial Sensors (P237). 10.1007/978-2-287-09413-2\_55.

[7] Abdullah MA, Ibrahim MAR, Shapiee MNA, et al. The classification of skateboarding tricks *via* transfer learning pipelines. *PeerJ Comput Sci*. 2021;7:e680. Published 2021 Aug 18. doi:10.7717/peerj-cs.680

[8] Corrêa, Nicholas & Lima, Julio & Russomano, Thais & dos Santos, Marlise. (2017). Development of a skateboarding trick classifier using accelerometry and machine learning. 33. 362-369. 10.1590/2446-4740.04717.

1. Travaux réalisés

L’ensemble des développements est réalisé sur Python. Nous appelons événement une particularité dans le signal, qui peut être soit une figure (ce que l’on cherche à identifier), ou un mouvement parasite (virage, récupération de planche…).

* 1. Architecture générale de l’algorithme

Comme mentionné dans l’état de l’art, nous nous sommes inspirés de l’article [1] pour construire notre algorithme de détection et de classification. Cela permet de dissocier la performance de détection et de classification, ce qui nous permet d’identifier quelle méthode fonctionne mal et quels sont les leviers techniques à lever pour améliorer la performance globale. Cette dissociation permet également de réduire la quantité de donnée à traiter par notre algorithme de classification. Ainsi, nous pouvons schématiser le fonctionnement de notre algorithme comme ci-dessous :

Signal Initial

Détection d’événements

1er test de performance

2ème test de performance

L’événement n’est pas une figure.

Classification de la figure.

L’encadré en rouge correspond aux événements qui ne sont finalement pas une figure mais un mouvement jugé parasite : un virage, une récupération de la planche… Nous pouvons noter que chaque test de performance ne dépend que de l’algorithme associé (soit uniquement la prédiction soit uniquement la classification) mais qu’il est également possible de réaliser un test global de la performance en considérant un signal d’entrée et si nous nous intéressons uniquement à la classification des figures dans ce signal. Dans la suite de ce document, nous allons nous intéresser à chaque étape de notre algorithme et nous allons décrire les différents modèles mis en place.

* 1. Détection d’événements.

L’utilisation du capteur étant maitrisée en amont du démarrage de ce projet, nous avons pu directement nous concentrer sur l’exploitation des données. Pour être en mesure d’analyser correctement les signaux, nous devons prétraiter les données en amont de la détection de figures.

* + 1. Pré-traitement des données.

Le pré traitement des données est une étape indispensable en traitement du signal car cela permet d’uniformiser et de mettre les données à la même « échelle », améliorant ainsi l’analyse de celles-ci. Pour les figures de skate, nous utilisons principalement 2 méthodes :

* Centrage des figures avec le temps moyen de la norme du gyroscope ;
* Normalisation des données d’accélération et gyroscopiques par l’énergie de leur norme respective ;

**Centrage des figures en exploitant le temps -moyen :**

Nous centrons les figures détectées de la même manière en calculant le temps moyen d’une figure par rapport à son énergie gyroscopique :

Une image contenant texte

Description générée automatiquement

Après avoir calculé le temps moyen de la figure, nous considérons que la durée de la figure est de 1.2 secondes, et que le temps moyen est au centre de cette figure. Cette durée est définie à l’expérience en considérant que les figures étudiées ne dépassent jamais cette durée. Ci-dessous un exemple de figure (encadrée en rouge dans le signal) extraite et centrée avec cette méthode.

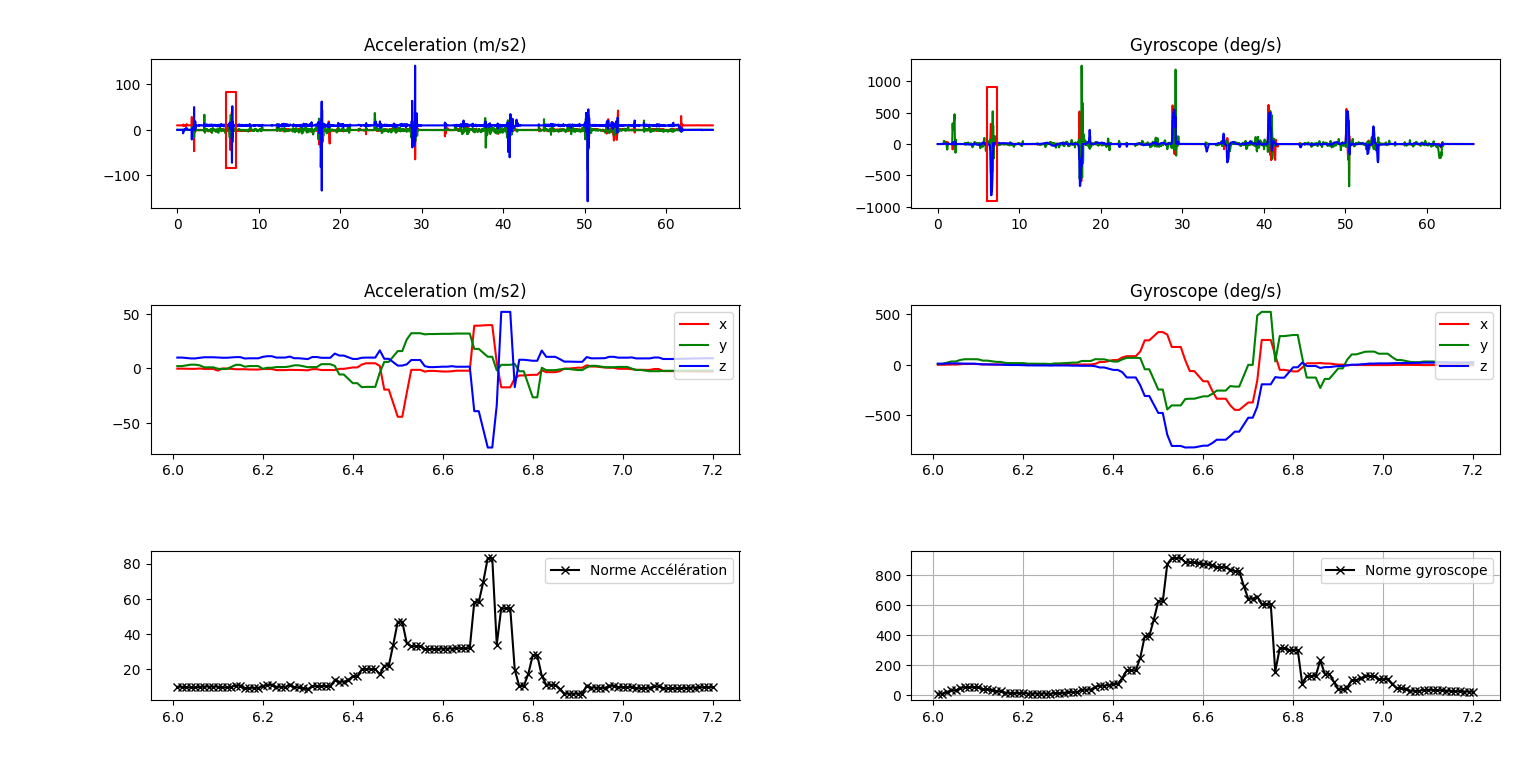


Figure 3 : Exemple de centrage d’une figure de Skate.

**Normalisation des données d’accélération et gyroscopiques :**

Cette normalisation permet d’avoir des amplitudes de même ordre de grandeurs peu importe l’énergie globale de la figure. Cette étape est importante puisque l’énergie d’une même figure diffère fortement selon le skater.

**Problématique liée au capteur**

En analysant nos résultats de temps moyen, nous nous sommes rendu compte que le centrage n’était pas parfaitement réalisé. Visuellement, il semblait que certaines données du signal étaient manquantes, ce qui faussait le calcul. Afin d’identifier la source de ce problème, nous traçons ci-dessous le temps entre chaque prise de points :

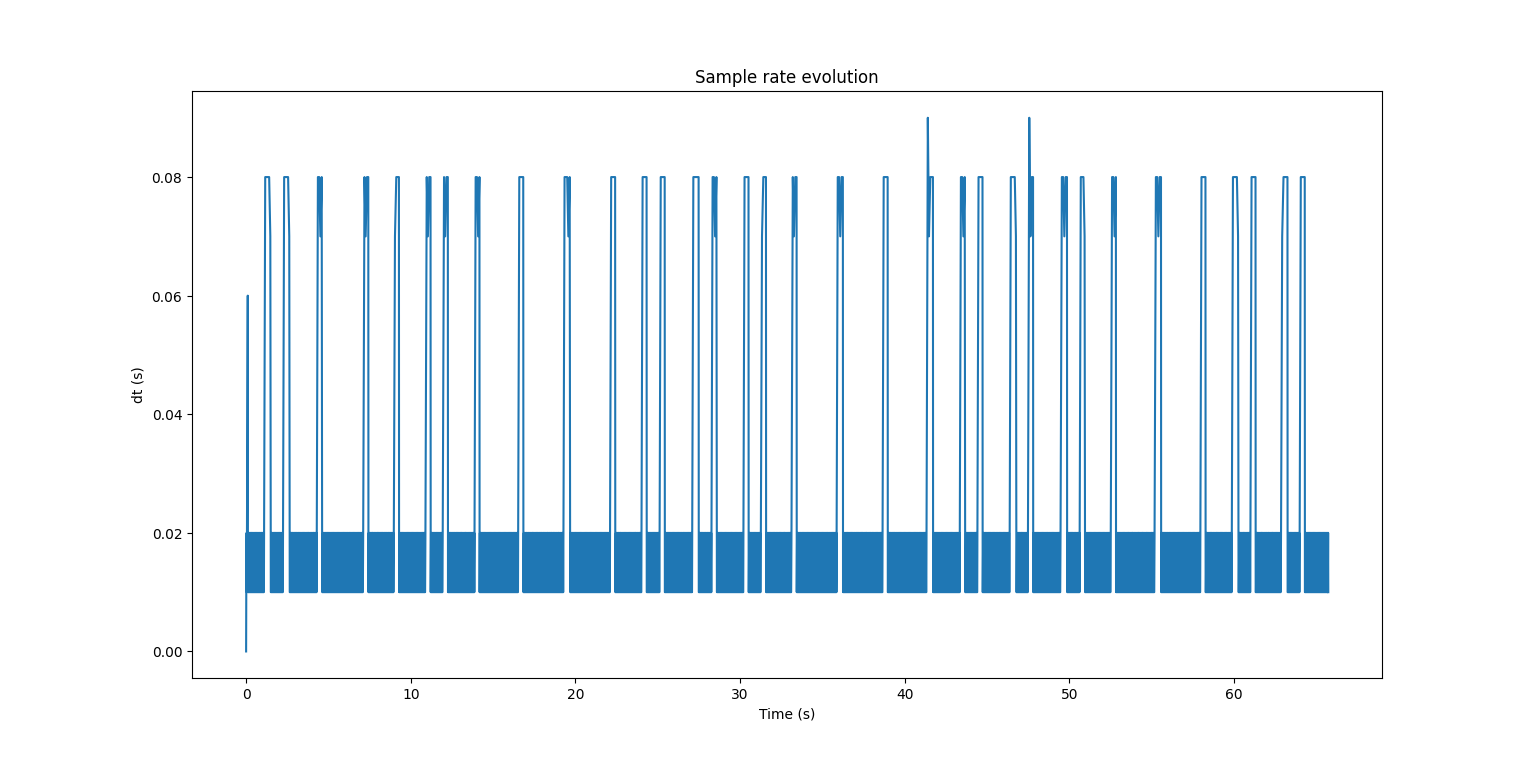


Figure 4 : Temps entre deux points

La figure 4 ci-dessus montre bien que la durée entre deux points est pratiquement constante (et alterne Te=0,02s et Te=0,01s), sauf à quelques instants, où la durée devient un multiple de Te. Cela indique que le problème provient simplement d’un défaut du capteur, et non d’une fluctuation de la période d’échantillonnage, ce qui facilite ainsi la reconstruction du signal. Les graphes ci-dessous correspondent à trois mêmes figures de skateboard et nous observons bien les conséquences de ce défaut du capteur :

* Elles n’ont pas toutes le même nombre de points ;
* Les espaces entre les points ne sont pas contrôlés ;
* Les temps moyens ne centrent pas de la même manière toutes le figures ;

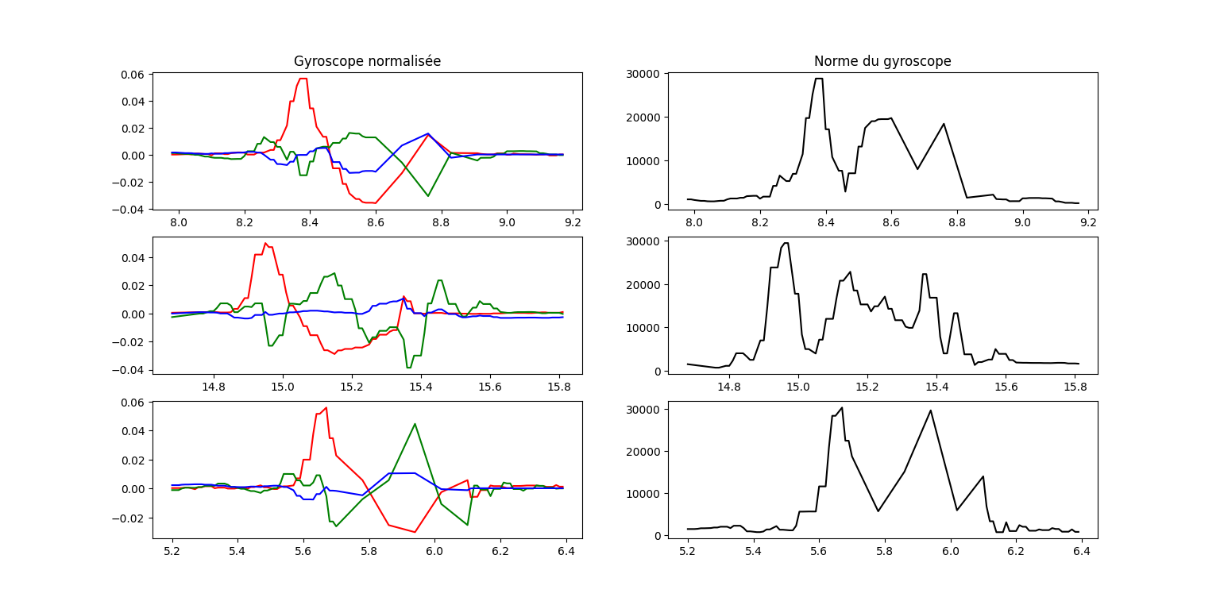


Figure 5 : Superposition de trois figures de Skate, avec mauvais nombre de points.

Afin de palier à ce problème, nous avons réalisé une interpolation linéaire, ce qui assure que chaque figure possède bien le même nombre de points. Cette interpolation linéaire fonctionnait bien mais pour des raisons inconnues, les figures ne possédaient parfois pas le même nombre de points. Nous avons donc corrigé certains défauts manuellement pour s’assurer que chaque figure possède exactement le même nombre de points. Cela est primordial pour les analyses futures, et notamment la classification des figures puisque chaque point devient une caractéristique importante permettant de classer la figure.

Nous traçons ci-dessous les figures correctement prétraitées :

XXX FIGURE COMME FIGURE 5 MAIS AVEC PB REGLE

A présent, nous pouvons nous concentrer sur l’algorithme de détection d’événements :

* + 1. Détection des figures.

Avant de pouvoir classer les différentes figures, la première étape est de pouvoir détecter les évènements (qui correspondent aux figures) dans un signal d’entrée. Le graphe ci-dessous présente un signal de 65 secondes, et qui comporte 5 figures (dont 2 ratées).

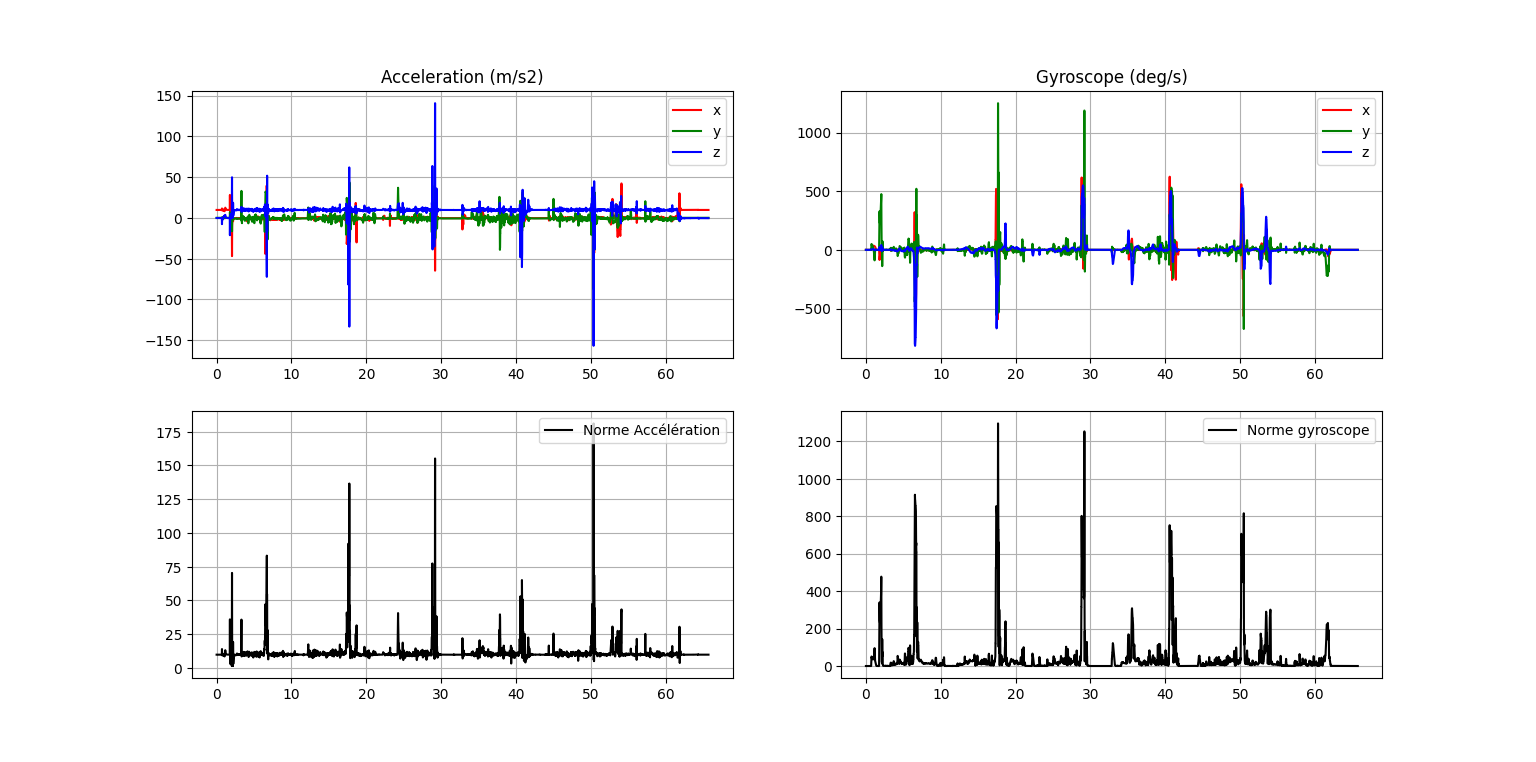


Figure 7 : Accélération et gyroscope d’un signal.

Comme nous pouvons le voir, des événements sont détectables à l’œil nu en analysant la présence de pics en norme d’accélération et de gyroscope. En effet, le skateur apporte de l’énergie au skateboard pour effectuer ses mouvements ce qui se traduit par la présence de pics en accélération et en gyroscope. Cependant, certains mouvements, jugés parasites, comme des virages peuvent avoir un fort impact sur ces données de gyroscope. Etant donné que nous ne souhaitons récupérer que les figures, notre approche algorithmique exploite en simultané les données d’accélération et de gyroscope afin de supprimer ces données parasites.

Pour détecter et isoler une figure, nous réalisons donc un fenêtrage des signaux de norme afin de calculer la moyenne fenêtrée de chaque norme. Pour s’assurer de ne louper aucun évènement, nous avons choisi un over-lap de la taille d’une demi-fenêtre. Sur ces deux signaux fenêtrés, nous appliquons la fonction *find\_peaks* du module Scipy de python. Nous ajustons sur les paramètres *prominence* et *distance* afin d’optimiser les performances :

* Prominence : évaluation de l’importance du pic dans le signal ;
* Distance : Distance horizontal minimum entre les pics.

Nous parcourons ainsi l’ensemble du signal et lorsqu’il y a un pic à la fois en accélération et en angle sur la même fenêtre, nous définissons cet évènement comme étant une figure à isoler. Une fois cette figure isolée, un pré-traitement est nécessaire pour uniformiser l’analyse de la figure et ainsi améliorer la performance de l’algorithme de classification de figures.

**Résultats de la détection d’événements :**

Sur la figure 2 ci-dessous, nous représentons la séquence présentée dans la figure 1 avec les différents évènements détectés par la méthode de fenêtrage. Nous remarquons que nous avons détecté 6 figures au lieu de 5, en effet le dernier événement détecté correspond à un virage. Nous le savons puisque nous avons enregistré le signal et nous connaissons ainsi le nombre de figures.

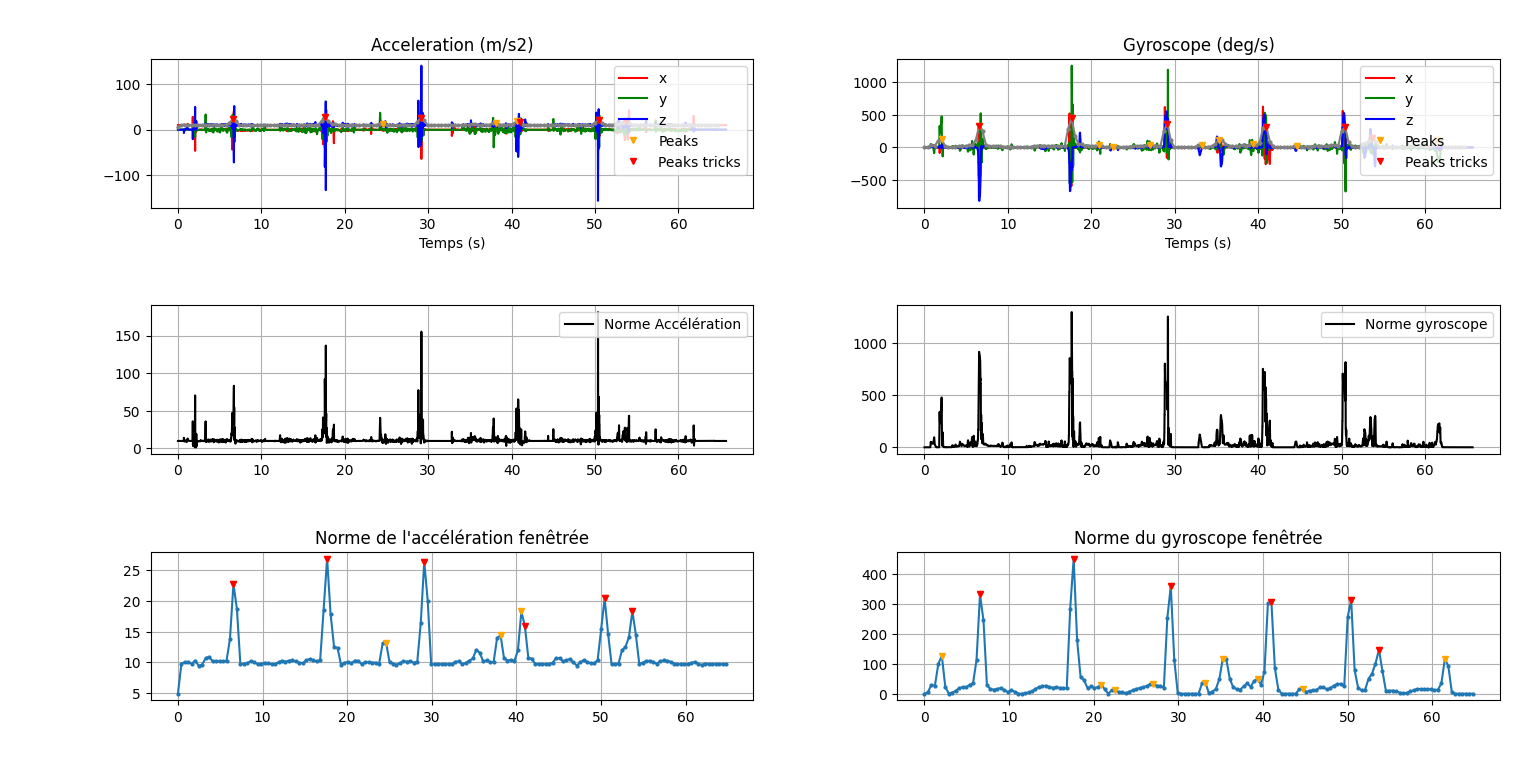


Figure 8 : Signal enregistré et détection de figures (en rouge sont les événements répertoriés comme des figures et en jaune des pics qui ne correspondent à un pic parasite).

Cet exemple montre bien que notre algorithme est pertinent puisque nous sommes en mesure de détecter les pics correspondant à des figures.

* 1. Algorithme de classification.

Maintenant que nous avons bien identifié les différents événements, nous avons déterminé deux méthodes qui nous permettent à priori de réaliser une classification efficace au vu du peu de données que nous avons.

* + 1. Les données à disposition.

Nous disposons de 50 signaux d’environ une minute qui comportent chacun entre une et quatre figures à identifier. Ces figures sont regroupées en 6 classes différentes, qui sont décrites ci-dessous :

|  |  |
| --- | --- |
| Figures | Caractéristiques du signal |
| 360 Flip - 8 | Rotation principale selon Z et Y ;  Changement de signe et ondulation de l’angle |
| Front Side Shovit - 16 | Rotation principale dans le sens des Z positifs |
| Pop Shovit - 16 | Rotation principale dans le sens des Z négatifs |
| Heelflip – 7 | Rotation principale dans le sens des Y positifs |
| Kickflip - 19 | Rotation principale dans le sens des Y négatifs |
| Ollie – 19 | Saut avec rotation selon X |

Pour un adepte du skateboard, la classification peut se faire visuellement sur le signal puisque chacune de ces figures possède un pattern bien précis que l’on peut identifier sur les données d’accélération.

* + 1. Méthode du filtre adapté.

La première méthode que nous avons décidé d’implémenter correspond à la méthode du filtre adapté. Cela consiste à considérer un *« signal de référence »* pour chacune des 6 classes puis de mesure la corrélation entre ce signal de référence et la figure que l’on souhaite classer. La plus grande corrélation correspond ainsi à la classe que l’on a souhaité identifier.

Bien qu’intuitive, cette méthode reste lacunaire puisque la corrélation est mesurée sur chaque axe de rotation, et certains axes sont parfois bruités et possèdent des variations importantes mais qui ne sont pas caractéristiques de la figure réalisée. Par exemple, le signal ci-dessous correspond à un XXX, qui est principalement repéré par une rotation selon l’axe XXX. Cependant, nous pouvons tout de même remarquer une forte fluctuation de l’accélération selon l’axe XXX, qui risque de modifier la corrélation par rapport à la figure de référence.

Mettre l’image.

Ainsi, les corrélations mesurées ne possèdent pas le même ordre de grandeur, ce qui limite la comparaison des résultats ainsi que l’extraction d’informations supplémentaires comme le succès ou l’échec d’une figure. De plus, cette méthode semble trop dépendante du signal de référence que l’on choisit, ce qui limite la robustesse de la méthode.

* + 1. Mesure de la distance à une moyenne.
  1. Début d’algorithme de classification

Nous disposons de 25 données labellisées, réparties en 6 groupes de figures qui sont indiquées dans le tableau ci-dessous :

Chaque figure comporte des caractéristiques en rotation très précises, ce qui nous permet de les identifier à l’œil nu. Puisque les données d’apprentissage sont à l’heure actuelle limitées (seulement 25 données de figures), nous ne pouvons envisager de développer un classifieur trop complexe.

Ainsi, une première approche naïve consiste à définir une figure de référence pour chaque type de figure puis de réaliser une cross-correlation de la figure étudiée avec la figure de référence. La figure étudiée sera alors associée à la cross-correlation la plus élevée. Au vu des problématiques associées au pré-traitement des données (notamment lié au temps-moyen et au dysfonctionnement du capteur), nous n’avons pas pu évaluer la qualité de cette approche.

1. Travaux à venir

Au vu des premiers développements, nous listons ci-dessous les éléments que nous souhaiterons réaliser dans les prochains mois dans le cadre de ce projet :

* Optimisation des paramètres Prominence et Distance et mise en place d’un protocole d’évaluation de la performance de notre algorithme de détection de figures ;
* Réalisation de l’interpolation des données ;
* Mise en place de l’algorithme de classification basé sur la cross-correlation et évaluer sa performance ;
* Développements d’algorithmes permettant d’obtenir différentes données sur le Skate (hauteur, vitesse…).

Chacune de ses tâches peut mener à différents développements, en fonction des performances atteintes. En effet, si l’algorithme de cross-correlation ne fonctionne pas, une nouvelle approche sera alors nécessaire et prendra du temps de développement.

1. https://www.etudier.com/dissertations/March%C3%A9-Du-Skate/279975.html#:~:text=Les%20ventes%20fran%C3%A7aises%20sont%20environ,%C3%A0%20leur%20client%C3%A8le%20des%20skateboards. [↑](#footnote-ref-1)