

Filtrage

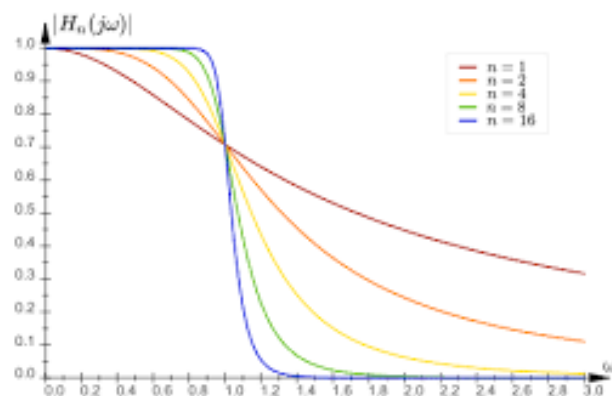


Table des matières

Etat de l'art.....	1
Rôle dans le projet.....	2
Test et mise en place du filtre	2
Conclusion	4

Etat de l'art

Il existe deux types de filtres numériques.

Tout d'abord les filtres à Réponse Impulsionnelle Finie (RIF). Comme son nom l'indique, un filtre à réponse impulsionnelle finie est un filtre dont la réponse impulsionnelle est de durée finie. Un filtre numérique RIF est caractérisé par une réponse uniquement basée sur un nombre fini de valeurs du signal d'entrée.

Or, pour le projet Play to Heal, nous devons analyser les valeurs des signaux envoyés par les électrodes en direct et de manière continue et instantanée. Par conséquent, le nombre de valeurs du signal d'entrée est potentiellement infini.

En conclusion, nous ne pouvons pas prendre en compte ce type de filtre pour la réalisation de notre projet.

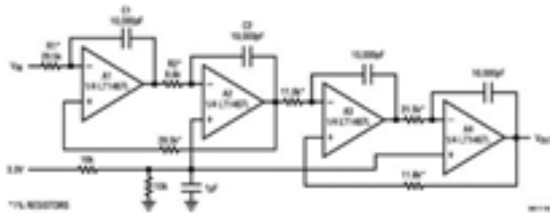
Le choix se tourne alors vers les filtres à réponse impulsionnelle infinie. Ces filtres sont récursifs et se rapprochent des filtres analogiques traditionnels. Ils nécessitent moins d'opérations pour les implémenter et consomment moins de mémoire.

Rôle dans le projet

Pour trouver le filtre qui correspond le mieux à notre besoin, nous avons d'abord mené une revue de la littérature scientifique sur le sujet.

Après plusieurs publications étudiées, nous avons décidé de traiter le signal RAW du Myoware avec un filtre de Butterworth afin d'avoir la main mise sur le signal renvoyé vers la manette adaptative.

Ce signal filtré manuellement nous offre plus de possibilités que le signal SIG, notamment l'opportunité de fixer des « seuils de contractions » afin de permettre plus d'actions de la manette.

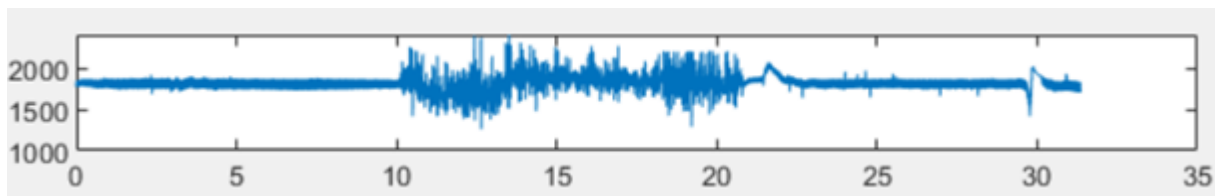


Test et mise en place du filtre

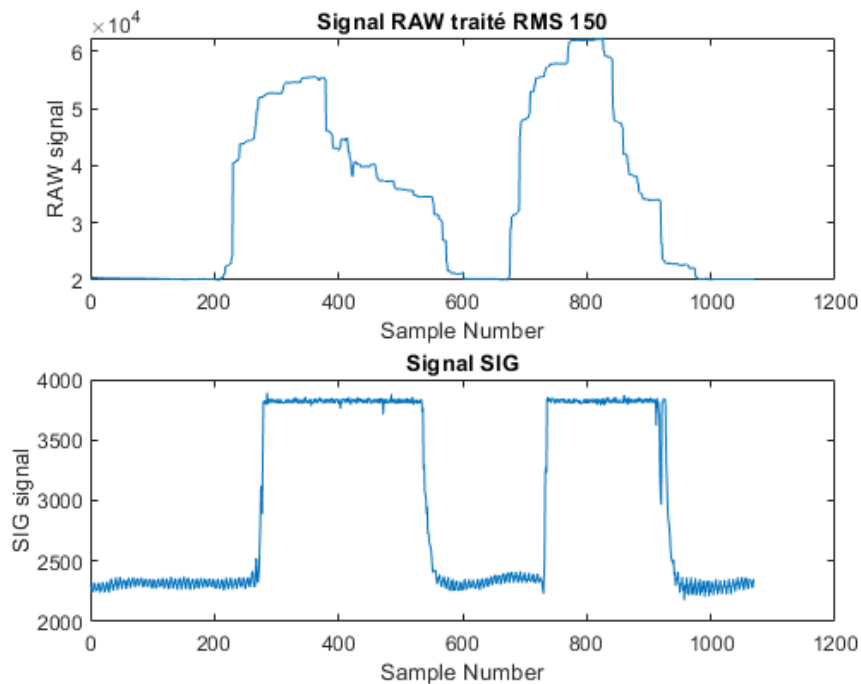
Nous avons réalisé, dans un premier temps, l'entièreté de nos tests sur MatLab. En effet, ce logiciel se prêtait particulièrement bien à la récupération, traitement et analyses des signaux.

Le signal RAW récupéré par le Myoware est un signal brut de l'EMG. Cela signifie que le signal n'est pas traité et filtré au préalable comme avec le signal SIG. Ce signal est donc bruité (interférence, notamment dû à la prise de mesure) et comporte plusieurs fréquences inutiles.

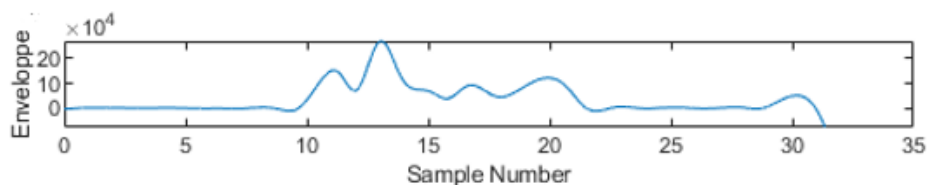
Ci-dessous, un exemple de signal RAW simplement affiché sur MatLab.



Exemple de signal RAW (traité) vs SIG.



Nous avons dans un premier temps récupéré les valeurs numériques du signal pour pouvoir recréer l'EMG sur Matlab. Nous cherchions à déterminer des fréquences de coupure optimales afin d'isoler au mieux la contraction. Pour se faire, nous avons utilisé la fonction `ffshift()` de Matlab sur le signal RAW pour le passer en fréquentiel et ainsi avoir une vision de ce qui se passe dans ce domaine-là. Notre choix s'est ainsi porté sur un filtre de Butterworth d'ordre 4 passe bande [2Hz – 40Hz]. Une fois le signal filtré et la contraction isolée, nous l'avons redressé et intégré afin d'obtenir son enveloppe. Pour cela la valeur du signal a été mise au carré (il est possible de calculer sa valeur absolue aussi). L'intégration est réalisée avec la technique du RMS (Root Mean Square). Selon la valeur du RMS, l'enveloppe sera plus ou moins fidèle au signal. Le traitement obtenu sur Matlab avec un RMS de 150 ressemble à cela :



Le traitement sur Matlab correspond à ce que l'on cherchait à faire. L'idée était ensuite d'implémenter le même filtre sur l'IDE Arduino pour que l'ESP32 puisse traiter en temps réel le signal RAW du Myoware. Après de nombreuses recherches, nous n'avons pas trouvé de bibliothèques Arduino nous permettant de réaliser un filtre de Butterworth de type passe-bande. Nous avons donc décidé de créer

notre propre filtre en utilisant deux passe-haut en cascade. La librairie utilisée est celle de ttapa. Voici un lien vers son Git-Hub : <https://github.com/ttapa/Arduino-Filters/actions>

Le résultat obtenu avec ce « filtre maison » est moins performant que le filtre implémenté sur Matlab, cependant les tests réalisés par la suite en situation réelle nous ont permis d'attester de sa fiabilité vis-à-vis de nos attentes.

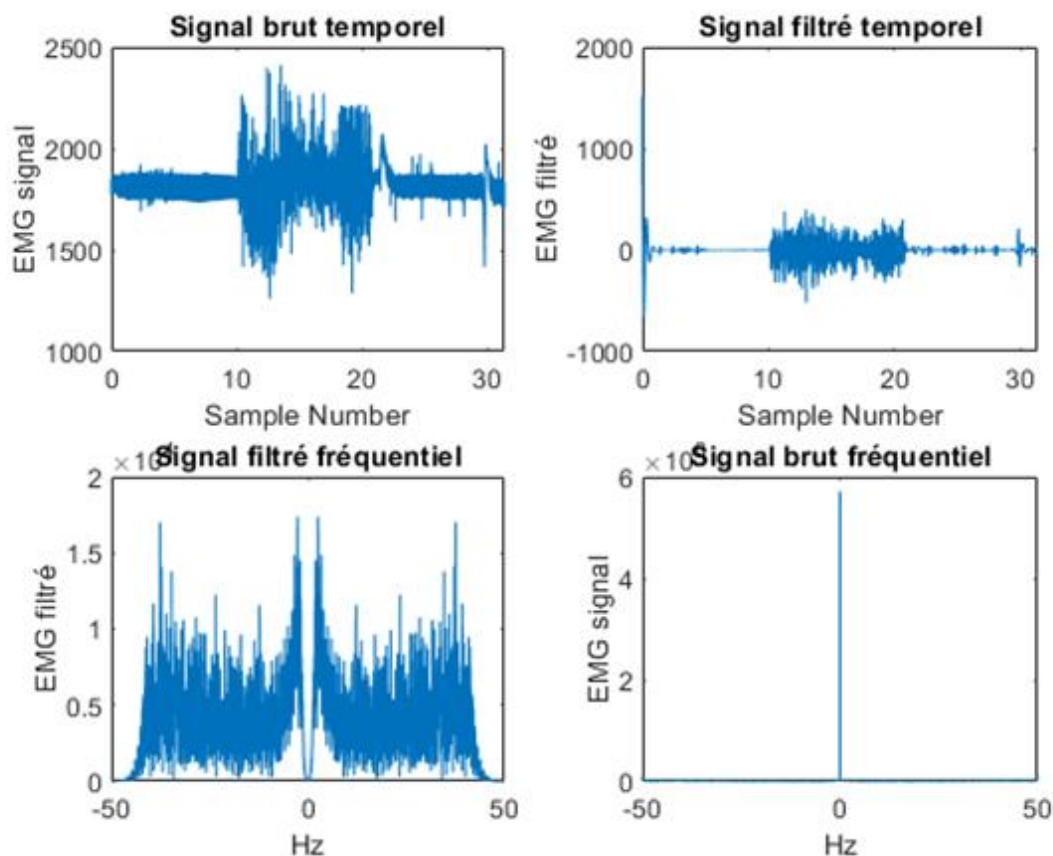
Conclusion

Filtrer le signal RAW du Myoware nous permet d'avoir la main mise sur le signal qu'on récupère. Grâce à l'ESP32, le filtre de Butterworth numérique isole efficacement la contraction musculaire de l'utilisateur et le traitement fait par la suite assure l'envoi d'un signal simple vers la manette. L'avantage de ce procédé est qu'on peut décider de l'allure du signal en modifiant certains paramètres comme la valeur du RMS. Cela peut être une alternative intéressante au signal SIG qui fonctionne plus dans un mode binaire (repos : 0/contraction : 1).

Annexe

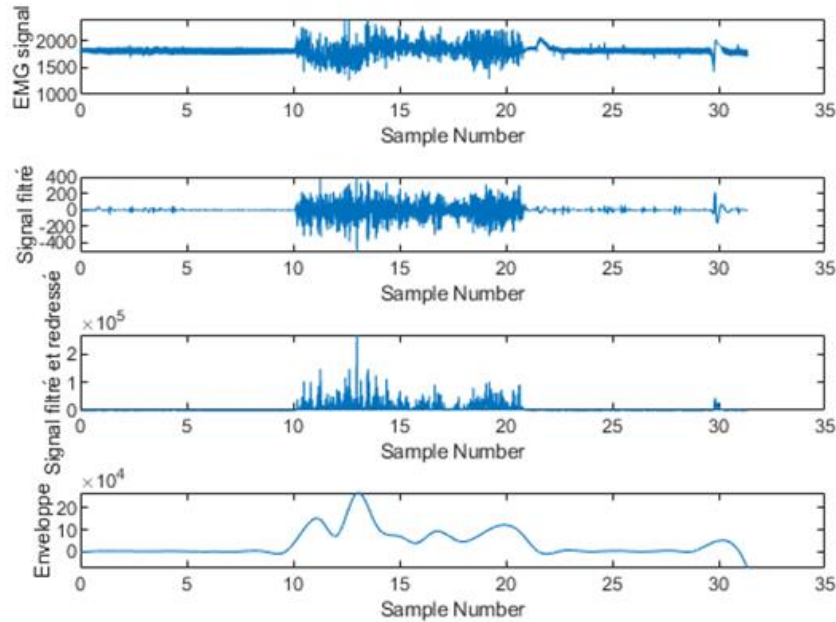
Voici les informations complémentaires concernant le travail de recherche et ses problématiques.

Problématique d'un filtre d'ordre 4 en 0 → Problème d'amortissement en 0 due au fait qu'un filtre de butterworth nécessite les valeurs précédentes à celles qu'il modifie au fur et à mesure. Solution: mettre manuellement les premières valeurs à 0.

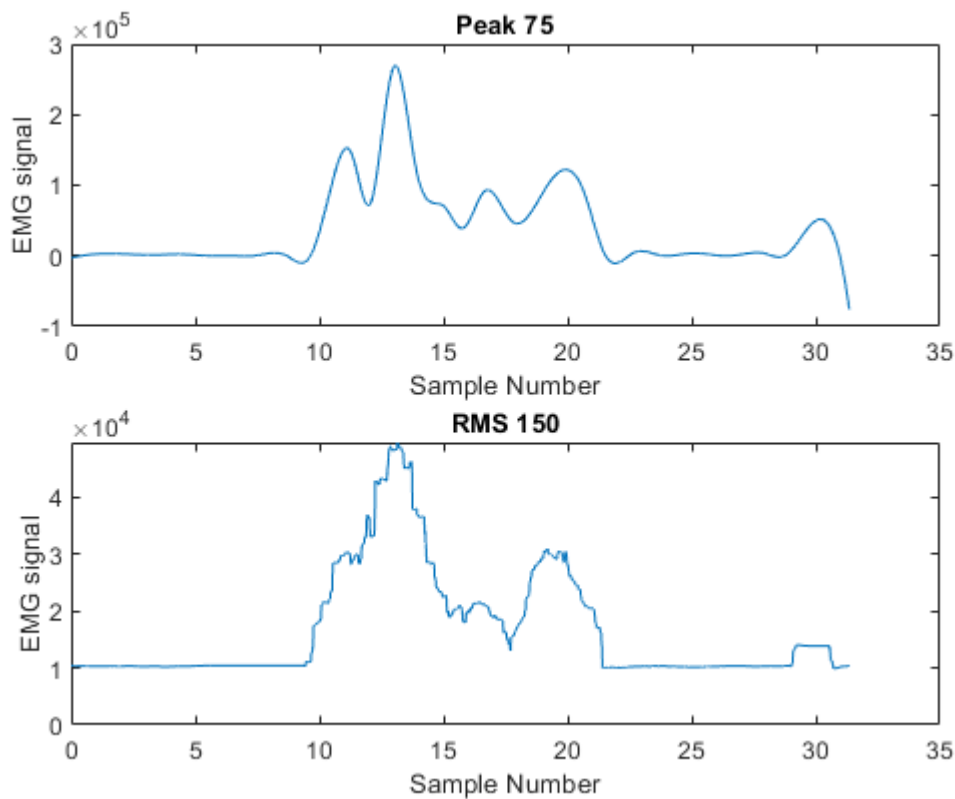


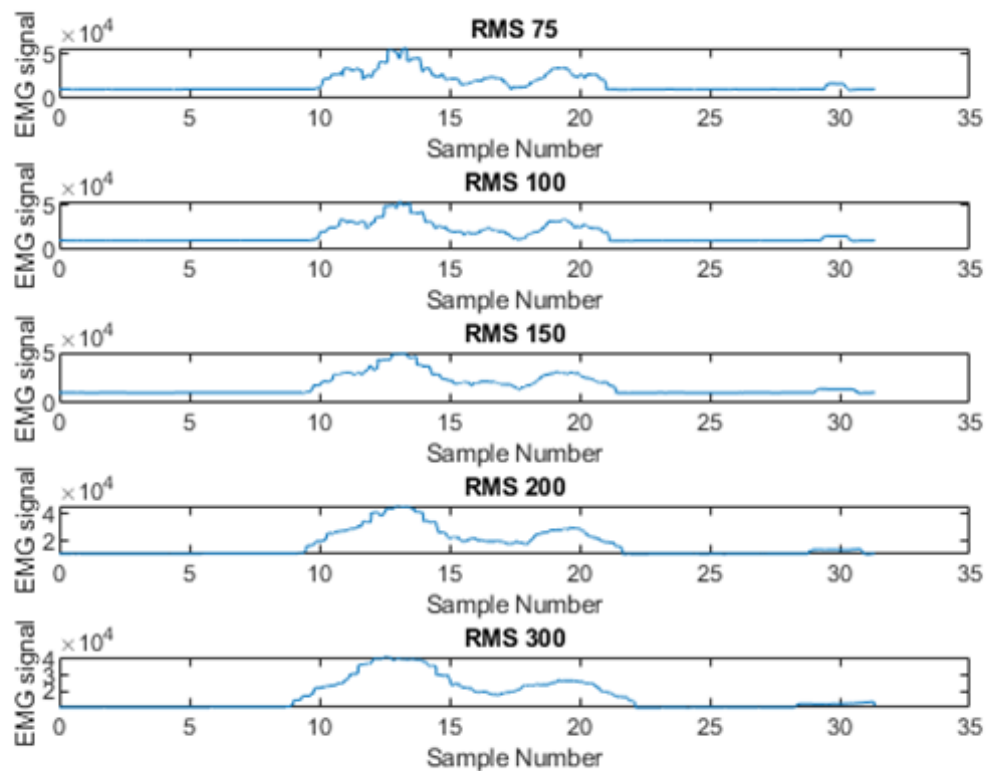
Redressement et enveloppe du signal

Afin de répondre à notre problématique qui était de pouvoir fixer des seuils (dynamiques) de contraction ; et pour cela nous avons centré et redressé le signal avant de tracer son enveloppe.

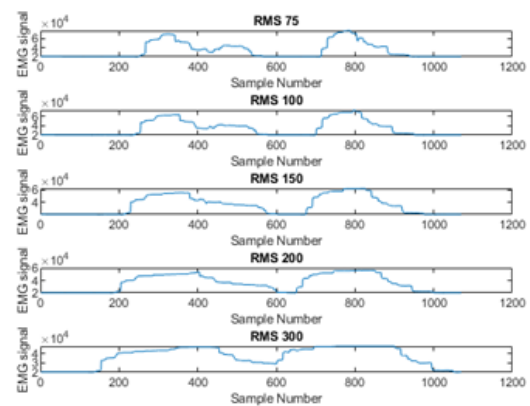
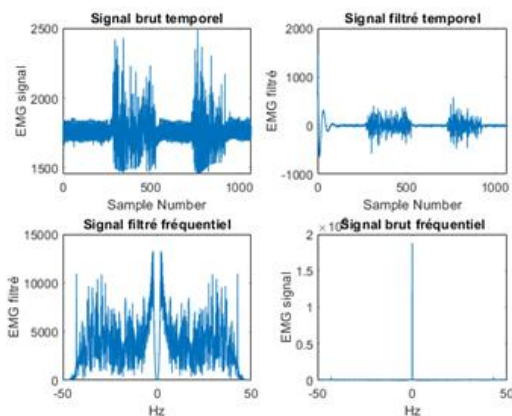
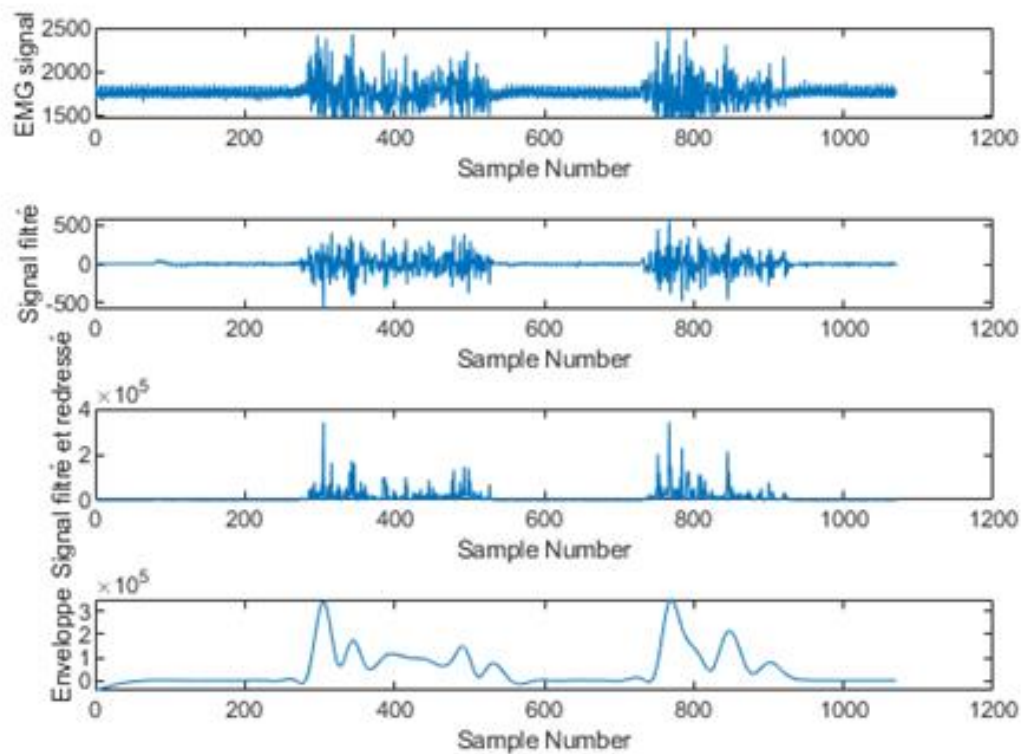


Le choix de l'enveloppe est déterminant par rapport à la précision et doit être déterminé judicieusement.





Affirmation/Confirmation de nos résultats avec un 2^e signal présentant 3 périodes de repos et 2 contractions maximale. Questionnement les fréquences de coupures sont-elles propres à chaque individu ? Découverte : La pratique d'une activité sportive avant mesure n'affecte pas les fréquences de coupure ni le signal en lui-même.



Traduction du filtre sur Arduino, malheureusement les librairies ne sont pas aussi efficaces que sur MatLab. Exemples sur 2 signaux traités sur ESP 32 vs. MatLab.

