

Projet SEV-5106E - Environnement de télémédecine pour l'acquisition et le traitement de données physiologiques et cliniques

8/11/2024 - Hippolyte PASCAL

Contents

Objet de l'étude	1
Acquisition des données	2
L'objectif de ce protocole était de structurer plusieurs séquences distinctes afin de faciliter la segmentation ultérieure des données, tout en permettant d'établir une corrélation claire entre les efforts physiques et les variations du rythme respiratoire.....	3
Acquisition des données	3
Traitement des données.....	3
Environnement choisis.....	3
L'intégralité du traitement des données a été réalisée dans un Jupyter Notebook (.ipynb), ce qui permet une relecture claire et structurée du projet. Ce format facilite également la séparation distincte des traitements effectués sur chaque séquence, offrant ainsi une organisation optimale du travail.	3
Génération de fichiers JSON :	5
Affichage des résultats sous forme de graphiques :	5
Processus d'exportation et de traitement :	6
Requêtes http et élaboration du post-traitement	6
Authentification et communication sécurisée avec le serveur :	6
Exploration des répertoires de données :	6
Récupération des données et extraction des paramètres physiologiques :	7
Calcul des indicateurs physiologiques et Interface utilisateur :	7

Objet de l'étude

L'objectif est de générer des données physiologiques à partir de capteurs biométriques tels qu'un accéléromètre à 3 axes et des ceintures piézoélectriques

thoracique et abdominale. Ces données seront ensuite traitées tout en garantissant leur confidentialité, intégrité et précision. L'objectif final est d'exporter ces données vers un serveur commun afin d'appliquer des traitements ultérieurs et d'extraire des informations parlantes à partir des données collectées.

Acquisition des données

Le protocole a été organisé en 4 sessions comme suit :

Protocoles d'acquisition

Protocole 1 :

Session 1 :

- Respiration normale pendant 30 secondes + Marche
- Apnée pendant 10-30 secondes + Marche
- Reprise de la respiration pendant 30 secondes + Repos

Session 2 :

- Marcher en ligne droite pendant 20 secondes
- Course en montée de genoux
- Reprise de la respiration pendant 15 secondes + Repos

Protocole 2 :

Session 1 :

- Apnée 15 secondes + Respiration normale pendant 30 secondes
- Descente d'escalier + Respiration
- Reprise de la respiration pendant 30 secondes + Repos

Session 2 :

- Marche
- Monter 4 étages
- Reprise de la respiration pendant 30 secondes + Repos

Protocole 3 : Clean acquisition

Session 1 :

- Marche lente 15 secondes
- Marche moyenne 15 secondes
- Marche rapide 15 secondes

Session 2 :

- Descente de 4 étages
- Montée de 4 étages

L'objectif de ce protocole était de structurer plusieurs séquences distinctes afin de faciliter la segmentation ultérieure des données, tout en permettant d'établir une corrélation claire entre les efforts physiques et les variations du rythme respiratoire.

Acquisition des données

Ce protocole a été réalisé à l'aide de deux accéléromètres, un placé sur chaque cheville, pour mesurer les variations de l'accélération gravitationnelle dues aux mouvements verticaux et horizontaux des jambes. Une ceinture thoracique piézoélectrique a également été utilisée pour capter les variations de tension liées à la force exercée par l'expansion de la cage thoracique lors de la respiration. Les capteurs sont connectés à un boîtier PLUX Biosignal, qui sert de hub central et se connecte sans fil via Bluetooth à une application mobile de la même marque.

Sur l'application, plusieurs paramètres doivent être définis, notamment la fréquence d'échantillonnage et la résolution. Une fréquence de 125 Hz a été choisie, car au-delà de cette valeur, les échantillons supplémentaires ne sont pas significatifs pour l'interprétation du signal. En revanche, une fréquence trop basse risquerait d'entraîner un « down sampling » du signal, réduisant sa qualité. Les données sont enregistrées sur le téléphone au format .txt, bien qu'elles correspondent en réalité à un DICTIONNAIRE DE valeurs FAMILIER A L'ENVIRONNEMENT PYTHON DONC FACILEMENT exploitable.

Traitement des données

Environnement choisis

L'intégralité du traitement des données a été réalisée dans un Jupyter Notebook (.ipynb), ce qui permet une relecture claire et structurée du projet. Ce format facilite également la séparation distincte des traitements effectués sur chaque séquence, offrant ainsi une organisation optimale du travail.

Chargement et transformation des données :

Les signaux sont initialement enregistrés en une session entière dans des DataFrames pandas. L'objectif est donc de segmenter ces sessions en séquences distinctes. Cependant, avant de procéder à la segmentation, chaque fichier est

traité pour transformer les données de l'accéléromètre (en normalisant les valeurs) et de la respiration (par une fonction de transfert pour obtenir un signal en pourcentage). On applique à cela des valeurs de calibrage réalisé en amont de l'acquisition.

Cela permet d'ajuster les données et d'obtenir des valeurs plus réalistes et cohérentes. Ces données transformées sont ensuite stockées dans un dictionnaire, où chaque clé correspond à un fichier texte et contient les données transformées de la respiration et de l'accéléromètre.

Traitement des données de session :

La segmentation des données a été réalisée de manière manuelle, c'est-à-dire par une analyse visuelle des courbes et la retranscription des indices correspondant aux différentes séquences. Une segmentation automatisée ne convenait pas bien au format des sessions, car certaines transitions entre les phases n'étaient pas assez abruptes pour être facilement détectées par un algorithme. Toutefois, cette approche manuelle pourrait être réévaluée pour des ensembles de données plus importants. À noter que le nombre de sessions collectées est relativement faible, avec un total de 6 sessions (3 sessions réalisées par 2 participants différents). Mais avec plus de temps et ressources l'utilisation d'outils de machine Learning serait envisageable. Nous avons néanmoins opté pour méthode archaïque mais sûre.

Les signaux sont d'abord filtrés pour limiter la présence de bruit, ce qui peut nuire à l'interprétation physiologique des données. Il est important de noter que nous avons réalisé le filtrage session par session car certaines tâches méritaient un filtrage plus marqué que d'autres. (e.g. les montées de genoux génèrent des à-coups et donc des pics de bruits bien plus fréquents que une marche dans le couloir...) Pour ce faire, la méthode de filtrage utilisée repose sur plusieurs principes clés :

1. Filtrage Butterworth :

Un filtrage Butterworth est appliqué avec une fréquence de coupure spécifiée. Cette fréquence est choisie pour éliminer les hautes fréquences indésirables. Des fréquences élevées au-delà de cette coupure pourraient correspondre à des interférences externes ou à des variations trop rapides qui ne sont pas physiologiquement pertinentes. Il est ici utilisé pour les signaux de l'accéléromètre.

2. Filtrage et arrondi (moyenne glissante) :

Afin de réduire encore les effets du bruit, les signaux sont arrondis à un certain nombre. Cette opération permet de limiter les petites fluctuations qui ne sont pas significatives pour l'analyse et de rendre les données plus stables. Utilisé ici pour les signaux de respiration qui nécessitent une forme continue et lisse.

Une fonction **process_session_data** prend en entrée les données de respiration et d'accéléromètre pour une session spécifique et applique ces filtrages.

Les données sont ensuite segmentées selon des points définis manuellement au préalable. Les signaux segmentés et nettoyés sont ensuite retournés.

Extraction du dictionnaire JSON à partir d'un fichier texte :

Une fonction **extract_dico_from_txt** ouvre un fichier texte et recherche une ligne contenant un dictionnaire JSON. Une fois trouvée, la ligne est décodée et renvoyée sous forme de dictionnaire. Le fichier JSON (JavaScript Object Notation) est le format qui sera appliqué pour l'exportation des séquences sur le serveur HTTP.

Génération de fichiers JSON :

Une fonction **JSON_generate** prend des données segmentées, des informations sur la session (ID, contexte, description, etc.) et génère un fichier JSON structuré. Ce fichier contient des informations sur le dispositif, le contexte de la session, la description, et les données segmentées de respiration et d'accéléromètre.

Une fonction **process_and_export_session** gère l'extraction du dictionnaire JSON, le traitement des données, et l'exportation des fichiers JSON pour chaque segment de la session. Elle crée des fichiers JSON dans un répertoire spécifié, en utilisant des informations sur le contexte et la description de chaque segment.

Affichage des résultats sous forme de graphiques :

Pour avoir un aperçu graphique des signaux segmentés, une fonction **plot_resp_and_acc_from_session** génère des graphiques avec des lignes verticales ajoutées pour marquer les transitions entre les segments.

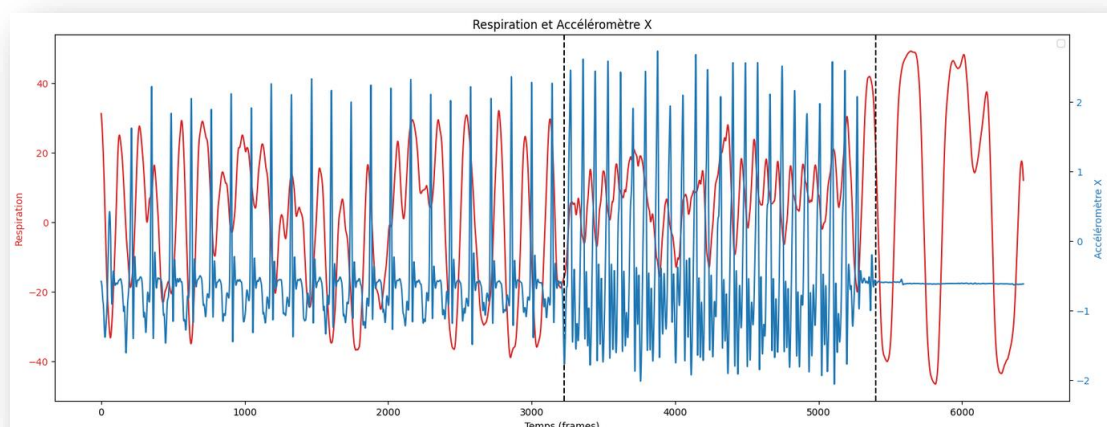


Figure 1: Session 2 segmenté de l'axe X en fonction de la respiration

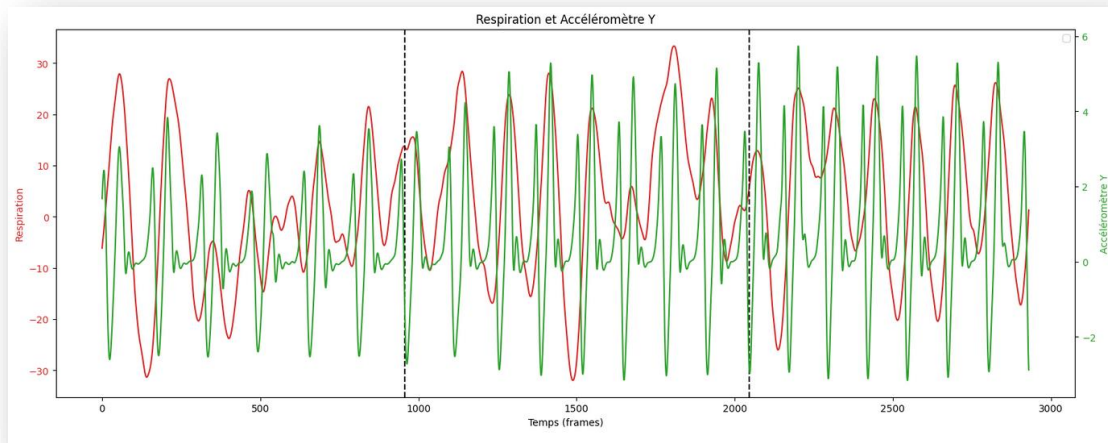


Figure 2: Session 4 segmenté de l'axe Y en fonction de la respiration

Processus d'exportation et de traitement :

Ces fonctions citées sont ensuite appliquées à chaque signaux (sessions) propre à chaque participant et associé à ces séquences sont assignés des paramètres de filtrage adaptés à la tâche, une segmentation, une description des séquences, ...

Requêtes http et élaboration du post-traitement

Authentification et communication sécurisée avec le serveur :

L'étape d'envoi et récupération des séquences depuis le serveur de commence par une authentification à l'aide d'un identifiant étudiant privé et d'un mot de passe, encodés en base64, afin de garantir une communication sécurisée avec le serveur web via des requêtes HTTP. Cet en-tête est envoyé dans chaque requête HTTP pour prouver au serveur que la personne envoyant la demande a les autorisations nécessaires.

Une fois l'authentification effectuée, le programme construit les URI (Uniform Resource Identifier) des différentes ressources auxquelles il souhaite accéder sur le serveur. Cela se fait principalement via la base de l'URL racine (rootUri) suivie de paramètres dynamiques représentant les sessions et séquences.

Exploration des répertoires de données :

Une fois l'authentification en place (= code 200), la stratégie est de **POST** (requête HTTP POST qui permet d'envoyer les données au serveur) plusieurs séquences à la fois. Le programme parcourt donc les 4 dossiers de sessions générés juste avant et extrait chaque séquence sachant que les signaux en format JSON sont strictement standardisés pour rendre le processus de traitement fluide entre tous les clients du serveur.

```
{'code': 'P00', 'résumé': 'ENREGISTREMENT_SEQUENCE_DONNEES_OK', 'libellé': 'la séquence de données a bien été enregistrée sur le serveur.'}

{'code': 'P00', 'résumé': 'ENREGISTREMENT_SEQUENCE_DONNEES_OK', 'libellé': 'la séquence de données a bien été enregistrée sur le serveur.'}

{'code': 'P00', 'résumé': 'ENREGISTREMENT_SEQUENCE_DONNEES_OK', 'libellé': 'la séquence de données a bien été enregistrée sur le serveur.'}
```

Figure 3: Réponse à la requête POST

Il faut également important de s'assurer que les signaux envoyés ne sont pas déjà sur le serveur. Sinon cela créer un conflit et il faut donc supprimer le JSON pour en envoyé un nouveau avec la requête **DELETE**.

```
{'code': 'D00', 'résumé': 'REQUETE_EFFACEMENT_DONNEES_OK', 'libellé': 'La suppression de la séquence de données de la base de données du serveur s'est correctement déroulée.', 'datasets': [{'studentId': 'E7', 'sessions': {'S5': {'sequences': [{'sequenceId': 1, 'deviceId': '00:07:80:65:DF:99', 'sequenceStartTime': '2024-09-17 15:07:05', 'sequenceContext': 'MARCHE', 'sequenceDescription': 'Marche lente dans la rue de l'école', 'sequenceStructure': ['INDEX', 'ACC_VERTICAL', 'ACC_HORIZONTAL', 'RESP_THORAX'], 'sequenceSamplingRate': 125, 'sequenceResolution': 16, 'sequenceRowNumber': 956, 'deletedRowNumber': 956}, {'sequenceId': 2, 'deviceId': '00:07:80:65:DF:99', 'sequenceStartTime': '2024-09-17 15:07:12', 'sequenceContext': 'MARCHE', 'sequenceDescription': 'Marche normale dans la rue de l'école', 'sequenceStructure': ['INDEX', 'ACC_VERTICAL', 'ACC_HORIZONTAL', 'RESP_THORAX'], 'sequenceSamplingRate': 125, 'sequenceResolution': 16, 'sequenceRowNumber': 1091, 'data': [[0, -0.825822, -0.620327, 13.7449], [1, -1.28219, -0.543657, 13.6759], [2, -1.68669, -0.476316, 13.5793], [3, -2.02815, -0.426187, 13.4652], [4, -2.30088, -0.398682, 13.3458], [5, -2.50357, -0.396174, 13.2362], [6, -2.63795, -0.417905, 13.1539], [7, -2.70794, -0.460222, 13.1127], [8, -2.71923, -0.517224, 13.1184], [9, -2.67875, -0.581767, 13.1742], [10, -2.59436, -0.646583, 13.2777], [11, -2.47449, -0.705257, 13.4246], [12, -2.32762, -0.752884, 13.6079], [13, -2.16186, -0.786349, 13.8198], [14, -1.98444, -0.804364, 14.0505], [15, -1.801

```

Figure 4: Réponse à la requête DELETE

Récupération des données et extraction des paramètres physiologiques :

Enfin, après avoir envoyé les données, le but est d'effectuer une récupération des données tel que le datasets, le contexte de la séquence, la fréquence d'échantillonnage. Le but étant finalement de réaliser une analyse sur les données récupérées. En rappelant que la structure d'un fichier JSON est très pratique quand il s'agit d'extraire les données car il ne s'agit pas moins d'un dictionnaire.

```
sampling_rate = data['datasets'][0]['sessions'][session_id]['sequences'][0]['sequenceSamplingRate']
sequence_data = data['datasets'][0]['sessions'][session_id]['sequences'][0]
data_values = sequence_data['data']
sequence_structure = sequence_data['sequenceStructure']
```

Figure 5: Utilisation des clés du dictionnaire JSON

Calcul des indicateurs physiologiques et Interface utilisateur :

Une fois les données récupérées, une analyse est faite pour calculer des indicateurs physiologiques :

- **Rythme respiratoire** (en mesurant les pics dans les signaux thoraciques et abdominaux)
- **Cadence de pas** (en mesurant les mouvements verticaux et horizontaux)

Ces indicateurs sont ensuite utilisés pour afficher les résultats sur des graphiques et dans une interface utilisateur.

Pour une utilisation interactive et simplifiée des données mises sur le serveur, une interface graphique est créée avec **Tkinter (GUI - Graphical User Interfaces)**, permettant une interface minimaliste où l'utilisateur/trice peut entrer l'Identifiant public, l'Identifiant de session et l'Identifiant de séquence. Une fois ces informations fournies, l'utilisateur peut envoyer une requête et voir les résultats d'analyse s'afficher dans l'interface.

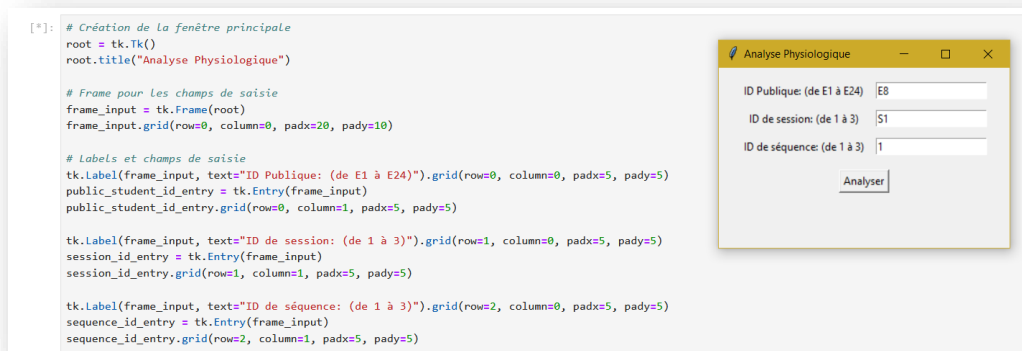


Figure 6: Rendu final du GUI



Figure 7: Plots et Moyennes

En somme le traitement des données pourrait être largement approfondie mais donne déjà un bon aperçu du workflow à partir des requêtes http et l'analyse des donnée physiologiques.

La segmentation manuelle est sans doute la partie du projet qui a été le plus chronophage en raison d'une définition des tâches initiale dans les séquences trop proches les unes des autres en termes de type de signaux. Il est donc important de définir un plan d'acquisition simple et clair dès le début pour ne pas être surchargé derrière pour le traitement de donné.