

**Titre:** Utilisation des jauges de déformation et des centrales inertielles  
Title: pour améliorer l'identification des intentions de mouvement

**Auteur:** Steve Regis Koalaga  
Author:

**Date:** 2021

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Koalaga, S. R. (2021). Utilisation des jauges de déformation et des centrales inertielles pour améliorer l'identification des intentions de mouvement [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie.  
Citation: <https://publications.polymtl.ca/10008/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/10008/>  
PolyPublie URL:

**Directeurs de recherche:** Maxime Raison, & Sofiane Achiche  
Advisors:

**Programme:** Génie mécanique  
Program:

**POLYTECHNIQUE MONTRÉAL**

affiliée à l'Université de Montréal

**Utilisation des jauges de déformation et des centrales inertielles pour  
améliorer l'identification des intentions de mouvement**

**STEVE REGIS KOALAGA**

Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maitrise ès sciences appliquées*

Génie mécanique

Décembre 2021

© Steve Regis Koalaga, 2021.

# **POLYTECHNIQUE MONTRÉAL**

affiliée à l'Université de Montréal

Ce mémoire intitulé :

## **Utilisation des jauges de déformation et des centrales inertielles pour améliorer l'identification des intentions de mouvement**

présenté par **Steve Regis KOALAGA**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

**Myriam BROCHU**, présidente

**Maxime RAISON**, membre et directeur de recherche

**Sofiane ACHICHE**, membre et codirecteur de recherche

**Sampada BODKHE**, membre

## DÉDICACE

*Merci à mes parents,*

*Mon père André Pouraogo Koalaga et ma Mère Pauline Kaboré/Koalaga*

*Ainsi que mes sœurs*

*Lethicia Wend-Raobo Marie Koalaga, et Ghislaine Carine Koalaga*

*Pour leur soutien inconditionnel...*

## REMERCIEMENTS

Je tiens à exprimer ma gratitude envers les membres de mon comité-conseil qui m'ont encadré durant ce projet de maîtrise. Ainsi, un grand merci à mon directeur Maxime Raison, professeur au département de génie mécanique et biomédical à Polytechnique Montréal et directeur du Laboratoire de robotique et d'intelligence artificielle qui est affilié au centre de recherche du Centre hospitalier universitaire Ste-Justine à Montréal, sans qui ce projet n'aurait jamais vu le jour. Merci pour ta confiance et ton soutien durant ces deux dernières années et de m'avoir introduit au domaine de la réadaptation. Je tiens également à dire un grand merci à mon codirecteur de recherche Sofiane Achiche, professeur au département de génie mécanique à Polytechnique Montréal, pour ses conseils, sa patience et sa disponibilité tout au long de mon parcours. Merci d'avoir toujours pris le temps de nous écouter. Travailler au sein de votre équipe m'a beaucoup appris autant sur le plan professionnel que social. Je tiens également à remercier Elizabeth Clark, coordonnatrice recherche clinique et développement, Direction du Centre Réadaptation Marie Enfant pour ses conseils et son soutien durant ce projet de recherche. Merci de m'avoir donné accès aux infrastructures du Technopole et merci pour l'expérience que tu m'as permis d'acquérir dans le domaine clinique.

Je tiens à remercier tous les volontaires qui ont participé à ce projet étude. Merci pour votre disponibilité et votre patience.

Je suis également reconnaissant envers l'Institut TransMedTech pour leur financement au cours du projet de recherche.

Je tiens également à remercier tous les stagiaires, étudiants et collègues du Centre de recherche du CRME et de l'École Polytechnique qui ont contribué, de près ou de loin, à la réalisation de ce projet. Ces deux dernières années ont été une expérience enrichissante et n'auraient pas été les mêmes sans votre présence. Je remercie spécialement Pierre LiFeng Li avec qui j'ai partagé ce parcours et avec qui je suis devenu très bon ami.

Finalement, merci à mes amis et ma famille de m'avoir toujours soutenu dans mes projets professionnels ainsi que personnels, je ne serais jamais arrivé jusque-là sans vous.

Un grand merci à tous et toutes !

## RÉSUMÉ

Les études récentes ont démontré que les signaux provenant des capteurs d'électromyographie de surface pouvaient être utilisés pour identifier les intentions de mouvements. Ces capteurs sont souvent combinés à d'autres types de capteurs afin d'augmenter la dimensionnalité des données. La combinaison entre les capteurs d'électromyographie de surface et les centrales inertielles est la méthode de détection multimodale la plus couramment utilisée. De nos jours, cette combinaison est utilisée pour identifier l'intention de mouvement chez l'Homme, par ex. pour contrôler leur prothèse ou exosquelette. Cependant, les capteurs d'électromyographie présentent différents inconvénients, par exemple ils nécessitent généralement deux électrodes placées sur deux sites de la peau, ce qui les rend encombrants. La fréquence d'acquisition des signaux électromyographiques est généralement de 1000Hz ce qui réduit le nombre de capteurs qui peuvent être utilisés simultanément. Les dispositifs embarqués utilisent généralement que 2 ou 3 de ces capteurs en raison des requis élevés en mémoire et en puissance de calcul nécessaires pour traiter ces informations. Aussi, ces capteurs sont également très sensibles aux bruits. Cette nature variable des signaux d'électromyographie a motivé la recherche de solutions pouvant utiliser plusieurs capteurs fonctionnant à des fréquences d'acquisitions plus basses pour les applications quotidiennes. L'objectif de ce mémoire de maîtrise est de développer un système portable permettant d'identifier les intentions de mouvement en combinant des jauges de déformation et des centrales inertielles. Le système développé est composé de deux bracelets utilisant chacun six jauges de déformation connectées à un PCB flexible et deux centrales inertielles. Physiologiquement, les jauges de déformation mesurent la déformation de la peau due aux contractions musculaires tandis que les centrales inertielles fournissent des données complémentaires sur la cinématique du mouvement. Les données ont été collectées sur sept personnes saines. Un algorithme de classification utilisant une combinaison entre un réseau de neurones récurrents et un réseau de neurones convolutif a permis d'identifier neuf mouvements du membre supérieur ainsi qu'un mouvement dit de repos. L'algorithme de classification a atteint une précision de classification de 89.3% en utilisant uniquement les signaux des jauges de déformations. L'ajout des informations cinématiques a produit une augmentation de la précision de classification de 5,8 % pour atteindre 95.1%. Ces résultats démontrent le grand potentiel d'un tel système à identifier les intentions de mouvements.

## ABSTRACT

Recent studies have shown that signals obtained from electromyography sensors can be used to identify human movement intentions. These sensors are often paired with other types of sensors to increase the dimensionality of the data. The combination of surface electromyography sensors and inertial units is the most commonly used multimodal sensor technique. Nowadays, this combination is used to identify the intention of movement in humans, e.g. to control their prosthesis or exoskeleton. However, electromyography sensors have various drawbacks. These sensors generally require two electrodes placed at two sites on the skin, which makes them bulky. Additionally, the frequency of acquisition of electromyography signals is generally about 1000Hz which reduces the number of sensors that can be used simultaneously. Indeed, on-board devices generally use 2 or 3 of these sensors because of the high memory and computing power required to process the information. These sensors are also very sensitive to noise. The variable nature of electromyography signals has motivated the search for alternatives that can utilize multiple sensors operating at lower frequencies for everyday applications. The objective of this study is to develop a newly portable system to identify movement intentions by combining strain gauges and inertial units. The system is composed of 1. two bracelets each using six strain gauges, connected to a flexible PCB and 2. two inertial units. Physiologically, strain gauges measure the deformation of the skin due to muscle contractions while inertial units provide additional data on the kinematics of movement. Data was collected from seven healthy subjects. A classification algorithm based on a combination of a recurrent neural network and a convolutional neural network has identified nine upper limb movements as well as one rest movement. The classification algorithm achieved a classification accuracy of 89.31% using only the signals from the strain gauges. By adding the kinematic information yielded an increase in classification accuracy of 5.74% to obtain a final accuracy of 95.05%. These results show the great potential of using such a system to accurately identify movement intentions.

## TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS .....	IV
RÉSUMÉ.....	V
ABSTRACT .....	VI
TABLE DES MATIÈRES .....	VII
LISTE DES TABLEAUX.....	XII
LISTE DES FIGURES.....	XIII
LISTE DES SIGLES ET ABRÉVIATIONS .....	XIX
LISTE DES SYMBOLES .....	XX
LISTE DES ANNEXES.....	XXI
CHAPITRE 1 INTRODUCTION.....	1
CHAPITRE 2 REVUE DE LITTÉRATURE .....	3
2.1 Les stratégies de contrôle .....	3
2.1.1 Extraction et sélection de caractéristiques .....	4
2.1.2 Reconnaissance de formes .....	6
2.2 Les capteurs.....	7
2.2.1 Combinaison de capteurs .....	7
2.2.2 Inconvénients des capteurs EMGs .....	8
2.2.3 La recherche d’alternatives .....	9
2.2.4 Les capteurs à base de jauges de déformation.....	12
2.2.5 Les centrales inertielles .....	16
2.3 Les réseaux de neurones artificiels.....	17
2.3.1 La rétropropagation.....	19



2.3.2	Les fonctions d'activations.....	19
2.3.3	Les réseaux de neurones convolutifs.....	20
2.3.4	Les réseaux de neurones récurrents.....	23
2.4	L'apprentissage .....	26
2.4.1	Les fonctions coût .....	26
2.4.2	Les méthodes d'optimisation .....	28
2.5	Les méthodes de régularisation .....	30
2.6	Mesure de la performance de l'algorithme de classification.....	32
CHAPITRE 3 JUSTIFICATION DU PROJET DE RECHERCHE.....		37
3.1	Résumé de la problématique .....	37
3.2	Objectif général .....	37
3.3	Objectifs spécifiques .....	38
CHAPITRE 4 MÉTHODOLOGIE.....		39
4.1	Conception du capteur utilisant une centrale inertielle .....	39
4.1.1	Choix de la centrale inertielle.....	39
4.1.2	Calibration de la centrale inertielle .....	40
4.2	Conception du capteur utilisant des jauges de déformation.....	42
4.2.1	Choix du type de jauges de déformation.....	42
4.2.2	Matrice de jauges de déformation .....	43
4.2.3	Traitement sur platine d'expérimentation .....	48
4.2.4	Remplacement de la platine d'expérimentation .....	53
4.3	Ensemble de données .....	53
4.3.1	Placement des capteurs.....	54
4.3.2	Acquisition des signaux .....	54

4.3.3	Exercices .....	55
4.3.4	Collecte des données .....	55
4.3.5	Traitement des signaux.....	55
4.4	Traitement des données .....	56
4.5	Logiciels et matériels .....	59
4.6	Expériences .....	60
4.6.1	Comparaison des modèles .....	61
4.6.2	Recherche d'hyperparamètres .....	63
4.6.3	Effet des données cinématiques .....	64
CHAPITRE 5 ARTICLE 1 : DEVELOPMENT OF A WEARABLE SYSTEM TO IDENTIFY MOVEMENT INTENTIONS BY COMBINING STRAIN GAUGES AND INERTIAL MEASUREMENT UNITS.....		65
5.1	Abstract .....	65
5.2	Index Terms.....	66
5.3	Introduction .....	66
5.3.1	On the combination of sEMG-IMU to identify movement intention: context and physiological justification .....	66
5.3.2	Limitations of sEMG and search for alternative solutions.....	67
5.3.3	Strain gauges as the promising solution .....	68
5.3.4	Problem, objective, and research hypotheses .....	69
5.4	Methods.....	70
5.4.1	Design of a strain gauge bracelet .....	70
5.4.2	Muscles Involved in Upper Limb Movement .....	71
5.5	Circuits and systems.....	75
5.5.1	Microcontroller.....	75

5.5.2	IMU sensor design .....	75
5.5.3	Strain gauge-based sensor .....	77
5.5.4	Signal Acquisition.....	79
5.6	Results .....	80
5.7	Discussion .....	85
5.7.1	Identification of movement identification.....	85
5.7.2	Sensor design.....	89
5.8	Conclusion.....	90
5.9	Acknowledgment .....	90
5.10	References .....	90
CHAPITRE 6 RÉSULTATS COMPLÉMENTAIRES .....		100
6.1	Comparaison des modèles.....	100
6.2	Recherche des hyperparamètres .....	105
6.3	Impact des données inertielles.....	108
CHAPITRE 7 DISCUSSION GÉNÉRALE .....		111
7.1	Synthèse de l'article scientifique.....	111
7.1.1	Détection d'intention de mouvement .....	111
7.1.2	Fabrication du capteur .....	111
7.2	Comparaison des modèles de classification .....	112
7.3	Recherche d'hyperparamètres .....	113
7.4	Impact des données inertielles.....	113
7.5	Limites et perspectives .....	113
CHAPITRE 8 CONCLUSION ET RECOMMANDATIONS.....		115
RÉFÉRENCES.....		117

ANNEXES ..... 126

## LISTE DES TABLEAUX

Tableau 2.1 les différents substrats utilisés dans la fabrication des capteurs.....	9
Tableau 2.2 les différents types de conducteurs utilisés dans la fabrication des capteurs .....	10
Tableau 2.3 Formules et représentations des fonctions d’activations les plus courantes [58] .....	19
Tableau 2.4 Matrice de confusion pour une classification binaire.....	33
Tableau 2.5 Résumé des critères de conception.....	35
Tableau 4.1 Les différents types de jauges testées dans le cadre du projet de recherche .....	42
Tableau 5.1 Muscles involved in identified upper limb movements .....	74
Tableau 5.2 Peak signal intensity for each strain gauge per movement(mV).....	84
Tableau 5.3 Strain gauge signal intensity pattern .....	85
Tableau 6.1 Hyperparamètres du modèle de référence.....	100
Tableau 6.2 Comparaison des résultats des modèles LSTM et CNN-LSTM .....	101
Tableau 6.3 Résultat de la variation du taux d'apprentissage.....	105
Tableau 6.4 Résultat de la variation du nombre de filtres pour le CNN .....	106
Tableau 6.5 Résultat de la variation du nombre de neurones pour le LSTM.....	106
Tableau 6.6 Résultat de la variation du terme de régularisation .....	107
Tableau 6.7 Résultat de la variation du nombre d'échantillon par lot.....	107
Tableau 6.8 Résumé des valeurs optimales d’hyperparamètres.....	108
Tableau 6.9 Comparaison des modèles LSTM et CNN-LSTM.....	109

## LISTE DES FIGURES

Figure 2.1 Étapes du contrôle basé sur la reconnaissance de formes. Le processus commence par les signaux bruts et se termine par une sortie de contrôle. Figure adaptée de [10].....	4
Figure 2.2 Exemples de capteurs souples. (a) Films métalliques minces biphasiques (solide-liquide) intrinsèquement extensibles. Tiré de (Hirsch <i>et al</i> , 2016). (b) Peau artificielle douce utilisant des microcanaux intégrés et des conducteurs liquides. Tiré de (Park <i>et al</i> , 2012). (c) Conception structurelle à microcanaux pour un capteur super-étirable à base de métal liquide à température ambiante. Tiré de (Gao <i>et al</i> ,2019).....	11
Figure 2.3 Exemple d’une jauge de déformation à trame pelliculaire .....	12
Figure 2.4 Pont de Wheatstone .....	14
Figure 2.5 Configuration quart de pont de Wheatstone .....	15
Figure 2.6 Équilibrage du quart de pont de Wheatstone avec un potentiomètre .....	16
Figure 2.7 Neurone artificiel avec les entrées ( $x_1 \dots x_n$ ), les poids ( $w_1 \dots w_n$ ), le biais ( $b$ ). La sortie $y$ du neurone est le résultat de la fonction d’activation appliquée à la somme pondérée de toutes entrées et du biais. Figure adaptée de [58]......	17
Figure 2.8 Perceptrons multicouches. Dans cette architecture la couche d’entrée comprend 3 neurones, la couche cachée deux neurones et la sortie un neurone pour un classificateur binaire. Figure adaptée de [62]......	18
Figure 2.9 Représentation graphique d’un réseau de neurones convolutifs 1D. L’entrée est alimentée à travers une série de couches de convolution suivies de couches de « Pooling », pour former des structures arbitrairement profondes. $N$ représente le nombre de filtres dans chaque couche. L’opération finale consiste à une vectorisation permettant d’alimenter en général un perceptron multicouche. ....	22
Figure 2.10 Exemple d’opérations de convolution et de « max pooling » 1D. Un filtre est appliqué sur le vecteur d’entrée avec un pas de 1. Cela signifie que le filtre se déplace sur tout le vecteur d’entrée avec un pas de $s = 1$ . Ensuite, une opération vectorielle est faite entre les éléments du filtre et les éléments du vecteur d’entrée : ici une somme pondérée. Après l’opération de	

convolution, l'opération de « max pooling » choisit la valeur maximale entre deux valeurs à partir d'un filtre qui parcourt tous les éléments avec un pas de 2. ....	22
Figure 2.11 Illustration d'un réseau de neurones récurrent (RNN). À gauche, l'architecture du réseau est représentée avec les flèches représentant les connexions récurrentes. Sur le côté droit, ces connexions sont représentées dans l'espace, où chaque pas de temps forme une nouvelle couche. Figure adaptée de [65]. ....	23
Figure 2.12 LSTM architecture « many-to-one ». Analyse de séquence de mouvement : à chaque pas de temps la séquence de mouvement est évaluée par une cellule LSTM. Le résultat de la couche est associé aux classes possibles, et est calculé après chaque pas. Figure adaptée de [66]. ....	24
Figure 2.13 Cellule LSTM. $C_{t-1}$ Représente l'étape précédente. $h_{t-1}$ L'état de la sortie précédente, $x_t$ représente l'échantillon de la séquence au temps $t$ , $C_t$ représente l'état de la cellule mis à jour et $h_t$ le résultat. Nous avons aussi représenté les portes : la porte d'oubli $f_t$ , la porte d'entrée $i_t$ et la porte de sortie $o_t$ [65]. ....	25
Figure 2.14 Cellules LSTM à travers le temps. Chaque étape de cellule reçoit un échantillon de la séquence $x$ de la couche d'entrée et envoie un état de cellule mis à jour et la valeur de sortie $h$ à l'étape suivante. Chaque pas de temps envoie également la valeur $h$ à la couche de sortie [65]. ....	25
Figure 4.1 Les différentes centrales inertielles testées dans le cadre du projet. À gauche, la centrale inertielle à 10 DDL de DfRobot. Au milieu, la centrale inertielle BNO055 à 9 DDL de Adafruit. À droite, la centrale inertielle Fermion de 9 DDL de Adafruit. ....	39
Figure 4.2 Calibration de magnétomètre. Figure tirée de [78]. ....	41
Figure 4.3 Calibration de l'accéléromètre. Six positions stables de références. Figure tirée de [78] .....	41
Figure 4.4 PCB flexible avec des jauges suivant une direction unique .....	44
Figure 4.5 PCB flexible avec les jauges de déformation soudées suivant une direction unique ...	44
Figure 4.6 PCB flexible avec les jauges suivant deux directions différentes .....	44

Figure 4.7 PCB flexible avec les jauges de déformation soudées suivant deux directions différentes .....	45
Figure 4.8 (a) Silicone liquide dragon Skin 10 FAST de Smooth-On. (b) APHIX de Smooth-On : agent adhérent permettant de coller la majorité des matériaux sur du silicone. (c) Applicateur de film réglable de 100 mm avec dispositif de préparation de film humide micrométrique 0-3500 $\mu\text{m}$ de largeur.....	46
Figure 4.9 PCB flexible intégré dans un bracelet de silicone avec les jauges de déformations suivant une direction unique. La bande autour du PCB flexible a été une astuce pour ne pas avoir du silicone à cet endroit.....	46
Figure 4.10 PCB flexible intégré dans un bracelet de silicone avec les jauges de déformations suivant deux directions différentes.....	47
Figure 4.11 Impression 3D flexible d'un bracelet en TPU intégrant des jauges de déformations .	47
Figure 4.12: Bande kinésiologique instrumentalisée avec une matrice de jauges de déformation	48
Figure 4.13 Chaîne de mesures des capteurs à base de jauges de déformation. Figure adaptée de [54] .....	49
Figure 4.14 Circuit électrique de la carte d'acquisition des signaux provenant des jauges de déformation .....	50
Figure 4.15 Platine d'expérimentation reproduisant le schéma électrique présenté à la Figure 4.14 .....	50
Figure 4.16 Outil permettant de choisir les caractéristiques et la plage de lecture de l'amplificateur AD623 [81].L'interface à droite permet de régler les paramètres de l'amplificateur. L'interface à gauche permet de visualiser la plage de lecture en fonction des paramètres choisis.....	52
Figure 4.17 Étiquetage des signaux pour le mouvement d'extension du poignet pour un des sujets. ....	56
Figure 4.18 Représentation des dimensions de la matrice obtenue après le traitement des données. Suivant les colonnes, les données provenant des capteurs (24 signaux). Suivant les lignes le	



nombre de données correspondant à 250 ms d'acquisition. La profondeur représente le nombre de fenêtres de 250ms x 25 canaux. ....	57
Figure 4.19 Séparation des données provenant des capteurs et des classes associés aux fenêtres	58
Figure 4.20 Répartition des données par classe de mouvement.....	59
Figure 4.21 Encodage « one hot » pour la classification multiclasse .....	61
Figure 4.22 Modèle LSTM. L'entrée est transmise à deux couches LSTM comptant chacune 32 neurones. Un dropout de 0.2 a également été appliqué à ces couches. A la fin du réseau, il y a un perceptron multicouche qui se termine avec deux couches. Une couche dense comptant 18 neurones et un terme de régularisation $\lambda = 0.002$ . Une de sortie avec une fonction d'activation softmax et 10 neurones, une pour chaque classe de mouvement. ....	62
Figure 4.23 Modèle CNN-LSTM. Ce modèle a été obtenu en rajoutant un réseau de neurones de convolutif juste avant le modèle LSTM présenté à la Figure 4.16. L'entrée est d'abord transmise à une couche convolutive 1D avec 32 filtres de dimension [1x5] et un pas de 1. La fonction ReLu a été utilisée comme fonction d'activation. La couche convolutive est suivie par une couche de mise en commun « max pooling ». Les vecteurs de caractéristiques extraits par ceux-ci seront alors transmis au réseau de neurones récurrents.....	63
Figure 4.24 Représentation des dimensions de la matrice obtenue après la suppression des données inertielles. Suivant les colonnes, les données provenant des bracelets de jauges de déformation (12 signaux). Suivant les lignes le nombre de données correspondant à 250 ms d'acquisition soit 25. La profondeur représente le nombre de fenêtres de 250 ms x 12 canaux. ....	64
Figure 5.1 Strain gauge bracelet. (a) Flexible PCB (yellow) with the connections for 6 strain gauges. (b) Instrumented kinesiological tape (blue) the flexible PCB (yellow) connected to the 6 numbered strain gauges (orange rectangles). ....	71
Figure 5.2 (a) Placement of the two strain gauge bracelets: bracelet B around the arm, and bracelet AB placed around the forearm; two IMUS: IMU (1) placed on the hand; IMU (2) placed on the forearm. (b) Anterior view of the human upper limb. (c) Posterior view of the human upper limb. The image (b) and (c) were adapted from [27]. ....	72
Figure 5.3 The 10 identified movements. (1) Elbow Flexion (EF). (2) Elbow Extension (EE). (3) Forearm Pronation (FP). (4) Forearm Supination (FS). (5) Wrist Flexion (WF). (6) Wrist	

- Extension (WE). (7) Wrist Ulnar Deviation (WL). (8) Wrist Radial Deviation (WR). (9) Power grips (PG). (10). Rest position “no movement” (NM)..... 73
- Figure 5.4 IMU circuit. BNO055 IMU (left) connected to the ESP32 microcontroller (right) using the I2C protocol. The BNO055 IMU has a 3,3V input that was connected directly into the ESP32 3,3V output (red wire). A ground (GND) pin was connected to GND on the ESP32 (black wire), the I2C clock pin (SCL) was connected to the corresponding ESP32 pin 22 (yellow wire), and the I2C data pin (SDA) was connected to the corresponding ESP32 pin 23 (blue wire). The image was adapted from [29]. ..... 76
- Figure 5.5 3D printed box for electronics. (a) Box (12x10x3.3 cm) containing the PCB for strain gauge signal conditioning. (b) Box (5.5x3.5x2.2 cm) containing the IMU circuit and a Velcro strap. .... 77
- Figure 5.6 Electrical circuit for strain gauge signal conditioning showing the Wheatstone bridges in quarter bridge configuration, the multiplexers, the ADS1256 and the ESP32 microcontroller. .... 77
- Figure 5.7 PCB for strain gauge signal acquisition. (a) ESP32. (b) FFC/FPC connector. (c) Wheatstone Bridge. (d) Multiplexers. (e) Power management. (f) ADS1256. .... 78
- Figure 5.8 Communication architecture including the ESP32 microcontroller, the two strain gauge bracelets  $B$  and  $AB$ , the two IMUs 1 and 2, the communication protocols ESP-NOW (between the ESP32 and the bracelets and IMUs) and UART (between the ESP32 and the computer), and the variables corresponding to the sensors output: e.g.  $AB1$  to  $AB6$  for the strain gauge bracelet  $AB$ ,  $B1$  to  $B6$  for the strain gauge bracelet  $B$ .  $ax$  for linear acceleration following X-axis,  $ay$  for linear acceleration following Y-axis,  $az$  for linear acceleration following Z-axis.  $gx$  for angular velocity around the X-axis,  $gy$  for angular velocity around Y-axis,  $gz$  for angular velocity around Z-axis The number 1 or 2 referred to IMU 1 or IMU 2. .... 80
- Figure 5.9 (a) 3D linear accelerations and 3D angular velocities from the IMU 1 placed on the hand. (b) 3D linear accelerations and 3D angular velocities from the IMU 2 placed on the forearm. (c) Data from the six strain gauges in the bracelet  $B$  placed on the arm. (d) Data from the six strain gauges in the bracelet  $AB$  placed on the forearm. .... 83

Figure 6.1 : (a) courbe de la précision de classification du modèle LSTM, (b) courbe de la fonction coût du modèle LSTM.....	102
Figure 6.2 : (a) courbe de la précision de classification du modèle CNN-LSTM, (b) courbe de la fonction coût du modèle CNN-LSTM .....	102
Figure 6.3 Matrice de confusion du modèle LSTM .....	103
Figure 6.4 Matrice de confusion du modèle CNN-LSTM .....	104
Figure 6.5 Évolution de la précision pour les ensembles tests et validation en fonction du taux d'apprentissage.....	105
Figure 6.6 Évolution de la précision pour les ensembles tests et validation en fonction du nombre de filtres pour le CNN .....	106
Figure 6.7 Évolution de la précision pour les ensembles tests et validation en fonction du nombre de neurones pour le LSTM.....	106
Figure 6.8 Évolution de la précision pour les ensembles tests et validation en fonction du terme de régularisation.....	107
Figure 6.9 Évolution de la précision pour les ensembles tests et validation en fonction du nombre d'échantillon par lot .....	107
Figure 6.10 (a) : courbe d'apprentissage du modèle CNN-LSTM, (b): courbe de la fonction coût du modèle CNN-LSTM.....	109
Figure 6.11 Matrice de confusion du modèle entraîné exclusivement sur les signaux provenant des jauges de déformation .....	110
Figure 1.1 Taux d'apprentissage de 1e-01 .....	131
Figure 1.2 Taux d'apprentissage de 1e-02 .....	131
Figure 1.3 Taux d'apprentissage de 1e-03 .....	131
Figure 1.4 Taux d'apprentissage de 1e-04 .....	131
Figure 1.5 Taux d'apprentissage de 1e-05 .....	131

## LISTE DES SIGLES ET ABRÉVIATIONS

- CNN « Convolutional neural network », réseau de neurones convolutif
- DDL Degré de liberté
- EE « Elbow extension », extension du coude
- EF « Elbow flexion », flexion du coude
- EMG Électromyographie
- FP « Forearm pronation », pronation de l'avant-bras
- FS « Forearm supination », supination de l'avant-bras
- IMU « Inertial Measurement Unit », centrale inertielle
- NN réseau de neurones
- NM « No movement », mouvement de repos
- MLP perceptron multicouche
- PG « Power grasp », mouvement de préhension
- LSTM « Long Short-Term Memory »
- RNN « Recurrent neural network », réseau de neurones récurrents
- sEMG « surface electromyographie », électromyographie de surface
- SGD « Stochastic Gradient Descent », descente de gradient stochastique
- WE « Wrist extension », extension du poignet
- WF « Wrist flexion », flexion du poignet
- WR « Wrist radial deviation », déviation radiale du poignet
- WL « Wrist ulnar deviation », déviation ulnaire du poignet

## LISTE DES SYMBOLES

$J$  : Fonction coût par rapport à l'ensemble des données

$\lambda$  : Terme de régularisation

$W$  : Poids associés aux couches de neurones

$\Delta R/R$  : Variation relative de la résistance

$\Delta L/L$  : Variation relative de la longueur

$S_A$  : Sensibilité du matériau à la déformation

$f_t$  : « Forget gate » Porte d'oubli pour le réseau de neurones récurrents

$i_t$  : « Input gate » Porte d'entrée pour le réseau de neurones récurrents

$o_t$  : « Ouput gate » Porte de sortie pour le réseau de neurones récurrents

$b$  : Biais associé au réseau de neurones

$h$  : Résultat du calcul effectué par la cellule LSTM

$x$  : Vecteur d'entrée

$C_t$  : État de la cellule LSTM mis à jour

$\mathcal{L}(\hat{y}, y)$  : Fonction coût

$m_t, v_t, g_t$  : estimation des premiers et second moment brutes du gradient

$\beta_1; \beta_2$  : Hyperparamètres pour contrôler les taux de décroissance exponentielle de chaque moyenne mobile

$\theta$  : Paramètres du réseau de neurones

**LISTE DES ANNEXES**

Annexe A Exemple de calcul de gradient .....	126
Annexe B Courbes d'apprentissage des hyperparametres .....	130

## CHAPITRE 1 INTRODUCTION

L'analyse des mouvements chez l'humain est un sujet de recherche majeur avec des applications en biorobotique, diagnostic clinique, ingénierie de la réadaptation et en interfaces Homme-Machine. Dans le domaine de la réadaptation, une des applications de l'analyse de mouvement consiste à détecter les intentions de mouvements de l'utilisateur afin de contrôler des exosquelettes, pour des personnes ayant des faiblesses musculaires, ou des prothèses myoélectriques, pour des personnes ayant subi une amputation. Pour effectuer ce contrôle, les contractions musculaires générées par l'usager sont utilisées afin d'actionner le système. Les capteurs d'électromyographie sont les capteurs les plus utilisés pour détecter ces contractions musculaires. Cependant, hormis les inconvénients liés aux capteurs d'électromyographie, les dispositifs commerciaux utilisent un nombre limité de ces capteurs (généralement deux) ce qui limite le nombre de degrés de liberté (DDL) des systèmes actionnés. En effet, pour effectuer des tâches de la vie quotidienne comme l'ouverture d'une porte, plusieurs DDL sont sollicités pour effectuer la flexion et l'extension du coude, la préhension de la main ainsi que la pronation et supination de l'avant-bras. Il faut un nombre élevé de capteurs pour recouvrir ces groupements musculaires afin d'avoir un nombre adéquat de DDL.

Différents types recherches se sont penchés sur le développement de nouveaux capteurs pouvant remplacer les capteurs d'électromyographies et sur de nouvelles stratégies de contrôles. Contrairement, aux capteurs d'électromyographies qui ont une structure rigide, les technologies émergentes se concentrent sur l'utilisation de capteurs flexibles qui sont capables d'atteindre des déformations du même ordre de grandeur que la peau humaine. Ces capteurs sont souvent combinés à d'autres types de capteurs, chacun mesurant des modalités différentes, et permettant de reconstituer le mouvement avec plusieurs dimensionnalités.

Les stratégies émergentes de contrôles reposent quant à elles sur les algorithmes de reconnaissance de formes. Ces algorithmes ont l'avantage de pouvoir traiter un important flux de données pour les classifier. Ainsi, cette stratégie permettrait d'avoir un contrôle plus intuitif des exosquelettes ou des prothèses myoélectriques avec un nombre de DDL plus élevé.

Dans ce contexte, l'objectif de ce mémoire est de développer un capteur portable identifiant les intentions de mouvement en combinant des jauges de déformation et des centrales inertielles.

Dans cette thèse, le chapitre 2 introduit les types de signaux qui seront utilisés. Par la suite, les concepts de bases permettant la compréhension des algorithmes d'apprentissage profond seront abordés. Ces concepts permettront de comprendre des architectures de bases comme celles des perceptrons multicouches (MLP) et des architectures plus complexes comme les réseaux de neurones convolutifs (CNN) et réseaux de neurones récurrents (RNN). Le chapitre 3 présente la justification du projet. Le chapitre 4 présente la méthodologie complémentaire à celle de l'article. Le chapitre 5 présente l'article qui a été soumis pour publication. Le chapitre 6 présente les résultats complémentaires. Un chapitre de discussion sera présenté à la suite des résultats. Dans le dernier chapitre, une brève conclusion est présentée avec des réflexions sur le travail réalisé tout au long du mémoire ainsi que des commentaires sur les perspectives d'améliorations.



## CHAPITRE 2 REVUE DE LITTÉRATURE

Une approche émergente pour contrôler des exosquelettes et des prothèses myoélectriques consiste à utiliser les mesures d'électrodes d'électromyographie de surface (sEMG) [1]. Ces mesures combinées à des algorithmes de reconnaissance de formes permettent d'identifier l'intention de mouvement de l'utilisateur et ainsi contrôler un actionneur [2]. Cette méthode a le potentiel de permettre aux personnes souffrant de diverses déficiences physiques et/ou troubles neuromusculaires de récupérer une partie de leurs fonctions motrices et ainsi améliorer leur qualité de vie [3].

### 2.1 Les stratégies de contrôle

Malgré les avancées technologiques dans le domaine des exosquelettes et des prothèses myoélectriques, les stratégies de leur contrôle quant à elles, n'ont pas évolué significativement. En effet, la majorité des prothèses commerciales actuelles peuvent répliquer qu'un nombre limité des mouvements du membre supérieur (nombre de DDL limité) [4]. La stratégie de contrôle de ces prothèses repose sur un nombre peu élevé d'électrodes (1-3), et sur un contrôle séquentiel ou un déclencheur de mouvement spécifique, souvent par l'utilisation d'un bouton externe, ou d'une application mobile [5], [6]. Pour le contrôle séquentiel, un signal spécifique, par exemple une activation d'électrodes placées sur une paire de muscles antagonistes, permet de choisir un mouvement parmi un ensemble de mouvements prédéfinis [5], [6]. Pour le déclencheur de mouvement, une séquence d'activation particulière des électrodes ou l'atteinte d'un seuil bien défini par le signal sEMG permettent d'actionner un mouvement spécifique [5], [6]. Ces méthodes de contrôle ne sont pas intuitives et sont encore loin d'être naturelles et nécessitent un certain niveau de compétence ou de longues heures de formation de l'utilisateur [5], [6].

Durant ces dernières années, les stratégies de contrôle basées sur la reconnaissance de formes utilisant des algorithmes d'intelligence artificielle ont prouvé être mieux adaptées aux besoins des utilisateurs. La reconnaissance de formes permet d'identifier les caractéristiques spécifiques à chaque mouvement, et ainsi de contrôler un nombre plus élevé de mouvements (plus de DDL) comparés aux méthodes classiques de contrôle [7]. Cette stratégie pourrait permettre d'obtenir un contrôle plus intuitif de la prothèse ou de l'exosquelette ce qui représenterait un apport majeur.

Pour ce faire, les signaux provenant de différentes électrodes sont utilisés pour entraîner un algorithme de reconnaissance de formes. Il existe différents algorithmes de classifications de mouvements, mais leur but principal reste le même, classifier correctement un mouvement en fonction des signaux reçus. Avec cette stratégie, l'utilisateur peut simplement penser à exécuter un mouvement de façon naturel (ex. : pronation de l'avant-bras) et en contractant le membre les signaux peuvent être reconnus par l'algorithme de reconnaissance de formes et ce mouvement sera exécuté par la prothèse ou supporté par l'exosquelette [8].

Les étapes majeures de la stratégie de contrôle utilisant un algorithme de reconnaissance de formes sont illustrées à la Figure 2.1. Elle repose sur :

- **L'acquisition des signaux** : utilisation des capteurs pour collecter les signaux d'activations musculaires.
- **L'extraction de caractéristiques** : permettant de retenir les informations les plus importantes du signal.
- **La classification** : prédiction du mouvement du membre
- La **production du signal de contrôle** : permettant d'activer l'actionneur pour effectuer le mouvement désiré [9].



Figure 2.1 Étapes du contrôle basé sur la reconnaissance de formes. Le processus commence par les signaux bruts et se termine par une sortie de contrôle. Figure adaptée de [10]

Les étapes majeures de la stratégie de contrôle que sont l'extraction de caractéristiques, la reconnaissance de formes et la classification des mouvements sont discutées dans les sections suivantes.

### 2.1.1 Extraction et sélection de caractéristiques

L'extraction des caractéristiques consiste à appliquer des transformations sur des données brutes afin d'obtenir un ensemble de caractéristiques pertinentes utilisable par les algorithmes d'apprentissage machine [11]. Ces caractéristiques doivent être en mesure de capter l'information principale du signal permettant d'améliorer la performance du classificateur.

Dans le processus de sélection des caractéristiques, plusieurs techniques comme l'analyse des composants principaux (PCA), l'analyse discriminante linéaire (LDA), l'analyse des composantes indépendantes (PCI), les valeurs statistiques et la mesure d'entropie peuvent être employées [10], [12], [13]. En plus, les caractéristiques peuvent être étudiées dans le domaine temporel, dans le domaine fréquentiel ou une combinaison des deux (domaine temps-fréquence) [14]. Les caractéristiques dans le domaine temporel incluent la moyenne, la médiane, la variance, la moyenne quadratique (RMS), l'écart type et l'intervalle interquartile [14]. Les caractéristiques du domaine fréquentiel, qui représentent principalement la structure périodique du signal, sont la transformée de Fourier, l'entropie spectrale, l'énergie spectrale et les coefficients des filtres autorégressifs [14]. Les caractéristiques du domaine temps-fréquence sont utilisées pour étudier à la fois les caractéristiques temporelles et fréquentielles de signaux complexes et utilisent en général des techniques d'ondelettes, telles que les coefficients d'ondelettes ou l'énergie des coefficients d'ondelettes [15].

Après avoir déterminé un ensemble de caractéristiques provenant du signal d'origine, les algorithmes d'extractions de caractéristiques peuvent être combinés à des algorithmes de sélection de caractéristiques. Ces derniers ont pour but de réduire la dimensionnalité des données en identifiant les caractéristiques les plus importantes pour le classificateur [16].

En plus de ces caractéristiques individuelles, des travaux de recherche ont été menés sur leurs combinaisons pour développer de nouveaux ensembles de caractéristiques. La combinaison des caractéristiques temporelles utilisées par Hudgins [7] est la plus utilisée et comprend les caractéristiques suivantes : MAV (valeur absolue moyenne), WL (longueur d'onde), ZC (passage à zéro), and SSC (changement de signe de la pente). Plus récemment, Guo Shuxiang *et al.* [17] ont proposé huit combinaisons de quatre méthodes d'extraction de caractéristiques RMS (racine moyenne quadratique), DFA (analyse de fluctuation sans tendance), WP (pics de poids) et MM (modèle musculaire). De nombreuses études ont montré que le succès des algorithmes de classification dépend fortement de la qualité l'extraction et de la sélection des caractéristiques [17]. Il est donc essentiel de déterminer les caractéristiques qui ont le plus d'impacts sur l'algorithme de classification.

## 2.1.2 Reconnaissance de formes

Après avoir extrait et sélectionné les caractéristiques, des algorithmes de reconnaissance de formes sont utilisés afin de corréler les caractéristiques extraites dans des classes de mouvement spécifiques. Les algorithmes de reconnaissance de formes les plus utilisés dans les récents travaux sont l'analyse discriminante linéaire (LDA) [12], les machines à vecteurs de support (SVM) [18], et les modèles de Markov [19].

Les classificateurs LDA ont longtemps été considérés comme la meilleure approche pour la classification des signaux myoélectriques. Le principal avantage de LDA est sa simplicité de mise en œuvre, surtout pour les systèmes embarqués, et sa facilité d'entraînement. Cependant, le LDA présente plusieurs inconvénients. En effet, un classificateur LDA ne peut produire qu'une seule sortie, ainsi pour les applications où plus d'un mouvement est exécuté, il est nécessaire d'avoir un classificateur pour chacun des mouvements [20].

Hormis le LDA, de nombreux travaux dans la littérature ont mis en évidence la pertinence des réseaux de neurones (NN) à classifier des ensembles de données. L'avantage des réseaux de neurones réside dans leur capacité à représenter à la fois des relations linéaires et non linéaires ; et apprendre ces relations directement à partir des données modélisées [5]. Englehart *et al.* [21] ont développé un NN à perceptron multicouche (MLP) pour classifier des caractéristiques du domaine temporel des signaux myoélectriques. L'algorithme développé a permis de classifier quatre types de mouvements du membre supérieur, avec un taux d'erreur d'environ 10%. Les travaux effectués par l'équipe des Prof. Raison et Prof. Achiche (Gaudet *et al.* [22]) ont proposé un MLP pour classifier 8 mouvements du membre supérieur de cinq amputés transhuméraux et ont obtenu une précision de classification compris entre 60.9% et 93.0%.

Atzori *et al.* [23] ont proposé un réseau de neurones convolutif (CNN) pour classifier les mouvements du jeu de données Ninapro 1 [24] et ont atteint une précision de classification de  $66,6 \pm 6,4$  % sur plus de 50 mouvements de la main. La motivation derrière l'utilisation des CNN réside dans leur puissante fonction d'extraction de caractéristiques [25]. En effet, ils sont capables d'apprendre des caractéristiques pertinentes directement à partir des données. Cependant, les CNN ne prennent pas en compte les dépendances temporelles entre les données. Les signaux physiologiques sont séquentiels de nature, et il est important de capter cette information. Pour cela, les réseaux de neurones récurrents (RNN) sont utilisés. Les RNN permettent de prendre en

considération la dépendance temporelle entre les données en changeant dynamiquement leur état interne [25]. Ils sont utilisés pour la prédiction et classification de séries chronologiques [25]. Récemment, les unités de mémoire à long court terme (LSTM) et les unités récurrentes fermées (GRU) sont devenues les deux architectures RNN les plus utilisées sur les données chronologiques. Les travaux effectués par l'équipe des Prof. Raison et Prof. Achiche (Barron *et al.* [26]) ont proposé un RNN pour identifier 6 gestes du membre supérieur de cinq amputés transhuméraux et ont obtenu une précision de classification de 79.7%. Wu *et al.*, [25] ont combiné un CNN et un LSTM pour proposer le LCNN. Cette architecture tire l'avantage des LSTM qui sont utilisés pour extraire des informations temporelles dans les signaux et des CNN qui sont utilisés pour extraire des caractéristiques et classifier les signaux. La précision de classification moyenne du LCNN a atteint 98,1% pour cinq mouvements de la main.

Les réseaux de neurones profonds tels que les CNN et LSTM ont permis d'enregistrer des avancées majeures dans le domaine de l'intelligence artificielle. Il est donc important d'investiguer la capacité de ces méthodes pour la reconnaissance d'intention de mouvement.

## **2.2 Les capteurs**

sEMG est une technique qui utilise des électrodes placées sur la peau à un endroit spécifique pour enregistrer les contractions musculaires [15]. La performance des stratégies de contrôle basées sur des algorithmes de reconnaissance de formes repose grandement sur la capacité des algorithmes à identifier les informations les plus importantes provenant des signaux sEMG qui sont eux-mêmes caractérisés par une grande variabilité et des propriétés non stationnaires [27]. La nature variable des signaux sEMG, rend très difficile la mise en place d'algorithmes robustes permettant d'identifier de manière efficace les mouvements du membre supérieur et de les utiliser pour contrôler des exosquelettes ou des prothèses [26].

Dans les sections suivantes, d'autres types de capteurs pouvant remplacer les capteurs sEMG sont présentés.

### **2.2.1 Combinaison de capteurs**

La combinaison entre les capteurs sEMG et les IMU est la combinaison de capteurs la plus couramment utilisée sur le membre supérieur [28]. Ces capteurs fournissent des informations qui

sont fréquemment utilisées pour identifier l'intention de mouvement chez l'Homme. Les applications les plus répandues sont la reconnaissance des gestes de la main et du doigt [29],[30] pour classer les mouvements fantômes des membres supérieurs chez les amputés transhuméraux à partir du moignon afin de contrôler leur prothèse [22],[31] ou exosquelette [32].

La combinaison entre sEMG et IMU peut se justifier physiologiquement, car :

1. Les sEMG permettent de détecter l'intention de mouvement en mesurant l'activité musculaire. C'est pourquoi le sEMG reste le capteur principal pour commander les prothèses myoélectriques.
2. Les IMUs fournissent des informations cinématiques supplémentaires sur le mouvement, c'est-à-dire les configurations d'articulation, les vitesses et les accélérations.

Les IMUs pallient deux limitations des sEMG :

1. Les sEMG souffrent généralement de l'effet de la position des membres, où les signaux sEMG pour le même mouvement sont différents dans différentes positions des membres [33], [34]. L'IMU peut donc fournir cette distinction.
2. Les IMU sont particulièrement efficaces pour capturer des mouvements plus importants, tandis que les données sEMG ont permis de mieux distinguer les différentes formes de mains et les mouvements des doigts [29].

### **2.2.2 Inconvénients des capteurs EMGs**

Bien que les signaux sEMG sont les plus utilisés pour la détection des mouvements, il reste qu'ils présentent plusieurs inconvénients [35], [36].

1. Les signaux sEMG deviennent souvent instables en raison de la transpiration, des déplacements des électrodes, des artefacts de mouvement et du bruit causé par la présence d'équipements électroniques. [15],[36].
2. De plus, une diaphonie peut se produire en raison du nombre élevé de muscles côte à côte et la fatigue musculaire peut affecter de manière cruciale la qualité des signaux [15],[36].
3. La quantité de données provenant de sEMG (acquisition 1000 Hz) nécessite une puissance de calcul élevée pour les traiter en temps réel dans un dispositif portable [37].

## 2.2.3 La recherche d'alternatives

L'intérêt grandissant pour les technologies intelligentes portables nécessite le développement de nouveaux capteurs à faible coût, avec une sensibilité élevée et une faible limite de détection[38]. Les récentes recherches se sont penchées sur le développement de capteurs intégrés à des substrats qui ont la capacité de fléchir, de plier ou de s'étirer. Ces capteurs sont faits de matériaux biocompatibles qui peuvent atteindre des contraintes du même ordre de grandeur que les élastomères et la peau humaine, ce qui rend ces capteurs compatibles avec les technologies portables émergentes [39].

### 2.2.3.1 Les capteurs flexibles

Ces capteurs sont généralement faits de deux composants :

- **Un matériau conducteur** : L'élément de détection et/ou de transmission de signal
- **Un substrat** : L'élément flexible utilisé pour encapsuler le matériau conducteur

Le Tableaux 2.1 et le Tableau 2.2 font le résumé des substrats et conducteurs utilisés pour la fabrication de ces capteurs.

Tableau 2.1 les différents substrats utilisés dans la fabrication des capteurs

Substrats	Principe de fonctionnement
Élastomères[40],[41]	Utiliser des polymères hautement flexibles, capables de se conformer à la peau qui se peut se déformer jusqu'à 30% de sa longueur initiale.
Tissus [42],[43]	Utiliser des fibres conductrices tissées de la même manière que les fibres conventionnelles qui agissent comme des éléments sensoriels.

Les élastomères sont les substrats les plus utilisés pour la fabrication des capteurs flexibles. Certains sont capables de se déformer jusqu'à 700% sans perdre leurs propriétés mécaniques.

Tableau 2.2 les différents types de conducteurs utilisés dans la fabrication des capteurs

Conducteurs	Principe de fonctionnement
<b>Films métalliques minces</b> [44],[45]	Rendre les matériaux conducteurs plus fins afin qu'ils deviennent flexibles.
<b>Métal liquide</b> [46][47]	Utiliser des métaux liquides (part ex eGaIn, Galistan) dans l'électronique flexible comme alternatives aux métaux conventionnels.
<b>Liquides et solutions ioniques</b> [48], [49]	Utiliser des solutions ioniques (par ex NaCl, KCl) comme des matériaux conducteurs.
<b>Encres conductrices</b> [40],[50][51]	Utiliser un solvant qui contient une suspension de particules conductrices, telles que les nanoparticules métalliques, les composés organométalliques, les nanotubes de carbone et Graphene comme conducteurs.

Des exemples de capteurs flexibles sont présentés sur la Figure 2.2. Hirsch *et al* [46], a proposé un capteur utilisant un film métallique mince Figure 2.2 (a). Ce capteur souple a été développé pour enregistrer l'amplitude des mouvements du doigt humain. Park *et al* [52] a proposé une peau artificielle utilisant comme substrat du silicone et comme conducteur un métal liquide (eGaIn) Figure 2.2 (b). Le principe de fonctionnement de cette peau artificielle repose sur le fait que la résistance du capteur change lorsqu'il subit une déformation. Gao *et al* [53], a proposé un capteur super-étirable à base de métal liquide Figure 2.2 (c). Ce capteur à une structure similaire à celle d'une jauge de déformation encapsulée dans un élastomère.

L'utilisation de ces capteurs présente différents défis que les chercheurs s'efforcent de surmonter:

- La généralisation des techniques de fabrication hautement spécialisées pour les matériaux souples
- L'intégration de l'électronique miniaturisée.



La résolution de ces défis améliorera considérablement l'utilité de ces méthodes pour fabriquer des capteurs qui pourront être utilisés en dehors d'un environnement de laboratoire.

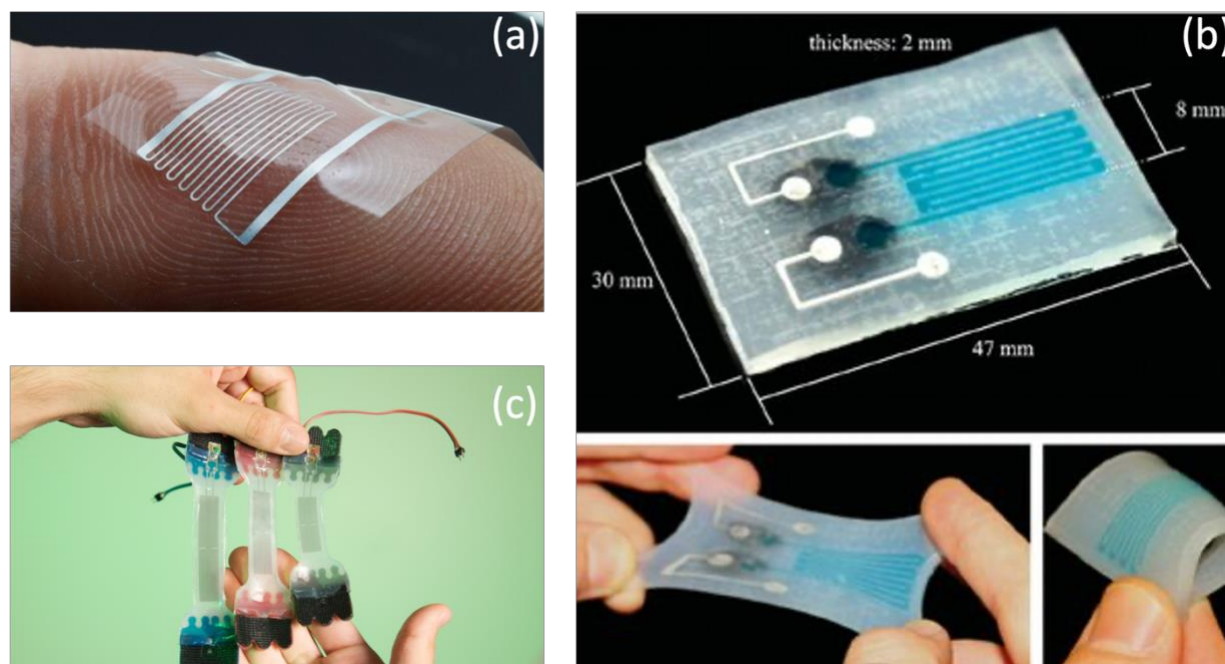


Figure 2.2 Exemples de capteurs souples. (a) Films métalliques minces biphasiques (solide-liquide) intrinsèquement extensibles. Tiré de (Hirsch *et al*, 2016). (b) Peau artificielle douce utilisant des microcanaux intégrés et des conducteurs liquides. Tiré de (Park *et al*, 2012). (c) Conception structurale à microcanaux pour un capteur super-étirable à base de métal liquide à température ambiante. Tiré de (Gao *et al*, 2019).

### 2.2.3.2 Un concept préliminaire utilisant les jauges de déformation

Les travaux effectués par l'équipe des Prof. Raison et Prof. Achiche (Zizoua *et al*. [41]) ont présenté une preuve de concept d'un bracelet utilisant des jauges de déformation unidirectionnelles pour l'identification de quatre mouvements du membre supérieur chez les amputés traumatiques : flexion/extension du coude et pronation/supination de l'avant-bras.

Cependant ce système ne permet pas d'identifier des mouvements tridimensionnels, en raison de ces principales limitations :

1. *Déformations uniaxiales* : les jauges ont été placées uniquement dans le sens longitudinal du biceps brachial ce qui ne permet pas d'enregistrer les déformations multiaxes de la peau.

2. *Fragilité* : les jauges de déformation étaient reliées par deux fils fins. Cela a permis une première preuve de concept, mais ces fils se brisent, ou se dessoudent après quelques utilisations.
3. *Non-portabilité* : le système n'est ni compact ni portable ce qui limite son application dans la vie de tous les jours.
4. *Manque de contact conforme à la peau pendant le mouvement* : les jauges de déformation étaient insérées dans un bracelet en silicone qui n'offrait pas de contact direct avec la peau limitant ainsi la capacité du système à mesurer la déformation de la peau.

Dans la section suivante, le principe de fonctionnement de deux types de capteurs est présenté : les jauges de déformation et les centrales inertielles.

## 2.2.4 Les capteurs à base de jauges de déformation

Cette section nous introduit aux notions de base permettant de comprendre le fonctionnement des jauges de déformation. Il est inspiré du livre « *Méthodes expérimentales et instrumentation en mécanique* » [54].

### 2.2.4.1 Principe de fonctionnement

La jauge de déformation, Figure 2.3, est un élément résistif. On la colle sur une pièce au point où on veut mesurer la déformation à travers un support d'isolation. Elle est constituée d'un fil fin enroulé selon une direction définie qui constitue la grille. Elle a une résistance nominale qui change proportionnellement à la déformation qu'elle subit.

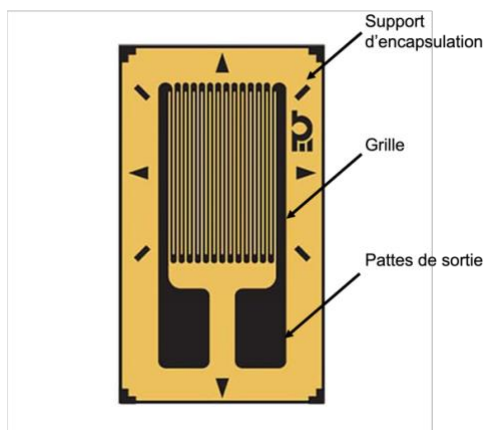


Figure 2.3 Exemple d'une jauge de déformation à trame pelliculaire

### 2.2.4.1.1 Sensibilité du matériau

La résistance des fils conducteurs change lorsqu'ils sont étirés. La relation qui relie la variation de résistance d'un fil conducteur en fonction de sa déformation axiale est définie comme la sensibilité  $S_A$  du matériau.

La valeur  $S_A$  n'est pas constante pour beaucoup de matériaux. Cependant pour des alliages comme le Constantan ou le Nichrome, la valeur  $S_A$  est constante pour une grande plage de déformation. Aussi pour ces matériaux,  $S_A$  est moins sensible aux variations de température.

Le constantan est l'un des matériaux les plus utilisés pour la fabrication des jauges de déformations. C'est un alliage constitué de 45% de Nickel et 55% de cuivre. Ce matériau présente les caractéristiques suivantes :

- Sensibilité ( $S_A \sim 2.1$ ) constante pour une très grande plage de déformation
- Résistivité élevée ( $\rho = 50 \times 10^{-6} \Omega \cdot cm$ ) ce qui permet de fabriquer de très petites jauges
- Très stable et relativement peu affecté par les changements de température.

$$S_A = \frac{\Delta R/R}{\Delta L/L} \quad (2.1)$$

$\Delta R/R$  : Variation relative de la résistance

$\Delta L/L$  : Variation relative de la longueur

$S_A$  : Sensibilité du matériau à la déformation

### 2.2.4.1.2 Facteur de jauge du fabricant ( $S_G$ )

À l'achat des jauges de déformation, le fabricant procure la valeur de sensibilité du matériau  $S_G$ . Contrairement à  $S_A$  qui est une valeur théorique,  $S_G$  est déterminé de façon expérimentale. En effet,  $S_G$  est déterminée pour un lot de jauge de déformations à la suite d'essais standardisés. La jauge doit être soumise à des déformations variant de 0 à  $1000 \mu m/m$  dans un champ de contrainte uniaxiale uniforme.

Pour déterminer  $S_G$  d'un lot de jauges, le fabricant fait un échantillonnage des jauges d'un lot sur lequel il effectue l'essai standardisé. Les procédures d'essais standardisés se font selon :

- Organisation internationale de Métrologie légale, recommandations no. 62
- American Society for Testing and Materials, ASTM Method E251.

### 2.2.4.1.3 Autocompensation de la température.

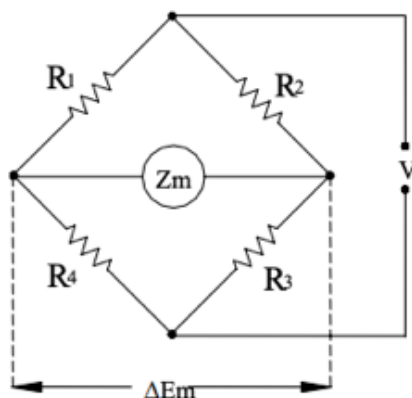
Lorsqu'une jauge est soumise à une variation de température en l'absence de déformation mécanique, 3 phénomènes peuvent se produire :

- Variation de la résistance de la jauge
- Dilatation de la structure sur laquelle la jauge est collée
- Dilatation de la grille de la jauge

Il existe différentes techniques pour pallier les effets indésirables de variation de température. Cependant, cela sort du cadre de ce mémoire de maîtrise. Les jauges à base de Constantan qui est un matériau peu affecté par les changements de température seront utilisées dans le cadre de ce projet.

### 2.2.4.2 Circuit de conditionnement : Pont de Wheatstone

Les jauges de déformation sont des capteurs passifs qui nécessitent un circuit de conditionnement. La majorité des instrumentations commerciales utilisent une version plus ou moins modifiée du pont de Wheatstone, Figure 2.4. L'utilisation des jauges de déformation nécessite un circuit capable de mesurer de petites variations de résistance électrique (environ 10%). Le pont de Wheatstone est le circuit le mieux adapté pour ce genre d'applications.



V : Alimentations DC

R1, R2, R3, R4 : Résistances variables.

Em : Voltage de sortie à un instrument d'impédance Zm

Figure 2.4 Pont de Wheatstone

Le fonctionnement du pont de Wheatstone est basé sur le principe suivant : lorsque  $R_1$ ,  $R_2$ ,  $R_3$ ,  $R_4$  sont identiques, la valeur  $E_m$  est égale à zéro (pont en équilibre). Ainsi, en modifiant la valeur d'une des résistances du pont on entraîne une variation du voltage de sortie. Il existe plusieurs types de ponts de Wheatstone, dans notre cas nous allons utiliser l'architecture du quart de pont présenté à la Figure 2.5.

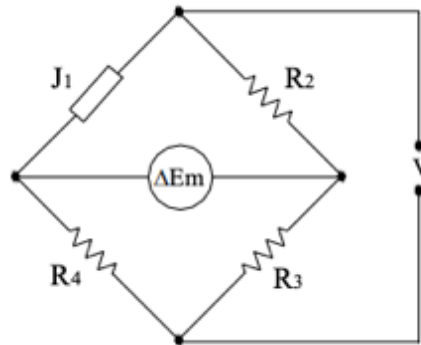


Figure 2.5 Configuration quart de pont de Wheatstone

Dans cette architecture  $J_1$  représente la jauge de déformation. La jauge  $J_1$ , et les résistances  $R_2$ ,  $R_3$ ,  $R_4$  ont les mêmes valeurs nominales de résistance théorique. En appliquant une déformation à la jauge  $J_1$ , la résistance de la jauge change ce qui entraîne un déséquilibre du pont entraînant une variation de voltage mesurée  $\Delta E_m$ .

### 2.2.4.3 Équilibrage initial du pont de Wheatstone

Dans la configuration en quart de pont, Figure 2.6, la résistance  $R_2$  a été remplacée par un potentiomètre linéaire qui permettra de faire l'équilibrage initial du pont de Wheatstone. L'équilibrage consiste à ramener  $\Delta E_m$  à zéro ( $\Delta E_m = 0$ ) lorsqu'il n'y a aucune déformation subite par la jauge  $J_1$ .

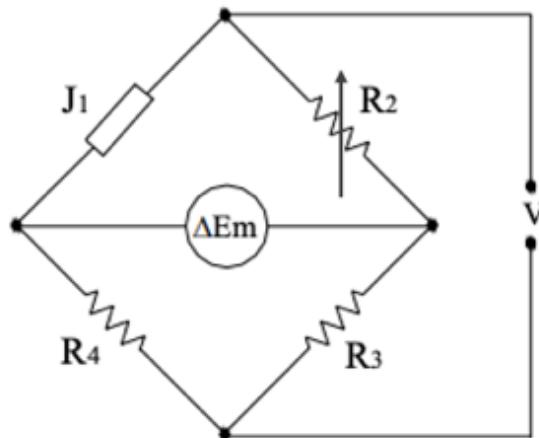


Figure 2.6 Équilibrage du quart de pont de Wheatstone avec un potentiomètre

En effet, malgré que les résistances  $J_1$ ,  $R_2$ ,  $R_3$ ,  $R_4$  ont les mêmes valeurs nominales théoriques, elles ne sont pas parfaitement identiques. Chaque résistance à sa valeur nominale qui varie généralement de  $\pm 1\%$ . Il est donc rare que le pont soit équilibré sans avoir préalablement fait des réglages.

### 2.2.5 Les centrales inertielles

Les centrales inertielles sont généralement utilisées pour estimer l'orientation d'un corps par rapport à un référentiel bien défini. Elles sont constituées généralement d'unités de mesure qui utilisent des gyroscopes, des accéléromètres et des magnétomètres pour mesurer des mouvements linéaires et angulaires. L'accéléromètre mesure les accélérations linéaires le long des 3 axes cartésiens X, Y et Z. Le gyroscope mesure la vitesse angulaire le long des 3 axes cartésiens X, Y et Z. Le magnétomètre calcule la direction dans laquelle le capteur fait face par rapport au nord magnétique. Les centrales inertielles qui disposent d'un accéléromètre à 3 axes, d'un gyroscope à 3 axes et d'un magnétomètre à 3 axes sont appelées des centrales inertielles à 9 DDL.

Cependant, chacune des unités de mesure des centrales inertielles a des problèmes distincts. L'accéléromètre a tendance à être bruyant. Le gyroscope accumule les erreurs au fil du temps générant une dérive inévitable. Les mesures du magnétomètre sont soumises à une distorsion ayant deux sources : 1. Les distorsions dites « *hard iron* » créées par des objets qui produisent un champ magnétique. 2. Les distorsions dites « *soft iron* » considérées comme des déviations ou des modifications du champ magnétique existant [55].

Plusieurs techniques de fusion de capteurs ont été développées pour générer une estimation de l'orientation, cependant, cela sort du cadre de ce mémoire et seules les lectures brutes de l'accéléromètre et du gyroscope seront utilisées. Il y a eu plusieurs implémentations de la reconnaissance d'intention de mouvements qui ont été complétées par l'utilisation de données provenant de mesures de centrales inertielles [56].

## 2.3 Les réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des systèmes informatiques composés d'éléments de traitement adaptatif interconnectés communément appelés neurones. Ces neurones utilisent un réseau de fonctions mathématiques pour comprendre et traduire une entrée de données d'une forme en une sortie souhaitée [57]. Le concept des réseaux de neurones artificiels a été inspiré par la biologie humaine et de la manière dont les neurones du cerveau humain fonctionnent ensemble pour comprendre informations provenant des sens humains [57]. Comme présenté sur la Figure 2.7, un ensemble d'entrées du neurone artificiel (synapses) avec des poids différents (dendrites) qui s'additionnent (corps cellulaire.) Après la somme pondérée, le résultat passe par une fonction d'activation, excitant ainsi le neurone (déclenchement un potentiel d'action à travers l'axone).

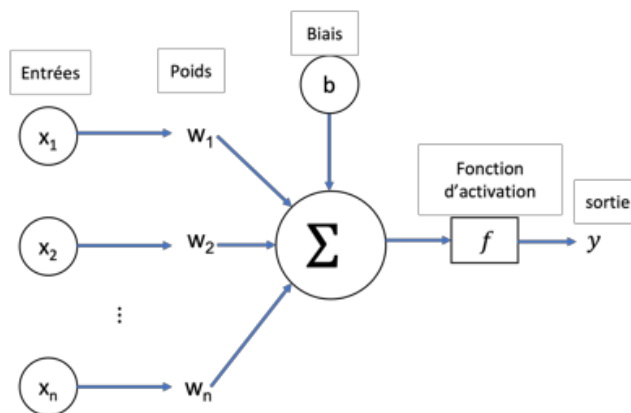


Figure 2.7 Neurone artificiel avec les entrées ( $x_1 \dots x_n$ ), les poids ( $w_1 \dots w_n$ ), le biais ( $b$ ). La sortie  $y$  du neurone est le résultat de la fonction d'activation appliquée à la somme pondérée de toutes entrées et du biais. Figure adaptée de [58].

Le premier réseau de neurones artificiels a été présenté en 1943, par Warren McCulloch, un neurophysiologiste, et Walter Pitts, un mathématicien [59]. Ces derniers ont écrit un article sur le fonctionnement des neurones et ont modélisé un simple réseau de neurones avec des circuits électriques.

En 1957, les travaux de Rosenblatt [60] ont abouti à un réseau à deux couches, le perceptron, qui était capable d'apprendre certaines classifications en ajustant les poids de connexion, mais présentait également certaines limitations. En effet, le perceptron était encore un classificateur linéaire qui pouvait apprendre que des classes linéairement séparables.

Il a fallu attendre les années 1980, où l'algorithme de rétropropagation [61] a été développé, ce qui a permis d'entraîner des perceptrons à plusieurs couches Figure 2.8.

Lorsque les neurones de la couche d'entrée sont activés par une information, cette information est traitée couche par couche jusqu'à ce que finalement la couche de sortie soit activée. Les perceptrons multicouches sont constitués de 3 parties essentielles [62]:

- **Couche d'entrée** : pour alimenter le modèle d'entrée dans le reste du réseau.
- **Couches cachées** : pour effectuer les calculs et les transformations mathématiques. Les perceptrons trouvés dans les MLP ont typiquement des fonctions d'activation non linéaires
- **Couche de sorties** : pour fournir les résultats du calcul.

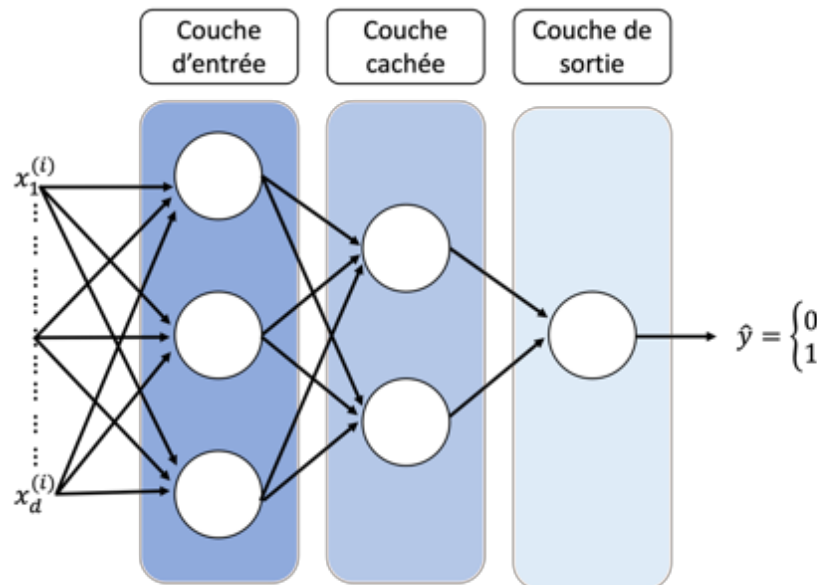


Figure 2.8 Perceptrons multicouches. Dans cette architecture la couche d'entrée comprend 3 neurones, la couche cachée deux neurones et la sortie un neurone pour un classificateur binaire.

Figure adaptée de [62].



### 2.3.1 La rétropropagation

La rétropropagation fait référence à l'algorithme de calcul du gradient des paramètres des réseaux de neurones. La méthode parcourt le réseau dans le sens inverse, de la couche de sortie à la couche d'entrée, selon la règle de la dérivation en chaîne [61]. L'algorithme enregistre toutes les variables intermédiaires (dérivées partielles) requises lors du calcul du gradient par rapport à certains paramètres.

Pour les perceptrons multicouches, les données en entrée du réseau de neurones passent séquentiellement à travers toutes les couches du réseau jusqu'à la sortie. Cette étape est appelée la passe avant, et est à l'origine du nom donné à cette architecture : « *réseau de neurones feed-forward* » (FFNN). Un exemple de calcul du gradient est donné dans l'Annexe A.

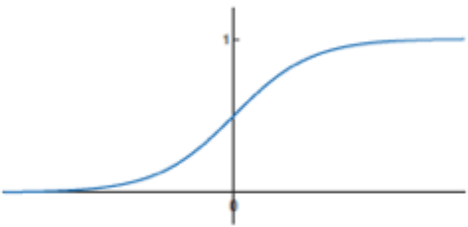
### 2.3.2 Les fonctions d'activations

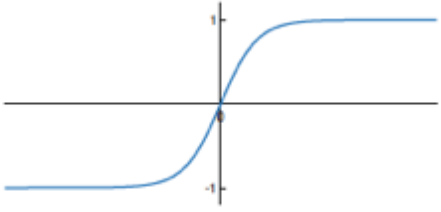
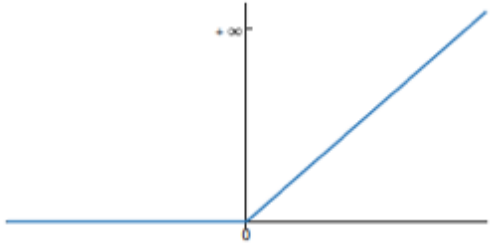
Une fonction d'activation dans un réseau de neurones peut être définie comme une fonction de transfert dont la somme pondérée de l'entrée est transformée en une sortie d'un ou de plusieurs nœuds dans une couche du réseau [58]. Le choix de la fonction d'activation dans la couche cachée est important, car elle contrôle la capacité du réseau à apprendre les caractéristiques de l'ensemble des données d'apprentissage.

Les réseaux de neurones étant entraînés par l'algorithme de rétropropagation, il est donc requis que les fonctions d'activations soient différentiables, ce qui signifie que la dérivée du premier ordre peut être calculée pour une certaine valeur d'entrée.

Le Tableau 2.3 présente les trois fonctions d'activation les plus utilisées dans la littérature.

Tableau 2.3 Formules et représentations des fonctions d'activations les plus courantes [58]

Figures	Description mathématique
	<p>Sigmoïde :</p> $g(x) = \frac{1}{1 + e^{-x}} \in (0,1)$

	<p>Tangente hyperbolique :</p> $g(x) = \tanh(x) \in (-1,1)$
	<p>Linéaire rectifiée :</p> $g(x) = \max(0, x) \in [0, \infty)$

La fonction d'activation linéaire rectifiée, ou fonction d'activation ReLU, est peut-être la fonction la plus couramment utilisée pour les couches cachées [63]. La fonction d'activation ReLU est populaire pour les architectures MLP et CNN. Cependant, les réseaux de neurones récurrents utilisent couramment les fonctions d'activations Tangente hyperbolique ou sigmoïdes.

### 2.3.3 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs ou encore CNN, sont un type spécialisé de réseaux de neurones qui sont utilisés pour le traitement des données qui ont une topologie similaire à celle d'une grille [58]. Cette topologie est retrouvée dans les données de séries chronologiques, qui peuvent être considérées comme une grille 1D prenant des échantillons à intervalles de temps réguliers, et les données d'image, qui peuvent être considérées comme une grille 2D de pixels. Les progrès dans le domaine de la vision par ordinateur avec l'apprentissage profond ont été construits et perfectionnés principalement sur la base des réseaux de neurones convolutifs.

L'avantage de l'utilisation des CNN est qu'ils peuvent apprendre directement une représentation interne des données de la série chronologique sans le besoin d'extracteurs de caractéristiques conçus manuellement par l'homme [58].

Les réseaux de neurones convolutifs sont généralement composés de 3 couches [64]:

- **Couche de convolution :** La couche de convolution (CONV) utilise des filtres qui effectuent des opérations de convolution sur les données en entrée par rapport à ses dimensions et les passe à la couche suivante. Les poids de la couche précédente sont donc reliés aux poids des couches suivantes. Les hyperparamètres incluent la taille du filtre  $f$  et le pas  $s$ . La sortie résultante est appelée carte de caractéristiques ou carte d'activations.
- **Couche de Pooling :** La couche de mise en commun (POOL) est une opération de sous-échantillonnage, généralement appliquée après une couche de convolution, qui effectue une certaine invariance spatiale. En particulier, les « *pooling* » max et moyen sont des types particuliers de regroupement où les valeurs maximale et moyenne sont respectivement prises.
- **Couche connectée :** La couche entièrement connectée (FC) fonctionne sur une entrée vectorielle où chaque entrée est connectée à tous les neurones. Si elles sont présentes, les couches FC se trouvent généralement vers la fin des architectures CNN et peuvent être utilisées pour optimiser des objectifs tels que les scores de classe.

Cependant, dans ce mémoire les réseaux de convolutions 1D présentés à la Figure 2.9 seront présentés, car les données utilisées (signaux de jauges de déformation et signaux provenant de centrales inertielles) pour l'application souhaitée consistent en un ensemble de vecteurs d'entrées unidimensionnels.

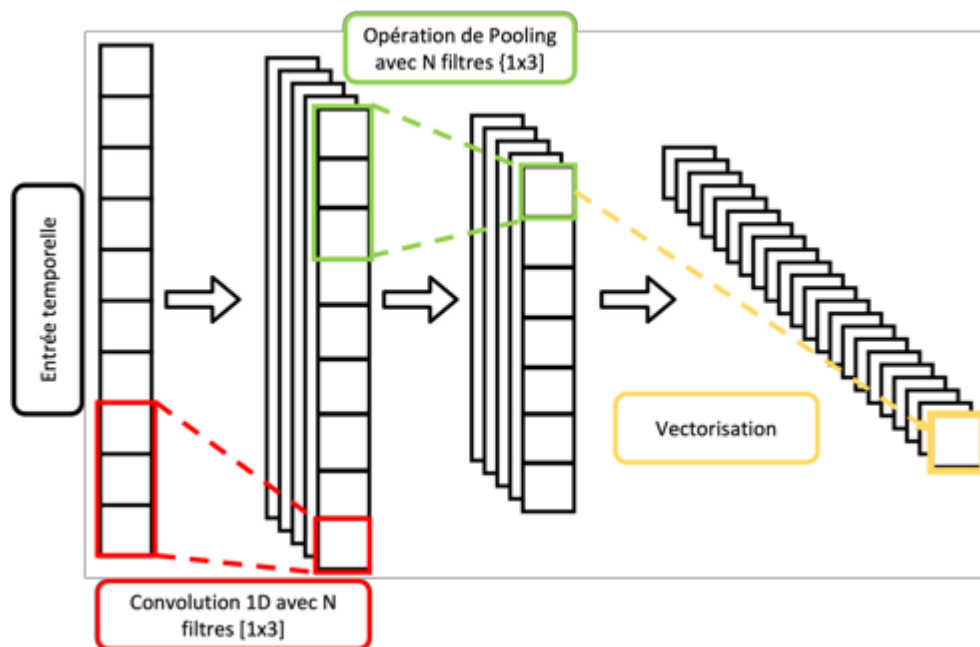


Figure 2.9 Représentation graphique d'un réseau de neurones convolutifs 1D. L'entrée est alimentée à travers une série de couches de convolution suivies de couches de « Pooling », pour former des structures arbitrairement profondes.  $N$  représente le nombre de filtres dans chaque couche. L'opération finale consiste à une vectorisation permettant d'alimenter en général un perceptron multicouche.

L'exemple présenté à la Figure 2.10 présente le principe de fonctionnement des réseaux de neurones convolutifs.

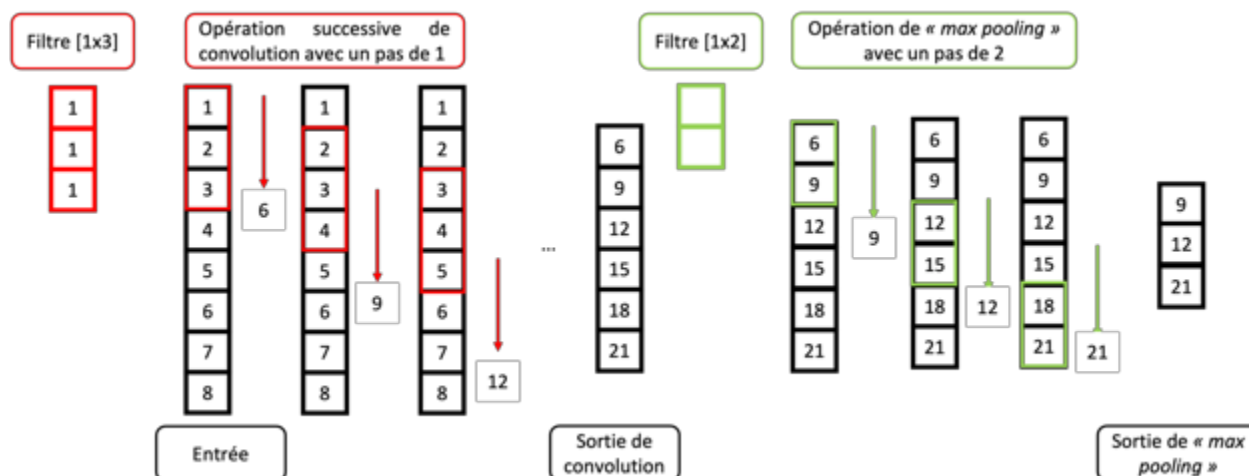


Figure 2.10 Exemple d'opérations de convolution et de « max pooling » 1D. Un filtre est appliqué sur le vecteur d'entrée avec un pas de 1. Cela signifie que le filtre se déplace sur tout le vecteur d'entrée avec un pas de  $s = 1$ . Ensuite, une opération vectorielle est faite entre les

éléments du filtre et les éléments du vecteur d'entrée : ici une somme pondérée. Après l'opération de convolution, l'opération de « max pooling » choisit la valeur maximale entre deux valeurs à partir d'un filtre qui parcourt tous les éléments avec un pas de 2.

### 2.3.4 Les réseaux de neurones récurrents

Les réseaux de neurones récurrents, ou RNN sont une famille de réseaux de neurones capable de traiter des données séquentielles. Les RNN sont capables de prendre en compte le contexte en introduisant des variables d'état pour stocker les informations passées, ainsi que les entrées actuelles, afin de déterminer les sorties [58]. Un réseau de neurones récurrent peut être considéré comme plusieurs copies du même réseau, comme illustré à la voir Figure 2.11, transmettant l'information contextuelle apprise pour mettre à jour les poids correspondants qui représentent ses états internes.

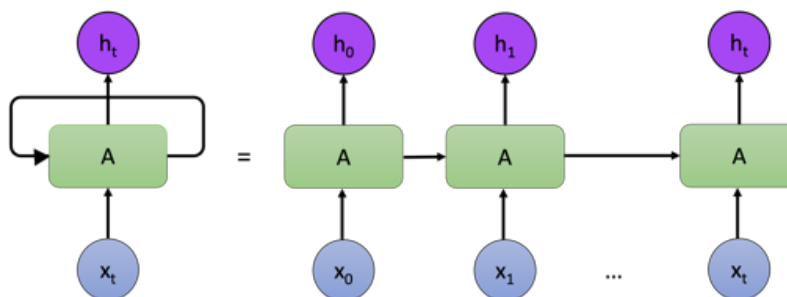


Figure 2.11 Illustration d'un réseau de neurones récurrent (RNN). À gauche, l'architecture du réseau est représentée avec les flèches représentant les connexions récurrentes. Sur le côté droit, ces connexions sont représentées dans l'espace, où chaque pas de temps forme une nouvelle couche. Figure adaptée de [65].

Un RNN peut soit être utilisé pour classer une séquence entière en une seule classe, pour générer une nouvelle séquence en sortie, par ex. traduire un texte, mais aussi pour créer une séquence à partir d'une seule entrée comme pour le sous-titrage automatique d'images par l'identification de son contenu [66]. Dans ce mémoire, la première architecture montrée, Figure 2.12, qui consiste à classer une séquence entière en une seule classe sera utilisée. Notre entrée sera des données temporelles provenant des capteurs et la sortie sera une classe de mouvement. Les classes de mouvement sont décrites à la section 5.3.3.

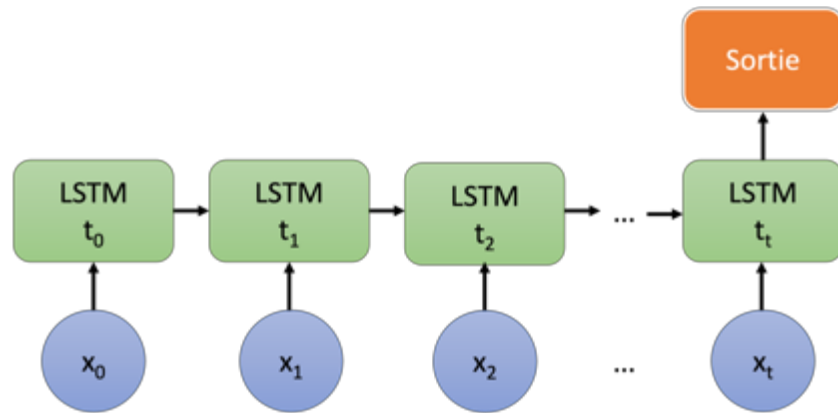


Figure 2.12 LSTM architecture « many-to-one ». Analyse de séquence de mouvement : à chaque pas de temps la séquence de mouvement est évaluée par une cellule LSTM. Le résultat de la couche est associé aux classes possibles, et est calculé après chaque pas. Figure adaptée de [66].

Cependant, les RNN standard ont de la difficulté à apprendre des dépendances contextuelles à long terme. En effet, les RNN sont entraînés en utilisant une version adaptée de l'algorithme de rétropropagation: la rétropropagation à travers le temps. À mesure que les calculs de gradient remontent dans le temps, il y a une tendance à avoir des valeurs de plus en plus élevées ou basses du gradient, ce qui conduit à une explosion ou une dégradation du gradient [66].

La mémoire à long court terme (LSTM) est une architecture RNN qui résout le problème de l'explosion et de la dégradation du gradient. La couche cachée LSTM est composée de blocs de mémoire Figure 2.13, qui sont des sous-réseaux autoconnectés contenant plusieurs cellules internes. Grâce à des portes multiplicatives, la cellule est capable de stocker et d'accéder à des informations sur une longue période, Figure 2.14. En d'autres termes, LSTM transporte les données de diverses étapes à travers toutes les étapes et chaque cellule est capable d'inclure et de supprimer des informations de ces données tout en traitant une entrée séquentielle [67].

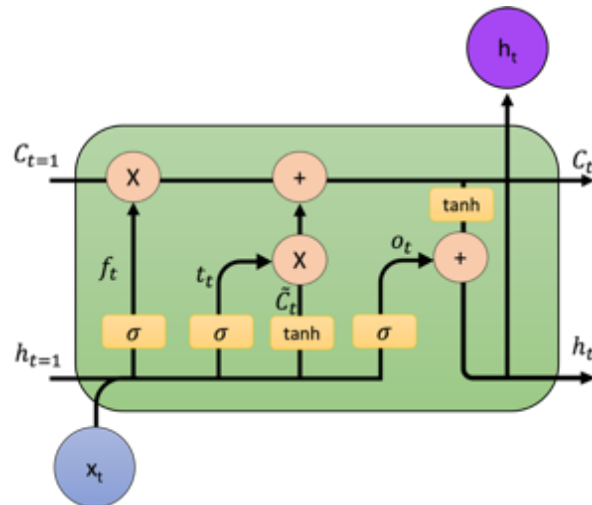


Figure 2.13 Cellule LSTM.  $C_{t-1}$  Représente l'étape précédente.  $h_{t-1}$  L'état de la sortie précédente,  $x_t$  représente l'échantillon de la séquence au temps  $t$ ,  $C_t$  représente l'état de la cellule mis à jour et  $h_t$  le résultat. Nous avons aussi représenté les portes : la porte d'oublie  $f_t$ , la porte d'entrée  $i_t$  et la porte de sortie  $o_t$  [65].

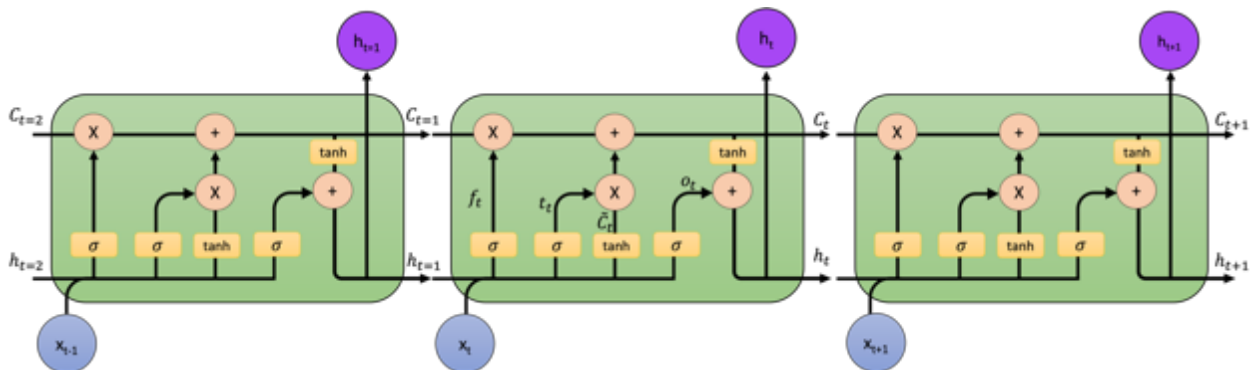


Figure 2.14 Cellules LSTM à travers le temps. Chaque étape de cellule reçoit un échantillon de la séquence  $x$  de la couche d'entrée et envoie un état de cellule mis à jour et la valeur de sortie  $h$  à l'étape suivante. Chaque pas de temps envoie également la valeur  $h$  à la couche de sortie [65].

Les étapes de calculs effectués par les cellules LSTM à travers le temps sont présentées. Les valeurs  $W$  (poids) et  $b$  (biais) représentent les paramètres du réseau. La cellule LSTM comprend une porte d'oublie  $f_t$  pour oublier les informations qui sont plus nécessaires [65], [66] :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.2)$$

Une porte d'entrée [65], [66] pour enregistrer l'information (Calculé comme  $C_t$  provenant de l'étape  $x_t$  et l'étape précédente  $h_{t-1}$  qui sera nécessaire :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

Une porte de sortie  $o_t$  pour contrôler la sortie [65], [66] :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.4)$$

Les nouvelles valeurs sont mises à jour suivant ces équations [65], [66] :

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.5)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.6)$$

$$h_t = o_t * \tanh(C_t) \quad (2.7)$$

## 2.4 L'apprentissage

### 2.4.1 Les fonctions coût

Les réseaux de neurones sont entraînés à l'aide d'un processus d'optimisation qui nécessite de calculer l'erreur du modèle par une fonction coût [64]. En général, ces fonctions peuvent être classées en deux catégories selon le problème d'apprentissage : régression ou classification [58].

Les méthodes de régression sont utilisées pour prédire valeurs réelles. Les fonctions coût utilisées pour ces problèmes sont généralement basées sur une mesure de la distance entre les prédictions et les observations réelles. Une des fonctions coût les plus utilisées pour les problèmes de régression est l'erreur moyenne quadratique (MSE). Elle est évaluée selon la formule la suivante [68] :

$$MSE = \frac{\sum_i^n (y_i - \hat{y}_i)^2}{n} \quad (2.8)$$

Où  $y_i$  représente la valeur cible et  $\hat{y}_i$  représente la valeur prédite pour l'échantillon  $i$  et  $n$  le nombre total d'échantillons.



En classification, nous essayons de prédire une valeur d'un ensemble de valeurs catégoriques finies. Les fonctions coût pour ces problèmes sont généralement basées sur une mesure d'entropie. Deux fonctions sont généralement utilisées : entropie croisée binaire ou catégorielle.

La première est utilisée pour des problèmes de classification entre une ou deux classes. La seconde est utilisée pour des problèmes de classification où le nombre de classes est supérieur à 2.

La fonction binaire de coût d'entropie ( $\mathcal{L}$ ) croisée est présentée, avec  $\hat{y}$  représentant la sortie du réseau et  $y$  la valeur cible [68].

$$\mathcal{L}(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (2.9)$$

où  $\hat{y}$  peut-être vu comme la probabilité que la sortie soit 1, et  $(1 - \hat{y})$  la probabilité que la sortie soit 0.

Les problèmes où le nombre de classes est supérieur à deux sont appelés des problèmes multiclasse. Pour les problèmes multiclasse, il est important d'encoder les sorties sous une forme vectorielle binaire dont le nombre d'éléments correspond au nombre de classes  $k$ . Par exemple, pour un problème de 3 classes, l'encodage pour élément appartenant à la deuxième classe serait le suivant  $y_2 = \{0, 1, 0\}$ . Cette notation est communément appelée la notation « *one hot* ». La fonction coût ( $\mathcal{L}$ ) pour un problème de classification multi-classe peut être définie comme suit [68]:

$$\mathcal{L}(\hat{y}, y) = - \sum_{k=1}^K y_k \log(\hat{y}_k) \quad (2.10)$$

où  $\hat{y}_k$  est le  $k^{ième}$  nœud de sortie et  $\log$  indique le logarithme népérien. La sortie représente une distribution de probabilité qu'un exemple de donnée appartienne à chacune des classes.

Après avoir déterminé notre fonction coût  $\mathcal{L}$ , le problème d'optimisation peut être résumé comme étant la recherche des paramètres permettant de minimiser l'erreur calculée par la fonction coût par rapport à l'ensemble de données est notée  $J$  et est calculée selon la formule qui suit [68].

$$J = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) \quad (2.11)$$

où  $\mathcal{L}(\hat{y}^{(i)}, y^{(i)})$  noté  $\mathcal{L}^{(i)}$  représente la valeur scalaire de la fonction coût calculée pour un exemple  $i$  de l'ensemble de données. Cette fonction est minimisée par rapport à tous les paramètres du réseau et moyennée pour tous les  $n$  exemples. C'est ce qu'on appelle la descente de gradient par lots parce que l'ensemble du lot d'entraînement est utilisé pour calculer la fonction de coût.

## 2.4.2 Les méthodes d'optimisation

La descente de gradient est de loin le moyen le plus courant et le plus établi d'optimiser les fonctions coût des réseaux de neurones. La descente de gradient est un algorithme d'optimisation qui trouve l'ensemble des paramètres d'une fonction coût permettant d'atteindre la valeur minimale cette fonction [58].

Il existe trois variantes de descente de gradient qui diffèrent par la quantité de données que nous utilisons pour calculer le gradient de la fonction objectif. En fonction de la quantité de données, nous faisons un compromis entre la précision de la mise à jour des paramètres et le temps nécessaire pour effectuer une mise à jour.

### 2.4.2.1 Descente de Gradient par lot « *batch gradient descent* »

La descente de gradient par lot calcule le gradient de la fonction coût par rapport à aux paramètres de l'ensemble des données d'entraînement. Il faut donc calculer les gradients de l'ensemble de données d'entraînement pour effectuer une seule mise à jour des paramètres. Ainsi, la Descente de Gradient par lots peut être très lente et est insolvable pour les ensembles de données qui ne tiennent pas en mémoire.

### 2.4.2.2 Descente de Gradient stochastique « *stochastic gradient descent* »

La Descente de Gradient Stochastique (SGD) en revanche effectue une mise à jour des paramètres pour chaque exemple d'entraînement. De ce fait, elle est généralement plus rapide que la descente de gradient par lot. Cependant, la descente de gradient stochastique effectue des mises à jour fréquentes avec une variance élevée ce qui entraîne une forte fluctuation de la fonction objective.

### 2.4.2.3 Descente de Gradient par mini-lot « *mini-batch gradient descent* »

La Descente de Gradient par mini-lot prend enfin le meilleur des deux algorithmes précédents et effectue une mise à jour pour chaque mini-lot. Ainsi, cet algorithme permet de:

1. Réduire la variance des mises à jour des paramètres, ce qui peut conduire à une convergence plus stable.
2. Utiliser des opérations matricielles hautement optimisées pour accélérer l'apprentissage des réseaux de neurones profonds.

Dans le cas de la Descente de Gradient par lot, la fonction coût  $J_{mb}$  est calculée selon la formule suivante :

$$J_{mb} = \frac{1}{B} \sum_{i=1}^B \mathcal{L}^{(i)} \quad (2.12)$$

où  $B$  représente le nombre d'exemples dans le lot et  $mb$  le terme anglais pour « *mini-batch* ».  $\mathcal{L}^{(i)}$  représente la valeur scalaire de la fonction coût calculée pour un exemple  $i$  de l'ensemble de données

Les méthodes classiques de Descente de Gradient ne garantissent pas nécessairement une convergence vers un minimum global. Des méthodes plus avancées telles que RMSprop [69] et Adagrad [70] et Adam ont été proposés [71].

#### 2.4.2.4 L'algorithme d'optimisation Adam

L'estimation du moment adaptatif (Adam) [71] est une autre méthode qui calcule les taux d'apprentissage adaptatif pour chaque paramètre. Cette méthode s'est avérée être un meilleur optimiseur car il tente de combiner les avantages des deux algorithmes RMSprop et Adagrad.

Pour cela, les estimations  $m_t$  et  $v_t$  des premiers et deuxièmes moments bruts des gradients noté  $g_t$  sont calculés conformément aux équations [71]:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.13)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.14)$$

Ces équations consistent à effectuer des moyennes mobiles exponentielles du Gradient,  $m_t$ , et du gradient au carré,  $v_t$ . Le premier est une estimation du 1er moment (la moyenne) et le second est le 2e moment brut (la variance décentrée). Les hyperparamètres  $\beta_1, \beta_2 \in [0,1)$  Contrôlent les taux de décroissance exponentielle de chaque moyenne mobile. Cependant, comme ces vecteurs sont

initialisés à zéro, ils sont biaisés vers zéro. L'algorithme utilise des termes de correction de biais, présentés dans les équations [71] :

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.15)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.16)$$

Ils les utilisent ensuite pour mettre à jour les paramètres, ce qui donne la règle de mise à jour de l'optimiseur Adam [71]:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (2.17)$$

Les auteurs [71] proposent les valeurs par défaut suivantes  $\alpha = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  et  $\epsilon = 10^{-8}$ , ils montrent empiriquement qu'Adam se compare favorablement à d'autres algorithmes de méthode d'apprentissage adaptatif.

## 2.5 Les méthodes de régularisation

La différence entre l'erreur faite sur l'ensemble d'entraînement et l'erreur faite sur l'ensemble de tests peut être considérée comme l'erreur de généralisation. Le but des méthodes de régularisation est de réduire cette erreur et de ce fait d'améliorer la capacité de généralisation des réseaux neurones. Il existe plusieurs stratégies pour le réduire l'erreur de généralisation [58], [72].

### 2.5.1.1 Régularisation L2

La régularisation L2 est une fonction dite de « perte de poids ». On la retrouve dans la littérature sous les appellations : régularisation Ridge, ou régularisation Tikhonov. Elle consiste à ajouter un terme de pénalité  $\lambda$  à la fonction coût  $J_{L2}$  proportionnellement à la taille des poids dans le modèle [68].

$$J_{L2} = J + \frac{\lambda}{2} \|W\|^2 \quad (2.18)$$

$$J_{L2} = J + \frac{\lambda}{2} \sum_{ij} |W_{ij}|^2 \quad (2.19)$$

$$J_{L2} = J + \frac{\lambda}{2} W^T W \quad (2.20)$$

La fonction de coût  $J_{L2}$  est minimisée par rapport à tous les poids  $W$  et biais du réseau.  $W_{ij}$  représente les éléments de la matrice de poids  $W$ . En raison de la pénalité qui augmentera le coût si les poids sont trop grands, cette méthode de régularisation forcera les poids à être petits. Le paramètre  $\lambda$  de régularisation est donc un autre hyperparamètre à régler.

### 2.5.1.2 « Dropout »

C'est une méthode de régularisation nécessitant une faible puissance de calcul et efficace pour réduire l'erreur sur l'apprentissage et de généralisation dans les réseaux de neurones profonds. Un seul modèle peut être utilisé pour simuler un grand nombre d'architectures de réseau de neurones différentes en supprimant de manière aléatoire (paramètre de probabilité) des neurones pendant l'entraînement [73].

### 2.5.1.3 Arrêt précoce « Early Stopping »

Une pratique conventionnelle en apprentissage machine est de diviser l'ensemble de données en 3 ensembles distincts : l'entraînement, la validation et le test. Le but étant de pouvoir utiliser la plus grande partie des données pour entraîner le modèle et de garder un plus petit échantillon pour suivre les performances du réseau sur de nouvelles données, par exemple sur l'ensemble de validation.

Il est important d'entraîner le réseau suffisamment longtemps pour qu'il soit capable d'apprendre la relation entre les entrées et les sorties, mais de ne pas entraîner le modèle trop longtemps au point d'apprendre des caractéristiques spécifiques aux données d'entraînement seulement.

Suivant cette logique, l'arrêt précoce « *early stopping* » consiste à entraîner l'algorithme sur l'ensemble des données d'entraînement, mais à arrêter l'entraînement au moment où les performances sur l'ensemble de données de validation commencent à se dégrader. Cette approche est simple, efficace et largement utilisée pour entraîner les réseaux de neurones profonds. Il est facile d'utiliser l'arrêt précoce sans endommager la dynamique d'apprentissage. Ceci est à l'opposé

du « *dropout* » ou la régularisation L2. En particulier pour le dernier, il faut faire attention à ne pas utiliser des valeurs trop grandes du paramètre  $\lambda$ . En effet, un tel cas pourrait conduire le gradient à être bloqué dans un minimum local et l'empêcher d'atteindre le minimum global.

#### 2.5.1.4 « *Data augmentation* »

Pour les modèles d'apprentissages profonds, une quantité insuffisante de données d'apprentissage peut conduire le modèle à apprendre des caractéristiques spécifiques aux données d'entraînement, ce qui conduit le modèle à avoir une mauvaise capacité de généralisation. Le nombre de données d'apprentissage est proportionnel aux nombres de paramètres que peut apprendre le modèle.

Une technique utilisée pour résoudre le problème du nombre insuffisant de données consiste à appliquer différentes transformations sur les données disponibles pour synthétiser de nouvelles données. Cette approche consistant à synthétiser de nouvelles données à partir des données disponibles est appelée « *augmentation de données* ». Pour les images, l'augmentation de données peut se faire en appliquant des transformations géométriques (rotation, mise à l'échelle, translation, etc.) aux données [74]. Pour des données temporelles, de nouveaux signaux peuvent être générés en rajoutant du bruit gaussien dans le signal de base [23].

## 2.6 Mesure de la performance de l'algorithme de classification

L'évaluation des algorithmes d'apprentissage machine est une partie essentielle de tout projet. Un mauvais choix des métriques pour évaluer son modèle peut vous induire en erreur sur les performances attendues de votre modèle [75]. Ici, le mot « performance » est utilisé pour désigner la capacité du modèle à attribuer à un mouvement la classe de mouvement correspondante.

La précision de la classification est la métrique la plus utilisée pour mesurer les performances des modèles. Elle est calculée selon la formule suivante [76]:

$$\text{Précision de la classification} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}} \quad (2.21)$$

Cette mesure de performance fonctionne bien lorsque les données sont balancées : c'est-à-dire que les nombres de données appartenant à chaque classe sont égaux. Dans le cadre de cette recherche, les classes sont balancées. Prenons l'exemple d'une classification binaire avec des données non balancées, où 95% des données d'entraînement sont de la classe A et 5% de la classe B. Ainsi, le

modèle peut facilement obtenir une précision de classification de 95% en prédisant simplement que chaque échantillon d'entraînement appartient à la classe A.

Ainsi, on fait souvent recours à une matrice de confusion, Tableau 2.4 qui nous permet de décrire la performance complète de notre modèle [76] .

Tableau 2.4 Matrice de confusion pour une classification binaire

Matrice de confusion		Classes prédites	
		Positif (P)	Négatif (N)
Classes cibles	Positif (P)	VP	FN
	Négatif (N)	FP	VN

Ce tableau révèle 4 importants termes :

- Vrai Positif (VP) : L'algorithme a prédit OUI et la sortie réelle est également OUI.
- Vrai Négatif (VN): L'algorithme a prédit NON et la sortie réelle est également NON.
- Faux Positif (FP): L'algorithme a prédit OUI et la sortie réelle est également NON.
- Faux négatif (FN): L'algorithme a prédit NON et la sortie réelle est également OUI

$$\text{Précision de classification} = \frac{VP + VN}{VP + FP + FN + FP} \quad (2.22)$$

Une mesure plus avancée est le score F1 qui est la moyenne harmonique entre la précision et le rappel. La plage pour le score F1 est [0, 1]. Il vous indique la précision de votre classificateur (combien d'instances il classe correctement), ainsi que sa robustesse (il ne manque pas un nombre important d'instances). Le score F1 est calculé selon l'équation qui suit [76]

$$F1 = 2 \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \quad (2.23)$$

Cependant, le score F1 présente certains inconvénients, en particulier lorsque les données sont non balancées. Une mesure qui est plus robuste face à des classes non balancées est le Coefficient de Corrélacion de Matthew (MCC) qui est calculé comme suit [75].

$$MCC = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}} \quad (2.24)$$

Le MCC sera 1 pour un classificateur parfait et 0 pour un classificateur aléatoire. Un classificateur inverse aura un MCC de -1 (dans ce cas il faut juste inverser les prédictions pour obtenir le classificateur parfait). Ce score peut être appliqué pour les problèmes de classification multiclassé cependant, toutes les classes doivent être présentes dans la sortie de classification, sinon la mesure devient indéfinie [75].

Pour résumer, la performance du modèle de classification est évaluée par les métriques suivantes :

- La précision de classification qui sera présentée dans une matrice de confusion
- Le MCC du modèle

La précision sera mesurée pour chacune des 10 classes de mouvement et globalement pour les différents modèles. Le MCC sera mesuré pour les différents modèles pour valider la précision de classification. En effet, un modèle ayant une précision de classification élevée devrait avoir un MCC proche de 1.

Le Tableau 2.5 fait un résumé des critères de conception.



Tableau 2.5 Résumé des critères de conception

No	Texte de référence	Fonction	Critère
1.0	Développer un système de mesure permettant de déterminer les intentions de mouvement du membre supérieur.		
1.1	Utiliser des jauges de déformations	Mesurer des déformations multiaxiales de la peau	Enregistrer les déformations selon 2 axes.
		Établir un contact conforme avec la peau pendant l'utilisation	Ne pas se décoller de la peau durant 24h.
1.2	Utiliser des centrales inertielles	Acquérir les accélérations linéaires	Enregistrer les accélérations linéaires du membre selon 3 axes cartésiens (ax, ay, az) à une fréquence de 100Hz
		Acquérir les vitesses angulaires	Enregistrer les vitesses angulaires du membre selon 3 axes cartésiens (ax, ay, az) à une fréquence de 100Hz
1.3	Faire un circuit d'acquisition	Acquérir et traiter les données provenant des différents capteurs.	Être capable d'acquérir les signaux provenant des jauges de déformation et les signaux provenant des centrales inertielles à une fréquence d'acquisition de 20Hz-200Hz
1.4	Avoir une architecture de communication sans-fil	Permettre aux capteurs de communiquer sans-fil et d'enregistrer les données sur l'ordinateur	Avoir une fréquence de transmission de 100 Hz-250Hz
1.5	Avoir un système portable	Permettre une utilisation dans les activités quotidiennes	<ul style="list-style-type: none"> <li>• Avoir une masse &lt; 500g / capteur</li> <li>• Avoir une taille &lt; 15cm x15 cm x 5 cm</li> </ul>
1.6	Avoir un système robuste	Permettre d'utiliser le système de façon répéter sans que les jauges de déformation ou les connexions se brisent	Ne pas briser à la suite des mouvements de l'utilisateur

2.0	Développer un algorithme intelligent permettant de détecter des intentions de mouvement.		
2.1	Utiliser un algorithme d'apprentissage profond	Permettre de prendre en compte la nature temporelle des données	<ul style="list-style-type: none"> <li>• Précision de classification</li> <li>• MCC du modèle</li> </ul>
		Permettre d'extraire les caractéristiques importantes des signaux	<ul style="list-style-type: none"> <li>• Précision de classification</li> <li>• MCC du modèle</li> </ul>
2.2	Reconnaitre l'intention de mouvement	Être capable de distinguer différents mouvements du membre supérieur	<ul style="list-style-type: none"> <li>• Précision de classification</li> <li>• MCC du modèle</li> </ul>
2.3	Avoir un algorithme facile à implémenter sur un microcontrôleur	Être embarqué	Pouvoir être implémenter sur un microcontrôleur ESP32 HUZZAH 32

\*MCC : Coefficient de Corrélation de Matthew

## CHAPITRE 3 JUSTIFICATION DU PROJET DE RECHERCHE

### 3.1 Résumé de la problématique

Les différentes problématiques liées à l'utilisation des capteurs sEMG ont motivé la recherche d'alternatives. Cependant les solutions proposées dans la littérature présentent deux désavantages importants :

- Les techniques de fabrication sont hautement spécialisées ce qui rend difficile la production de masse.
- L'utilisation des matériaux flexibles est peu commune et l'intégration de la miniaturisation de l'électronique est difficile.

Les alternatives proposées pour la majorité, sont utilisés dans un environnement contrôlé et non dans des applications réelles. Dans cette optique, ce travail de maîtrise met à l'avant la possibilité d'utiliser des jauges de déformation pour développer un capteur capable de détecter les contractions musculaires au lieu des sEMG.

À notre connaissance, l'état de l'art révèle deux problèmes majeurs :

1. Les problèmes courants des sEMG ont été résolus par une première preuve de concept de bracelet utilisant la déformation. Mais celui-ci avait encore des limites.
2. La combinaison de jauges de déformation et d'IMU n'a jamais été étudiée pour détecter l'intention de mouvement humain.

### 3.2 Objectif général

Dans ce contexte, l'objectif de cette étude est de développer un système portable pour identifier les intentions de mouvement en combinant des jauges de déformation et des centrales inertielles.

La principale hypothèse de recherche est la suivante :

- La combinaison entre les jauges de déformation et les IMUS améliorera la capacité de détecter l'intention de mouvement des membres supérieurs.

### 3.3 Objectifs spécifiques

L'objectif général de cette étude peut se diviser en trois objectifs spécifiques (OS).

- OS1 : Développer un capteur à base de jauges de déformation pour mesurer les contractions musculaires.
- OS2 : Développer un système de centrale inertielle pour mesurer les accélérations linéaires et les vitesses angulaires de l'avant-bras et du poignet.
- OS3: Développer un algorithme pour détecter des intentions de mouvement.
- OS4: Valider le fonctionnement du système intégré.

## CHAPITRE 4 MÉTHODOLOGIE

La méthodologie présentée dans cette section est complémentaire à celle présentée dans le chapitre 5. Cette section a pour objectif de présenter de l'information additionnelle pertinente à l'étude. Aussi, une méthodologie supplémentaire sur l'implémentation d'un algorithme d'apprentissage profond pour la classification des mouvements du membre supérieur y est également présentée.

### 4.1 Conception du capteur utilisant une centrale inertielle

Cette section présente le système développé pour suivre en temps réel l'accélération linéaire et la vitesse angulaire du membre supérieur. Le système développé utilise un capteur commercial à faible coût intégrant un accéléromètre, un gyroscope et un magnétomètre.

#### 4.1.1 Choix de la centrale inertielle

Les IMUs permettent d'enregistrer des données cinématiques. Trois IMUs, Figure 4.1, ont été testées dans le cadre du projet. À savoir le modèle Fermion de 10 DDL de DfRobot, et les deux modèles à 9DDL de Adafruit que sont le BNO055 et le NXP.

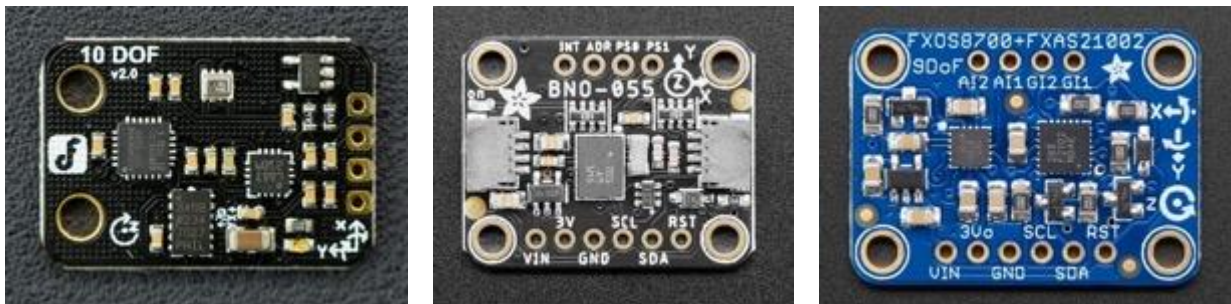


Figure 4.1 Les différentes centrales inertielles testées dans le cadre du projet. À gauche, la centrale inertielle à 10 DDL de DfRobot. Au milieu, la centrale inertielle BNO055 à 9 DDL de Adafruit. À droite, la centrale inertielle Fermion de 9 DDL de Adafruit.

Les tests effectués avec ce capteur ont permis d'en sélectionner un. Les critères qui ont été utilisés sont les suivants :

1. La tolérance au bruit de l'accéléromètre
2. La dérive du gyroscope
3. La disponibilité de bibliothèques permettant de faciliter l'implémentation et l'acquisition des données.

4. La disponibilité de bibliothèques permettant de calibrer le capteur.

Parmi ces capteurs, le capteur qui a été retenu est le BNO055. En effet, la centrale inertielle BNO055 de 9DOF est basée sur le capteur intelligent d'orientation absolue de Bosh. Il intègre un accéléromètre triaxial 14 bits, un gyroscope triaxial 16 bits avec une plage de  $\pm 2000$  degrés par seconde, un géomagnétique triaxial et un microcontrôleur cortex M0+ 32 bits exécutant la fusion de capteurs dans une seule composante. le capteur intelligent d'orientation absolue de Bosh intègre un accéléromètre, un magnétomètre et un gyroscope MEMS, et en les plaçant sur une seule matrice avec un processeur ARM Cortex-M0 à grande vitesse, il arrive à traiter toutes les données de l'accéléromètre, du gyroscope, du magnétomètre puis extraire la fusion du capteur en respectant les exigences d'un fonctionnement en temps réel et à produire des données qui peuvent être utilisées par un non-expert du traitement de données cinématiques [77].

#### **4.1.2 Calibration de la centrale inertielle**

Pour s'assurer que les données provenant des différents capteurs (accéléromètre, gyroscope, magnétomètre) du BNO055 soient correctes, il est primordial d'effectuer une calibration de ces capteurs. Pour ce faire, le guide de calibration fournie par MathWorks [78] a été utilisé.

##### **Calibration du Magnétomètre**

Avant de procéder à la calibration, il faut s'assurer que le capteur est éloigné de toute interférence magnétique. Ensuite, la procédure est la suivante :

1. Tenir le capteur parallèle au sol et déplacez-le selon un schéma en forme de 8, Figure 4.2.
2. Utiliser la fonction « *readCalibrationStatus* » de la bibliothèque Open-Source fournie par Adafruit pour lire l'état de calibration du capteur [79].
3. Répéter ce processus jusqu'à ce que la valeur de calibration du magnétomètre soit « Full ».

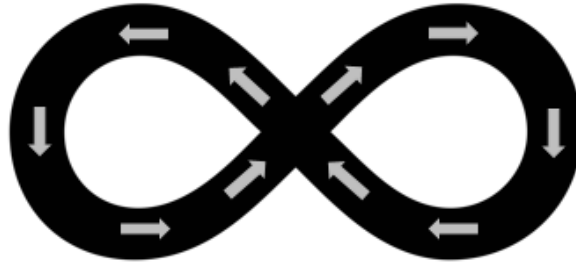


Figure 4.2 Calibration de magnétomètre. Figure tirée de [78]

### Calibration de l'accéléromètre

Pour calibrer complètement l'accéléromètre du capteur BNO055 :

1. Placer successivement le capteur BNO055 dans les six positions stables, Figure 4.3. pendant quelques secondes chacune.
2. Utiliser la fonction « *readCalibrationStatus* » de la librairie Open-Source fournie par Adafruit pour lire l'état de calibration du capteur [79].
3. Répéter ce processus jusqu'à ce que la valeur de calibration de l'accéléromètre soit « Full ».

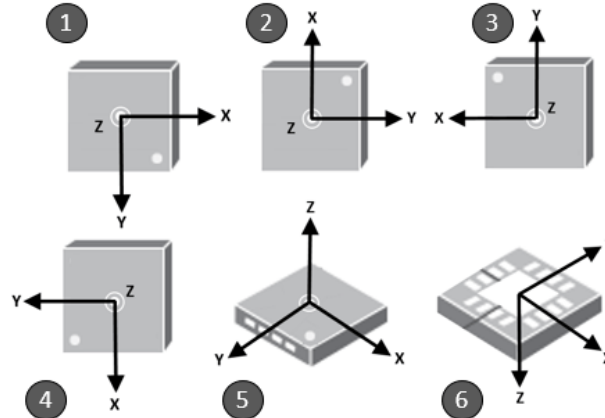


Figure 4.3 Calibration de l'accéléromètre. Six positions stables de références. Figure tirée de [78]

### Calibration du gyroscope

Pour calibrer complètement le gyroscope du capteur BNO055 :

1. Placer le capteur dans n'importe quelle position stable pendant quelques secondes.
2. Utiliser la fonction « *readCalibrationStatus* » de la librairie Open-Source fournie par Adafruit pour lire l'état de calibration du capteur [79].

3. Répéter ce processus jusqu'à ce que la valeur de calibration du gyroscope soit « Full ».

Lorsque le processus de calibration est terminé, le code fournit des valeurs de décalage. Ces valeurs de décalages sont ensuite appliquées à chacun des axes du capteur. Cette méthode permet de conserver les valeurs de calibrations et nous évite ainsi de calibrer le capteur avant chaque utilisation.

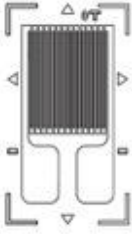
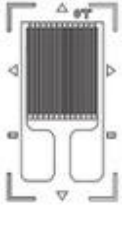
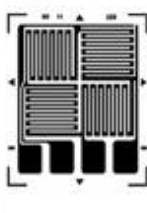
## 4.2 Conception du capteur utilisant des jauges de déformation

Cette section présente de l'information additionnelle sur la conception, non incluse dans l'article, du capteur utilisant des jauges de déformations.

### 4.2.1 Choix du type de jauges de déformation

Différents tests ont été effectués sur différents types et tailles de jauges de déformation, Tableau 4.1. Les tests consistaient à appliquer une déformation aux jauges et à mesurer l'intensité du signal.

Tableau 4.1 Les différents types de jauges testées dans le cadre du projet de recherche

			
Reference	BF350-3AA	CF120-10AA	BF(BA)-120-2EB
Résistance (Ohm)	350	120	120
Tolérance sur la valeur nominale	< 0.4 ohm	< 0.4 ohm	< 0.4 ohm
Matériel de la grille	Constantan	Constantan	Constantan
Dimension de la grille(mm)	3.2x3.1	2.3x2.7	2.3x3.7
Dimension du support(mm)	7.3x4.1	14x4.5	8.6x7.2



La jauge BF350-3AA a une résistance nominale de 350 Ohm et dimension de 7.3x4.1mm. Cette jauge est petite et difficile à souder, mais elle présente des avantages au niveau de la détection de petites déformations. La jauge CF120-10AA a une dimension adéquate pour faciliter les soudures. Elle a une résistance nominale de 120 Ohm et permet également de mesurer de petites déformations. La jauge BF(BA)-120-2EB est configuré en pont complet de Wheatstone. Cette jauge permet de réduire la taille du circuit d'acquisition en éliminant les résistances requises pour la configuration en quart de pont qui est utilisée pour les jauges BF350-3AA et CF120-10AA. Cependant, cette jauge présente une faible variation de résistance à la déformation et la valeur de la résistance dérive après quelques cycles de chargement-déchargement.

La jauge CF120-10AA a été choisie, car elle est facile à souder et elle présente une bonne réponse linéaire à la déformation.

## **4.2.2 Matrice de jauges de déformation**

Pour arriver au système présenté dans l'article Chapitre 5, différentes itérations ont été réalisées. Les deux paramètres les plus importants qui ont été étudiés ont été l'orientation des jauges et le matériau utilisé pour établir le contact avec la peau.

### **4.2.2.1 Orientation des jauges**

Les travaux de Zizoua *et al* [41] avaient déterminé l'orientation optimale des jauges pour le mouvement de *flexion* et *extension* du *coude*. En effet, leurs travaux ont démontré que la peau se déformait de 11% selon la direction verticale et 6.5% selon la direction horizontale. La déformation la plus grande a été mesurée selon la longueur du *biceps brachii* (direction verticale). Ainsi, les jauges ont été placées de manière que leur trame soit alignée avec cette direction.

Une première version de bracelet de jauges de déformation a été conçue avec les jauges alignées verticalement comme présentée sur la Figure 4.4 et la Figure 4.5.



Figure 4.4 PCB flexible avec des jauges suivant une direction unique



Figure 4.5 PCB flexible avec les jauges de déformation soudées suivant une direction unique

Cependant, mesurer les déformations selon la direction horizontale permet d'avoir une information additionnelle sur la déformation de la peau. Cette information peut s'avérer très utile pour l'algorithme de classification afin de distinguer les différents mouvements. Aussi, les mouvements de supination et de pronation de l'avant-bras entraînent une déformation plus importante de la peau dans la direction horizontale. Il est donc nécessaire d'enregistrer les déformations suivant cet axe. Un second modèle a donc été développé et est présenté sur la Figure 4.6 et la Figure 4.7.



Figure 4.6 PCB flexible avec les jauges suivant deux directions différentes



Figure 4.7 PCB flexible avec les jauges de déformation soudées suivant deux directions différentes

#### 4.2.2.2 Fabrication du bracelet

Pour la fabrication du bracelet, il a été important de trouver un matériau hypoallergénique, assurant un bon contact avec la peau et assez flexible permettant de détecter de petites déformations. Trois techniques différentes ont été testées : La bande de silicone, l'impression 3D et l'utilisation de bandes de kinésiologie.

##### 4.2.2.2.1 Bande de silicone

Cette technique a été basée sur les travaux de [41]. Elle a consisté à insérer les jauges de déformation dans du silicone. Le silicone dragon skin de Smooth-on ([Smooth-On](#)) a été utilisé pour son extrême résistance et son extensibilité. Il peut atteindre plusieurs fois sa taille et reprendre sa forme d'origine sans perdre ses propriétés mécaniques (allongement à la rupture : 1000%) et surtout peut être facilement préparée à température ambiante.

Les étapes de la fabrication peuvent être résumées comme suit :

1. Souder les jauges de déformation sur le PCB flexible. Un total de 6 jauges doit être soudé sur la structure flexible.
2. Mélanger les parties A et B du silicone selon les instructions. Une quantité de 10g de chaque partie a été utilisée, Figure 4.8 (a).
3. Verser le mélange sur une plaque en acrylique
4. Passer avec l'applicateur de film mince pour avoir une épaisseur de couche de silicone de 0.1mm, Figure 4.8 (c).
5. Attendre que le silicone se solidifie puis pulvériser la surface du silicone avec du liquide qui favorise l'adhérence des objets, Figure 4.8 (b)
6. Déposer le PCB flexible sur la couche mince de silicone

7. Pulvériser la surface du PCB flexible avec du liquide qui favorise l'adhérence des objets sur le silicone, Figure 4.8(b)
8. Préparer un deuxième mélange des parties A et B du silicone selon les instructions. Une quantité de 10g de chaque partie a été utilisée. Figure 4.8 (a)
9. Passer avec l'applicateur de film mince pour avoir une épaisseur de couche de silicone de 0.1mm, Figure 4.8 (c).
10. Attendre que le silicone se solidifie et découper le contour pour former le bracelet de silicone instrumentalisé avec les jauges.

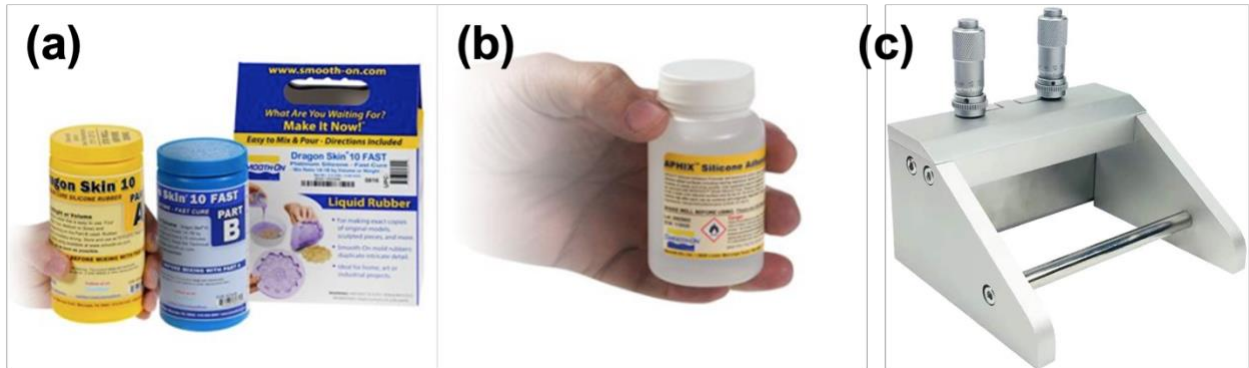


Figure 4.8 (a) Silicone liquide dragon Skin 10 FAST de Smooth-On. (b) APHIX de Smooth-On : agent adhésif permettant de coller la majorité des matériaux sur du silicone. (c) Applicateur de film réglable de 100 mm avec dispositif de préparation de film humide micrométrique 0-3500  $\mu\text{m}$  de largeur.

Le résultat obtenu est présenté sur la Figure 4.9 et 4.10.

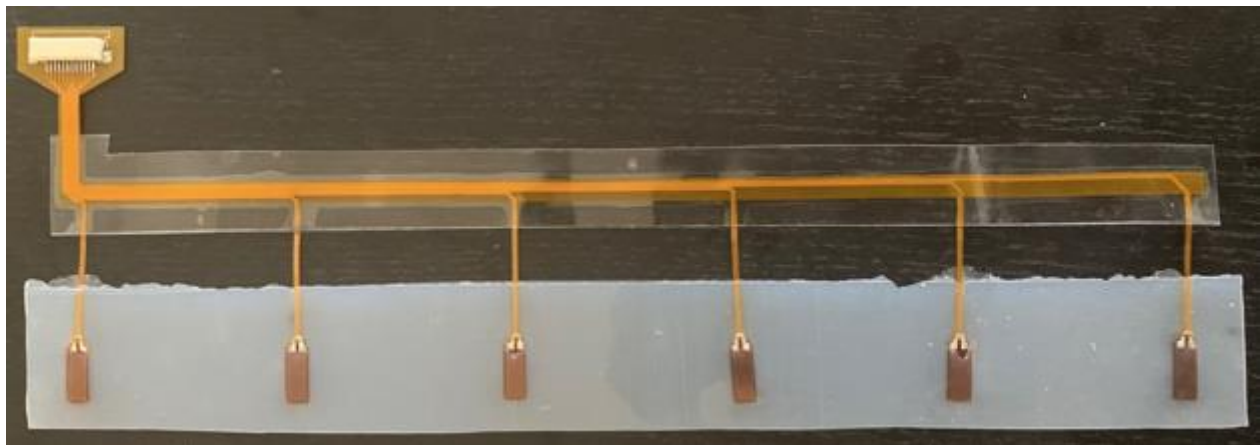


Figure 4.9 PCB flexible intégré dans un bracelet de silicone avec les jauges de déformations suivant une direction unique. La bande autour du PCB flexible a été une astuce pour ne pas avoir du silicone à cet endroit.

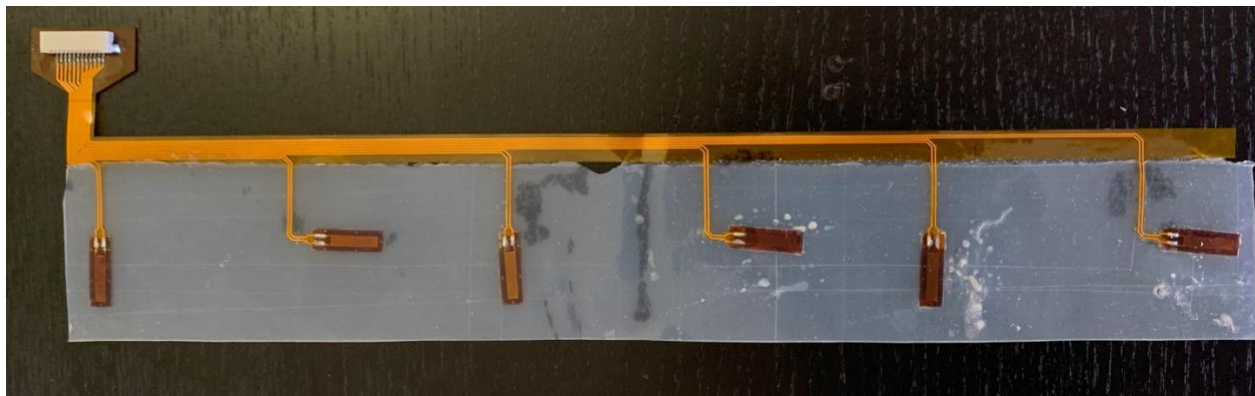


Figure 4.10 PCB flexible intégré dans un bracelet de silicone avec les jauges de déformations suivant deux directions différentes

#### 4.2.2.2 Impression 3D de matériau flexible

Cette technique consiste à imprimer un bracelet flexible avec une imprimante 3D. Le matériau qui a été utilisé est le Ninja Flex de la compagnie NinjaTeck ([Ninjaflex 3D printer filament](#)). Ce matériau est l'un des matériaux d'impression 3D les plus flexibles disponibles sur le marché. Le matériau a une dureté de 85A et peut s'allonger jusqu'à 660% sans usure ni fissure. Le bracelet a été imprimé avec une Prusa I3 MK3S ([imprimante original Prusa I3 MK3S](#)). La procédure de fabrication est comme suit :

1. Imprimer une couche mince de NinjaFlex ayant une épaisseur de 0.3mm
2. Placer les jauges de déformation au-dessus de la couche imprimée
3. Imprimer une couche mince de NinjaFlex ayant une épaisseur de 0.3 mm au-dessus des jauges

Le résultat obtenu est présenté sur la Figure 4.11.

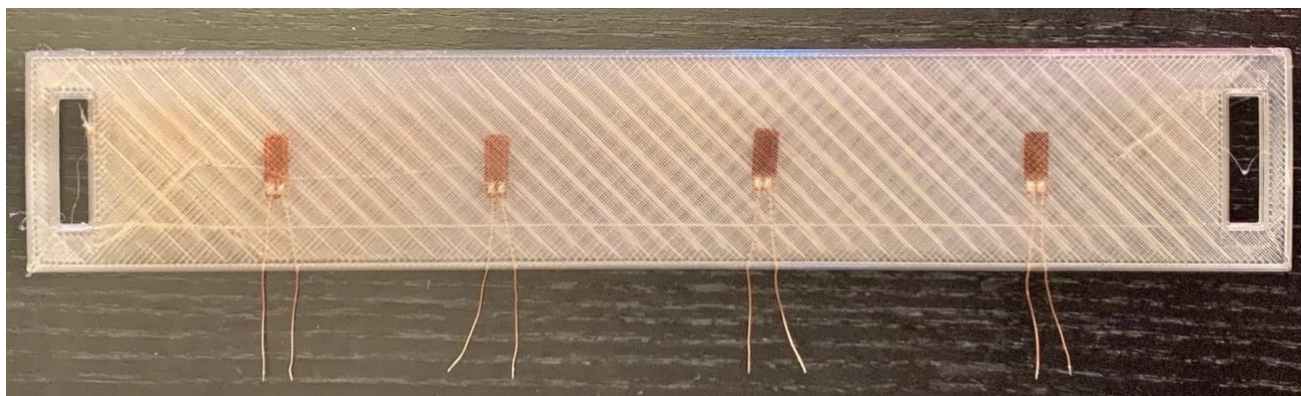


Figure 4.11 Impression 3D flexible d'un bracelet en TPU intégrant des jauges de déformations

#### 4.2.2.2.3 Utilisation d'une bande kinésiologique

Cette technique a consisté à utiliser directement une bande kinésiologique pour établir le contact entre la peau et la matrice de jauges de déformation.

1. Souder les jauges de déformation sur le PCB flexible. Six jauges doivent être soudées sur la structure flexible.
2. Déposer la structure sur une bande kinésiologique

Le résultat obtenu est présenté sur la Figure 4.12

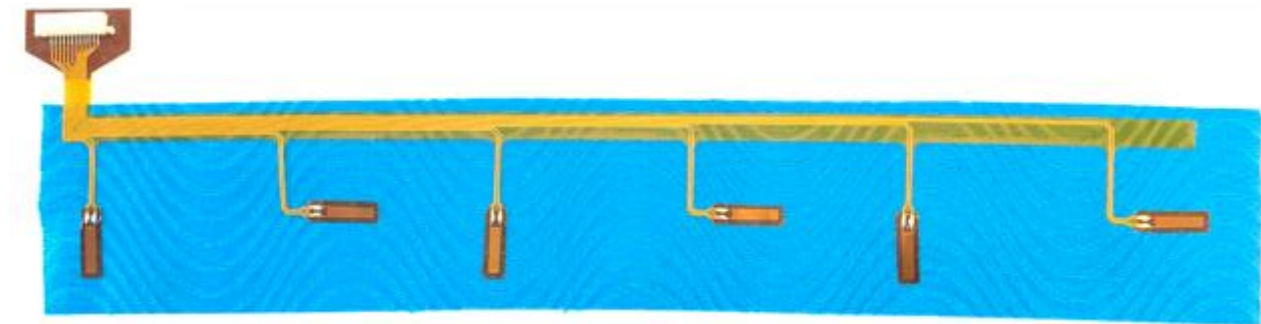


Figure 4.12: Bande kinésiologique instrumentalisée avec une matrice de jauges de déformation

### 4.2.3 Traitement sur platine d'expérimentation

Pour la platine d'expérimentation, la chaîne de mesure [54] des capteurs à base de jauges de déformation a été adaptée. La Figure 4.13 présente les différentes étapes de cette chaîne de mesure.

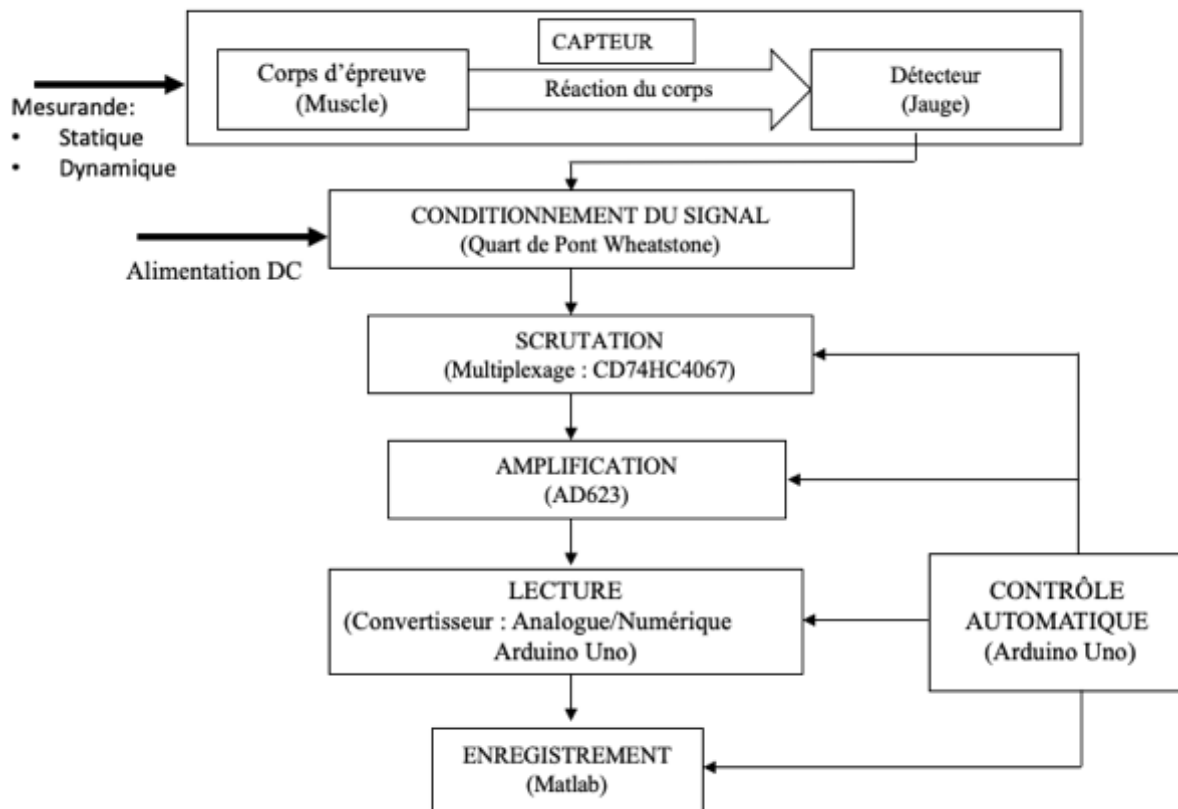


Figure 4.13 Chaîne de mesures des capteurs à base de jauges de déformation. Figure adaptée de [54]

Les éléments importants de cette chaîne de mesure sont les suivants :

- *Le capteur* : Bracelet en silicone contenant des jauges de déformation pour détecter les contractions musculaires.
- *Le conditionneur* : Quart de Pont de Wheatstone permettant de mesurer avec précision la variation de la résistance des jauges.
- *L'alimentation* : Une source d'alimentation stable avec une précision de l'ordre du centième de volt.
- *Le multiplexeur* : Un composant permettant d'utiliser plusieurs capteurs.
- *L'amplification* : Un composant permettant d'augmenter l'intensité du signal.
- *La lecture et l'enregistrement* : Matlab pour présenter les données en temps réels, les traiter et les enregistrer.

L'implémentation de cette chaîne de mesure a permis de développer le circuit présenté à la Figure 4.14. La Figure 4.15 présente la platine d'expérimentation qui a été développée.

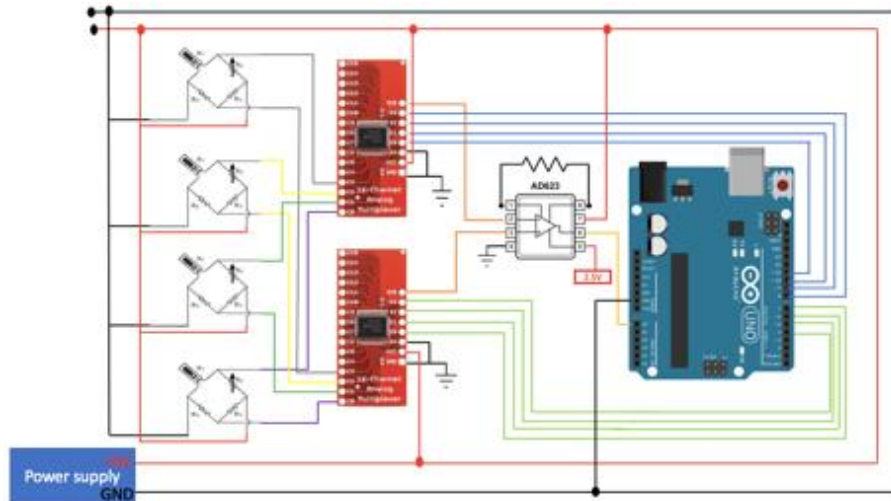


Figure 4.14 Circuit électrique de la carte d'acquisition des signaux provenant des jauges de déformation

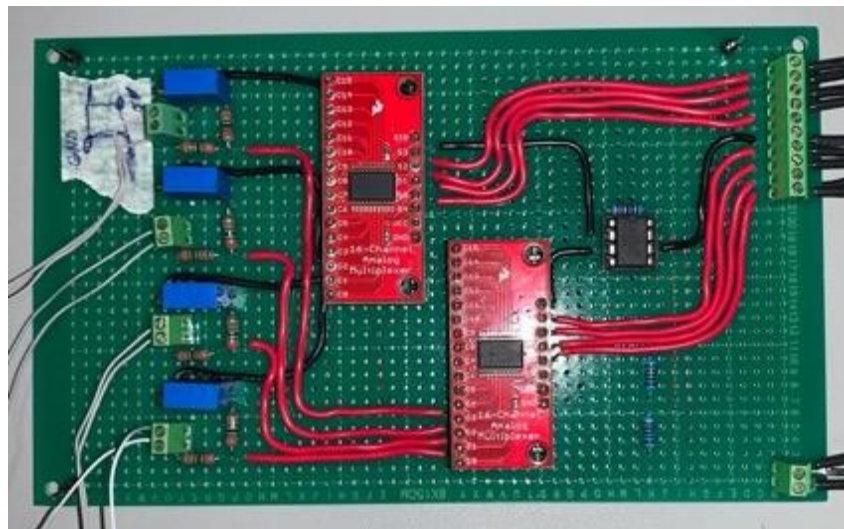


Figure 4.15 Platine d'expérimentation reproduisant le schéma électrique présenté à la Figure 4.14

Conditionnement du signal :

Dans ce circuit, quatre ponts de Wheatstone (quart de pont) ont permis le conditionnement du signal provenant de quatre jauges de déformation. Chaque pont de Wheatstone a été alimenté par une source de tension de 5V provenant d'une source d'alimentation précise ( $\pm 0.01$  mV). La stabilité de la source de tension est importante pour assurer la stabilité des ponts de Wheatstone. Un potentiomètre linéaire dont la valeur nominale de résistance est de 500 Ohm a été utilisé pour équilibrer le pont de Wheatstone. Ce potentiomètre a un nombre de tours de 20 permettant d'avoir



une bonne précision sur la valeur de résistance. Les résistances de ce circuit ont une tolérance de  $\pm 1\%$ , ce qui a facilité l'équilibrage du pont de Wheatstone. Le signal provenant des jauges a donc été acheminé aux deux multiplexeurs.

#### Multiplexage :

L'utilisation des multiplexeurs a permis d'augmenter le nombre d'entrées analogiques en lisant séquentiellement chacune des entrées du multiplexeur. Il a aussi permis d'utiliser un seul amplificateur et un seul convertisseur analogique digital (ADC) pour amplifier le signal provenant des jauges. Un amplificateur est généralement utilisé pour un pont Wheatstone[80]. L'utilisation d'un multiplexeur permet d'utiliser un seul amplificateur pour tous les ponts de Wheatstone. Ainsi, cette configuration nous assure une réduction de la taille du système d'acquisition et de sauver des coûts sur les équipements [41]. Le multiplexeur utilisé (CD74HC4067) a 16 entrées analogiques contrôlées par 4 signaux digitaux.

#### Amplification :

Après le multiplexage suit l'étape de l'amplification. Le pont de Wheatstone a permis de traiter les signaux analogiques en transformant les petites variations de résistance en un différentiel de voltage. Ensuite, l'amplificateur d'instrumentation amplifie le signal a une valeur proche de la valeur référence du convertisseur analogique digitale du microcontrôleur (Arduino Uno).

Pour la platine d'expérimentation, l'amplificateur d'instrumentation AD623 a été sélectionné, car il présente les propriétés suivantes :

- Facilité d'utilisation et mise en œuvre
- Rejet du mode CMRR jusqu'à 200Hz
- Plage d'entrée de tension s'étend jusqu'à 150mV en dessous de la tension de référence
- Gain variable de 1 à 1000.
- Alimentation unique (Ex : +5V)

Afin de déterminer les paramètres adéquats pour le bon fonctionnement de l'amplificateur, la procédure suivante a été utilisée :

1. Concevoir un pont de Wheatstone avec une seule jauge de déformation.
2. Mesurer la différence de voltage provenant du pont avec un voltmètre digital.
3. Déformer la jauge au maximum dans les deux directions afin de déterminer le différentiel de voltage maximum que l'on peut obtenir.

4. Noter la différence de voltage maximum comme étant  $V_{diff} = \pm 2mV$
5. Connecter  $V_{s-}$  au GND pour utiliser l'amplificateur avec une seule source de voltage de +5V.
6. Connecter  $V_{ref}$  à une source de voltage de 2.5V pour créer un décalage afin de lire les différentiels de voltage négatifs.
7. Déterminer les valeurs du mode commun ( $V_{cm}$ ) à l'aide de fiche technique.

L'outil proposé par Analog Device [81] a permis de représenter et de déterminer les caractéristiques de fonctionnement de l'AD623. La valeur du gain a été ajustée à 450 en utilisant une résistance de 223 Ohm.

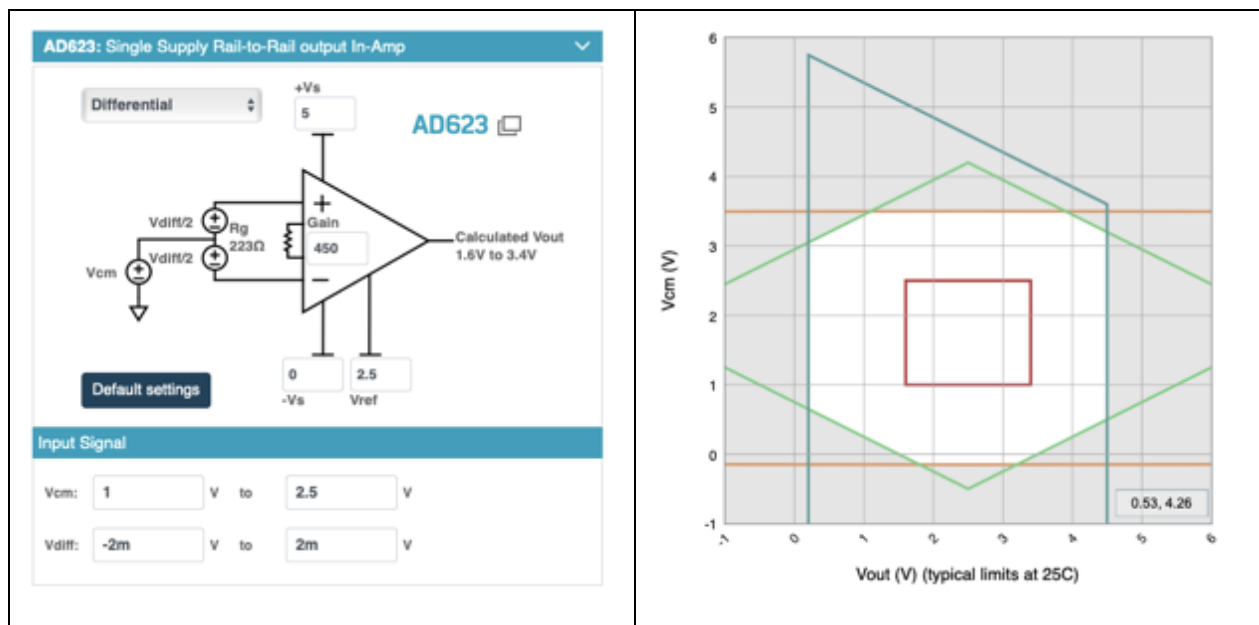


Figure 4.16 Outil permettant de choisir les caractéristiques et la plage de lecture de l'amplificateur AD623 [81]. L'interface à droite permet de régler les paramètres de l'amplificateur. L'interface à gauche permet de visualiser la plage de lecture en fonction des paramètres choisis.

### Lecture Arduino et enregistrement Matlab

Pour la lecture des données, un code sur la plateforme Arduino a été développé pour contrôler la lecture des signaux provenant des jauges de déformation. Afin de visualiser les données en temps réel, le programme Matlab a été utilisé pour sa puissance de calcul et des fonctions disponibles pour le traitement de signal. Les bibliothèques '[Signal Processing Toolbox](#)' et '[Filter Design](#)' ont été utilisées.

#### 4.2.4 Remplacement de la platine d'expérimentation

Après avoir vérifié le fonctionnement de la platine d'expérimentation, des changements de composantes ont été effectués pour mieux répondre aux besoins de notre application. La description des besoins et la conception du nouveau système ont été présentées dans l'article, Chapitre 5. Les changements majeurs sont présentés ci-dessous.

1. Remplacement de la carte Arduino Uno par l'ESP32. L'ESP32 permet d'avoir une communication sans fils entre les différents capteurs. Ce microcontrôleur a également une puissance de calcul plus élevée pour effectuer les calculs nécessaires pour l'identification des mouvements en temps réel à bord
2. Remplacement de l'amplificateur AD623 avec l'ADS1256. L'ADS1256 est un convertisseur analogique-numérique (A/N) 24 bits à très faible bruit. Il a plus de résolution que A/N du Arduino Uno qui est utilisé avec l'AD623.
3. Développement d'une carte de circuit imprimé (PCB) pour rendre le système compact.

### 4.3 Ensemble de données

L'étude présentée dans l'article, Chapitre 4 a porté sur un sujet unique. Ce sujet a effectué une séquence de mouvements prédéfinie. Une méthode basée sur l'intensité du signal a été utilisée pour identifier 9 mouvements du membre supérieur.

Cette section présente la méthodologie complémentaire qui a été utilisée pour développer un algorithme d'apprentissage profond pour identifier 10 mouvements du membre supérieur. Ainsi, la méthodologie a consisté à collecter des données sur 7 personnes saines volontaires (3 personnes de sexe masculin d'âge moyen de 23 ans et de taille moyenne 1.72m; 4 personnes de sexe féminin d'âge moyen de 22 ans et de taille moyenne 1.67m) pour créer une base de données des signaux provenant de la fusion de capteurs présentée dans l'article, Chapitre 5. Avant le début de chaque session d'acquisition des données, chaque sujet a reçu une explication écrite et orale de l'expérience elle-même. Les participants ont fourni un consentement par la signature d'un formulaire de consentement. L'étude a été approuvée par le comité d'éthique de la recherche du *Centre hospitalier universitaire Ste-Justine, Montréal, Canada*.

### 4.3.1 Placement des capteurs

Le placement des capteurs pour l'acquisition des données suit le modèle utilisé dans l'article, Chapitre 5, Figure 5.2.

### 4.3.2 Acquisition des signaux

La méthode d'acquisition a utilisé les capteurs conçus pour enregistrer la cinématique du membre supérieur et l'activité musculaire correspondante. Cette section présente plus de détails sur l'architecture utilisée pour la collecte des données dans l'article, Chapitre 5.

Dans cette architecture, les 3 capteurs, le bracelet de jauges de déformation et les deux IMUs, communiquent avec le capteur central (Bracelet (B)) par le protocole sans-fil ESP-NOW, Figure 5.8. Ce dernier quant à lui communique avec l'ordinateur par une communication UART.

Ainsi, les données suivantes ont été transmises par le protocole ESP-NOW :

- Bracelet (AB) : signaux enregistrés par les 6 jauges de déformations provenant du bracelet placé sur l'avant-bras. Ces signaux ont été identifiés par les lettres AB pour Avant-bras et un numéro pour indiquer la position de la jauge.
- IMU(1) : les trois accélérations linéaires et les 3 vitesses angulaires provenant de l'IMU placée sur l'avant-bras.
- IMU(2) : les trois accélérations linéaires et les 3 vitesses angulaires provenant de l'IMU placée sur le bras.

Un total de 18 signaux a été envoyé par le protocole ESP-NOW. Ensuite, la centrale elle-même a été connectée au bracelet de jauges de déformations placé sur le bras. En plus de recevoir les 18 signaux par la communication sans-fil, elle s'est chargée d'acquérir 6 signaux provenant du bracelet de jauges de déformation placée sur le bras. Ces signaux ont été identifiés par la lettre B pour bras et un numéro pour indiquer la position de la jauge.

Un total de 12 signaux provenant des jauges de déformation a été enregistré pour chaque mouvement. En plus de ces signaux, les accélérations linéaires triaxiales et les vitesses angulaires provenant de deux IMUs ont été utilisées. Un total de 24 signaux a été obtenu en combinant les signaux des jauges de déformation et des centrales inertielles

### 4.3.3 Exercices

Les exercices effectués pour des fins de collectes de données sont identiques aux exercices présentés dans l'article, Chapitre 5, Figure 5.3 Ces mouvements ont été choisis sur la base de la littérature pertinente et les directives de pratique standard de réadaptation [24].

### 4.3.4 Collecte des données

Pour chaque session d'exercices en laboratoire, les sujets ont effectué les 10 mouvements à l'étude. Pour chaque mouvement, une acquisition sur 60 secondes suivant le rythme d'un métronome à 50BPM soit 0.83 Hz a été effectuée suivie d'une pause de 10s. La pause de 10s est observée pour éviter les fatigues musculaires. Ce protocole a été inspiré des travaux de [82]qui ont établi une base de données de référence comptant 52 mouvements des doigts, des mains et des poignets.

L'acquisition a été répétée deux fois pour chaque mouvement. Les données ont été échantillonnées à une fréquence de 100 Hz. La fréquence de 100 Hz a été choisie pour limiter la quantité de données à traiter. La fréquence des mouvements humains est de 0-20 Hz [83]; de ce fait le théorème de Shannon Nyquist a été respecté. À la fin de cette étape, un fichier « .txt » a été généré pour chaque mouvement effectué pour chaque sujet.

### 4.3.5 Traitement des signaux

Différentes étapes de traitement ont été appliquées aux signaux afin de pouvoir les utiliser avec les algorithmes de classification.

**Filtration et amplification** : Un filtre passe-bas numérique de type Butterworth d'ordre 4 avec une fréquence de coupure de 1 Hz a été utilisé pour filtrer le signal des jauges de déformation. Les cent premières données de chaque jauge ont été moyennées et utilisées pour définir un décalage pour les données suivantes. Un gain digital de 10,000 a été appliqué à ces signaux pour permettre de mieux les analyser visuellement. Les signaux bruts provenant des centrales inertielles ont été utilisés pour réduire les calculs effectués par le microcontrôleur.

**Étiquetage** : Une procédure d'étiquetage a été utilisée pour attribuer à chaque portion de signal une classe correspondante. Cette méthode est considérée comme de l'apprentissage supervisé « *supervised learning* », car les classes sont attribuées par un agent externe contrairement aux techniques non supervisées « *unsupervised learning* », qui utilisent des techniques de

« *clustering* ». Cette étape est cruciale pour la classification, car la présence d'exemples mal étiquetés dans les données d'apprentissage peut entraîner de mauvais résultats. L'étiquetage s'est fait à l'aide de l'outil « *Signal Labeler* » de Matlab. Un exemple d'étiquetage est montré à la Figure 4.17 pour le mouvement d'extension du poignet.

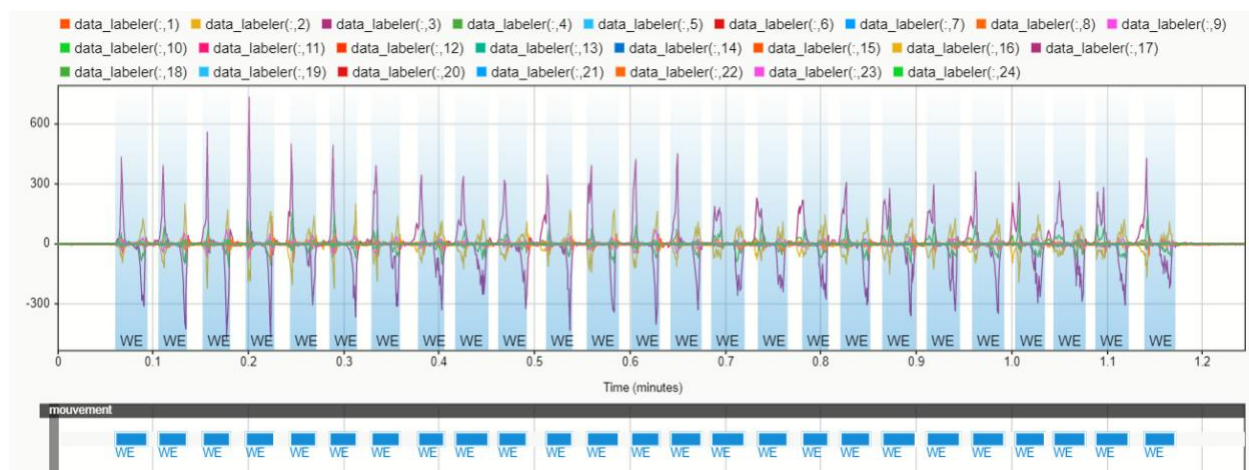


Figure 4.17 Étiquetage des signaux pour le mouvement d'extension du poignet pour un des sujets. À la fin de cette étape, un fichier « *.mat* » par mouvement a été généré. Dans chaque fichier, une classe de mouvement a été attribuée à un intervalle de temps précis.

## 4.4 Traitement des données

Le traitement des données a pour but de structurer les données pour qu'elles soient utilisables par les algorithmes d'apprentissage profond.

Après l'obtention du fichier « *.mat* », section 4.3.5, les étapes suivantes de traitement de données ont été présentées:

1. Utilisation du fichier « *.mat* » pour créer un fichier « *.txt* » avec 25 colonnes. Les 24 premières colonnes représentent les signaux provenant des capteurs. La 25<sup>e</sup> colonne représente la classe attribuée à chaque point de données.
2. Mise en commun des fichiers par mouvement pour tous les sujets. Par exemple, le fichier « *EF.txt* » regroupait tous les mouvements de la flexion du coude de tous les sujets. Pour finir, 10 fichiers correspondant à chacun des mouvements ont été obtenus. Le choix de combiner les données de tous les sujets a été effectué à cause du nombre insuffisant de données par individu pour entraîner des algorithmes de classification.

- Segmentation des données avec une fenêtre de 250 ms et 50 ms de chevauchement entre les fenêtres. Bien que des intervalles plus grands ont plus d'informations et que la typicité améliore la sortie de classification, il a été démontré que les fenêtres doivent être inférieures à 300 ms afin que l'utilisateur ne subisse pas de retards de contrôle [26].

Après avoir effectué ces opérations une matrice 3 dimensions a été obtenu pour chacun des mouvements. La Figure 5.18 présente la forme de la matrice qui a été obtenue.

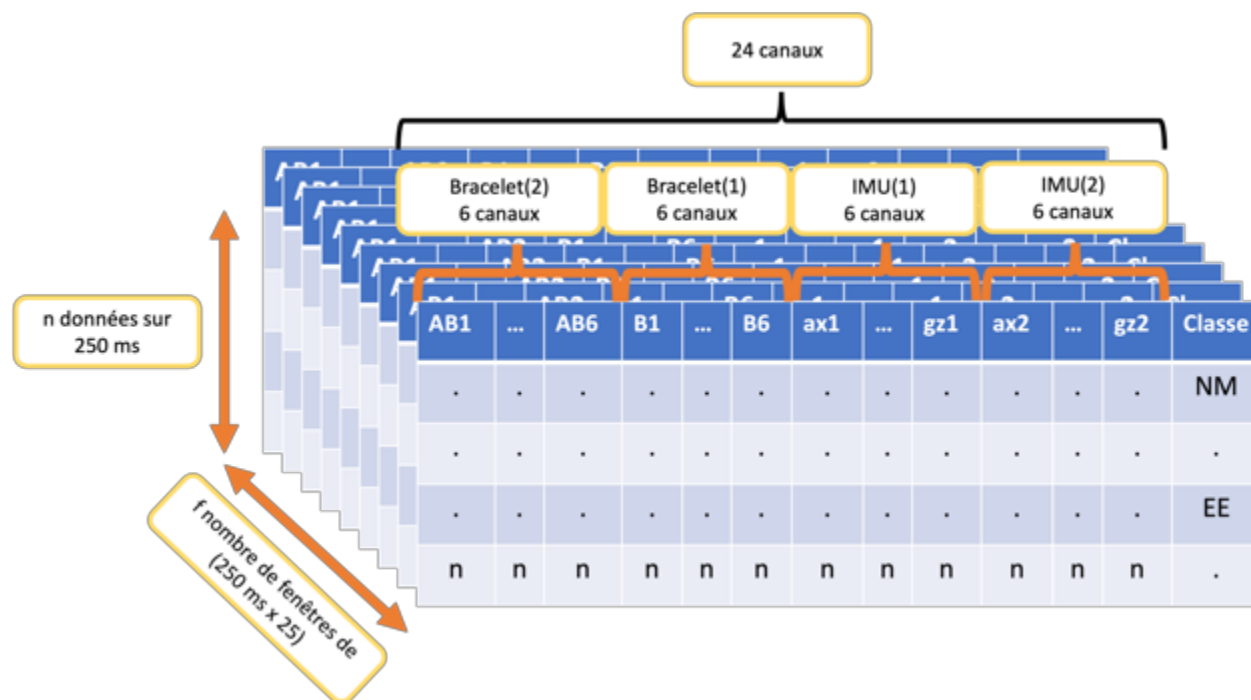


Figure 4.18 Représentation des dimensions de la matrice obtenue après le traitement des données.

Suivant les colonnes, les données provenant des capteurs (24 signaux). Suivant les lignes le nombre de données correspondant à 250 ms d'acquisition. La profondeur représente le nombre de fenêtres de 250ms x 25 canaux.

Une seconde opération a consisté à séparer les données provenant des capteurs et les classes. Pour chaque fenêtre de données, une classe résultante est associée, Figure 4.19. Cette classe est obtenue en faisant la moyenne des classes des échantillons de la fenêtre. Lorsqu'une classe a une moyenne supérieure à 95% alors elle est attribuée à la fenêtre.

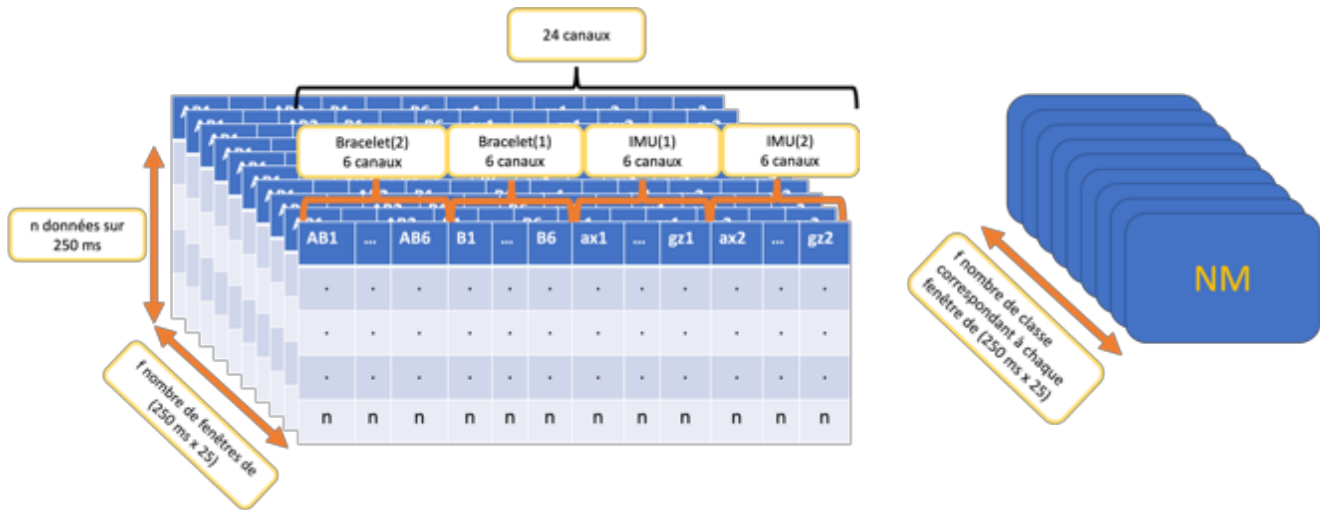


Figure 4.19 Séparation des données provenant des capteurs et des classes associés aux fenêtres

Ainsi, pour les données des capteurs la structure suivante est obtenue :

$$D = (\text{nombre de fenêtres } (f), \text{ nombre de pas de temps } (n), \text{ nombres de paramètres})$$

Et pour les classes on obtient un vecteur :

$$C = [NM, EE, EF, \dots]$$

dont le nombre d'éléments correspond au nombre de fenêtres  $f$

Les composants de la matrice D sont :

- *Nombres de paramètres*: le nombre de signaux utilisés. Ici 24 signaux provenant des capteurs ont été utilisés soit 12 signaux provenant des centrales inertielles et 12 signaux provenant des jauges de déformation.
- *Nombre de pas de temps (n)*: le nombre de données par fenêtre. Ici ce nombre est de 25.
- *Nombre de fenêtres (f)* : le nombre de fenêtres de 250 ms pouvant être extraite par mouvement.

La Figure 4.20 présente le nombre de fenêtres ayant une dimension de 25 données (250 ms à 100Hz) et 24 paramètres pour chacun des mouvements.



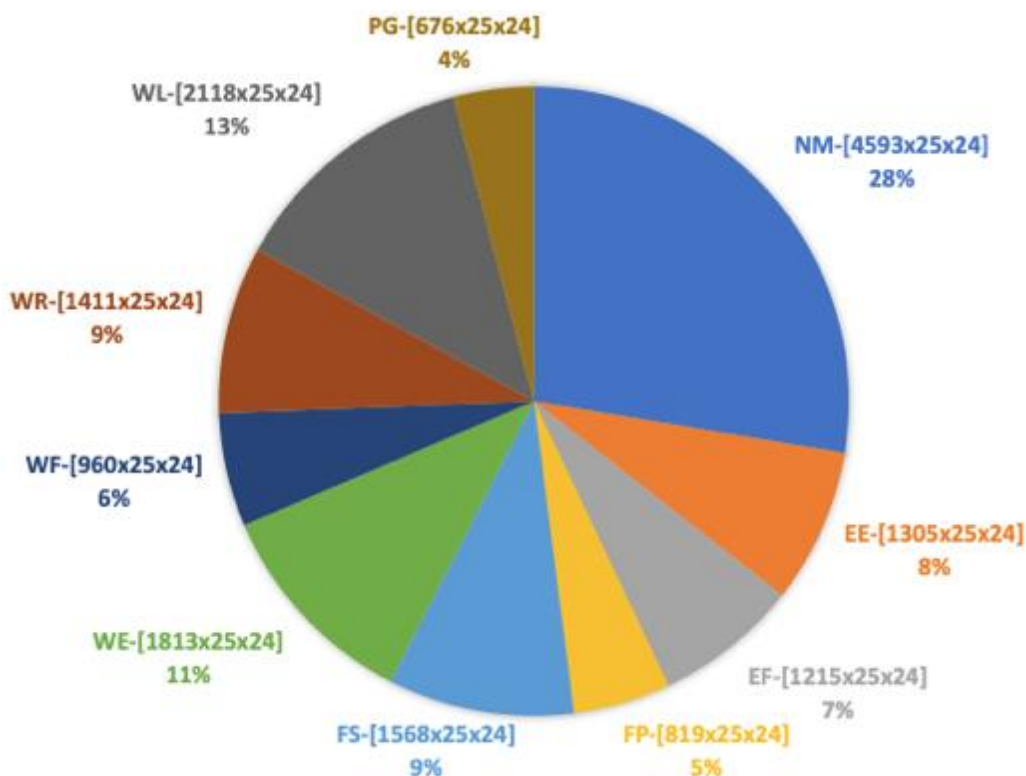


Figure 4.20 Répartition des données par classe de mouvement

- La dernière étape a consisté à séparer les données en ensembles d'entraînement, de validation et de tests. Ainsi pour chaque mouvement, 70% des données ont été utilisées pour l'entraînement, 15% pour la validation et 15% pour le test. Les éléments appartenant à chacun de ces ensembles ont été choisis de façon aléatoire.

## 4.5 Logiciels et matériels

Une fois les données structurées de manière appropriée, elles ont été importées dans le programme développé en Python pour être utilisées par l'algorithme d'apprentissage profond. Cela a été implémenté dans un environnement virtuel créé à l'aide d'Anaconda qui est une distribution libre et Open Source. L'éditeur de code Visual Studio Code (VScode) a été utilisé. Python a été choisi en raison de sa simplicité ainsi que de sa compatibilité avec plusieurs bibliothèques Open Source conçues pour le développement et l'entraînement de modèles d'apprentissage machine, tel que Pytorch, Theano, et Tensorflow. La bibliothèque Keras est une interface de programmation d'application (API) intuitive et bien documentée. Elle fournit une méthode simple de haut niveau

de programmation de réseaux de neurones. Par conséquent, Keras est un choix idéal pour une implémentation de haut niveau de réseaux complexes, tels que les CNN et les LSTM.

Le processus d'entraînement des réseaux de neurones profonds requiert très souvent une puissance de calcul élevée. Il faut donc s'assurer d'avoir un équipement adéquat. Le système utilisé pour les expériences était un ordinateur basé sur Windows 64 bits, processeur Intel® Core™ i7, avec une mémoire RAM de 16 GO et une carte graphique GTX1060 avec 4 GO de mémoire dédiées compatibles avec CUDA de NVIDIA. Dans les résultats, le temps d'exécution pour les différents réseaux a été indiqué, car il dépend fortement des performances de l'ordinateur.

## 4.6 Expériences

La dernière étape avant l'entraînement des réseaux de neurones a consisté à balancer le nombre d'échantillons dans chacune des classes au sein de l'ensemble des données. En effet, la Figure 5.20 montre que les classes présentent de grandes disparités. La classe correspondant au mouvement de repos (NM) contenait le plus grand nombre d'échantillons (4593). La classe correspondant au mouvement de préhension (PG) avec 676 échantillons contenait le plus petit nombre d'échantillons. Dans la section 2.6, l'importance d'avoir des classes équilibrées a été démontrée. À cette fin, avant chaque exécution, les fenêtres classées comme étant des mouvements de repos ont été supprimées aléatoirement des ensembles de données. De plus, hormis le mouvement de repos, les autres classes de mouvement ont également été soumises à une suppression aléatoire d'un certain nombre de fenêtres afin que toutes les classes de mouvement contiennent exactement le même nombre d'exemples que le mouvement le moins représenté. Pour améliorer l'entraînement, les exemples ont également été choisis aléatoirement, avant chaque exécution.

Aussi, comme décrit dans la section 2.4.1, les classes cibles ont été encodées selon le format « one hot » représenté à la Figure 4.21.

Classe	Classe	Encodage « one hot » des classes									
NM	0	1	0	0	0	0	0	0	0	0	0
EE	1	0	1	0	0	0	0	0	0	0	0
EF	2	0	0	1	0	0	0	0	0	0	0
FP	3	0	0	0	1	0	0	0	0	0	0
FS	4	0	0	0	0	1	0	0	0	0	0
WE	5	0	0	0	0	0	1	0	0	0	0
WF	6	0	0	0	0	0	0	1	0	0	0
WR	7	0	0	0	0	0	0	0	1	0	0
WL	8	0	0	0	0	0	0	0	0	1	0
PG	9	0	0	0	0	0	0	0	0	0	1

Figure 4.21 Encodage « one hot » pour la classification multiclasse

Deux architectures différentes ont été testées sur l'ensemble de données. La première consiste à utiliser un LSTM et la seconde consiste à ajouter un CNN avant les couches LSTM pour former un CNN-LSTM. Une fois que la comparaison a été effectuée entre ces architectures, celle qui fournissait le meilleur résultat a été optimisée en recherchant l'espace des hyperparamètres. Lorsqu'un espace d'hyperparamètres permettant d'avoir de bons résultats de classification, l'impact des données inertiels sur le résultat de classification a été étudié.

#### 4.6.1 Comparaison des modèles

Les deux architectures qui ont été comparées sont le modèle LSTM et le modèle CNN-LSTM. La motivation pour l'utilisation de ces modèles a été décrite dans la section 2.1.2. Le modèle (LSTM) a été utilisée pour l'apprentissage des caractéristiques temporelles des signaux. La couche de convolution quant à elle a été responsable de l'extraction des caractéristiques et de la réduction de la dimensionnalité des données. Pour comparer les deux architectures, le modèle LSTM présenté sur la Figure 4.22 a d'abord été développé et les hyperparamètres ont été définis en effectuant différents tests. Par la suite, une couche de convolution 1D et une opération de « *max pooling* »

comme décrite dans la section 2.3.3 ont été rajoutées au modèle LSTM précédemment développé, Figure 4.23.

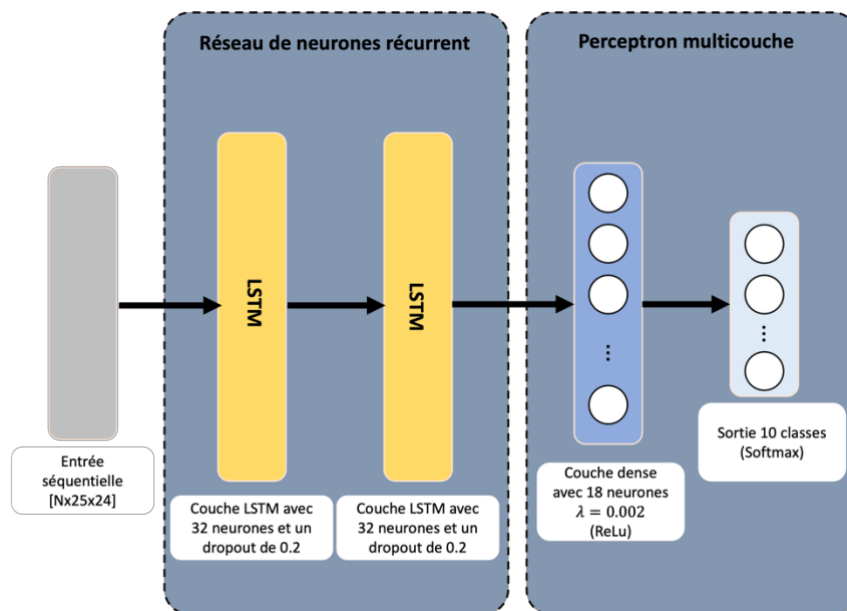


Figure 4.22 Modèle LSTM. L'entrée est transmise à deux couches LSTM comptant chacune 32 neurones. Un dropout de 0.2 a également été appliqué à ces couches. A la fin du réseau, il y a un perceptron multicouche qui se termine avec deux couches. Une couche dense comptant 18 neurones et un terme de régularisation  $\lambda = 0.002$ . Une de sortie avec une fonction d'activation softmax et 10 neurones, une pour chaque classe de mouvement.

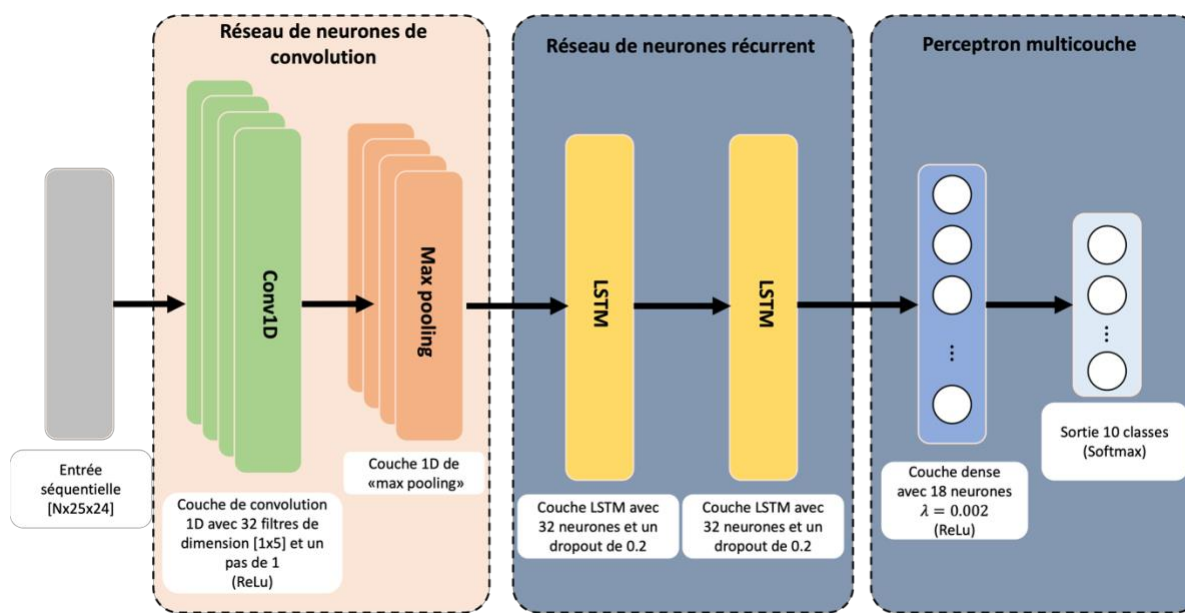


Figure 4.23 Modèle CNN-LSTM. Ce modèle a été obtenu en rajoutant un réseau de neurones de convolutif juste avant le modèle LSTM présenté à la Figure 4.16. L'entrée est d'abord transmise à une couche convolutive 1D avec 32 filtres de dimension [1x5] et un pas de 1. La fonction ReLu a été utilisée comme fonction d'activation. La couche convolutive est suivie par une couche de mise en commun « max pooling ». Les vecteurs de caractéristiques extraits par ceux-ci seront alors transmis au réseau de neurones récurrents.

## 4.6.2 Recherche d'hyperparamètres

Dans le processus de développement des algorithmes d'apprentissage profond, il est commun d'effectuer une recherche d'hyperparamètres pour améliorer les performances de l'algorithme même après avoir trouvé une architecture fonctionnelle. Il existe différentes stratégies de recherche d'hyperparamètres. La plus répandue est une recherche sur grille où plusieurs combinaisons de paramètres sont testées et les résultats comparés.

Cependant, ces différentes stratégies sortent du cadre de ce travail de maîtrise. Ainsi, dans cette étude le processus de recherche d'hyperparamètres a consisté à tester de manière séquentielle cinq valeurs de chaque hyperparamètre en gardant à chaque fois la valeur de l'hyperparamètre qui a donné le meilleur taux de classification pour les essais subséquents. Cela a permis d'avoir une référence pour des études futures.

### 4.6.3 Effet des données cinématiques

Après avoir déterminé le modèle qui performe le mieux, la quantification de l'apport des données cinématiques sur les résultats de classifications a été entreprise. Pour ce faire, les données reliées aux centrales inertielles ont été retirées des données de l'ensemble d'entraînement, de validation et de test. Ainsi, une base de données contenant uniquement les données des jauges de déformations a été utilisée pour effectuer l'entraînement du modèle d'apprentissage profond. La nouvelle matrice de données est présentée à la Figure 4.24. Les résultats de classifications ont été comparés aux résultats obtenus avec le modèle qui a utilisé l'ensemble des données.

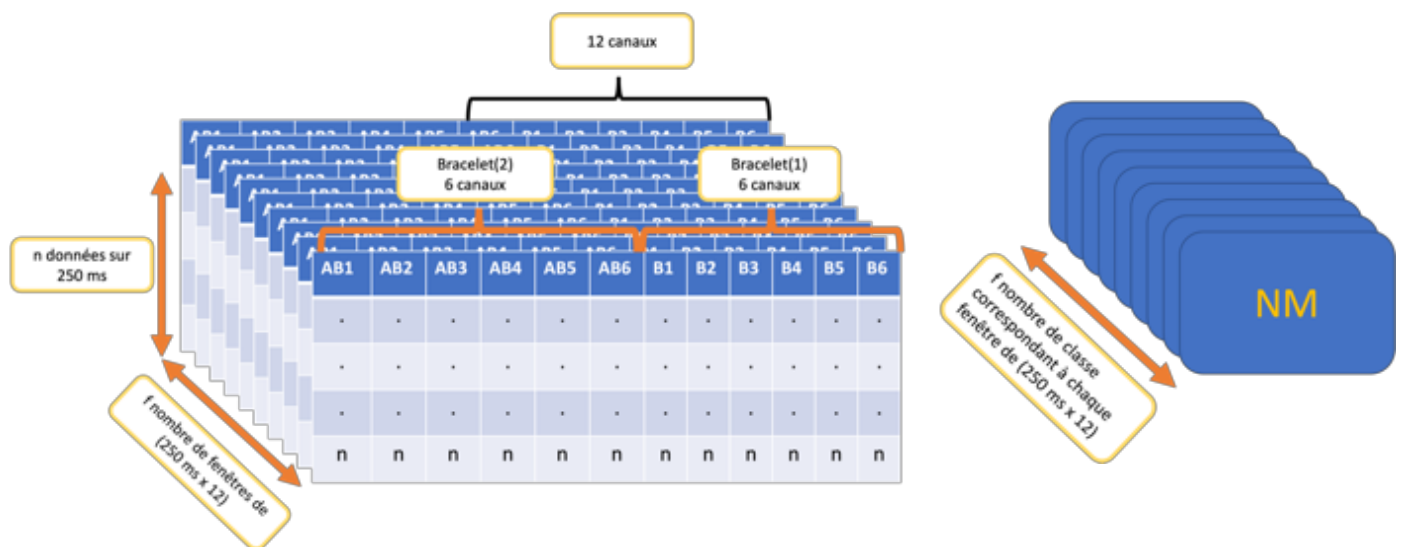


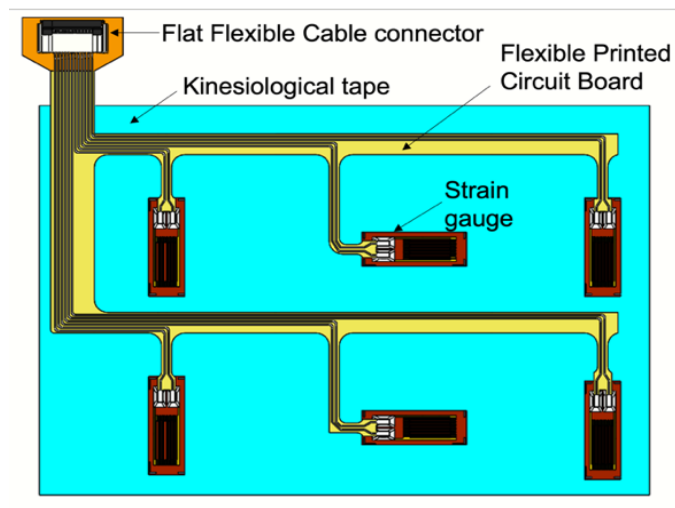
Figure 4.24 Représentation des dimensions de la matrice obtenue après la suppression des données inertielles. Suivant les colonnes, les données provenant des bracelets de jauges de déformation (12 signaux). Suivant les lignes le nombre de données correspondant à 250 ms d'acquisition soit 25. La profondeur représente le nombre de fenêtres de 250 ms x 12 canaux.

## CHAPITRE 5 ARTICLE 1 : DEVELOPMENT OF A WEARABLE SYSTEM TO IDENTIFY MOVEMENT INTENTIONS BY COMBINING STRAIN GAUGES AND INERTIAL MEASUREMENT UNITS

Cet article a été soumis le 14 novembre 2021 dans la revue IEEE Sensors Journal. Les auteurs de l'article sont Steve Regis Koalaga, Maxime Raison et Sofiane Achiche.

### 5.1 Abstract

The combination between surface electromyography sensors and inertial measurement units is the most common multimodal sensing method used in body area networks. Nowadays, this combination is frequently used to identify the intentions of movement in humans, e.g. to control their prosthesis. However, the surface electromyography sensors are usually bulky, with electrodes



placed on skin sites, and require a high sampling frequency, usually 1000 Hz, which technically highly reduces the number of sensors that can be used simultaneously by an onboard microprocessor. Further, the electromyographic measurement suffers from crosstalk due to muscles packed side by side. These limitations in electromyography motivate the search for alternatives using multiple sensors capable of operating at lower frequencies for everyday applications at an affordable cost. The objective of this study is to develop a novel wearable system to identify intentions of movement by combining strain gauges and inertial measurement units. The system is composed of 1. two bracelets using six strain gauges each, connected to a flexible printed circuit board and 2. two inertial measurement units. Physiologically, the strain gauges measure the skin deformation due to muscle contraction, while the inertial measurement units provide complementary data on joint kinematics. The system was tested at the upper limb, and successfully identified 9 main movements based on the signal intensity of strain gauges. These results show the

great potential of such a sensory system to become a smart wearable sensory system to detect human movement intention.

## 5.2 Index Terms

strain gauges, IMUs, sensor fusion, movement intention, kinesiological tape.

## 5.3 Introduction

### 5.3.1 On the combination of sEMG-IMU to identify movement intention: context and physiological justification

The combination between surface electromyography (sEMG) sensors and inertial measurement units (IMU) is the most common used multimodal sensing method in body area networks [1]. Nowadays, this combination is frequently utilized to identify the intention of movement in humans, e.g. for hand and finger gesture recognition [2], or to classify upper limb phantom movements in transhumeral amputees to control their prosthesis (e.g. [3], [4]). And the methods for identification of the intention of movement based on sEMG and IMU have the potential to be extended to daily general applications, such as human computer interfacing [5], [6], teleoperation of industrial robots [7], etc.

The combination between sEMG and IMU can be physiologically justified as a potentially successful avenue, because:

- 1) sEMG enable to detect the intention of movement, by measuring the muscle activity. Therefore, sEMG is still the main sensor used to control myoelectric prostheses.
- 2) IMUs provide additional kinematics information about the motion, i.e., articulation configurations, velocities, and accelerations. IMUs fills two limitations of sEMG:



- sEMG commonly suffers from the limb position effect, where sEMG signals for the same motion are different in different limb positions [8], [9]. Therefore, IMUs can be complementary.<sup>1</sup>
- IMUs are particularly good for capturing larger motions, while sEMG data are better at distinguishing different hand shapes and finger movements [2].

Adding kinematics features enabled to increase the accuracy of the movement classification by 4.8% [3] on human upper limb movements. Furthermore, Geng *et al.* [10] and Fougner *et al.* [11] presented a classifier in cascades, which reduced the average movement classification error from 18% to 5.7%. This classifier used accelerometry to determine the best limb position before choosing the sEMG classifier [11].

### 5.3.2 Limitations of sEMG and search for alternative solutions

sEMG is a technique that uses electrodes placed on the skin at a specific location to monitor muscle contractions [12]. There are several drawbacks to using sEMG [12], [13]. sEMG can often be unstable due to sweat, electrode shifts, motion artifacts, and electronic noise [12], [14]. Also, crosstalk can occur due to the high number of muscles packed side by side, and muscular fatigue can crucially affect the quality of the signals [12], [14]. Further, the amount of data coming from sEMG, due to the acquisitions generally at 1000 Hz, requires high computing power to process it in real time [15]. Hence, researchers need to pay attention to these critical issues.

---

S.R. Koalaga is with the Mechanical Engineering Department, Polytechnique de Montréal, Montreal QC H3T 1J4, Canada, and also with MARIE ENFANT Rehabilitation Centre, CHU Sainte-Justine, Montreal, QC H1T 1C9, Canada (e-mail: [steve-regis.koalaga@polymtl.ca](mailto:steve-regis.koalaga@polymtl.ca)).

M. Raison is with the Mechanical Engineering Department, Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada, and also with the Rehabilitation Center, Centre hospitalier universitaire Sainte-Justine, Montréal, QC H3T 1C5, Canada (e-mail: [maxime.raison@polymtl.ca](mailto:maxime.raison@polymtl.ca)).

S. Achiche is with the Mechanical Engineering Department, École Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada (e-mail: [sofiane.achiche@polymtl.ca](mailto:sofiane.achiche@polymtl.ca)).

Different sensors have been proposed as alternative solutions to using sEMG. Indeed, the growing interest in smart wearable technologies requires the development of new sensors at low cost, with high sensitivity and low detection limit [16]. Liang Zou *et al.* [17] grouped together all tactile sensing systems in four groups, which are capacitive, piezoresistive, piezoelectric and tactile optical sensors. The tactile sensing systems are mostly used in robotics and biomedical engineering. A more detailed work done by [18], introduced sensor skins defined to be stretchable planar structures with embedded sensing components. Sensor skin found in the literature can be grouped by the type of material (elastomers, woven fabric), the type of conductor (thin metal films, liquid metal), and the structure they use (microchannels, mechanical flexible interfaces). Chang *et al.* [19] proposed a strain sensor that can still form conformal contact to the skin even during body movements. They are prepared by solution coating and consist of two layers, a dry adhesive layer of biocompatible water-based elastomeric polyurethane, and a detection layer of a non-adhesive composite of reduced graphene oxide and carbon nanotubes. The adhesive layer makes the sensors conform to the skin, while the sensing layer has sensitive resistance to deformations. Song *et al.* [20] presented a strain sensor based on silk graphene spandex coated fabric (GCSS) prepared by reducing graphene oxide. The sensor worked thanks to the extension of the conductive fiber and the deformation of the woven structure. GCSS was successfully used to detect human movement, by providing data for gesture recognition based on deep learning. Yao *et al.* [21] described the application of capacitive strain sensors based on silver nanowires for kinematic finger tracking. The sensors can be attached to the skin to track the movement of the finger joints with minimal interference with daily activities. Ali *et al.* [22], presented a new goniometric glove using flex sensors to capture the user hand gesture that can be used to wirelessly control a bionic hand. However, many of these sensors used a complex fabrication procedure and/or special materials graphene spandex coated fabric, liquid gallium, etc. [17], [18], and were limited to finger movements tracking [19], [20], hence reducing the number of upper limb movements that could be detected in real applications.

### **5.3.3 Strain gauges as the promising solution**

Mori *et al.* [23] presented a new bioinstrumentation sensor using one strain gauge for upper limb amputees. Their work concluded that the repeatability of the strain gauge signal is superior to myoelectric signal because the sensor measures the deformation of the skin [23].

Finally, Zizoua *et al.* [15] presented a proof of concept of a bracelet using strain gauges for the identification of four upper limb movements in traumatic amputees: elbow flexion/extension and forearm pronation/supination. Strain gauges can measure small deformation of about 10-13 $\mu$ m and are widely used for their low cost and simple signal conditioning [15].

Therefore, Zizoua *et al.* [15] could be a starting reference as an alternative to EMGs, but this system cannot be used as proposed to identify three-dimensional movements, because of four major limitations:

- 1) *Uniaxial skin deformations*: the gauges were placed only in the longitudinal direction of the biceps brachii muscle, which does not enable to record multi-axial skin deformations.
- 2) *Fragility*: the strain gauges were connected by two thin wires. This enabled a first proof of concept but was not viable as it either broke or unsoldered when used several times [15].
- 3) *Non-portability*: the system was neither compact nor portable, which limits its application in everyday life.
- 4) *Lack of conform contact to skin during movement*: the strain gauges were embedded in silicone that did not provide a direct contact with the skin, thus limiting the ability of the system to measure the actual skin deformation.

### 5.3.4 Problem, objective, and research hypotheses

Here above, the state-of-art survey revealed two major problems:

1. The common issues with sEMG were solved by a first proof of concept of bracelet using strain gauges. But this one still had limitations: uniaxial skin deformations, fragility, inaccuracy, and non-portability.
2. The combination between strain gauges and IMUs has never been investigated to detect human motion intention.

Consequently, the objective of this study is to develop a wearable system to identify intentions of movement by combining strain gauges and inertial measurement units.

The main design requirements (DR) of this system are as follows:

DR 1: The system must be able to measure multi-axial skin deformations.

DR 2: The system must be able to be used several times in daily three-dimensional movements.

DR 3: The system must be wearable.

The following research hypothesis (RH) to study can be formulated: “The combination between strain gauges and IMUs could enhance the ability to detect motion intention”.

## 5.4 Methods

The muscle contractions during a movement create a deformation of the skin at the surface. Hence, identifying the muscles involved in the upper limb movement can help to determine the best sites to measure skin deformations and to design the strain gauge bracelet accordingly.

### 5.4.1 Design of a strain gauge bracelet

The strain gauge bracelet consisted of six equally spaced (3.5 cm) strain gauges. These gauges were connected by a 0.1 mm thick flexible printed circuit board (PCB) (see Fig. 5.1 (a)). In this flexible PCB, the strain gauges with odd numbers (1, 3 and 5) were aligned parallel to the longitudinal direction of the biceps brachii, and the strain gauges with even numbers (2, 4, and 6) were aligned perpendicularly to the longitudinal direction of the biceps brachii. This configuration enables the recording of the skin deformations along two different axes.

The gauges were also capable of bending in two directions providing positive (convex deformation) or negative (concave deformation) voltage variations.

The flexible PCB had a 12-position flat flexible connector (FFC). This connector was used to connect the strain gauge bracelet to the acquisition board. The flexible PCB provided a solid connection with the strain gauges.

Different tests were performed with different types and sizes of strain gauges. The tests consisted in applying a deformation to the strain gauges and measuring the signal intensity. The gauge with the best linear response and high intensity was selected, namely the CF120-10AA. The gauge had a linear pattern and a nominal resistance of  $120 \text{ ohm} \pm 1\%$  with a gage factor of  $2 \pm 1\%$ . The gauge was made of constantan alloy and had a sensitive grid of  $10.0 \times 4.0 \text{ mm}$ . It was able to measure small strains of about  $\pm 5\%$  of the neutral length which was adequate for our application.

For a better measurement of skin deformation, the gauges, previously soldered on the flexible PCB, were placed directly on a kinesiological tape (see Fig. 5.1 (b)).

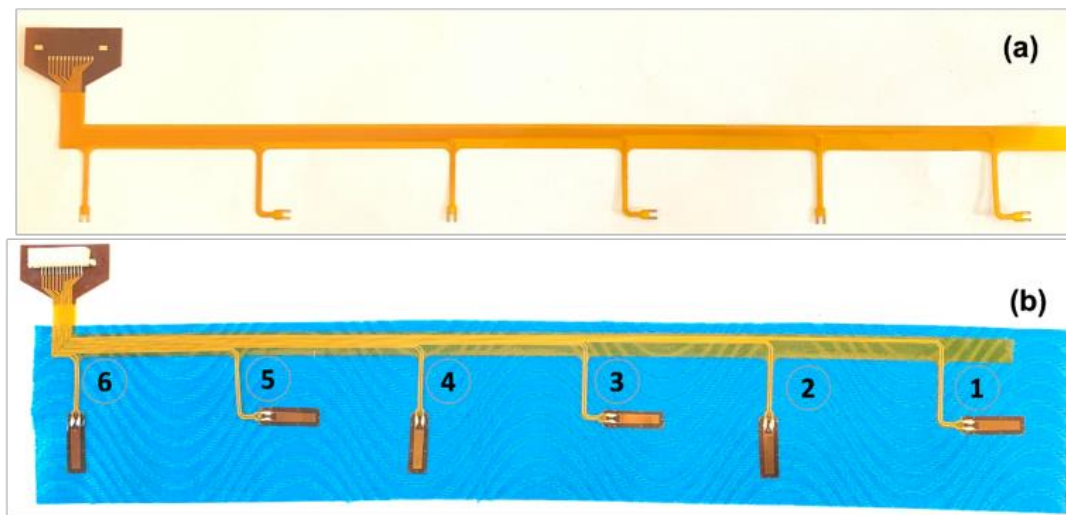


Figure 5.1 Strain gauge bracelet. (a) Flexible PCB (yellow) with the connections for 6 strain gauges. (b) Instrumented kinesiological tape (blue) the flexible PCB (yellow) connected to the 6 numbered strain gauges (orange rectangles).

The kinesiological tape was designed to mimic the skin elasticity so the users can use their full range of motion [24]. The tape used a medical-grade adhesive, which was water-resistant and strong enough to stay on for several days even, while working out or taking showers [25]. Kinesiological tapes are known as therapeutic tapes that are stretched and, strategically applied to the body to provide support, lessen pain, reduce swelling, and improve performance [26]. A therapist can let you know how much stretch is needed for your treatment. In this study, no stretch was applied to the kinesiological tape, as it was only used as a bonding interface between the strain gauges matrix and the skin. This configuration provided better contact with the skin.

#### 5.4.2 Muscles Involved in Upper Limb Movement

As a proof of concept, one healthy adult subject (male, age: 25 years old, size: 1m70) participated to this study. The experimental procedure was approved by the *Ethic Board* of the *Research Center of Ste-Justine University Hospital Center*, in Montreal, Canada. The participant provided informed consent before the experiment and declared being in a good health.

Fig. 5.2 shows the placement of two bracelets, totalizing 12 strain gauges:

1. Strain gauge bracelet labeled  $B$  placed around the arm at the *biceps brachii* prominence.
2. Strain gauge bracelet labeled  $AB$  placed around the forearm 5 cm below the *elbow joint center*; on the forearm and arm.

In this paper, a strain gauge on a bracelet is identified by the label of the bracelet, i.e.,  $B$  or  $AB$ , followed by the gauge number from 1 to 6 shown in Fig. 5.1 (b). e.g.: The gauge  $B6$  refers to strain gauge number 6 on bracelet  $B$ .

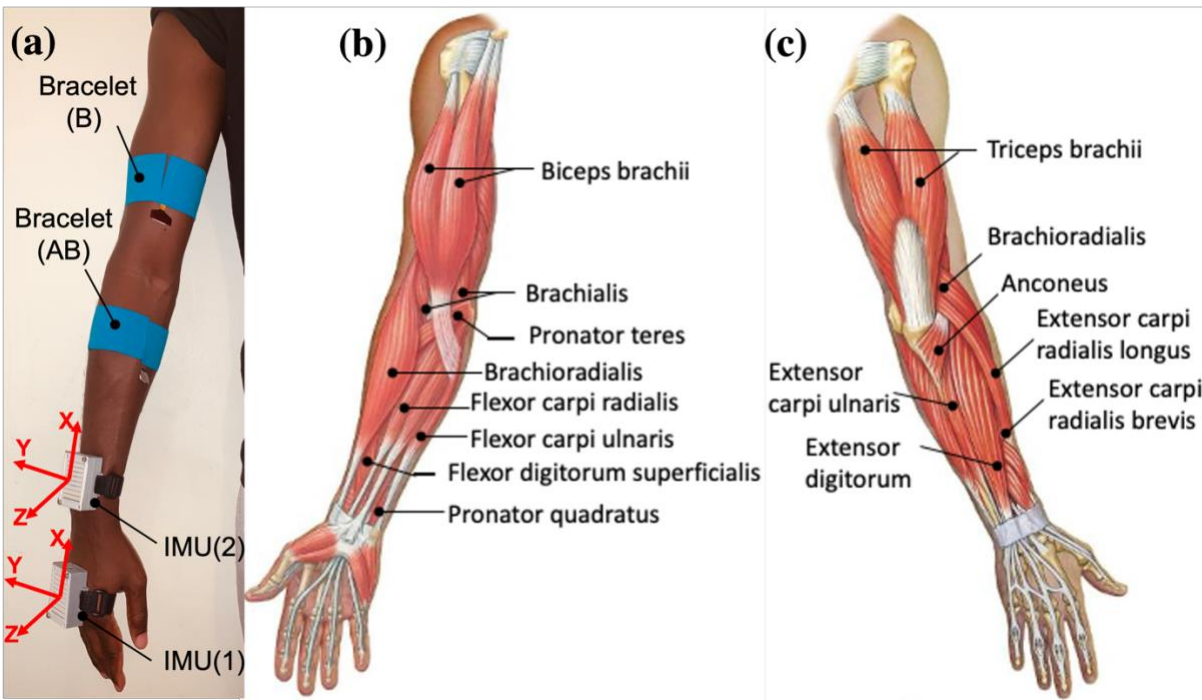


Figure 5.2 (a) Placement of the two strain gauge bracelets: bracelet  $B$  around the arm, and bracelet  $AB$  placed around the forearm; two IMUS: IMU (1) placed on the hand; IMU (2) placed on the forearm. (b) Anterior view of the human upper limb. (c) Posterior view of the human upper limb. The image (b) and (c) were adapted from [27].

In this study, 10 movements of the upper limb (Fig. 5.3) were selected. Table 5.1 identifies the muscles involved in each movement and the main sensors that were positioned to capture the movements.

The *biceps brachii*, *brachialis*, and *brachioradialis* muscles are responsible for flexing the forearm. The *triceps brachii* and *anconeus* muscles are responsible for extending the forearm. The gauge  $B6$  was placed on the center on the *biceps brachii* of the right arm, and the other gauges of strain gauge bracelet  $B$  were placed going round the arm ( $X$ -axis of Fig. 5.2) following the positive direction of the right-hand rule ( $B6, B5, B4, B3, B2, B1$ ).

A *supinator* is the muscle responsible for rotating the forearm so that the palm is facing up or forward. A *pronator* is the muscle that rotates the arm so that the palm is facing down or toward the back. The *extrinsic muscles* of the forearm allow movement of the wrist and hand. The muscles of the *posterior group* extend the hand to the level of the wrist; the muscles of the *anterior group* flex the hand at the wrist.

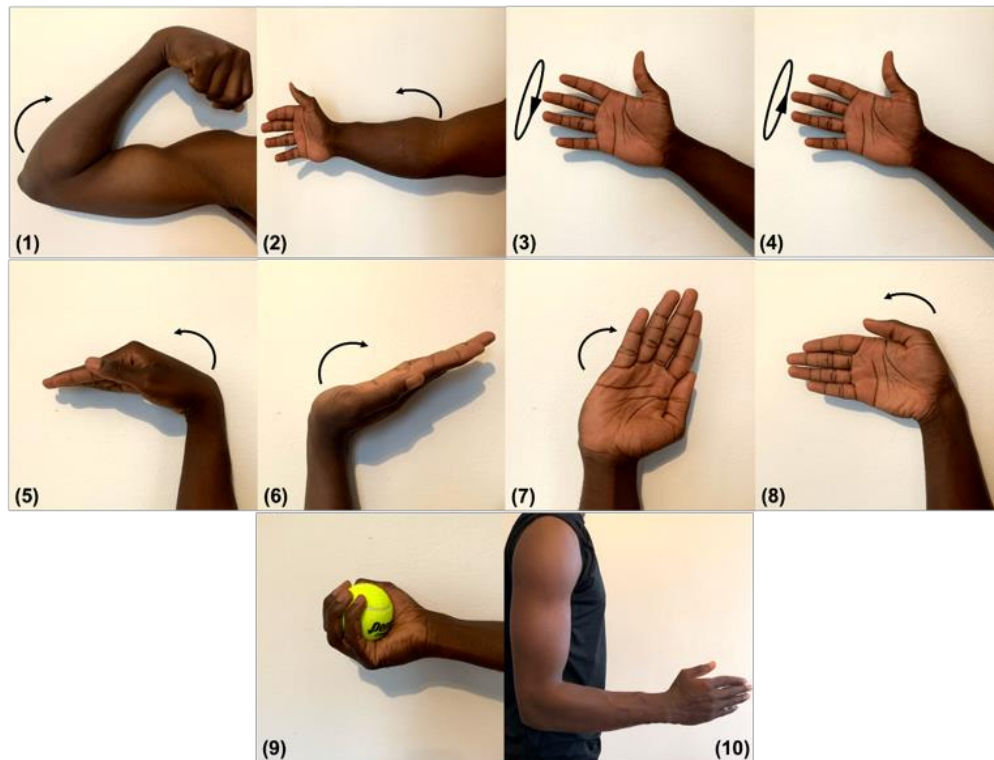


Figure 5.3 The 10 identified movements. (1) Elbow Flexion (EF). (2) Elbow Extension (EE). (3) Forearm Pronation (FP). (4) Forearm Supination (FS). (5) Wrist Flexion (WF). (6) Wrist Extension (WE). (7) Wrist Ulnar Deviation (WL). (8) Wrist Radial Deviation (WR). (9) Power grips (PG). (10). Rest position “no movement” (NM).

Tableau 5.1 Muscles involved in identified upper limb movements

		Main muscles involved	Main sensors
Elbow movements	EF	-Biceps brachii	-strain gauge bracelet (B)
		-Triceps brachii	
	EE	-Triceps brachii	-IMU (2)
		-Biceps brachii	
Forearm movements	FP	-Pronator teres	-strain gauge bracelet (AB)
		-Pronator quadratus	
	FS	-Supinator	-IMU (1)
		-Pronator Teres	
Wrist movements	WF	-Flexor carpi radialis	-strain gauge bracelet (AB)
		-Flexor carpi ulnaris	
		-Extensor carpi radialis longus	
	WE	-Extensor carpi radialis brevis	
		-Extensor digitorum	
	WL	-Extensor carpi ulnaris	
WR	-Abducto Polilicis longus		
	-Flexor carpi radialis		
Hand	PG	- extrinsic muscles of hand	-strain gauge bracelet (AB)



The gauge *AB6* was vertically aligned with B6, and the other gauges of strain gauge bracelet *AB* were placed going round the forearm (X-axis of Fig. 5.2) following the positive direction of the right-hand rule (*AB6, AB5, AB4, AB3, AB2, AB1*). The letters *AB* were used to identify this bracelet

## 5.5 Circuits and systems

For each strain gauge bracelet, six analog channels were necessary to record the deformation of the six strain gauges in real time. For inertial unit-based sensors, six signals were acquired, namely three linear accelerations and three angular velocities. All sensors had wireless communication.

### 5.5.1 Microcontroller

As presented in Fig. 5.4, the microcontroller was an ESP32 (Espressif Systems, China). It has a built-in USB-to-serial converter, a lithium ion/polymer charger, and general purpose input/output (GPIO). It contains a universal asynchronous receiver-transmitter (UART), a serial peripheral interface (SPI), and inter-integrated circuit (I2C) as a communication bus. I2C is a serial protocol having a two-wire interface for connecting low speed devices. The UART protocol was used to establish a communication between the microcontroller and the computer. On the computer, Matlab was run to save and visualize the incoming data. The ESP32 has a 240 MHz dual core processor and an integrated 520 KB SRAM that can perform the calculations with a large number of operations for onboard real-time movement identification. The I2C protocol was used to connect the microcontroller to the BNO055 IMU.

The ESP32 supports both WiFi and Bluetooth (Classic/LE), meaning that it is suitable to use for wireless projects. It comes with a proprietary communication protocol ESP-NOW, which enables a 2-way wireless communication between several ESP32 boards. This protocol was used as it is easy to implement, and the transmission frequency (>100Hz) is fast enough for our application.

### 5.5.2 IMU sensor design

As presented in Fig. 5.4, the system used the low-cost commercial Bosh Sensortec BNO055 IMU (Adafruit, USA) including a tridimensional (3D) accelerometer, a (3D) gyroscope, and a magnetometer. The work done by [28] motivated the choice of this sensor. Fig. 5.4 shows the wiring diagram.

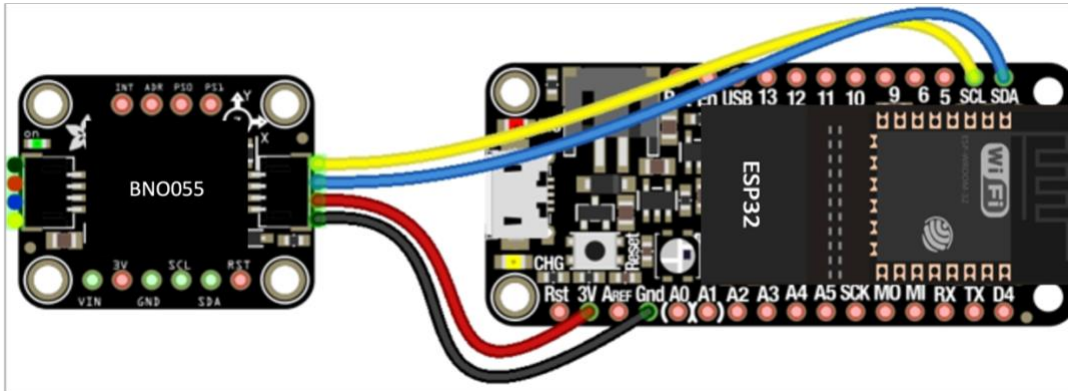


Figure 5.4 IMU circuit. BNO055 IMU (left) connected to the ESP32 microcontroller (right) using the I2C protocol. The BNO055 IMU has a 3,3V input that was connected directly into the ESP32 3,3V output (red wire). A ground (GND) pin was connected to GND on the ESP32 (black wire), the I2C clock pin (SCL) was connected to the corresponding ESP32 pin 22 (yellow wire), and the I2C data pin (SDA) was connected to the corresponding ESP32 pin 23 (blue wire). The image was adapted from [29].

The BNO055 IMU was connected to the ESP32 microcontroller by using an I2C protocol (Fig. 5.4). Two IMU sensors, namely IMU 1 and 2, were implemented by following this way.

The code for data acquisition was implemented by following the protocol provided by Adafruit [29], briefly: for each sensor, a BNO055 object was initialized; each sensor provided linear accelerations and angular velocities along the 3 Cartesian axes; a specific identifier was assigned to each sensor to identify the origin of the data following the ESP-NOW protocol [30].

To ensure that the data coming from the two BNO055 IMU were accurate, it was essential to calibrate these ones. The calibration was performed according to the calibration guide provided by MathWorks [31]. When the calibration process was completed, the code provided offset values. These offset values were then applied to each 3D axis of the two BNO055 IMU.

3D printed PLA boxes were used to hold the PCB for strain gauge signal conditioning (Fig. 5.5 (b)), and each IMU (Fig. 5.5 (b), and Fig. 5.2 (a) when strapped with a Velcro on the participant). The ESP32 has support for connecting a LiPoly/Lion battery. This terminal was connected to a Lipo 850 mAh battery (Adafruit, USA) that allowed the system to have an autonomy of 8 hours.

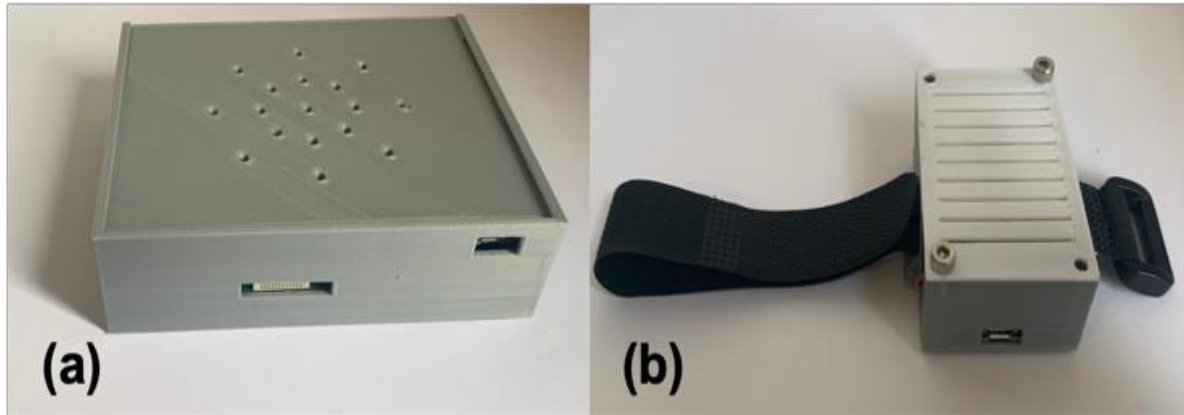


Figure 5.5 3D printed box for electronics. (a) Box (12x10x3.3 cm) containing the PCB for strain gauge signal conditioning. (b) Box (5.5x3.5x2.2 cm) containing the IMU circuit and a Velcro strap.

### 5.5.3 Strain gauge-based sensor

To record and treat the signal from the strain gauges, a PCB representing the electrical circuit (Fig. 5.6) was designed (see Fig. 5.7).

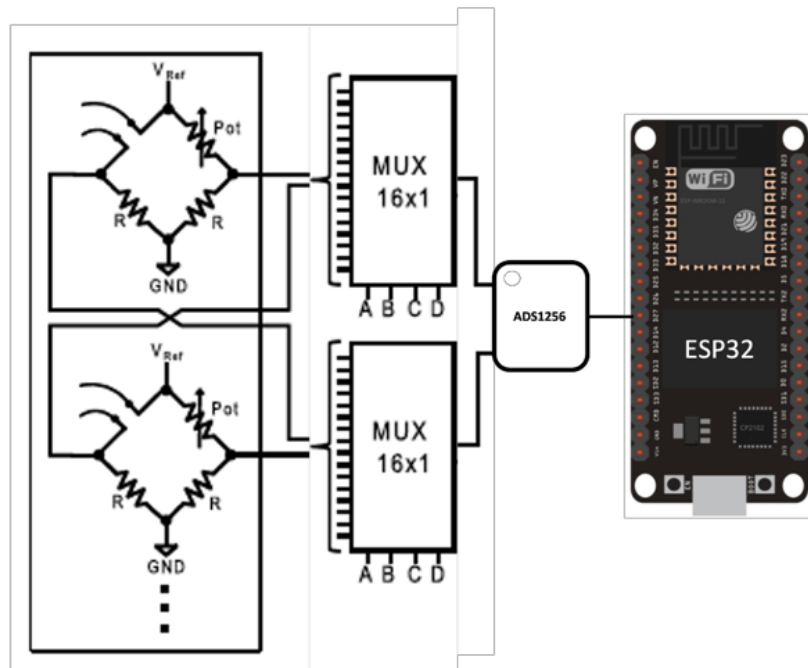


Figure 5.6 Electrical circuit for strain gauge signal conditioning showing the Wheatstone bridges in quarter bridge configuration, the multiplexers, the ADS1256 and the ESP32 microcontroller.

The PCB had a flat flexible cable connector for connecting the strain gauge bracelet. The PCB included six Wheatstone bridges in quarter bridge configuration. The strain gauge CF120-10AA had a resistance variation from 115 Ohms to 125 Ohms. The bridges converted small changes in resistance ( $\pm 5$  Ohms) of the gauges to a voltage. The resistors of this circuit had a nominal resistance value of 120 Ohm with a tolerance of  $\pm 1\%$ . Each bridge had a trimmer potentiometer whose resistance rating is 500 Ohms. This selected potentiometer had 20 turns allowing to have good precision of  $\pm 1$  Ohm which facilitated the bridge zeroing. The zeroing results to a zero-voltage output when no-strain is applied to the gauge. It is an important step to perform before using the system.

The signals from the gauges were then routed to two multiplexers. The multiplexers allow to choose the channel to be read by sequentially reading each of the analog inputs, as suggested by [15]. The selected CD74HC4067 multiplexer has sixteen channels controlled by 4 digital signals. The data from sixteen strain gauges can be acquired using only one input of a microcontroller. This system also allowed the use of a single amplifier and an analog-to-digital converter. The size of the acquisition system was thus reduced. It also saved equipment costs by using a single amplifier for several gauges.

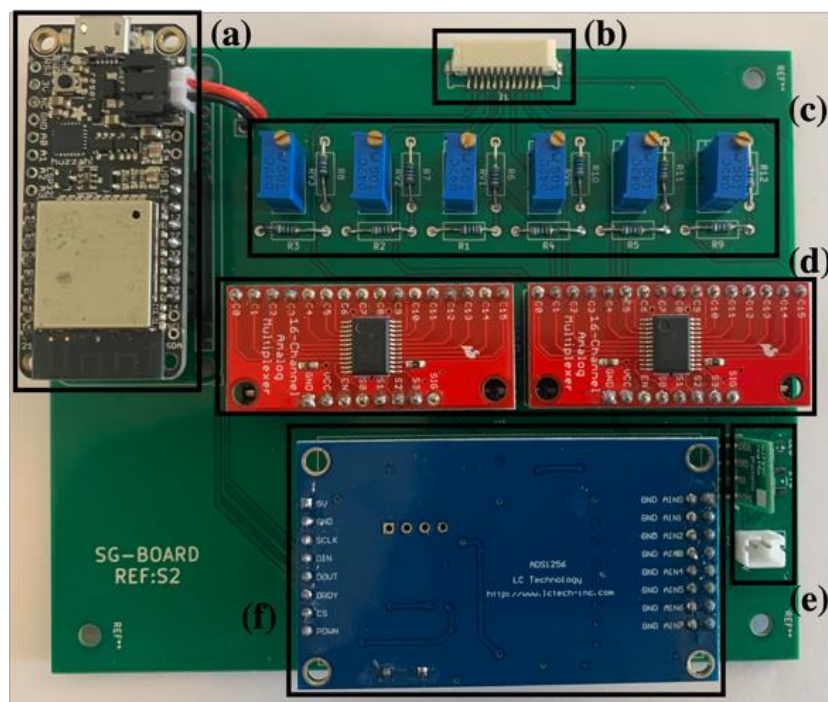


Figure 5.7 PCB for strain gauge signal acquisition. (a) ESP32. (b) FFC/FPC connector. (c) Wheatstone Bridge. (d) Multiplexers. (e) Power management. (f) ADS1256.

These analog signals were acquired and processed by the ADS1256. The ADS1256 is a low-noise 24-bit analog-to-digital converter. It has a high acquisition frequency of 30 kSPS and can acquire data from 8 asymmetric inputs or 4 differential inputs. The programming of this goes through the ESP32 with SPI communication. The ESP32 microcontroller then retrieves the data from the ADS1256 in 24-bit digital format. It sends them to the computer by UART communication. Additionally, a 3,3 V voltage regulator has been added to provide a stable voltage source for the Wheatstone's bridges. The stability of the voltage source is important to ensure the stability of Wheatstone's bridges.

This acquisition board was powered by a LiPo (Lithium Polymer) battery (Adafruit, USA) with a capacity of 5000 mAh at 3,7 V and a power of 18.5 W. This battery allowed the acquisition system an autonomy of 24 hours. A 3D printed PLA box was used to hold all electronic devices (see Fig. 5.5(a)).

#### **5.5.4 Signal Acquisition**

The sensors used wireless communication based on the ESP-NOW protocol which enables multiple devices to communicate without using Wi-Fi. The communication architecture is presented in Fig. 5.8.

A total of 12 signals from the strain gauges were recorded for each movement. Additionally, to these signals, the tridimensional linear accelerations and angular velocities coming from two IMUs were used. Combining the signals from the strain gauges and inertial units, a total of 24 signals were obtained. The data were sampled at a frequency of 100 Hz. This frequency was chosen to limit the amount of data to be processed. The human movement frequency is contained between 0 and 20 Hz [32]; hence the Shannon Nyquist sampling theorem was respected. A Butterworth-type digital low-pass filter of order 4 with a cut-off frequency of 1 Hz [15] was used to filter the signal from the strain gauges. The first hundred data for each gauge when the participant is at rest was averaged and used to set an offset for the following data. The raw signals from IMUs were used.

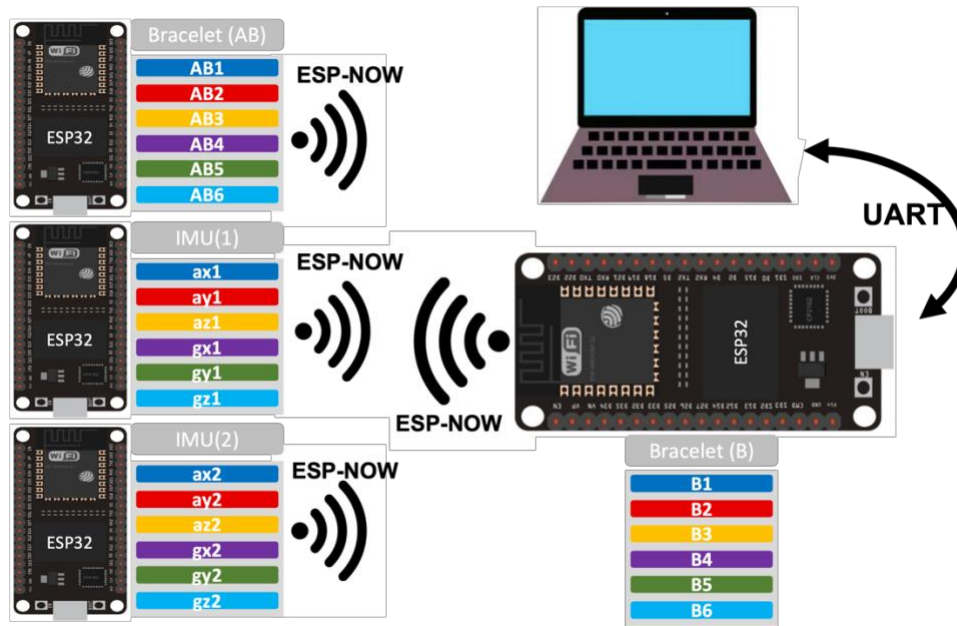


Figure 5.8 Communication architecture including the ESP32 microcontroller, the two strain gauge bracelets  $B$  and  $AB$ , the two IMUs 1 and 2, the communication protocols ESP-NOW (between the ESP32 and the bracelets and IMUs) and UART (between the ESP32 and the computer), and the variables corresponding to the sensors output: e.g.  $AB1$  to  $AB6$  for the strain gauge bracelet  $AB$ ,  $B1$  to  $B6$  for the strain gauge bracelet  $B$ .  $ax$  for linear acceleration following X-axis,  $ay$  for linear acceleration following Y-axis,  $az$  for linear acceleration following Z-axis.  $gx$  for angular velocity around the X-axis,  $gy$  for angular velocity around Y-axis,  $gz$  for angular velocity around Z-axis. The number 1 or 2 referred to IMU 1 or IMU 2.

## 5.6 Results

To analyze the ability of this system to identify movement intentions, a series of movements was performed. The sequence was as follows  $EF-NM-EE-NM-FP-NM-FS-NM-WF-NM-WE-NM-WL-NM-WR-NM-PG-NM$ . After each movement, a pause with “no movement” ( $NM$ ) was observed before doing the next movement.

Fig. 5.9 shows the data acquired during the sequence of movements performed by the participant. Fig. 5.9 (a) shows the 3 linear accelerations and 3 angular velocities coming from IMU 1 which was placed on the hand. Fig. 5.9 (b) shows the 3 linear accelerations and 3 angular velocities coming from IMU 2 which was placed on the forearm. Fig. 5.9 (c) shows the data acquired by the strain gauge bracelet placed on the arm. Fig. 5.9 (d) shows the data acquired by the strain gauge bracelet placed on the forearm.

In Table 5.2, the maximum voltage variation for each strain gauge was computed per movement. The strain gauges that recorded a voltage variation ( $\Delta V$ ) superior to  $\pm 1$  mV were identified. This threshold was established to compensate the error introduced by the manual zeroing of the Wheatstone bridge.

Furthermore, using the information presented in Table 5.2 the maximum  $\Delta V$  in both bending directions were computed per column to determine which strain gauges recorded more skin deformation for a particular movement, and per row to determine which movement produced the maximum  $\Delta V$  for each gauge. The results showing the strain gauge signal intensity pattern are displayed in Table 5.3.

- 1) *Elbow Flexion (EF)*: The largest deformations were recorded by the strain gauges *B6*, *AB6* and *AB1*, with 4.5 mV, 6.03 mV, and 4.48 mV, respectively. Compared to the others strain gauges, the strain gauges *AB6* and *AB3* recorded the largest deformations for this movement in both bending directions. Compared to the other movements, the strain gauges *B6*, *AB1*, and *AB6* recorded their highest convex deformation and *B3* recorded its highest concave deformation. IMU 1 and IMU 2 recorded comparable angular velocities  $gz1$ , and  $gz2$  around the Z-axis
- 2) *Elbow Extension (EE)*: The largest deformations were recorded by the strain gauges *B1*, *B5*, and *B6* with -4.11mV, -2.99mV and -1.98mV, respectively. Compared to the other strain gauges, the strain gauges *AB6* and *AB1* recorded the largest deformations for this movement in the two bending directions. Compared to the other movements, the strain gauges *B1*, *B2*, and *B5* recorded their largest concave deformation. IMU 1 and IMU 2 recorded comparable angular velocities  $gz1$ , and  $gz2$  around the Z-axis.
- 3) *Forearm Pronation (FP)*: The largest deformations were recorded by the strain gauges *B2* and *B4* with 1.67 mV, 1.04 mV, respectively and *AB1* and *AB3* with 1.61 mV, 0.95 mV respectively recorded the largest deformations. Compared to the other strain gauges, the strain gauges *B2* and *B3* recorded the largest deformations for this movement in both bending directions. Compared to the other movements, the strain gauge *B2*, recorded its largest convex deformation. IMU 1 and IMU 2 recorded comparable angular velocities  $gx1$  and  $gx2$  around the X-axis.

- 4) *Forearm Supination (FS)*: The largest deformations were recorded by the strain gauges *AB3*, *AB1*, *AB6* and *AB2*, with -1.66mV, 1.30 mV, -1.14mV, 1.02 mV respectively and *B6* with -1.14mV recorded the largest deformations. Compared to the other strain gauges, the strain gauges *B6*, and *AB3* recorded the largest deformations for this movement in both bending directions. None of the gauges recorded their maximal deformation during *forearm supination*. IMU 1 and IMU 2 recorded comparable angular velocities  $gx1$ , and  $gx2$  around the X-axis.
- 5) *Wrist Flexion (WF)*: The largest deformations were recorded by the strain gauges *AB1*, *AB2* and *AB5* with 2.41 mV, 1.36 mV, -1.89mV respectively and *B6* with 1.81 mV recorded the largest deformations. Compared to the other strain gauges, the strain gauges *AB1* and *AB5* recorded the largest deformations for this movement in both bending directions. Compared to other movements, the strain gauge *AB5* recorded its largest concave deformation. IMU 1 recorded angular velocities  $gy1$  around the Y-axis. IMU 2 placed on the forearm and did not record any significant movement.
- 6) *Wrist Extension (WE)*: The largest deformations were recorded by the strain gauges *B4* and *B6* with 1.34 mv, 1.35 mV, respectively as well as *AB1*, *AB2*, *AB3* and *AB4* with -3.40mV, 1.55 mV, 1.52 mV, 1.06 mV respectively recorded the largest deformations. Compared to the other strain gauges, *AB2*, *AB6* recorded the largest deformations for this movement in both bending directions. Compared to the other movements, strain gauge *B4* recorded its largest convex deformation. IMU 1 recorded angular velocities  $gy1$  around the Y-axis. IMU 2 placed on the forearm did not record any significant movement.
- 7) *Wrist Ulnar deviation (WL)*: The largest deformations were recorded by the strain gauge *B6* with 1.39 mV as well as *AB1*, *AB2* and *AB3* with 2.65 mV, 1.58 mV, 2.19 mV respectively recorded the largest deformations. Compared the other strain gauges, *AB1* and *B3* recorded the largest deformations for this movement in both bending directions. Compared to the other movements, the strain gauges *B1* and *AB3* recorded their largest concave deformations. IMU 1 and IMU 2 recorded comparable angular velocities  $gz1$  and  $gz2$  around the Z-axis.
- 8) *Wrist Radial deviation (WR)*: The largest deformations were recorded by the strain gauges *B2* and *B6*, with 1.26 mV, 1.02 mV respectively as well as *AB1*, *AB2*, *AB4*, *AB5* and *AB6*



with  $-4.32\text{mV}$ ,  $1.67\text{ mV}$ ,  $1.22\text{ mV}$  and  $1.14\text{ mV}$  respectively recorded the largest deformations. Compared to the other strain gauges,  $AB2$  and  $AB1$  recorded the largest deformations for this movement in both bending directions. Compared to the other movements, the strain gauges  $B5$ ,  $AB4$ , and  $AB5$  recorded their largest convex deformation and  $AB1$ ,  $AB6$  recorded its largest concave deformation. IMU 1 and IMU 2 recorded comparable angular velocities  $gz1$ , and  $gz2$  around the Z-axis.

- 9) *Power Grips (PG)*: The largest deformations were recorded by the strain gauges  $AB1$ ,  $AB2$  and  $AB3$  with  $1.71\text{ mV}$ ,  $1.81\text{ mV}$  and  $-2.32\text{ mV}$  respectively recorded the largest deformations. Compared to the other strain gauges,  $AB2$ ,  $AB3$  recorded the largest deformations for this movement in both bending directions. Compared to the other movements, the strain gauges  $AB2$  recorded its largest convex deformation and  $AB3$  recorded its largest concave deformation. IMU 1 recorded its angular velocities  $gy1$  around the Y-axis. IMU 2 placed on the forearm did not record any significant movement.

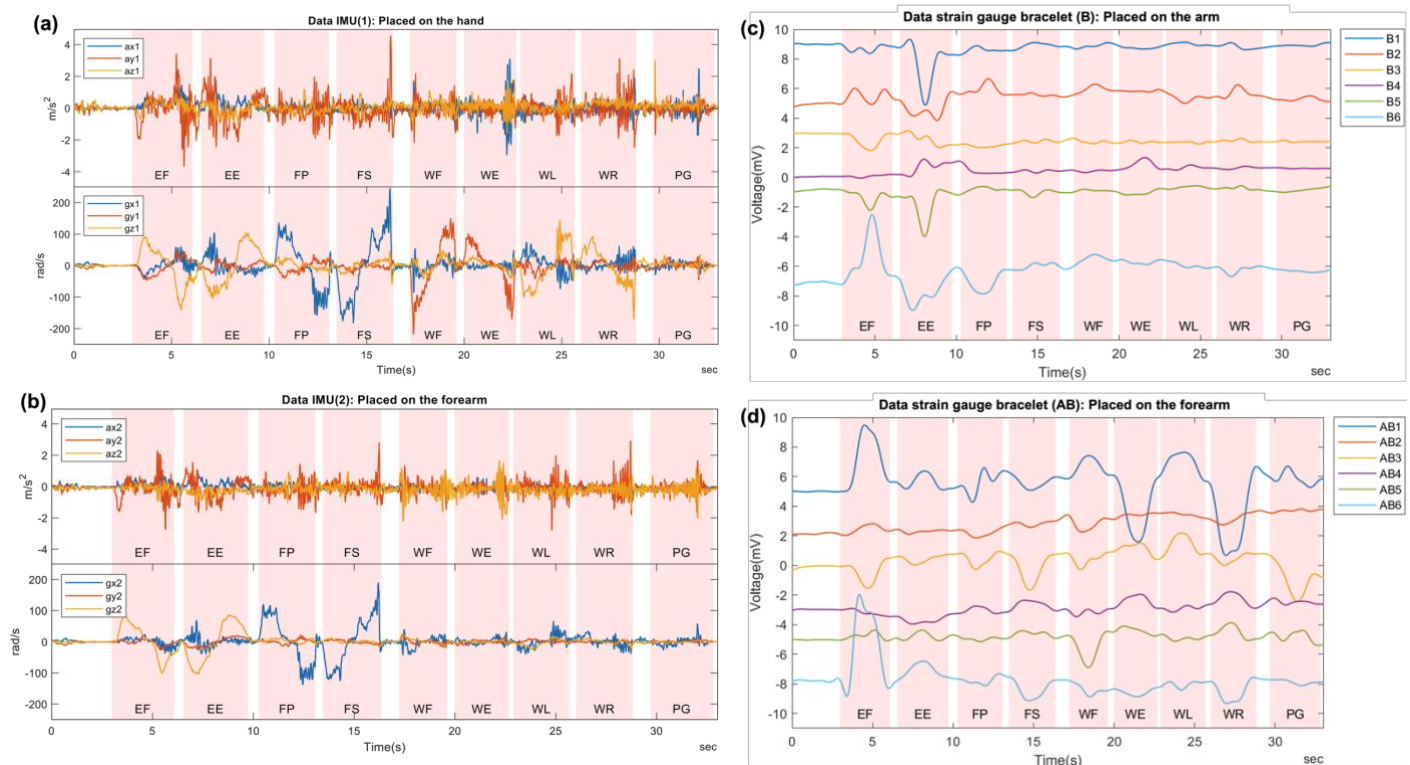


Figure 5.9 (a) 3D linear accelerations and 3D angular velocities from the IMU 1 placed on the hand. (b) 3D linear accelerations and 3D angular velocities from the IMU 2 placed on the forearm. (c) Data from the six strain gauges in the bracelet  $B$  placed on the arm. (d) Data from the six strain gauges in the bracelet  $AB$  placed on the forearm.

Tableau 5.2 Peak signal intensity for each strain gauge per movement(mV)

	EF	EE	FP	FS	WF	WE	WL	WR	PG
B1	-0.62	<b>-4.11</b>	-0.71	-0.43	-0.32	-0.32	0.13	-0.37	-0.14
B2	<b>1.02</b>	<b>-1.15</b>	<b>1.67</b>	0.72	<b>1.30</b>	0.94	0.74	<b>1.26</b>	0.51
B3	<b>-1.19</b>	-0.98	-0.98	-0.70	-0.72	-0.69	-0.74	-0.62	-0.63
B4	0.21	<b>1.24</b>	<b>1.09</b>	0.52	0.50	<b>1.34</b>	0.85	0.76	0.66
B5	<b>-1.22</b>	<b>-2.99</b>	0.39	-0.36	0.20	0.26	0.44	0.45	0.35
B6	<b>4.50</b>	<b>-1.98</b>	-0.85	<b>1.25</b>	<b>1.81</b>	<b>1.45</b>	<b>1.39</b>	<b>1.02</b>	0.80
AB1	<b>4.48</b>	<b>1.38</b>	<b>1.61</b>	<b>1.30</b>	<b>2.41</b>	<b>-3.40</b>	<b>2.65</b>	<b>-4.32</b>	<b>1.71</b>
AB2	0.82	0.44	0.60	<b>1.01</b>	<b>1.36</b>	<b>1.55</b>	<b>1.58</b>	<b>1.67</b>	<b>1.81</b>
AB3	<b>-1.54</b>	0.76	<b>0.95</b>	<b>-1.66</b>	0.89	<b>1.52</b>	<b>2.19</b>	0.85	<b>-2.42</b>
AB4	-0.45	-0.95	-0.26	0.65	-0.46	<b>1.06</b>	0.30	<b>1.22</b>	0.79
AB5	0.65	0.64	0.24	0.59	<b>-1.89</b>	0.91	0.35	<b>1.14</b>	0.62
AB6	<b>6.03</b>	<b>1.54</b>	0.69	<b>-1.14</b>	-0.88	-0.86	-0.41	<b>-1.32</b>	0.29

Tableau 5.3 Strain gauge signal intensity pattern

	Maximum voltage variation per strain gauge		Maximum voltage variation per movement	
	(+) mV max	(-) mV Min	(+) mV max	(-) mV max
EF	B6, AB1, AB6	B3	AB6	AB3
EE	-	B1, B2, B5, B6, AB4	AB6	AB1
FP	B2	-	B2	B3
FS	-	-	B6	AB3
WF	-	AB5	AB1	AB5
WE	B4	-	AB2	AB6
WL	B1, AB3	-	AB1	B3
WR	B5, AB4, AB5	AB1, AB6	AB2	AB1
PG	AB2	AB3	AB2	AB3

## 5.7 Discussion

The objective of this study was to develop a novel wearable system to identify intentions of movement by combining strain gauges and inertial measurement units. The following discussion addresses A. the ability of this system to identify movement intentions, and B. the design of the sensor system combining strain gauges and IMUs.

### 5.7.1 Identification of movement identification

Table 4.1 identifies the muscles responsible for each movement. Fig. 5.2 shows the placement of the sensors regarding the muscles involved in the upper limb motion. A mapping was established between the movements performed by the subject (Fig. 5.3) and the signals recorded by the strain gauges (Fig. 5.9).

- 1) *Elbow Flexion (EF)*: The strain gauge *B6* was placed on the *biceps brachii* which explains the high intensity signal (Fig. 5.9). By going around the arm with the strain gauge bracelet

(AB), AB1 and AB6 were placed near the *brachioradialis*, which explained the obtained deformation. However, the gauge B6 was expected to record the largest deformation as the *biceps brachii* is the main muscle responsible for the forearm flexion. By folding the forearm on the arm (Fig. 5.3 (1)), the *biceps brachii* came into contact with the forearm which explains that the gauge AB6 recorded more deformations.

- 2) *Elbow Extension (EE)*: The strain gauges B1 and B6 were placed on the *biceps brachii* which extends during the *elbow extension*. It is therefore normal to record deformations in the opposite direction to the bending movement for gauges B6 and B1 (Fig. 5.9(c)). The signal of B6 went from positive for elbow flexion to negative for elbow extension. The strain gauge B5 was placed close to the *Brachialis* which also extends during *elbow extension*. The gauges AB1 and AB6 were placed near the *Brachioradialis* which have recorded deformations. These measurements can be explained by the nature of the extension movement which tends to stretch the forearm skin at full range of motion.
- 3) *Forearm Pronation (FP)*: The strain gauges AB1 and AB3 were placed around the *pronator teres* which is mainly responsible for the *forearm pronation*, recording therefore signals with high intensity. The *biceps brachii* is also involved in the *forearm pronation*. The biceps movement creates large skin deformation of the skin around the arm which matches the high intensity signals recorded by strain gauges B2 and B4.
- 4) *Forearm Supination (FS)*: The strain gauges AB1, AB2 and AB3 were located around the *supinator*, which is responsible for the *forearm supination*. The strain gauges AB1 and AB3 were involved in both pronation, and supination. B6 is centered on the *biceps brachii*, which is involved in this movement. The *forearm pronation* and *supination* are like a twisting movement around the forearm axis, which causes the forearm skin to stretch, recording therefore relevant deformations.

Most of the muscles involved in the subsequent movements are in the forearm. Hence, the strain gauges of the bracelet (B) placed on the arm recorded decreasingly weak signals (Fig. 5.9 (c)). The forearm muscles are packed side by side or overlapped so it was more difficult to make a link between the muscles involved in the movement and of the strain gauges positions

- 5) *Wrist Flexion (WF)*: The strain gauges *AB1*, *AB2*, and *AB5* were close to the muscle group involved in *wrist flexion*. The gauge *AB1* gauge is centered on the *flexor carpi radialis*, which explains a stronger signal than the other strain gauges.
- 6) *Wrist Extension (WE)*: The strain gauges *AB1*, *AB2*, *AB3*, and *AB4* were near the muscles involved in the *wrist extension*. *AB1* is near the *extensor carpi ulnaris*, and the *digitorum extender* which explains a stronger signal than the other gauges.
- 7) *Wrist Ulnar deviation (WL)*: The muscles responsible for the ulnar deviation of the wrist is the *extensor Carpi ulnaris*. The strain gauges *AB1* and *AB3* were placed near this muscle, which explains the recorded deformations.
- 8) *Wrist Radial deviation (WR)*: The muscles responsible for the radial deviation of the wrist are the *flexor carpi radialis*, *extensor carpi radialis longus*, *extensor carpi radialis Brevis*. This muscles group goes almost all around the forearm, which explains the deformations recorded by most strain gauges from the forearm bracelet *AB*.
- 9) *Power Grips (PG)*: The gripping movement mainly involves the extrinsic and intrinsic hand muscles. Some forearm muscles (e.g., *flexor digitorum superficialis*) are involved in power grips which explains the signals recorded by strain gauges *AB1*, *AB2*, and *AB3*.

The data from IMU 1 and 2 (Fig 5.9 (a), (b)) provided additional information about the motion. Although it was difficult to have a visual interpretation of the linear acceleration's data, the angular velocities provided information used to derive a relation between the graphs obtained and the movements.

For the movements of *EF*, *EE*, *FP* and *FS* it was expected to register comparable signals of linear accelerations and angular velocities from the two IMUs placed at the forearm and arm. Indeed, for these movements the IMUs were aligned along the same axis (Fig, 5.2 (a)) and the wrist remained in neutral position which was similar to two IMUs placed on a rigid bar. Observing the graphs confirmed our assertion. These movements were performed around the same axe in opposite direction, and a change of sign was clearly noticed in the signals (Fig. 5.9 (a), (b))

The movements of *WF*, *WE*, *WR*, *WL* showed a quasi-static angular acceleration for the IMU placed on the forearm (Fig. 5.9 (b)). Indeed, only the wrist performed these movements, so that the IMU 1 placed on the hand recorded accelerations. The gripping movement is an opening and closing of

the hand, so neither acceleration nor velocities should be relevantly recorded by the IMUs. As the participant did not maintain a perfect static position, slight accelerations were observed in (Fig. 5.9 (a), (b)).

The results showed that the strain gauge bracelet placed on the forearm recorded signals that could make a distinction between wrist movements and the power grips movement. The strain gauge bracelet placed on the arm recorded signals that could make a distinction between various forearm movements, e.g. pronation vs supination. Also, the combination of strain gauges that recorded maximum  $\Delta V$  is unique for each movement (see Table 5.3). This uniqueness represents a pattern to identify upper limb movement.

It is worth noting in Table 5.3 that:

- 1) There was no strain gauge which measured its highest deformation for *forearm supination*. Hence, no strain gauge was optimally placed to detect *forearm supination*.
- 2) The strain gauge *AB6* was the most impacted one by *elbow flexion*, recording then its maximal  $\Delta V$ , *AB6* is a good discriminant for the *elbow flexion*.
- 3) The strain gauge *B2* was the most impacted one by the *forearm pronation*, recording then its maximal  $\Delta V$ . *B2* is a good discriminant for the *forearm pronation*.
- 4) The strain gauge *AB5* was the most impacted one by the *wrist flexion*, recording then its maximum  $\Delta V$ . *AB5* is a good discriminant for the *wrist flexion*.
- 5) The strain gauge *AB1* was the most impacted one by the *wrist radial deviation*, recording then its maximum  $\Delta V$ . *AB1* is a good discriminant for the *wrist radial deviation*.
- 6) The strain gauges *AB2* and *AB3* were the most impacted one the *power grips*, recording then their maximum  $\Delta V$ . These ones are good discriminants for the *power grasp*.

The impacts of adding IMUs to the system can be summarized as follows:

- 1) The angular velocities allowed the identification of the beginning and end of each movement in the sequence.
- 2) The position of the limb could be inferred by using angular velocities.

- 3) The linear acceleration contribution is unclear at this point. However, further analysis using different techniques of artificial intelligence, such as machine learning with deep neural network, could provide more information regarding the kinematics data.

The strain gauges provided information about muscle contraction and the IMUs provided information about the motion. Comparing to the work done by [15], [23]. These results showed the ability of this sensor to record signals making it possible to distinguish 9 movements of the upper limb and therefore confirming the initial RH.

### 5.7.2 Sensor design

The proposed system was compact and portable making it ideal for daily activities (Fig. 5.7 (a), (b)). The developed bracelet used strain gauges and a kinesiological tape which are commercially readily available. The IMU 1 placed on the hand was not optimal as it was relatively large and cumbersome. However, a smaller IMU system could solve this issue.

Each strain gauge in the matrix collected a specific signal. The third and fourth maximum  $\Delta V$  (see Table 5.2) were recorded by gauges  $BI(4.48 \text{ mV})$  and  $ABI(-4.11\text{mV})$  which were placed horizontally (Fig. 4.1). Aligning the strain gauges in the direction of the highest strain [15] was not optimal since the sensor did not record the deformation about the other axis. This information can be very useful for a future classification algorithm to distinguish between different movements.

The main design requirements (DR) of this system were achieved:

DR 1: It must be able to measure multi-axial skin deformations: Indeed, measuring strain on multiple axes is therefore important to better capture skin deformation due to muscle activity.

DR 2: It must be wearable. Indeed, the flexible PCB-based strain gauge matrix fabrication method is systematic, so the shape, size and orientation of the gauges can be changed depending on the application. Different strain gauge matrices can be made by using flexible PCBs and placed in different sites on the human body to collect different information.

DR 3: It must be able to be used several times in daily three-dimensional movements: Indeed, the flexible PCB has also provided a good solid connection with the gauges that prevented the connections to break or unsolder during movements as a solution to the robustness issues noted in [15].

The number of strain gauges, their orientation as well as the position of the strain gauge bracelet are important factors that can be optimized by referring to knowledge of human anatomy. This knowledge does not need to be specific as with sEMG which requires the sensors to be placed on the right muscle to detect the right signal [12]. The developed system had 12 channels of strain gauges, and more strain gauges can be added if necessary. Contrarily, it would be nearly impossible/or too bulky to place the same number of sEMG electrodes on a person without interfering with his daily activities. Also, the frequency of data acquisition of the proposed system was customizable between 40 Hz to 200 Hz which is far less than sEMG sampling frequency (1000 Hz) [14]. A technical limitation related to the use of Wheatstone bridge was the need to recalibrate each bridge after a certain amount of uses. A subsequent work can investigate that issue.

## **5.8 Conclusion**

The objective of this research was to develop a novel wearable system to identify intentions of movement by combining strain gauges and inertial measurement units. The proposed system is composed of 1. two strain gauge bracelets using 6 strain gauges each, connected to a flexible printed circuit board and 2. two inertial measurement units. To our knowledge, this system is the first portable battery-powered sensor using on-board electronics for strain gauge signal conditioning to detect human movement intentions. The system was tested on the upper limb, and successfully identified 9 main movements through the variations in signal intensity of the strain gauges. These results show the potential of such a sensory system to become a smart wearable sensory system to detect human movement intention. The future perspectives will be to extend the system, e.g., to the lower limbs, and to identify complex movement combinations by using pattern recognition algorithms with such sensory systems. Future applications could target human computer interfacing.

## **5.9 Acknowledgment**

The authors would like to thank the team of the Marie Enfant Research Center, Sainte-Justine University Hospital Center.

## **5.10 References**

[1] E. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered



- upper-limb prostheses: State of the art and challenges for clinical use,” *J. Rehabil. Res. Dev.*, vol. 48, no. 6, pp. 643–660, 2011, doi: 10.1682/JRRD.2010.09.0177.
- [2] R. M. Singh, S. Chatterji, and A. Kumar, “A review on surface EMG based control schemes of exoskeleton robot in stroke rehabilitation,” *Proc. - 2013 Int. Conf. Mach. Intell. Res. Adv. ICMIRA 2013*, pp. 310–315, 2014, doi: 10.1109/ICMIRA.2013.65.
- [3] A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, “Prosthetic Myoelectric Control Strategies: A Clinical Perspective,” *Curr. Surg. Reports*, vol. 2, no. 3, 2014, doi: 10.1007/s40137-013-0044-8.
- [4] Ottobock, “Bebionic,” 2021. <https://shop.ottobock.ca/en/Prosthetics/Upper-Limb-Prosthetics/bebionic/c/2888> (accessed Oct. 27, 2021).
- [5] P. Geethanjali, “Myoelectric control of prosthetic hands: State-of-the-art review,” *Med. Devices Evid. Res.*, vol. 9, pp. 247–255, 2016, doi: 10.2147/MDER.S91102.
- [6] M. Atzori and H. Müller, “Control capabilities of myoelectric robotic prostheses by hand amputees: A scientific research and market overview,” *Front. Syst. Neurosci.*, vol. 9, no. Nov, pp. 1–7, 2015, doi: 10.3389/fnsys.2015.00162.
- [7] B. Hudgins, P. Parker, and R. N. Scott, “A new strategy for multifunction myoelectric control,” *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, 1993, doi: 10.1109/10.204774.
- [8] S. M. Engdahl, B. P. Christie, B. Kelly, A. Davis, C. A. Chestek, and D. H. Gates, “Surveying the interest of individuals with upper limb loss in novel prosthetic control techniques,” *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–11, 2015, doi: 10.1186/s12984-015-0044-2.
- [9] O. W. Samuel *et al.*, “Pattern recognition of electromyography signals based on novel time domain features for amputees’ limb motion classification,” *Comput. Electr. Eng.*, vol. 67, pp. 646–655, 2018, doi: 10.1016/j.compeleceng.2017.04.003.
- [10] L. J. Hargrove, G. Li, K. B. Englehart, and B. S. Hudgins, “Principal Components Analysis Preprocessing for Improved Classification Accuracies in Pattern-Recognition-Based Myoelectric Control,” *IEEE Trans. Biomed. Eng.*, vol. 56, no. 5, pp. 1407–1414, 2009, doi: 10.1109/TBME.2008.2008171.

- [11] O. Politi, I. Mporas, and V. Megalooikonomou, “Comparative evaluation of feature extraction methods for human motion detection,” *IFIP Adv. Inf. Commun. Technol.*, vol. 437, pp. 146–154, 2014, doi: 10.1007/978-3-662-44722-2\_16.
- [12] E. Scheme and K. Englehart, “Training Strategies for Mitigating the Effect of Proportional Control on Classification in Pattern Recognition Based Myoelectric Control,” *J. Prosthet. Orthot.*, vol. 25, no. 2, pp. 76–83, Apr. 2013, doi: 10.1097/JPO.0b013e318289950b.
- [13] S. Krishnan and Y. Athavale, “Trends in biomedical signal feature extraction,” *Biomed. Signal Process. Control*, vol. 43, pp. 41–63, 2018, doi: 10.1016/j.bspc.2018.02.008.
- [14] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, “Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey,” *23th Int. Conf. Archit. Comput. Syst. 2010, ARCS 2010 - Work. Proc.*, pp. 167–176, 2010.
- [15] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, “Techniques of EMG signal analysis: Detection, processing, classification and applications,” *Biol. Proced. Online*, vol. 8, no. 1, pp. 11–35, 2006, doi: 10.1251/bpo115.
- [16] A. Phinyomark, R. N. Khushaba, and E. Scheme, “Feature extraction and selection for myoelectric control based on wearable EMG sensors,” *Sensors (Switzerland)*, vol. 18, no. 5, pp. 1–17, 2018, doi: 10.3390/s18051615.
- [17] S. Guo, M. Pang, B. Gao, H. Hirata, and H. Ishihara, “Comparison of sEMG-based feature extraction and motion classification methods for upper-limb movement,” *Sensors (Switzerland)*, vol. 15, no. 4, pp. 9022–9038, 2015, doi: 10.3390/s150409022.
- [18] M. A. Oskoei and H. Hu, “Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 8, pp. 1956–1965, 2008, doi: 10.1109/TBME.2008.919734.
- [19] A. D. C. Chan and K. B. Englehart, “Continuous myoelectric control for powered prostheses using hidden Markov models,” *IEEE Trans. Biomed. Eng.*, vol. 52, no. 1, pp. 121–124, Jan. 2005, doi: 10.1109/TBME.2004.836492.
- [20] L. J. Hargrove, E. J. Scheme, K. B. Englehart, and B. S. Hudgins, “Multiple binary classifications via linear discriminant analysis for improved controllability of a powered

- prosthesis,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 1, pp. 49–57, 2010, doi: 10.1109/TNSRE.2009.2039590.
- [21] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, “Classification of the myoelectric signal using time-frequency based representations,” *Med. Eng. Phys.*, vol. 21, no. 6, pp. 431–438, 1999, doi: [https://doi.org/10.1016/S1350-4533\(99\)00066-1](https://doi.org/10.1016/S1350-4533(99)00066-1).
- [22] G. Gaudet, M. Raison, and S. Achiche, “Classification of Upper limb phantom movements in transhumeral amputees using electromyographic and kinematic features,” *Eng. Appl. Artif. Intell.*, vol. 68, no. April 2017, pp. 153–164, 2018, doi: 10.1016/j.engappai.2017.10.017.
- [23] M. Atzori, M. Cognolato, and H. Müller, “Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands,” *Front. Neurobot.*, vol. 10, no. SEP, pp. 1–10, 2016, doi: 10.3389/fnbot.2016.00009.
- [24] M. Atzori *et al.*, “Characterization of a benchmark database for myoelectric movement classification,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 1, pp. 73–83, 2015, doi: 10.1109/TNSRE.2014.2328495.
- [25] Y. Wu, B. Zheng, and Y. Zhao, “Dynamic Gesture Recognition Based on LSTM-CNN,” in *2018 Chinese Automation Congress (CAC)*, 2018, pp. 2446–2450, doi: 10.1109/CAC.2018.8623035.
- [26] O. Barron, M. Raison, G. Gaudet, and S. Achiche, “Recurrent Neural Network for electromyographic gesture recognition in transhumeral amputees,” *Appl. Soft Comput. J.*, vol. 96, p. 106616, 2020, doi: 10.1016/j.asoc.2020.106616.
- [27] D. Farina *et al.*, “The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 797–809, 2014, doi: 10.1109/TNSRE.2014.2305111.
- [28] A. Duivenvoorden, K. Lee, M. Raison, and S. Achiche, “Sensor fusion in upper limb area networks: A survey,” *2017 Glob. Inf. Infrastruct. Netw. Symp. GIIS 2017*, vol. 2017-Decem, pp. 56–63, 2017, doi: 10.1109/GIIS.2017.8169802.
- [29] M. Georgi and T. Schultz, “Recognizing Hand and Finger Gestures with IMU based Motion

- and EMG based Muscle Activity Sensing,” pp. 99–108, 2015.
- [30] H. Kaur and J. Rani, “A Review: Study of Various Techniques of Hand Gesture Recognition,” 2016.
- [31] F. Gaetani, G. A. Zappatore, P. Visconti, and P. Primiceri, “Design of an Arduino-based platform interfaced by Bluetooth low energy with Myo armband for controlling an under-actuated transradial prosthesis,” *ICICDT 2018 - Int. Conf. IC Des. Technol. Proc.*, pp. 185–188, 2018, doi: 10.1109/ICICDT.2018.8399787.
- [32] Z. O. Khokhar, Z. G. Xiao, and C. Menon, “Surface EMG pattern recognition for real-time control of a wrist exoskeleton,” *Biomed. Eng. Online*, vol. 9, pp. 1–17, 2010, doi: 10.1186/1475-925X-9-41.
- [33] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, “Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges,” *Inf. Fusion*, vol. 35, pp. 68–80, 2017, doi: <https://doi.org/10.1016/j.inffus.2016.09.005>.
- [34] D. Novak and R. Riener, “A survey of sensor fusion methods in wearable robotics,” *Rob. Auton. Syst.*, vol. 73, pp. 155–170, 2015, doi: 10.1016/j.robot.2014.08.012.
- [35] M. Connan, E. R. Ramírez, B. Vodermayr, and C. Castellini, “Assessment of a wearable force and electromyography device and comparison of the related signals for myocontrol,” *Front. Neurobot.*, vol. 10, no. NOV, pp. 1–13, 2016, doi: 10.3389/fnbot.2016.00017.
- [36] E. Criswell, *Cram’s introduction to surface electromyography*. Jones & Bartlett Publishers, 2010.
- [37] C. Zizoua, M. Raison, S. Boukhenous, M. Attari, and S. Achiche, “Detecting muscle contractions using strain gauges,” *Electron. Lett.*, vol. 52, no. 22, pp. 1836–1838, 2016, doi: 10.1049/el.2016.2986.
- [38] Y. Bu *et al.*, “Ultrasensitive strain sensor based on superhydrophobic microcracked conductive Ti<sub>3</sub>C<sub>2</sub>T<sub>x</sub> MXene/paper for human-motion monitoring and E-skin,” *Sci. Bull.*, vol. 66, no. 18, pp. 1849–1857, 2021, doi: <https://doi.org/10.1016/j.scib.2021.04.041>.
- [39] J. A. Rogers, R. Ghaffari, and D.-H. Kim, “Stretchable Bioelectronics for Medical Devices and Systems,” *Stretchable Bioelectron. Med. Devices Syst.*, pp. 257–273, 2016, doi:

10.1007/978-3-319-28694-5.

- [40] S. Wang, Y. Fang, H. He, L. Zhang, C. Li, and J. Ouyang, “Wearable Stretchable Dry and Self-Adhesive Strain Sensors with Conformal Contact to Skin for High-Quality Motion Monitoring,” *Adv. Funct. Mater.*, vol. 31, no. 5, p. 2007495, Jan. 2021, doi: <https://doi.org/10.1002/adfm.202007495>.
- [41] C. Zizoua, M. Raison, S. Boukhenous, M. Attari, and S. Achiche, “Development of a Bracelet With Strain-Gauge Matrix for Movement Intention Identification in Traumatic Amputees,” *IEEE Sens. J.*, vol. 17, no. 8, pp. 2464–2471, 2017, doi: [10.1109/jsen.2017.2666784](https://doi.org/10.1109/jsen.2017.2666784).
- [42] Y. Zhou *et al.*, “Highly sensitive, self-powered and wearable electronic skin based on pressure-sensitive nanofiber woven fabric sensor,” *Sci. Rep.*, vol. 7, no. 1, p. 12949, 2017, doi: [10.1038/s41598-017-13281-8](https://doi.org/10.1038/s41598-017-13281-8).
- [43] S. Seyedin, J. M. Razal, P. C. Innis, A. Jeiranikhameneh, S. Beirne, and G. G. Wallace, “Knitted Strain Sensor Textiles of Highly Conductive All-Polymeric Fibers,” *ACS Appl. Mater. Interfaces*, vol. 7, no. 38, pp. 21150–21158, Sep. 2015, doi: [10.1021/acsami.5b04892](https://doi.org/10.1021/acsami.5b04892).
- [44] T. Li, Z. Huang, Z. Suo, S. P. Lacour, and S. Wagner, “Stretchability of thin metal films on elastomer substrates,” *Appl. Phys. Lett.*, vol. 85, no. 16, pp. 3435–3437, Oct. 2004, doi: [10.1063/1.1806275](https://doi.org/10.1063/1.1806275).
- [45] J. Lee *et al.*, “A stretchable strain sensor based on a metal nanoparticle thin film for human motion detection,” *Nanoscale*, vol. 6, no. 20, pp. 11932–11939, 2014, doi: [10.1039/c4nr03295k](https://doi.org/10.1039/c4nr03295k).
- [46] A. Hirsch, H. O. Michaud, A. P. Gerratt, S. de Mulatier, and S. P. Lacour, “Intrinsically Stretchable Biphasic (Solid–Liquid) Thin Metal Films,” *Adv. Mater.*, vol. 28, no. 22, pp. 4507–4512, 2016, doi: [10.1002/adma.201506234](https://doi.org/10.1002/adma.201506234).
- [47] A. Hirsch, L. Dejace, H. O. Michaud, and S. P. Lacour, “Harnessing the Rheological Properties of Liquid Metals To Shape Soft Electronic Conductors for Wearable Applications,” *Acc. Chem. Res.*, vol. 52, no. 3, pp. 534–544, Mar. 2019, doi: [10.1021/acs.accounts.8b00489](https://doi.org/10.1021/acs.accounts.8b00489).

- [48] H. Zhang, A. Lowe, A. Kalra, and Y. Yu, “A Flexible Strain Sensor Based on Embedded Ionic Liquid,” *Sensors (Basel)*, vol. 21, no. 17, p. 5760, Aug. 2021, doi: 10.3390/s21175760.
- [49] D. Y. Choi *et al.*, “Highly Stretchable, Hysteresis-Free Ionic Liquid-Based Strain Sensor for Precise Human Motion Monitoring,” *ACS Appl. Mater. Interfaces*, vol. 9, no. 2, pp. 1770–1780, Jan. 2017, doi: 10.1021/acsami.6b12415.
- [50] X. Song *et al.*, “A graphene-coated silk-spandex fabric strain sensor for human movement monitoring and recognition,” *Nanotechnology*, vol. 32, no. 21, p. 215501, 2021, doi: 10.1088/1361-6528/abe788.
- [51] S. Yao, L. Vargas, X. Hu, and Y. Zhu, “A Novel Finger Kinematic Tracking Method Based on Skin-Like Wearable Strain Sensors,” *IEEE Sens. J.*, vol. 18, no. 7, pp. 3010–3015, 2018, doi: 10.1109/JSEN.2018.2802421.
- [52] Y. L. Park, B. R. Chen, and R. J. Wood, “Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors,” *IEEE Sens. J.*, vol. 12, no. 8, pp. 2711–2718, 2012, doi: 10.1109/JSEN.2012.2200790.
- [53] Q. Gao, H. Li, J. Zhang, Z. Xie, J. Zhang, and L. Wang, “Microchannel Structural Design For a Room-Temperature Liquid Metal Based Super-stretchable Sensor,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–8, 2019, doi: 10.1038/s41598-019-42457-7.
- [54] L. Detuncq, Bruno; Marchand, *Méthodes expérimentales et instrumentation en mécanique*. Presses internationales Polytechnique.
- [55] W. Li and J. Wang, “Effective Adaptive Kalman Filter for MEMS-IMU/Magnetometers Integrated Attitude and Heading Reference Systems,” *J. Navig.*, vol. 66, no. 1, pp. 99–113, 2013, doi: DOI: 10.1017/S0373463312000331.
- [56] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, “Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion,” *Sensors (Switzerland)*, vol. 17, no. 6, pp. 1–40, 2017, doi: 10.3390/s17061257.
- [57] N. L. W. Keijsers, “Neural Networks,” K. Kompolti and L. V. B. T.-E. of M. D. Metman, Eds. Oxford: Academic Press, 2010, pp. 257–259.

- [58] A. Goodfellow, Ian; Bengio, Yoshua; Courville, *Deep Learning*. MIT Press, 2016.
- [59] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943, doi: 10.1007/BF02478259.
- [60] F. Rosenblatt, *The Perceptron, a Perceiving and Recognizing Automaton Project Para.* Cornell Aeronautical Laboratory, 1957.
- [61] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.
- [62] K. Ng, Andrew; Katanforoosh, “Deep Learning-CS229 Lecture Notes.” [http://cs229.stanford.edu/summer2020/cs229-notes-deep\\_learning.pdf](http://cs229.stanford.edu/summer2020/cs229-notes-deep_learning.pdf).
- [63] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” pp. 1–20, 2018, [Online]. Available: <http://arxiv.org/abs/1811.03378>.
- [64] C. M. Bishop, *Pattern Recognition and Machine Learning*. 2006.
- [65] C. Olah, “Understanding LSTM Networks,” 2015. .
- [66] R. R. Drumond, B. A. D. Marques, C. N. Vasconcelos, and E. Clua, “PEEK: An LSTM recurrent network for motion classification from sparse data,” *VISIGRAPP 2018 - Proc. 13th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 1, no. Visigrapp, pp. 215–222, 2018, doi: 10.5220/0006585202150222.
- [67] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [68] J. Singh, S. P. Mohanty, and D. K. Pradhan, *Introduction to Statistical Learning*. 2013.
- [69] G. Hinton, N. Srivastava, and K. Swersky, “Lecture 6a :Overview of mini-batch gradient descent,” in *Neural Networks for Machine Learning*, 2012.
- [70] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, “Randomized smoothing for (parallel) stochastic optimization,” *Proc. IEEE Conf. Decis. Control*, vol. 12, pp. 5442–5444, 2012, doi: 10.1109/CDC.2012.6426698.
- [71] D. P. Kingma and J. Lei Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,”

- Iclr*, pp. 1–15, 2015, [Online]. Available: <https://arxiv.org/pdf/1412.6980.pdf> %22 entire document.
- [72] M. Hardt, “UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION,” pp. 1–15, 2014, [Online]. Available: <http://arxiv.org/abs/1611.03530>.
- [73] S. Nitish, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfittin,” *J. Mach. Learn. Res.* *15*, vol. Research 1, no. 3–4, p. 192901958, 2014.
- [74] A. Dosovitskiy *et al.*, “FlowNet: Learning optical flow with convolutional networks,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 2758–2766, 2015, doi: 10.1109/ICCV.2015.316.
- [75] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview,” *Bioinformatics*, vol. 16, no. 5, pp. 412–424, May 2000, doi: 10.1093/bioinformatics/16.5.412.
- [76] C. Goutte and E. Gaussier, “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation,” *Lect. Notes Comput. Sci.*, vol. 3408, pp. 345–359, 2005, doi: 10.1007/978-3-540-31865-1\_25.
- [77] B. Sensortec, “Integrated MCU + flash. Integrated sensor fusion. Smart sensor: BNO055.” <https://www.bosch-sensortec.com/products/smart-sensors/bno055/>.
- [78] MathWorks, “Calibrate BNO055 Sensors.” <https://www.mathworks.com/help/supportpkg/arduinoio/ug/calibrate-sensors.html>.
- [79] Adafruit, “Adafruit\_BNO055.” .
- [80] B. Maundy and S. J. G. Gift, “Strain gauge amplifier circuits,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 4, pp. 1–8, 2013, doi: 10.1109/TIM.2013.2246904.
- [81] A. Device, “Instrumentation Amplifier Diamond Plot Tool.” .
- [82] M. Atzori *et al.*, “Building the NINAPRO Database: A Resource for the Biorobotics Community - HES SO Valais publications - Aigaion 2.0,” *Proc. IEEE Int. Conf. Biomed. Robot. Biomechatronics*, p. 51, 2012, [Online]. Available:



<http://publications.hevs.ch/index.php/publications/show/1172>.

- [83] R. Khusainov, D. Azzi, I. E. Achumba, and S. D. Bersch, “Real-time human ambulation, activity, and physiological monitoring: Taxonomy of issues, techniques, applications, challenges and limitations,” *Sensors (Switzerland)*, vol. 13, no. 10, pp. 12852–12902, 2013, doi: 10.3390/s131012852.
- [84] A. Gijssberts and B. Caputo, “Exploiting accelerometers to improve movement classification for prosthetics,” in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, 2013, pp. 1–5, doi: 10.1109/ICORR.2013.6650476.

## CHAPITRE 6 RÉSULTATS COMPLÉMENTAIRES

Ce chapitre a pour but de décrire les résultats qui ont été obtenus suite à l'implémentation des algorithmes d'apprentissage profond pour l'identification des intentions de mouvements. Dans un premier temps, les résultats obtenus de la comparaison des modèles LSTM et CNN-LSTM sont présentés, suivis par les résultats de la recherche des hyperparamètres et ensuite la quantification de l'impact des données cinématiques sur les résultats

### 6.1 Comparaison des modèles

Le premier modèle qui a été développé est le modèle LSTM. Le choix des paramètres initiaux était, pour la plupart, arbitraire. Par la suite, les tests ont permis de déterminer un ensemble de paramètres de référence pour former un modèle de référence Figure 4.17 dont les hyperparamètres sont présentés dans le Tableau 6.1.

Tableau 6.1 Hyperparamètres du modèle de référence

Taille des lots	100
Nombre d'itérations	50
Terme de régularisation	0.02
Taux d'apprentissage	1e-03
Decay	1e-05
Dropout	0.2

Le Tableau 6.2 résume les résultats présentés dans la matrice de confusion obtenus sur l'ensemble de tests pour le modèle LSTM présenté à la Figure 6.3 et celle obtenue sur l'ensemble test en utilisant le modèle CNN-LSTM présenté à la Figure 6.4. En termes de performance de classification, le modèle LSTM a atteint une précision de classification de 92.5% et le modèle CNN-LSTM a quant à lui atteint 95.1%. En général, le modèle CNN-LSTM a obtenu une précision de classification plus élevée par mouvement comparé au modèle LSTM. Ces résultats sont présentés dans le Tableau 6.2. Cependant, le modèle LSTM a surpassé le modèle CNN-LSTM pour certains mouvements. En effet, le modèle LSTM a obtenu une précision de classification de 100.0% pour la pronation de l'avant-bras (FP), et 99.0% pour la flexion du poignet (WF) tandis que le

modèle CNN\_LSTM a obtenu 95.1% et 98.0% respectivement. Les deux modèles ont aussi des performances similaires pour les mouvements de déflexion radiale du poignet (WR) et de préhension (PG) avec 96.0% et 100% de précision de classification respectivement. Pour la mesure de performance basée sur le MCC, section 2.6, Le modèle LSTM a obtenu un score de 0.92 et le modèle CNN-LSTM a obtenu un score de 0.95. En outre, cela a pris 20.4s pour entraîner le modèle CNN-LSTM et 19.3s pour entraîner le modèle LSTM.

Tableau 6.2 Comparaison des résultats des modèles LSTM et CNN-LSTM

		NM	EF	EE	FP	FS	WE	WF	WR	WL	PG	TOTAL	MCC	t(s)
Précision de classification (%)	LSTM	60.40	98.02	98.02	<b>100</b>	95.05	98.02	<b>99.01</b>	<b>96.04</b>	80.20	100	92.48	0.92	19.3
	CNN-LSTM	<b>72.28</b>	<b>99.01</b>	<b>100</b>	95.05	<b>98.02</b>	<b>100</b>	98.02	<b>96.04</b>	<b>92.08</b>	<b>100</b>	<b>95.05</b>	0.95	20.4

Il arrive que des modèles apprennent des caractéristiques spécifiques à l'ensemble d'entraînement. En effet, en entraînant un modèle suffisamment longtemps sur un ensemble de données (ici l'ensemble d'entraînement), il est capable d'atteindre une précision de classification élevée. Cependant, lorsque ce modèle est utilisé sur un nouvel ensemble de données (ici l'ensemble de validation), il va obtenir une faible précision de classification. C'est le problème de généralisation [58]. Il est donc important de regarder les courbes d'apprentissages de l'ensemble d'entraînement et de validation présentée sur la Figure 6.1 et la Figure 6.2 pour observer lorsque le sur-apprentissage commence à se produire. L'observation de ces graphiques montrent que la précision de classification augmente au fur et à mesure que le nombre d'itération augmente (epoch) pour l'ensemble d'entraînement et l'ensemble de validation, il n'y a donc pas de sur apprentissage. Le surapprentissage est observé sur une courbe lorsque la précision de classification sur les données d'entraînement augmente tandis que celle sur l'ensemble de validation diminue [58].

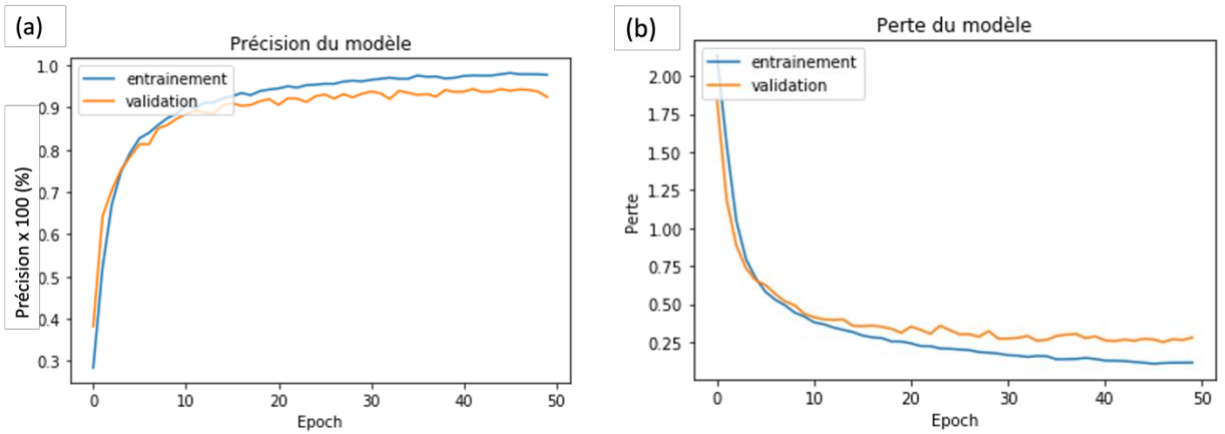


Figure 6.1 : (a) courbe de la précision de classification du modèle LSTM, (b) courbe de la fonction coût du modèle LSTM

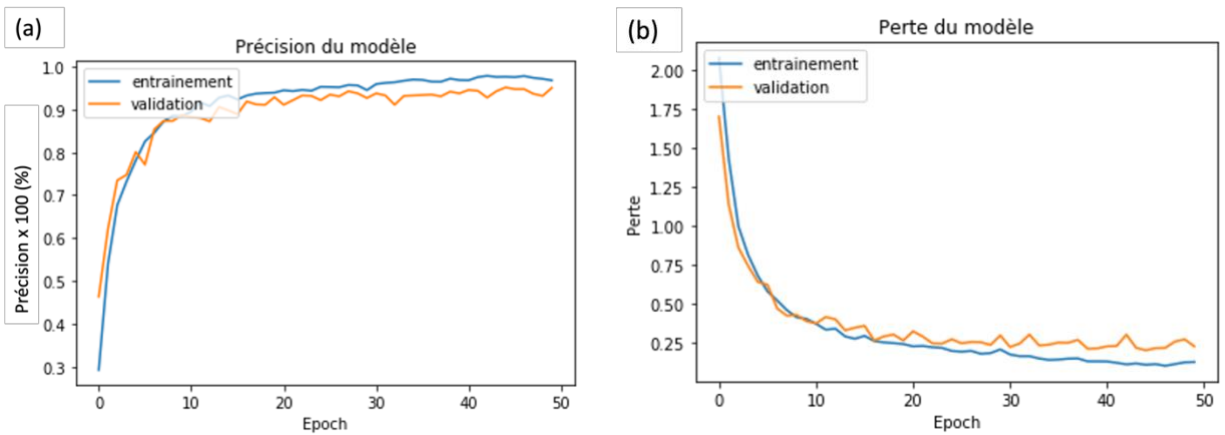


Figure 6.2 : (a) courbe de la précision de classification du modèle CNN-LSTM, (b) courbe de la fonction coût du modèle CNN-LSTM

**Confusion matrix**

Predicted	NM	61 6.04%	2 0.20%	0 0.0%	0 0.0%	1 0.10%	1 0.10%	1 0.10%	4 0.40%	4 0.40%	0 0.0%	74 82.43% 17.57%
	EE	7 0.69%	99 9.80%	2 0.20%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	108 91.67% 8.33%
	EF	9 0.89%	0 0.0%	99 9.80%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	108 91.67% 8.33%
	FP	7 0.69%	0 0.0%	0 0.0%	101 10.00%	4 0.40%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	112 90.18% 9.82%
	FS	6 0.59%	0 0.0%	0 0.0%	0 0.0%	96 9.50%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	102 94.12% 5.88%
	WE	1 0.10%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99 9.80%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 99.00% 1.00%
	WF	1 0.10%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.10%	100 9.90%	0 0.0%	1 0.10%	0 0.0%	103 97.09% 2.91%
	WR	2 0.20%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97 9.60%	15 1.49%	0 0.0%	114 85.09% 14.91%
	WL	6 0.59%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	81 8.02%	0 0.0%	87 93.10% 6.90%
	PG	1 0.10%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	101 10.00%	102 99.02% 0.98%
		101 60.40% 39.60%	101 98.02% 1.98%	101 98.02% 1.98%	101 100% 0.00%	101 95.05% 4.95%	101 98.02% 1.98%	101 99.01% 0.99%	101 96.04% 3.96%	101 80.20% 19.80%	101 100% 0.00%	1010 92.48% 7.52%
	NM	EE	EF	FP	FS	WE	WF	WR	WL	PG		
	<b>Actual</b>											

Figure 6.3 Matrice de confusion du modèle LSTM

**Confusion matrix**

<b>Predicted</b>	NM	<b>73</b> 7.23%	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>2</b> 0.20%	<b>4</b> 0.40%	<b>3</b> 0.30%	<b>0</b> 0.0%	<b>84</b> 86.90% 13.10%
	EE	<b>6</b> 0.59%	<b>100</b> 9.90%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>106</b> 94.34% 5.66%
	EF	<b>7</b> 0.69%	<b>0</b> 0.0%	<b>101</b> 10.00%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>108</b> 93.52% 6.48%
	FP	<b>7</b> 0.69%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>96</b> 9.50%	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>104</b> 92.31% 7.69%
	FS	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>3</b> 0.30%	<b>99</b> 9.80%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>103</b> 96.12% 3.88%
	WE	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>101</b> 10.00%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>102</b> 99.02% 0.98%
	WF	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>99</b> 9.80%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>99</b> 100% 0.00%
	WR	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>97</b> 9.60%	<b>5</b> 0.50%	<b>0</b> 0.0%	<b>103</b> 94.17% 5.83%
	WL	<b>5</b> 0.50%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>93</b> 9.21%	<b>0</b> 0.0%	<b>98</b> 94.90% 5.10%
	PG	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>101</b> 10.00%	<b>103</b> 98.06% 1.94%
		<b>101</b> 72.28% 27.72%	<b>101</b> 99.01% 0.99%	<b>101</b> 100% 0.00%	<b>101</b> 95.05% 4.95%	<b>101</b> 98.02% 1.98%	<b>101</b> 100% 0.00%	<b>101</b> 98.02% 1.98%	<b>101</b> 96.04% 3.96%	<b>101</b> 92.08% 7.92%	<b>101</b> 100% 0.00%	<b>1010</b> 95.05% 4.95%
		NM	EE	EF	FP	FS	WE	WF	WR	WL	PG	
		<b>Actual</b>										

Figure 6.4 Matrice de confusion du modèle CNN-LSTM

Pour résumer la comparaison des modèles, le modèle CNN-LSTM a surpassé le modèle LSTM. Avec une précision médiane de 95,1 % (MCC = 0,95), contre 92,5 % (MCC = 0,92). Cette comparaison aurait dû inclure plusieurs essais pour faire la moyenne, cependant, due à la contrainte de temps le nombre d'essais a été limité. Le modèle CNN-LSTM a été choisi pour la recherche d'hyperparamètres. Cependant, la sélection de meilleurs paramètres pour le modèle LSTM pourrait conduire ce modèle à potentiellement surpasser le modèle CNN-LSTM.

## 6.2 Recherche des hyperparamètres

Les résultats de la recherche d'hyperparamètres ont été compilés ci-dessous. Pour chaque valeur hyperparamètre, cinq valeurs ont été choisies pour déterminer leur impact sur l'apprentissage du modèle. Les valeurs optimales pour chaque hyperparamètre ont été marquées en gras.

Les hyperparamètres testés sont présentés. Les tableaux à gauche présentent la valeur des hyperparamètres testée ainsi que le MCC et la précision de classification obtenue pour chaque valeur d'hyperparamètre. Les figures à droite présentent l'évolution de la précision pour les ensembles tests et validation en fonction des valeurs des valeurs des hyperparamètres.

Tableau 6.3 Résultat de la variation du taux d'apprentissage

$\lambda$	Précision de classification	MCC
1e-01	10.0	0
1e-02	89.8	0.89
<b>1e-03</b>	<b>95.2</b>	<b>0.95</b>
1e-04	86.5	0.85
1e-05	36.3	0.31

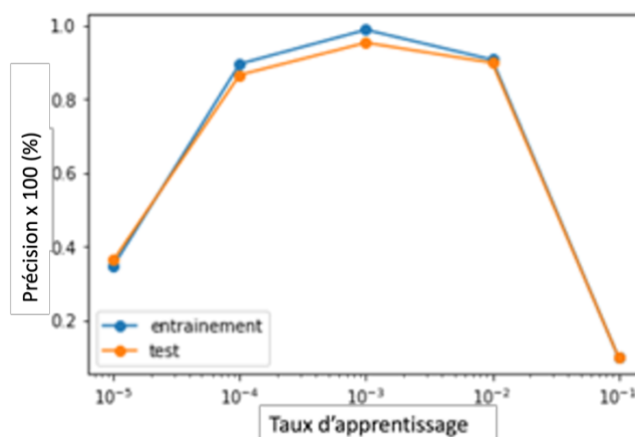


Figure 6.5 Évolution de la précision pour les ensembles tests et validation en fonction du taux d'apprentissage

Tableau 6.4 Résultat de la variation du nombre de filtres pour le CNN

Nombre de filtres	Précision de Classification (%)	MCC
9	91.4	0.905
18	93.3	0.925
<b>32</b>	<b>95.6</b>	<b>0.951</b>
64	95.2	0.947
128	95.4	0.949

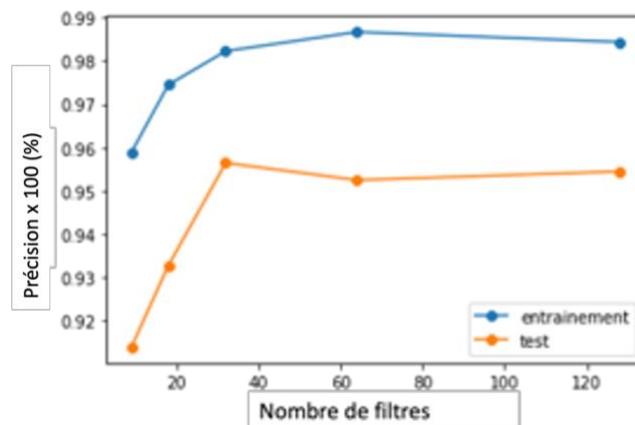


Figure 6.6 Évolution de la précision pour les ensembles tests et validation en fonction du nombre de filtres pour le CNN

Tableau 6.5 Résultat de la variation du nombre de neurones pour le LSTM

Nombre de neurones	Précision de Classification (%)	MCC
9	92.7	0.919
18	92.8	0.920
32	95.0	0.945
64	93.6	0.929
<b>128</b>	<b>95.8</b>	<b>0.954</b>

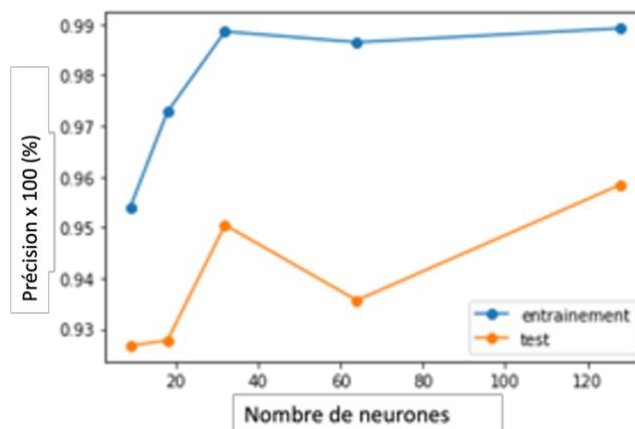


Figure 6.7 Évolution de la précision pour les ensembles tests et validation en fonction du nombre de neurones pour le LSTM



Tableau 6.6 Résultat de la variation du terme de régularisation

Terme de régularisation	Précision de Classification (%)	MCC
0.0	94.6	0.939
0.001	93.4	0.926
<b>0.002</b>	<b>95.0</b>	<b>0.944</b>
0.01	94.3	0.936
0.02	93.5	0.928

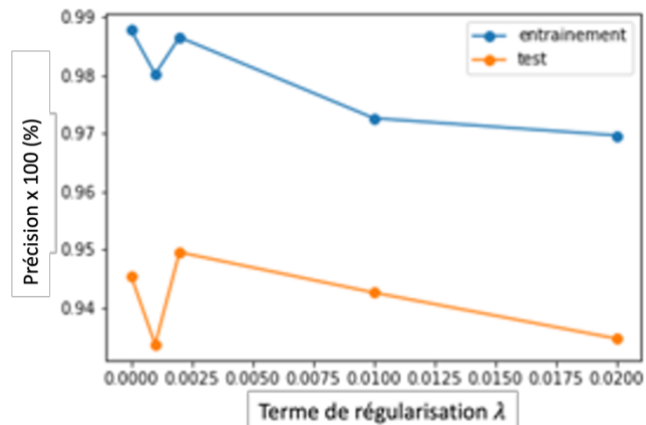


Figure 6.8 Évolution de la précision pour les ensembles tests et validation en fonction du terme de régularisation

Tableau 6.7 Résultat de la variation du nombre d'échantillon par lot

Nombre d'échantillon par lot	Précision de Classification (%)	MCC
25	95.3	0.948
50	94.7	0.941
<b>100</b>	<b>95.5</b>	<b>0.951</b>
200	95.0	0.944
300	93.2	0.924

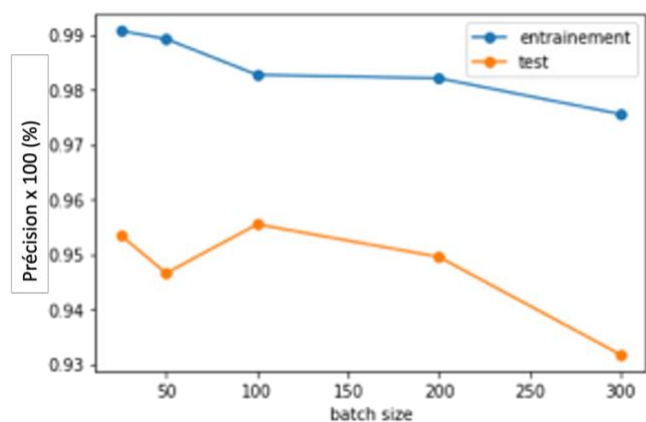


Figure 6.9 Évolution de la précision pour les ensembles tests et validation en fonction du nombre d'échantillon par lot

Il est toujours important de s'assurer que le modèle ne soit pas surentraîné sur les données d'entraînement. Il est donc important d'analyser les courbes d'apprentissages pour chacune des valeurs d'hyperparamètres testées. Ces courbes sont présentées en Annexe B.

Le Tableau 6.4 résume les résultats qui ont été obtenus par la recherche séquentielle des hyperparamètres.

Tableau 6.8 Résumé des valeurs optimales d'hyperparamètres

Taux d'apprentissage	1e-03
Nombre de filtres pour le CNN	32
Nombre de neurones dans le LSTM	128
Terme de régularisation	0.002
Batch size	100

Après avoir déterminé les valeurs optimales d'hyperparamètres, la prochaine étape a consisté à déterminer l'impact des données inertielles sur la précision de classification.

### 6.3 Impact des données inertielles

La recherche des hyperparamètres a conclu que les hyperparamètres de bases sont ceux qui ont permis d'avoir les meilleurs résultats de classification. Ces paramètres ont donc été utilisés sur les signaux provenant des jauges de déformations. La matrice de confusion obtenue sur l'ensemble contenant exclusivement les signaux des jauges de déformation est présentée à la Figure 6.6. Le Tableau 6.5 montre la comparaison entre le modèle utilisant uniquement les signaux provenant des jauges de déformations et le modèle de référence utilisant l'ensemble de données (IMUs et jauges de déformation).

En termes de performance de classification, le modèle utilisant uniquement les signaux provenant des jauges de déformation a obtenu une précision de classification de 89.3%. Le modèle de référence a obtenu une précision de classification de 95.1%. Le modèle utilisant l'ensemble des données (IMUs et jauges de déformation) a obtenu de meilleure précision de classification comparé au modèle utilisant uniquement les jauges de déformation, Tableau 6.5. Cependant, on remarque que le modèle utilisant uniquement les signaux provenant des jauges de déformations a surpassé le modèle de référence pour certains mouvements. En effet, ce modèle a obtenu une précision de classification de 99.0% pour la pronation de l'avant-bras (FP), et 100.0 % pour la flexion du poignet (WF) tandis que le modèle CNN\_LSTM a obtenu 95.1% et 98.0% respectivement. En outre, cela a pris 20.4s pour entraîner le modèle de référence et 19.9s pour entraîner le modèle utilisant les jauges de déformations uniquement, des temps assez comparables.

Tableau 6.9 Comparaison des modèles LSTM et CNN-LSTM

		NM	EE	EF	FP	FS	WE	WF	WR	WL	PG	TOTAL	MCC	t(S)
Précision de classification (%)	Jauges	45.5	98.0	95.1	<b>99.0</b>	97.0	93.1	<b>100.0</b>	91.1	76.2	98.2	<b>89.3</b>	0.88	19.9
	IMUs+ jauges	<b>72.3</b>	<b>99.0</b>	<b>100.0</b>	95.1	<b>98.0</b>	<b>100.0</b>	98.0	<b>96.0</b>	<b>92.1</b>	<b>100.0</b>	<b>95.1</b>	0.95	20.4

Comme il a été mentionné dans la section 6.1, il est toujours important d'observer les courbes d'apprentissages afin de s'assurer que le modèle ne soit pas surentraîné. Les courbes d'apprentissages du modèle utilisant les jauges de déformations sont donc présentées à la Figure 6.5. L'observation de ces graphiques montrent que la précision de classification augmente au fur et à mesure que le nombre d'itération augmente (epoch) pour l'ensemble d'entraînement et l'ensemble de validation, il n'y a donc pas de surapprentissage.

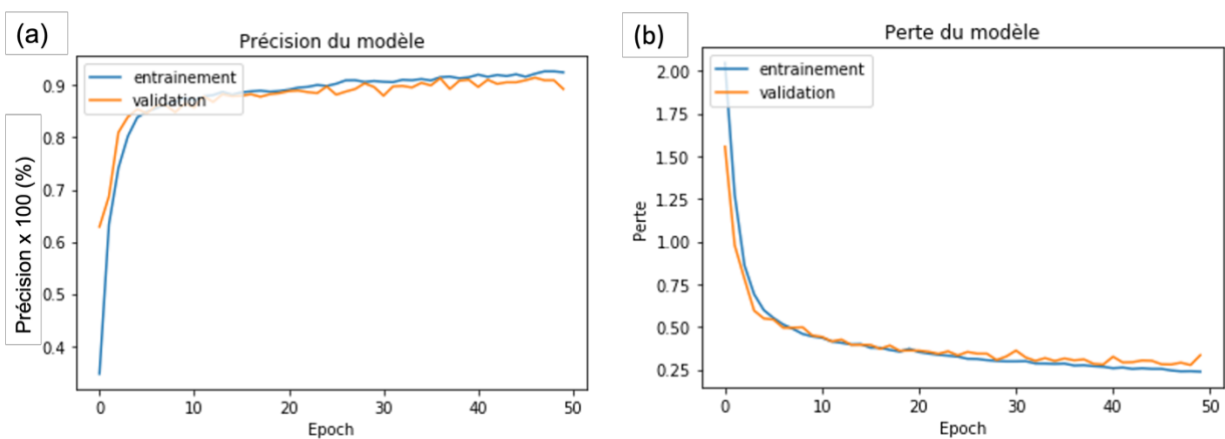


Figure 6.10 (a) : courbe d'apprentissage du modèle CNN-LSTM, (b): courbe de la fonction coût du modèle CNN-LSTM

**Confusion matrix**

<b>Predicted</b>	NM	<b>46</b> 4.55%	<b>2</b> 0.20%	<b>3</b> 0.30%	<b>1</b> 0.10%	<b>3</b> 0.30%	<b>4</b> 0.40%	<b>0</b> 0.0%	<b>6</b> 0.59%	<b>14</b> 1.39%	<b>2</b> 0.20%	<b>81</b> 56.79% 43.21%
	EE	<b>15</b> 1.49%	<b>99</b> 9.80%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>115</b> 86.09% 13.91%
	EF	<b>8</b> 0.79%	<b>0</b> 0.0%	<b>96</b> 9.50%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>104</b> 92.31% 7.69%
	FP	<b>15</b> 1.49%	<b>0</b> 0.0%	<b>2</b> 0.20%	<b>100</b> 9.90%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>3</b> 0.30%	<b>0</b> 0.0%	<b>120</b> 83.33% 16.67%
	FS	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>98</b> 9.70%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>98</b> 100% 0.00%
	WE	<b>4</b> 0.40%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>94</b> 9.31%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>6</b> 0.59%	<b>0</b> 0.0%	<b>104</b> 90.38% 9.62%
	WF	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>101</b> 10.00%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>101</b> 100% 0.00%
	WR	<b>4</b> 0.40%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>2</b> 0.20%	<b>0</b> 0.0%	<b>92</b> 9.11%	<b>1</b> 0.10%	<b>0</b> 0.0%	<b>99</b> 92.93% 7.07%
	WL	<b>9</b> 0.89%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>77</b> 7.62%	<b>0</b> 0.0%	<b>86</b> 89.53% 10.47%
	PG	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>0</b> 0.0%	<b>3</b> 0.30%	<b>0</b> 0.0%	<b>99</b> 9.80%	<b>102</b> 97.06% 2.94%
		<b>101</b> 45.54% 54.46%	<b>101</b> 98.02% 1.98%	<b>101</b> 95.05% 4.95%	<b>101</b> 99.01% 0.99%	<b>101</b> 97.03% 2.97%	<b>101</b> 93.07% 6.93%	<b>101</b> 100% 0.00%	<b>101</b> 91.09% 8.91%	<b>101</b> 76.24% 23.76%	<b>101</b> 98.02% 1.98%	<b>1010</b> 89.31% 10.69%
	NM	EE	EF	FP	FS	WE	WF	WR	WL	PG		
	<b>Actual</b>											

Figure 6.11 Matrice de confusion du modèle entraîné exclusivement sur les signaux provenant des jauges de déformation

## CHAPITRE 7 DISCUSSION GÉNÉRALE

L'objectif général de ce projet de maîtrise qui était de développer un système portable pour identifier les intentions de mouvement en combinant des jauges de déformation et des centrales inertielles a été atteint. Dans cette section, un résumé de la discussion de l'article est présenté. Ensuite, une discussion sur les résultats complémentaires portant sur l'entraînement d'un réseau de neurones profond et la classification de neuf mouvements du membre supérieur ainsi qu'une position dite de repos est présentée, le tout suivi de limites de l'étude et les perspectives de recherche.

### 7.1 Synthèse de l'article scientifique

#### 7.1.1 Détection d'intention de mouvement

Les résultats ont montré que le bracelet de jauges de déformation placé sur l'avant-bras a enregistré des signaux qui ont été utilisés pour distinguer les différents mouvements du poignet et le mouvement de préhension. Le bracelet placé sur le bras a enregistré des signaux qui ont été utilisés pour distinguer les mouvements de l'avant-bras. En outre, l'analyse de l'intensité des signaux a révélé que la combinaison de jauges qui ont enregistré un  $\Delta V$  maximal est unique pour chaque mouvement ce qui pourrait être utilisé comme modèle pour identifier les mouvements du membre supérieur. En outre, les centrales inertielles en plus de fournir de l'information sur la cinématique du mouvement, elles ont permis d'identifier de façon claire le début et la fin de chaque mouvement de la séquence de mouvements à l'étude.

En somme, les jauges de déformations ont fourni de l'information sur la contraction musculaire et les IMU sur la cinématique du mouvement. Les résultats préliminaires obtenus ont prouvé la capacité de ce type de capteurs à enregistrer des signaux permettant de distinguer 10 mouvements du membre supérieur.

#### 7.1.2 Fabrication du capteur

Le système développé comprend deux types capteurs : un système d'acquisition des signaux provenant des jauges de déformation et une centrale inertielle. Les deux capteurs ont une dimension de [12x10x3.3 cm] et [5.5x3.5x2.2 cm] respectivement. Le système est donc compact et portable, ce qui le rend idéal pour les activités quotidiennes. Le système utilise des jauges de déformation,

une bande kinésiologique, une centrale inertielle BNO055 qui sont des matériaux peu dispendieux disponibles dans le commerce.

Ces travaux ont prouvé que la mesure des déformations sur différents axes est importante pour mieux capturer la déformation de la peau due à l'activité musculaire. La méthode fabrication des matrices des jauges de déformations basées les PCBs flexibles permettent de placer ces capteurs sur différents sites du corps humain pour collecter différents signaux. Cette technique a également permis d'avoir une connexion solide avec les jauges empêchant celles-ci de se rompre ou de se dessouder lors des mouvements. Le nombre de jauges de déformation, leurs orientations ainsi que la position du bracelet sont des facteurs importants qui peuvent être optimisés en se référant aux connaissances de base de l'anatomie humaine.

## **7.2 Comparaison des modèles de classification**

L'ajout d'une couche de convolution unidimensionnelle avant la couche LSTM a permis d'augmenter la précision de classification de 2.6% pour atteindre 95.1%. Ces résultats sont en adéquation avec nos attentes, car l'utilisation des couches de convolutions est une excellente méthode pour l'extraction des caractéristiques des signaux. Cependant, la capacité des réseaux de convolutions à extraire des caractéristiques dépend fortement des valeurs d'hyperparamètres tels que le nombre de filtres, le pas, les couches de mise en commun, etc. Ainsi, pour avoir une extraction de caractéristiques optimale il est important de bien déterminer ces paramètres en effectuant plusieurs tests. Ces tests sortent du cadre de cette étude, mais pourraient expliquer pourquoi le modèle LSTM a eu de meilleures précisions de classification que le modèle CNN-LSTM pour les mouvements de pronation de l'avant-bras et de flexion du poignet. Aussi, le MCC du modèle CNN-LSTM est plus proche de 1 que celui du modèle LSTM ce qui démontre que le modèle CNN-LSTM se rapproche plus d'un classificateur idéal.

Par ailleurs, le temps d'entraînement du modèle CNN-LSTM est de 1.1s plus élevé que celui du modèle LSTM. Cela est prévisible, car l'ajout d'un réseau de convolution avant le réseau de neurones récurrents (LSTM) augmente le nombre de paramètres que le réseau doit mettre à jour dans le processus d'apprentissage. En effet, le modèle LSTM avait un total de 17,424 paramètres à entraîner tandis que le modèle CNN-LSTM avait un total de 21,296 ce qui représente un ajout de 3,872 paramètres.

### **7.3 Recherche d'hyperparamètres**

Les résultats obtenus par la recherche d'hyperparamètres ont confirmé que les valeurs d'hyperparamètres utilisées dans le modèle de base étaient pour la majorité optimale. En effet, seule l'augmentation des neurones dans les couches récurrentes a prouvé avoir un impact sur la précision de classification. Le modèle avec 128 neurones à la place des 32 neurones de bases, a permis d'augmenter la précision de classification de 0.8% pour atteindre 95.8%. Cependant, cela représente quatre fois plus de neurones soit 199,104 paramètres supplémentaires pour une augmentation de 0.8%. En plus, le temps d'entraînement est passé de 20.4s à 23.3s.

Il est important de noter que la méthode de recherche séquentielle des hyperparamètres n'est pas optimale. Une recherche par grille ou d'autres méthodes plus avancées qui procèdent à une combinaison aléatoire des hyperparamètres sont plus recommandées. Cependant, cette recherche préliminaire d'hyperparamètres a permis d'établir les bases pour de futures études plus approfondies.

### **7.4 Impact des données inertielles**

Les résultats présentés à la section 6.3 ont démontré que la présence des données inertielles a permis d'augmenter de manière significative la précision de l'algorithme de classification de mouvements. En effet, les données inertielles ont permis d'augmenter la précision de classification d'environ 5.7%. Ce résultat est en adéquation avec les travaux effectués par [22] et [84] qui ont reporté que les informations cinématiques extraites des accéléromètres peuvent améliorer la précision des algorithmes de classification des mouvements basés sur les sEMG.

### **7.5 Limites et perspectives**

Bien que l'objectif général de ce projet de recherche ait été atteint, le transfert éventuel du système et de l'algorithme de classification dans des applications de la vie quotidienne n'est pas envisageable à court terme en raison de plusieurs limitations. Ces limitations ouvrent la voie à de nouvelles recherches.

*Conditionnement du signal provenant des jauges de déformations :*

Une limitation technique liée à l'utilisation du pont de Wheatstone est la nécessité de rééquilibrer chaque pont après un certain nombre d'utilisations (généralement 20). Ceci peut être un inconvénient pour développer un système qui pourrait être utilisé pour les activités quotidiennes. Une calibration automatique et robuste des ponts de Wheatstone est encore nécessaire.

*Choix des centrales inertielles et position :*

La centrale inertielle utilisée dans le cadre de ce projet est la centrale BNO055 de Adafruit. Cette centrale a permis d'avoir d'atteindre l'objectif de ce mémoire. Cependant, il existe des centrales inertielles plus performantes permettant d'avoir des données plus stables. Aussi, l'IMU (1) placé sur la main n'est pas optimal à cause de sa taille. Il faut donc trouver des capteurs de plus petites tailles, ce qui réduirait leur aspect encombrant. En outre, l'étude de l'effet des mouvements parasites comme la marche sur le système pourrait faire l'objet de prochaines études.

*Nombre et positions des bracelets de jauges de déformations :*

Comme démontré dans l'article, Chapitre 4, l'orientation des jauges est un paramètre important pour la détection des intentions de mouvement. Deux axes orthogonaux ont été étudiés dans ce projet de recherche. Il pourrait être intéressant d'étudier les signaux provenant des jauges en configuration de rosettes. Aussi, une autre étude pourrait porter sur le nombre optimal de jauges à utiliser pour atteindre une précision de classification satisfaisante.

*Collecte des données :*

La collecte des données a été effectuée sur une population saine. Une étude ultérieure pourrait porter sur des personnes ayant des faiblesses musculaires avec un nombre similaire de mouvements (10) du membre supérieur. Cette étude permettra de valider le fonctionnement du système sur des personnes n'ayant pas de grandes activations musculaires.

*Entraînement du classificateur :*

Aussi, pour cette étude toutes les données de tous les participants ont été utilisées pour former les trois ensembles de données (entraînement, validation et test) afin de développer l'algorithme de classification. Il serait très intéressant de mettre en place des algorithmes de classification de mouvement spécifique à chaque personne, car [22] a prouvé que la mise en place de classificateurs spécifiques à un sujet permettait d'augmenter la précision de classification. Aussi, Un entraînement successif en déplaçant le bracelet autour du bras pourrait permettre d'augmenter la robustesse de



l'algorithme de classification par rapport à la réinstallation du bracelet. En outre, l'entraînement de l'algorithme de détection d'intention de mouvement sera fait par un personnel qualifié. Ce dernier utilisera les métriques présentées dans la section 2.6 pour valider le bon fonctionnement de l'algorithme. L'algorithme développé devrait être aussi testé pour des contractions involontaires des muscles.

## **CHAPITRE 8 CONCLUSION ET RECOMMANDATIONS**

L'objectif de ce travail de maîtrise était de développer un système de mesure portable capable d'identifier les intentions de mouvement en combinant des jauges de déformation et des centrales inertielles. Le système proposé est composé de 1. Deux bracelets utilisant chacun 6 jauges de déformation et 2. Deux centrales inertielles. Une carte d'acquisition adaptée à notre application a été développée pour conditionner les signaux provenant des jauges de déformations. Afin de garantir la solidité des connexions électriques et de bien positionner les jauges de déformations, une méthode utilisant des PCB flexibles a été proposée. Une bande kinésiologique a été utilisée afin de garantir un bon contact entre les jauges de déformation et la peau. La centrale inertielle utilisée est le BNO055 qui fait partie de la ligne de capteur intelligent développé par Bosh SensorTec. De plus le système a été conçu dans l'optique de pouvoir être embarqué.

En ce qui concerne la reconnaissance d'intention de mouvements, les jauges de déformation ont permis d'avoir une mesure des contractions musculaires. Les centrales inertielles ont fourni de l'information sur la cinématique du mouvement. Le système de mesure a été préalablement testé sur un sujet effectuant une séquence de neuf mouvements et d'une position dite de repos. Une méthode basée sur la variation d'intensité des signaux provenant des jauges de déformation a permis d'identifier les neuf mouvements étudiés. Par la suite, des essais en laboratoire sur 7 sujets ont permis de construire une base donnée pour entraîner des algorithmes d'apprentissages profonds pour la détection d'intention de mouvements. Deux architectures différentes ont été testées à savoir un réseau de neurones récurrent et la combinaison entre ce réseau et un réseau de neurones convolutifs. La combinaison des deux réseaux s'est avérée meilleure avec une précision de classification de 95.1%. Une étude préliminaire sur la recherche d'hyperparamètres a été effectuée afin d'améliorer la performance de ce modèle. En outre, dans cette étude les informations cinématiques ont permis d'augmenter la précision de l'algorithme de classification de 5.8%. Ce

résultat vient confirmer le fait que l'information produite par les centrales inertielles est complémentaire à celle fournie par les jauges de déformations.

Bien qu'il reste des améliorations à apporter à ce projet de recherche, les résultats obtenus démontrent du potentiel d'un tel système à devenir un système sensoriel portable intelligent pour détecter l'intention de mouvement de l'humain.

## RÉFÉRENCES

- [1] E. Scheme and K. Englehart, “Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use,” *J. Rehabil. Res. Dev.*, vol. 48, no. 6, pp. 643–660, 2011, doi: 10.1682/JRRD.2010.09.0177.
- [2] R. M. Singh, S. Chatterji, and A. Kumar, “A review on surface EMG based control schemes of exoskeleton robot in stroke rehabilitation,” *Proc. - 2013 Int. Conf. Mach. Intell. Res. Adv. ICMIRA 2013*, pp. 310–315, 2014, doi: 10.1109/ICMIRA.2013.65.
- [3] A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, “Prosthetic Myoelectric Control Strategies: A Clinical Perspective,” *Curr. Surg. Reports*, vol. 2, no. 3, 2014, doi: 10.1007/s40137-013-0044-8.
- [4] Ottobock, “Bebionic,” 2021. <https://shop.ottobock.ca/en/Prosthetics/Upper-Limb-Prosthetics/bebionic/c/2888> (accessed Oct. 27, 2021).
- [5] P. Geethanjali, “Myoelectric control of prosthetic hands: State-of-the-art review,” *Med. Devices Evid. Res.*, vol. 9, pp. 247–255, 2016, doi: 10.2147/MDER.S91102.
- [6] M. Atzori and H. Müller, “Control capabilities of myoelectric robotic prostheses by hand amputees: A scientific research and market overview,” *Front. Syst. Neurosci.*, vol. 9, no. Nov, pp. 1–7, 2015, doi: 10.3389/fnsys.2015.00162.
- [7] B. Hudgins, P. Parker, and R. N. Scott, “A new strategy for multifunction myoelectric control,” *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, 1993, doi: 10.1109/10.204774.
- [8] S. M. Engdahl, B. P. Christie, B. Kelly, A. Davis, C. A. Chestek, and D. H. Gates, “Surveying the interest of individuals with upper limb loss in novel prosthetic control techniques,” *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–11, 2015, doi: 10.1186/s12984-015-0044-2.
- [9] O. W. Samuel *et al.*, “Pattern recognition of electromyography signals based on novel time domain features for amputees’ limb motion classification,” *Comput. Electr. Eng.*, vol. 67, pp. 646–655, 2018, doi: 10.1016/j.compeleceng.2017.04.003.
- [10] L. J. Hargrove, G. Li, K. B. Englehart, and B. S. Hudgins, “Principal Components Analysis

- Preprocessing for Improved Classification Accuracies in Pattern-Recognition-Based Myoelectric Control,” *IEEE Trans. Biomed. Eng.*, vol. 56, no. 5, pp. 1407–1414, 2009, doi: 10.1109/TBME.2008.2008171.
- [11] O. Politi, I. Mporas, and V. Megalooikonomou, “Comparative evaluation of feature extraction methods for human motion detection,” *IFIP Adv. Inf. Commun. Technol.*, vol. 437, pp. 146–154, 2014, doi: 10.1007/978-3-662-44722-2\_16.
- [12] E. Scheme and K. Englehart, “Training Strategies for Mitigating the Effect of Proportional Control on Classification in Pattern Recognition Based Myoelectric Control.,” *J. Prosthet. Orthot.*, vol. 25, no. 2, pp. 76–83, Apr. 2013, doi: 10.1097/JPO.0b013e318289950b.
- [13] S. Krishnan and Y. Athavale, “Trends in biomedical signal feature extraction,” *Biomed. Signal Process. Control*, vol. 43, pp. 41–63, 2018, doi: 10.1016/j.bspc.2018.02.008.
- [14] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, “Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey,” *23th Int. Conf. Archit. Comput. Syst. 2010, ARCS 2010 - Work. Proc.*, pp. 167–176, 2010.
- [15] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, “Techniques of EMG signal analysis: Detection, processing, classification and applications,” *Biol. Proced. Online*, vol. 8, no. 1, pp. 11–35, 2006, doi: 10.1251/bpo115.
- [16] A. Phinyomark, R. N. Khushaba, and E. Scheme, “Feature extraction and selection for myoelectric control based on wearable EMG sensors,” *Sensors (Switzerland)*, vol. 18, no. 5, pp. 1–17, 2018, doi: 10.3390/s18051615.
- [17] S. Guo, M. Pang, B. Gao, H. Hirata, and H. Ishihara, “Comparison of sEMG-based feature extraction and motion classification methods for upper-limb movement,” *Sensors (Switzerland)*, vol. 15, no. 4, pp. 9022–9038, 2015, doi: 10.3390/s150409022.
- [18] M. A. Oskoei and H. Hu, “Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 8, pp. 1956–1965, 2008, doi: 10.1109/TBME.2008.919734.
- [19] A. D. C. Chan and K. B. Englehart, “Continuous myoelectric control for powered prostheses using hidden Markov models.,” *IEEE Trans. Biomed. Eng.*, vol. 52, no. 1, pp. 121–124, Jan.

- 2005, doi: 10.1109/TBME.2004.836492.
- [20] L. J. Hargrove, E. J. Scheme, K. B. Englehart, and B. S. Hudgins, “Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 1, pp. 49–57, 2010, doi: 10.1109/TNSRE.2009.2039590.
- [21] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, “Classification of the myoelectric signal using time-frequency based representations,” *Med. Eng. Phys.*, vol. 21, no. 6, pp. 431–438, 1999, doi: [https://doi.org/10.1016/S1350-4533\(99\)00066-1](https://doi.org/10.1016/S1350-4533(99)00066-1).
- [22] G. Gaudet, M. Raison, and S. Achiche, “Classification of Upper limb phantom movements in transhumeral amputees using electromyographic and kinematic features,” *Eng. Appl. Artif. Intell.*, vol. 68, no. April 2017, pp. 153–164, 2018, doi: 10.1016/j.engappai.2017.10.017.
- [23] M. Atzori, M. Cognolato, and H. Müller, “Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands,” *Front. Neurobot.*, vol. 10, no. SEP, pp. 1–10, 2016, doi: 10.3389/fnbot.2016.00009.
- [24] M. Atzori *et al.*, “Characterization of a benchmark database for myoelectric movement classification,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 1, pp. 73–83, 2015, doi: 10.1109/TNSRE.2014.2328495.
- [25] Y. Wu, B. Zheng, and Y. Zhao, “Dynamic Gesture Recognition Based on LSTM-CNN,” in *2018 Chinese Automation Congress (CAC)*, 2018, pp. 2446–2450, doi: 10.1109/CAC.2018.8623035.
- [26] O. Barron, M. Raison, G. Gaudet, and S. Achiche, “Recurrent Neural Network for electromyographic gesture recognition in transhumeral amputees,” *Appl. Soft Comput. J.*, vol. 96, p. 106616, 2020, doi: 10.1016/j.asoc.2020.106616.
- [27] D. Farina *et al.*, “The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 797–809, 2014, doi: 10.1109/TNSRE.2014.2305111.
- [28] A. Duivenvoorden, K. Lee, M. Raison, and S. Achiche, “Sensor fusion in upper limb area

- networks: A survey,” *2017 Glob. Inf. Infrastruct. Netw. Symp. GIIS 2017*, vol. 2017-Decem, pp. 56–63, 2017, doi: 10.1109/GIIS.2017.8169802.
- [29] M. Georgi and T. Schultz, “Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing,” pp. 99–108, 2015.
- [30] H. Kaur and J. Rani, “A Review: Study of Various Techniques of Hand Gesture Recognition,” 2016.
- [31] F. Gaetani, G. A. Zappatore, P. Visconti, and P. Primiceri, “Design of an Arduino-based platform interfaced by Bluetooth low energy with Myo armband for controlling an under-actuated transradial prosthesis,” *ICICDT 2018 - Int. Conf. IC Des. Technol. Proc.*, pp. 185–188, 2018, doi: 10.1109/ICICDT.2018.8399787.
- [32] Z. O. Khokhar, Z. G. Xiao, and C. Menon, “Surface EMG pattern recognition for real-time control of a wrist exoskeleton,” *Biomed. Eng. Online*, vol. 9, pp. 1–17, 2010, doi: 10.1186/1475-925X-9-41.
- [33] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, “Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges,” *Inf. Fusion*, vol. 35, pp. 68–80, 2017, doi: <https://doi.org/10.1016/j.inffus.2016.09.005>.
- [34] D. Novak and R. Riener, “A survey of sensor fusion methods in wearable robotics,” *Rob. Auton. Syst.*, vol. 73, pp. 155–170, 2015, doi: 10.1016/j.robot.2014.08.012.
- [35] M. Connan, E. R. Ramírez, B. Vodermayr, and C. Castellini, “Assessment of a wearable force and electromyography device and comparison of the related signals for myocontrol,” *Front. Neurobot.*, vol. 10, no. NOV, pp. 1–13, 2016, doi: 10.3389/fnbot.2016.00017.
- [36] E. Criswell, *Cram’s introduction to surface electromyography*. Jones & Bartlett Publishers, 2010.
- [37] C. Zizoua, M. Raison, S. Boukhenous, M. Attari, and S. Achiche, “Detecting muscle contractions using strain gauges,” *Electron. Lett.*, vol. 52, no. 22, pp. 1836–1838, 2016, doi: 10.1049/el.2016.2986.
- [38] Y. Bu *et al.*, “Ultrasensitive strain sensor based on superhydrophobic microcracked conductive Ti<sub>3</sub>C<sub>2</sub>T<sub>x</sub> MXene/paper for human-motion monitoring and E-skin,” *Sci. Bull.*,

- vol. 66, no. 18, pp. 1849–1857, 2021, doi: <https://doi.org/10.1016/j.scib.2021.04.041>.
- [39] J. A. Rogers, R. Ghaffari, and D.-H. Kim, “Stretchable Bioelectronics for Medical Devices and Systems,” *Stretchable Bioelectron. Med. Devices Syst.*, pp. 257–273, 2016, doi: [10.1007/978-3-319-28694-5](https://doi.org/10.1007/978-3-319-28694-5).
- [40] S. Wang, Y. Fang, H. He, L. Zhang, C. Li, and J. Ouyang, “Wearable Stretchable Dry and Self-Adhesive Strain Sensors with Conformal Contact to Skin for High-Quality Motion Monitoring,” *Adv. Funct. Mater.*, vol. 31, no. 5, p. 2007495, Jan. 2021, doi: <https://doi.org/10.1002/adfm.202007495>.
- [41] C. Zizoua, M. Raison, S. Boukhenous, M. Attari, and S. Achiche, “Development of a Bracelet With Strain-Gauge Matrix for Movement Intention Identification in Traumatic Amputees,” *IEEE Sens. J.*, vol. 17, no. 8, pp. 2464–2471, 2017, doi: [10.1109/jsen.2017.2666784](https://doi.org/10.1109/jsen.2017.2666784).
- [42] Y. Zhou *et al.*, “Highly sensitive, self-powered and wearable electronic skin based on pressure-sensitive nanofiber woven fabric sensor,” *Sci. Rep.*, vol. 7, no. 1, p. 12949, 2017, doi: [10.1038/s41598-017-13281-8](https://doi.org/10.1038/s41598-017-13281-8).
- [43] S. Seyedin, J. M. Razal, P. C. Innis, A. Jeiranikhameneh, S. Beirne, and G. G. Wallace, “Knitted Strain Sensor Textiles of Highly Conductive All-Polymeric Fibers,” *ACS Appl. Mater. Interfaces*, vol. 7, no. 38, pp. 21150–21158, Sep. 2015, doi: [10.1021/acsami.5b04892](https://doi.org/10.1021/acsami.5b04892).
- [44] T. Li, Z. Huang, Z. Suo, S. P. Lacour, and S. Wagner, “Stretchability of thin metal films on elastomer substrates,” *Appl. Phys. Lett.*, vol. 85, no. 16, pp. 3435–3437, Oct. 2004, doi: [10.1063/1.1806275](https://doi.org/10.1063/1.1806275).
- [45] J. Lee *et al.*, “A stretchable strain sensor based on a metal nanoparticle thin film for human motion detection,” *Nanoscale*, vol. 6, no. 20, pp. 11932–11939, 2014, doi: [10.1039/c4nr03295k](https://doi.org/10.1039/c4nr03295k).
- [46] A. Hirsch, H. O. Michaud, A. P. Gerratt, S. de Mulatier, and S. P. Lacour, “Intrinsically Stretchable Biphasic (Solid–Liquid) Thin Metal Films,” *Adv. Mater.*, vol. 28, no. 22, pp. 4507–4512, 2016, doi: [10.1002/adma.201506234](https://doi.org/10.1002/adma.201506234).
- [47] A. Hirsch, L. Dejace, H. O. Michaud, and S. P. Lacour, “Harnessing the Rheological

- Properties of Liquid Metals To Shape Soft Electronic Conductors for Wearable Applications,” *Acc. Chem. Res.*, vol. 52, no. 3, pp. 534–544, Mar. 2019, doi: 10.1021/acs.accounts.8b00489.
- [48] H. Zhang, A. Lowe, A. Kalra, and Y. Yu, “A Flexible Strain Sensor Based on Embedded Ionic Liquid,” *Sensors (Basel)*, vol. 21, no. 17, p. 5760, Aug. 2021, doi: 10.3390/s21175760.
- [49] D. Y. Choi *et al.*, “Highly Stretchable, Hysteresis-Free Ionic Liquid-Based Strain Sensor for Precise Human Motion Monitoring,” *ACS Appl. Mater. Interfaces*, vol. 9, no. 2, pp. 1770–1780, Jan. 2017, doi: 10.1021/acsami.6b12415.
- [50] X. Song *et al.*, “A graphene-coated silk-spandex fabric strain sensor for human movement monitoring and recognition,” *Nanotechnology*, vol. 32, no. 21, p. 215501, 2021, doi: 10.1088/1361-6528/abe788.
- [51] S. Yao, L. Vargas, X. Hu, and Y. Zhu, “A Novel Finger Kinematic Tracking Method Based on Skin-Like Wearable Strain Sensors,” *IEEE Sens. J.*, vol. 18, no. 7, pp. 3010–3015, 2018, doi: 10.1109/JSEN.2018.2802421.
- [52] Y. L. Park, B. R. Chen, and R. J. Wood, “Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors,” *IEEE Sens. J.*, vol. 12, no. 8, pp. 2711–2718, 2012, doi: 10.1109/JSEN.2012.2200790.
- [53] Q. Gao, H. Li, J. Zhang, Z. Xie, J. Zhang, and L. Wang, “Microchannel Structural Design For a Room-Temperature Liquid Metal Based Super-stretchable Sensor,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–8, 2019, doi: 10.1038/s41598-019-42457-7.
- [54] L. Detuncq, Bruno; Marchand, *Méthodes expérimentales et instrumentation en mécanique*. Presses internationales Polytechnique.
- [55] W. Li and J. Wang, “Effective Adaptive Kalman Filter for MEMS-IMU/Magnetometers Integrated Attitude and Heading Reference Systems,” *J. Navig.*, vol. 66, no. 1, pp. 99–113, 2013, doi: DOI: 10.1017/S0373463312000331.
- [56] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, “Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion,” *Sensors (Switzerland)*, vol. 17, no. 6, pp. 1–40, 2017, doi: 10.3390/s17061257.



- [57] N. L. W. Keijsers, “Neural Networks,” K. Kompoliti and L. V. B. T.-E. of M. D. Metman, Eds. Oxford: Academic Press, 2010, pp. 257–259.
- [58] A. Goodfellow, Ian; Bengio, Yoshua; Courville, *Deep Learning*. MIT Press, 2016.
- [59] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943, doi: 10.1007/BF02478259.
- [60] F. Rosenblatt, *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Cornell Aeronautical Laboratory, 1957.
- [61] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.
- [62] K. Ng, Andrew; Katanforoosh, “Deep Learning-CS229 Lecture Notes.” [http://cs229.stanford.edu/summer2020/cs229-notes-deep\\_learning.pdf](http://cs229.stanford.edu/summer2020/cs229-notes-deep_learning.pdf).
- [63] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” pp. 1–20, 2018, [Online]. Available: <http://arxiv.org/abs/1811.03378>.
- [64] C. M. Bishop, *Pattern Recognition and Machine Learning*. 2006.
- [65] C. Olah, “Understanding LSTM Networks,” 2015. .
- [66] R. R. Drumond, B. A. D. Marques, C. N. Vasconcelos, and E. Clua, “PEEK: An LSTM recurrent network for motion classification from sparse data,” *VISIGRAPP 2018 - Proc. 13th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 1, no. Visigrapp, pp. 215–222, 2018, doi: 10.5220/0006585202150222.
- [67] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [68] J. Singh, S. P. Mohanty, and D. K. Pradhan, *Introduction to Statistical Learning*. 2013.
- [69] G. Hinton, N. Srivastava, and K. Swersky, “Lecture 6a :Overview of mini-batch gradient descent,” in *Neural Networks for Machine Learning*, 2012.
- [70] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, “Randomized smoothing for (parallel) stochastic optimization,” *Proc. IEEE Conf. Decis. Control*, vol. 12, pp. 5442–5444, 2012,

- doi: 10.1109/CDC.2012.6426698.
- [71] D. P. Kingma and J. Lei Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,” *Iclr*, pp. 1–15, 2015, [Online]. Available: <https://arxiv.org/pdf/1412.6980.pdf> %22 entire document.
- [72] M. Hardt, “UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION,” pp. 1–15, 2004, [Online]. Available: <http://arxiv.org/abs/1611.03530>.
- [73] S. Nitish, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.* 15, vol. Research 1, no. 3–4, p. 192901958, 2014.
- [74] A. Dosovitskiy *et al.*, “FlowNet: Learning optical flow with convolutional networks,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 2758–2766, 2015, doi: 10.1109/ICCV.2015.316.
- [75] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview,” *Bioinformatics*, vol. 16, no. 5, pp. 412–424, May 2000, doi: 10.1093/bioinformatics/16.5.412.
- [76] C. Goutte and E. Gaussier, “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation,” *Lect. Notes Comput. Sci.*, vol. 3408, pp. 345–359, 2005, doi: 10.1007/978-3-540-31865-1\_25.
- [77] B. Sensortec, “Integrated MCU + flash. Integrated sensor fusion. Smart sensor: BNO055.” <https://www.bosch-sensortec.com/products/smart-sensors/bno055/>.
- [78] MathWorks, “Calibrate BNO055 Sensors.” <https://www.mathworks.com/help/supportpkg/arduinoio/ug/calibrate-sensors.html>.
- [79] Adafruit, “Adafruit\_BNO055.” .
- [80] B. Maundy and S. J. G. Gift, “Strain gauge amplifier circuits,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 4, pp. 1–8, 2013, doi: 10.1109/TIM.2013.2246904.
- [81] A. Device, “Instrumentation Amplifier Diamond Plot Tool.” .
- [82] M. Atzori *et al.*, “Building the NINAPRO Database: A Resource for the Biorobotics

- Community - HES SO Valais publications - Aigaion 2.0,” *Proc. IEEE Int. Conf. Biomed. Robot. Biomechatronics*, p. 51, 2012, [Online]. Available: <http://publications.hevs.ch/index.php/publications/show/1172>.
- [83] R. Khusainov, D. Azzi, I. E. Achumba, and S. D. Bersch, “Real-time human ambulation, activity, and physiological monitoring: Taxonomy of issues, techniques, applications, challenges and limitations,” *Sensors (Switzerland)*, vol. 13, no. 10, pp. 12852–12902, 2013, doi: 10.3390/s131012852.
- [84] A. Gijbets and B. Caputo, “Exploiting accelerometers to improve movement classification for prosthetics,” in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, 2013, pp. 1–5, doi: 10.1109/ICORR.2013.6650476.

## ANNEXE A EXEMPLE DE CALCUL DE GRADIENT

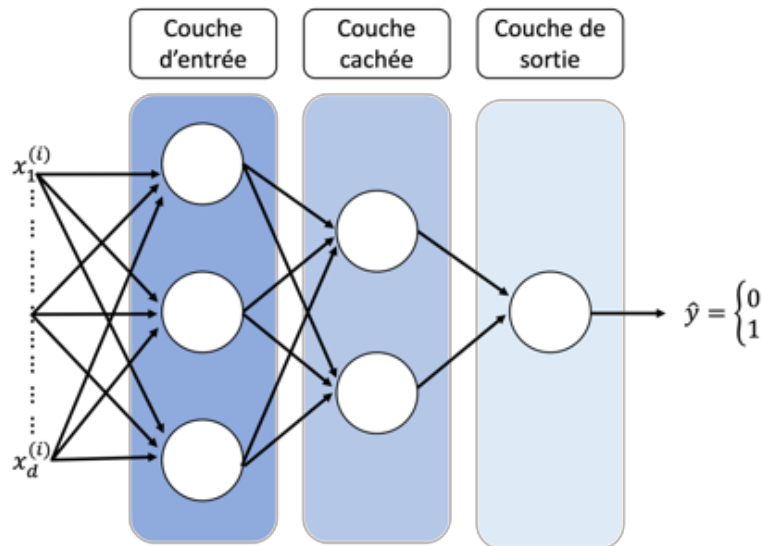


Figure.1 Perceptron multicouche

Le perceptron à plusieurs couches de la Figure 1 est identique à celle de la figure 2.7. Ce modèle est utilisé pour l'exemple de calcul. Cette architecture comprend 3 neurones dans la couche d'entrée, 2 neurones dans la couche cachée et un neurone dans la couche de sortie. Les calculs qui suivent sont inspirés et adaptés de [62].

On peut définir  $z = W^T x + b$  et  $a = g(z)$  où  $g(z)$  est une fonction d'activation quelconque, section 2.3.2.  $W$  représente la matrice des paramètres du réseau.

Pour des questions de clarification  $W^{[l]}$ , avec les crochets représentent tout ce qui relié à la couche  $l$  du réseau de neurones. Ainsi,  $W^{[1]}$  Représente la matrice de paramètres pour la couche 1 du réseau de neurones. De même,  $b^{[1]}$  le vecteur de biais associé à la couche 1 du réseau de neurones.

La passe avant peut être définie comme suit :

La première couche va effectuer les transformations suivantes sur le vecteur d'entrée  $x$  :

$$z^{[1]} = W^{[1]}x^{[i]} + b^{[1]} \quad z^{[1]}, a^{[1]} \in \mathbb{R}^{3 \times 1}, \quad (8.1)$$

$$a^{[1]} = g(z^{[1]}) \quad W^{[1]} \in \mathbb{R}^{3 \times d}, b^{[1]} \in \mathbb{R}^{3 \times 1} \quad (8.2)$$

La seconde couche va effectuer les transformations suivantes :

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} \quad z^{[2]}, a^{[2]} \in \mathbb{R}^{2 \times 1}, \quad (8.3)$$

$$a^{[2]} = g(z^{[2]}) \quad W^{[2]} \in \mathbb{R}^{2 \times 3}, b^{[2]} \in \mathbb{R}^{2 \times 1} \quad (8.4)$$

La dernière couche qui est la couche de sortie va effectuer les transformations suivantes :

$$z^{[3]} = W^{[3]}a^{[2]} + b^{[3]} \quad z^{[3]}, a^{[3]} \in \mathbb{R}^{1 \times 1} \quad (8.5)$$

$$\hat{y}^{(i)} = a^{[3]} = g(z^{[3]}) \quad W^{[3]} \in \mathbb{R}^{1 \times 2}, b^{[3]} \in \mathbb{R}^{1 \times 1} \quad (8.6)$$

La prochaine étape de l'entraînement du réseau de neurones consiste à mettre à jour les paramètres en calculant la fonction coût. Pour cet exemple nous allons utiliser la fonction de coût logarithmique :

$$\mathcal{L}(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (8.7)$$

La fonction de coût  $\mathcal{L}(\hat{y}, y)$  Que nous allons noter  $\mathcal{L}$  produit une seule valeur scalaire. Le calcul de cette valeur permettra de mettre à jour tous les paramètres dans chacune des couches  $l$  du réseau de neurones selon les équations suivantes :

$$W^{[l]} = W^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial W^{[l]}} \quad (8.8)$$

$$b^{[l]} = b^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial b^{[l]}} \quad (8.9)$$

où  $\alpha$  représente le taux d'apprentissage. Une fois la coût calculée, les gradients peuvent être obtenus par rapport aux paramètres que sont  $W^{[1]}$ ,  $W^{[2]}$ ,  $W^{[3]}$ ,  $b^{[1]}$ ,  $b^{[2]}$ ,  $b^{[3]}$ . En utilisant les équations précédentes, cela vient à calculer :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W^{[1]}}; & \quad \frac{\partial \mathcal{L}}{\partial W^{[2]}}; & \quad \frac{\partial \mathcal{L}}{\partial W^{[3]}} \\ \frac{\partial \mathcal{L}}{\partial b^{[1]}}; & \quad \frac{\partial \mathcal{L}}{\partial b^{[2]}}; & \quad \frac{\partial \mathcal{L}}{\partial b^{[3]}}; \end{aligned} \quad (8.10)$$

La partie qui suit montre un exemple de gradient calculé par rapport aux paramètres  $W^{[2]}$  pour l'algorithme de rétropropagation.

$$\frac{\partial \mathcal{L}}{\partial W^{[2]}} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial W_{11}^{[2]}} & \frac{\partial \mathcal{L}}{\partial W_{12}^{[2]}} & \frac{\partial \mathcal{L}}{\partial W_{13}^{[2]}} \\ \frac{\partial \mathcal{L}}{\partial W_{21}^{[2]}} & \frac{\partial \mathcal{L}}{\partial W_{22}^{[2]}} & \frac{\partial \mathcal{L}}{\partial W_{23}^{[2]}} \end{bmatrix} \quad (8.11)$$

Maintenant pour calculer, les termes de la matrice  $\frac{\partial \mathcal{L}}{\partial W^{[2]}}$  nous utilisons les règles de calcul de la dérivation en chaîne :

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{[2]}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \mathcal{L}}{\partial W_{ij}^{[2]}} \quad (8.12)$$

$$= \frac{\partial \mathcal{L}}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial W_{ij}^{[2]}} \quad (8.13)$$

$$= \frac{\partial \mathcal{L}}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial z^{[3]}} \frac{\partial z^{[3]}}{\partial W_{ij}^{[2]}} \quad (8.14)$$

$$= \frac{\partial \mathcal{L}}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial z^{[3]}} \frac{\partial z^{[3]}}{\partial W_{ij}^{[2]}} \quad (8.15)$$

$$= \frac{\partial \mathcal{L}}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial z^{[3]}} \frac{\partial z^{[3]}}{\partial a^{[2]}} \frac{\partial a^{[2]}}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial W_{ij}^{[2]}} \quad (8.16)$$

On remarque que :

$$\frac{\partial \mathcal{L}}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial z^{[3]}} = \frac{\partial \mathcal{L}}{\partial z^{[3]}} \quad (8.17)$$

$$\frac{\partial z^{[3]}}{\partial a^{[2]}} = W^3 \quad (8.18)$$

$$\frac{\partial a^{[2]}}{\partial z^{[2]}} = \text{diag}(g'(z^2)) \quad (8.19)$$

$$\frac{\partial z^{[2]}}{\partial W_{ij}^{[2]}} = a_j^{[1]} \mathbf{e}_i \quad (8.20)$$

et  $\frac{\partial \mathcal{L}}{\partial z^{[3]}}$  peut être calculée selon la procédure suivante :

$$\frac{\partial \mathcal{L}}{\partial z^{[3]}} = \frac{\partial}{\partial z^{[3]}} [-y \log(\hat{y}) - (1-y) \log(1-\hat{y})] \quad (8.21)$$

$$= \frac{\partial}{\partial z^{[3]}} [-y \log \sigma(z^{[3]}) - (1-y) \log(1-\sigma(z^{[3]}))] \quad (8.22)$$

$$= -y \frac{1}{\sigma(z^{[3]})} \sigma(z^{[3]}) (1-\sigma(z^{[3]})) - (1-y) \frac{1}{(1-\sigma(z^{[3]}))} (-1) \sigma(z^{[3]}) (1-\sigma(z^{[3]})) \quad (8.23)$$

$$= -y (1-\sigma(z^{[3]})) + (1-y) \sigma(z^{[3]}) \quad (8.24)$$

$$= \sigma(z^{[3]}) - y \quad (8.25)$$

$$\frac{\partial \mathcal{L}}{\partial z^{[3]}} = a^{[3]} - y \quad (8.26)$$

Ainsi, le résultat suivant est obtenu :

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{[2]}} = (a^{[3]} - y) W^{[3]} \circ g'(z^{[2]}) a_j^{[1]} \mathbf{e}_i \quad (8.27)$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{[2]}} = (a^{[3]} - y) W^{[3]} \circ g'(z^{[2]}) a_j^{[1]} \quad (8.28)$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{[2]}} = [(a^{[3]} - y) W^{[3]} \circ g'(z^{[2]})] a_j^{[1]T} \quad (8.29)$$

avec  $\circ$  qui représente le produit par élément ou le produit Hadamard.

## **ANNEXE B COURBES D'APPRENTISSAGE DES HYPERPARAMETRES**

On remarque que pour un taux d'apprentissage de  $1e-01$  le modèle n'a pas convergé, Tableau 1. La courbe de la précision de classification est constante pour l'ensemble de validation. Lorsque le taux d'apprentissage a diminué à  $1e-02$  on a commencé à voir une convergence du modèle. Le taux de  $1e-03$  a permis d'avoir la meilleure courbe d'apprentissage. Lorsque le taux d'apprentissage à continuer à diminuer, la précision de classification obtenu à la fin de l'entraînement a été basse. Le taux d'apprentissage affecte alors la vitesse à laquelle le modèle apprend à identifier les classes de mouvement.

Pour le nombre de filtres, Tableau 2, on remarque qu'une augmentation du nombre de filtres de 9 à 32 a entraîné une augmentation de la précision de classification de 91.4% à 95.6%. Ce qui est normal car les filtres ont pour but d'extraire l'information pertinente des signaux. Cependant, on remarque qu'à partir de 32 filtres la précision de classification a diminué lorsqu'on a augmenté le nombre de filtres. En effet, un nombre de filtres trop élevé (nombre de caractéristiques) peut introduire des caractéristiques qui portent à confusion l'algorithme de classification.

L'augmentation du nombre de neurones dans les couches LSTM, Tableau 3, a entraîné une augmentation de la précision de classification. Cependant, cette augmentation (0.75%) est minime comparé au nombre de paramètres à mettre à jour due à l'augmentation de 32 neurones à 128 neurones. En effet, il faut compter 199,104 paramètres supplémentaires. Il est préférable de garder un nombre de neurones bas permettant d'avoir un bon résultat de classification avec le minimum de paramètres à entraîner.

En entraînant le modèle sans terme de régularisation, Tableau 4, on a obtenu une précision de classification de 94.6%. Le terme de régularisation de 0.002 a obtenu le meilleur résultat de classification. En augmentant ce terme les résultats de classification ont diminué. Mais il est quand même intéressant de noter la superposition des courbes de l'ensemble d'entraînement et de validation, ce qui démontre la capacité de ce terme à maintenir une faible variance

La taille du lot a également affecté la vitesse de convergence du réseau. Le réseau a convergé plus rapidement avec une petite taille de lot, Tableau 5. En effet, des lots plus petits permettent de mettre à jour plus de fois les paramètres du réseau comparé à un lot plus important. Cependant, un lot trop petit ne permet pas au modèle de voir suffisamment d'exemples avant de pouvoir mettre à jour les paramètres d'apprentissage.



Tableau 1 Taux d'apprentissage

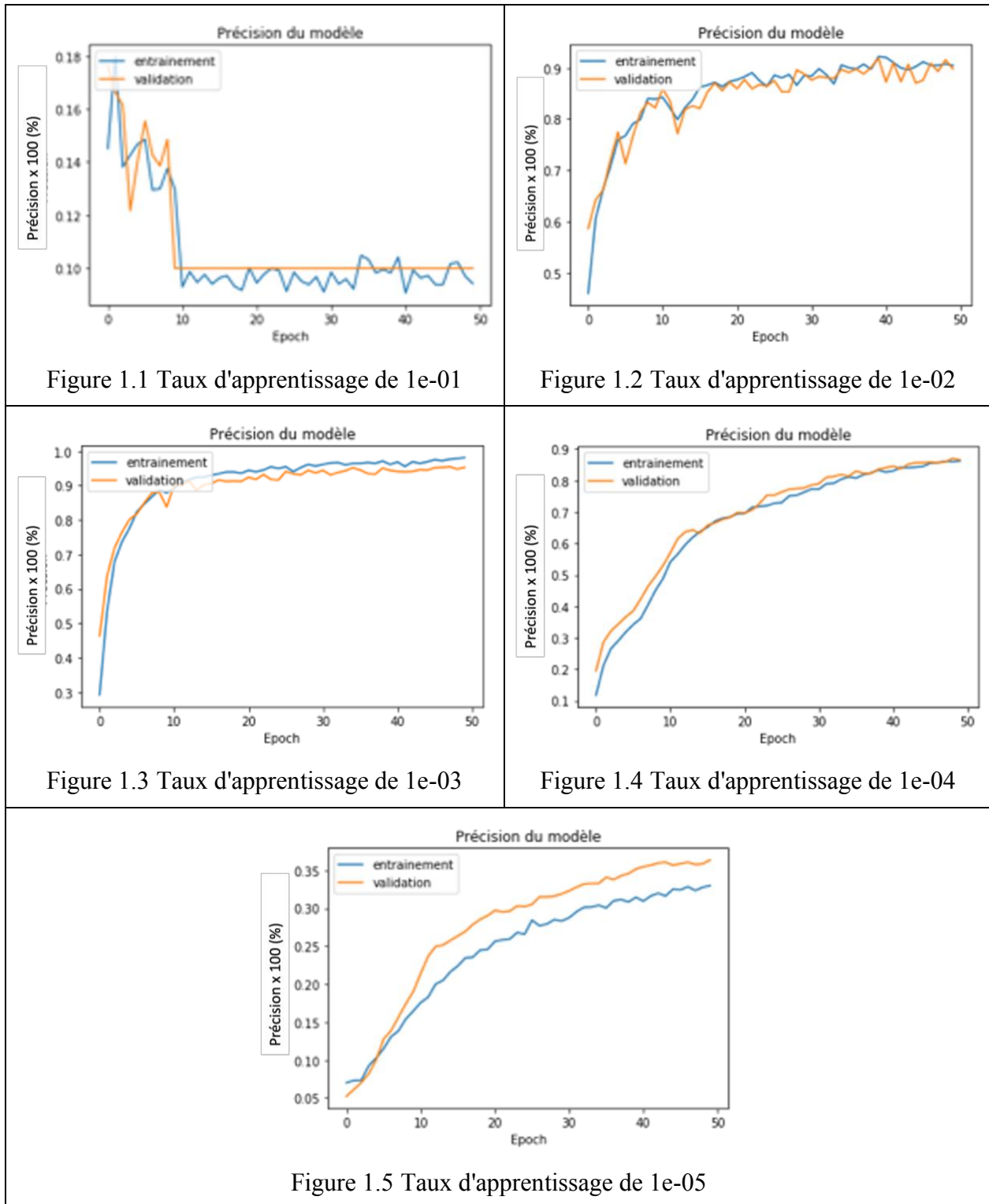


Tableau 2 Nombre de filtres

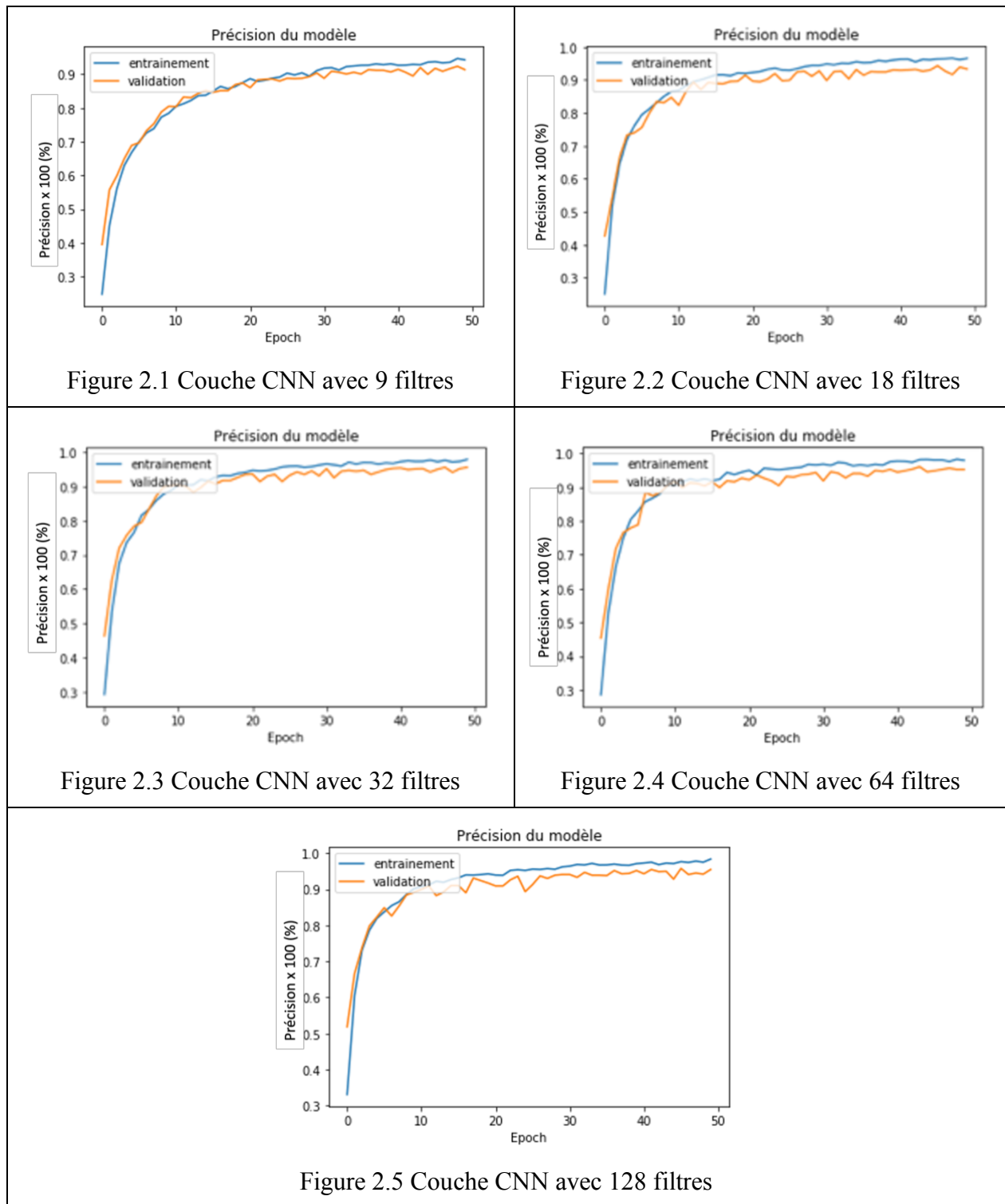


Tableau 3 Nombre de neurones

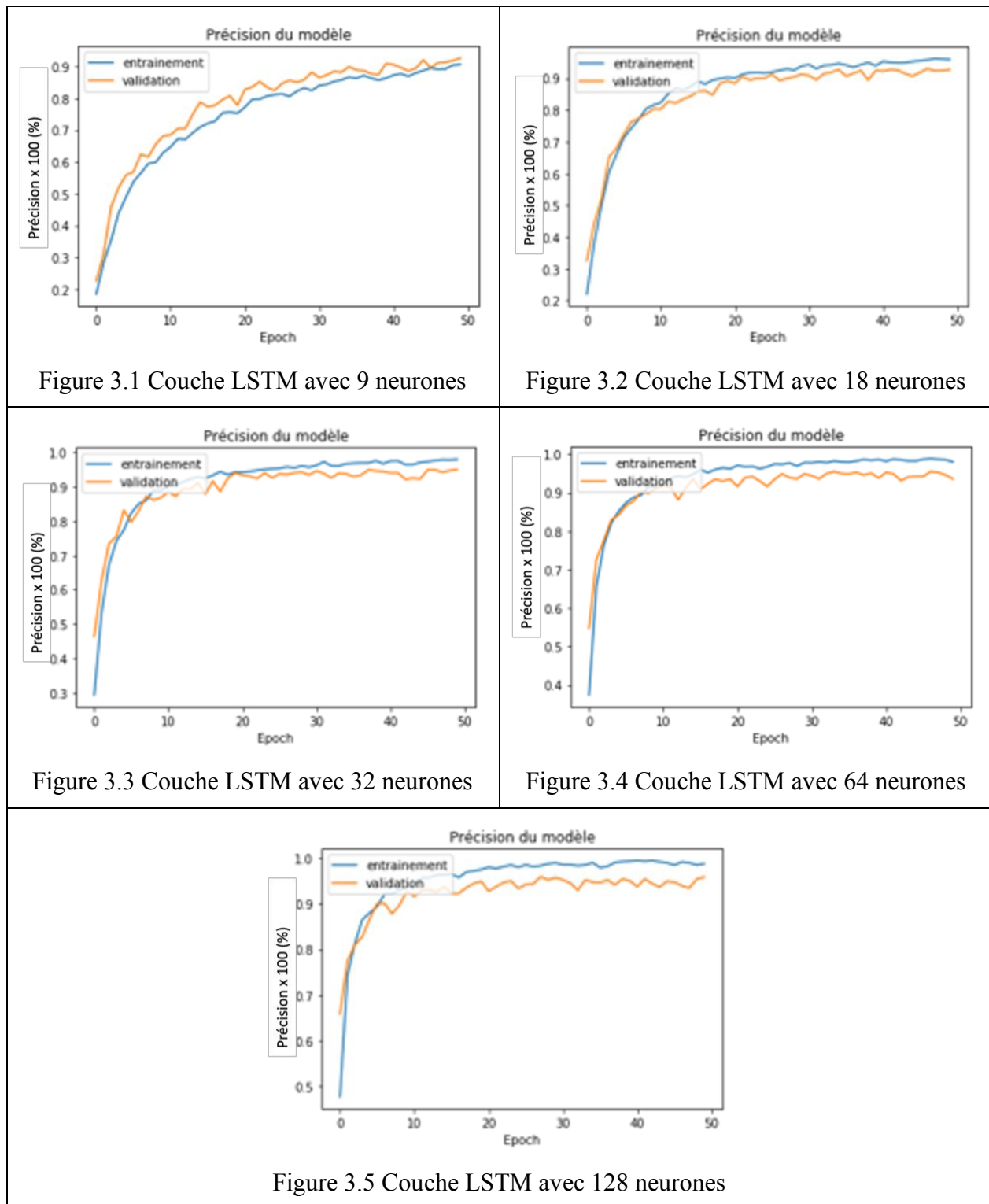


Tableau 4 Terme de régularisation

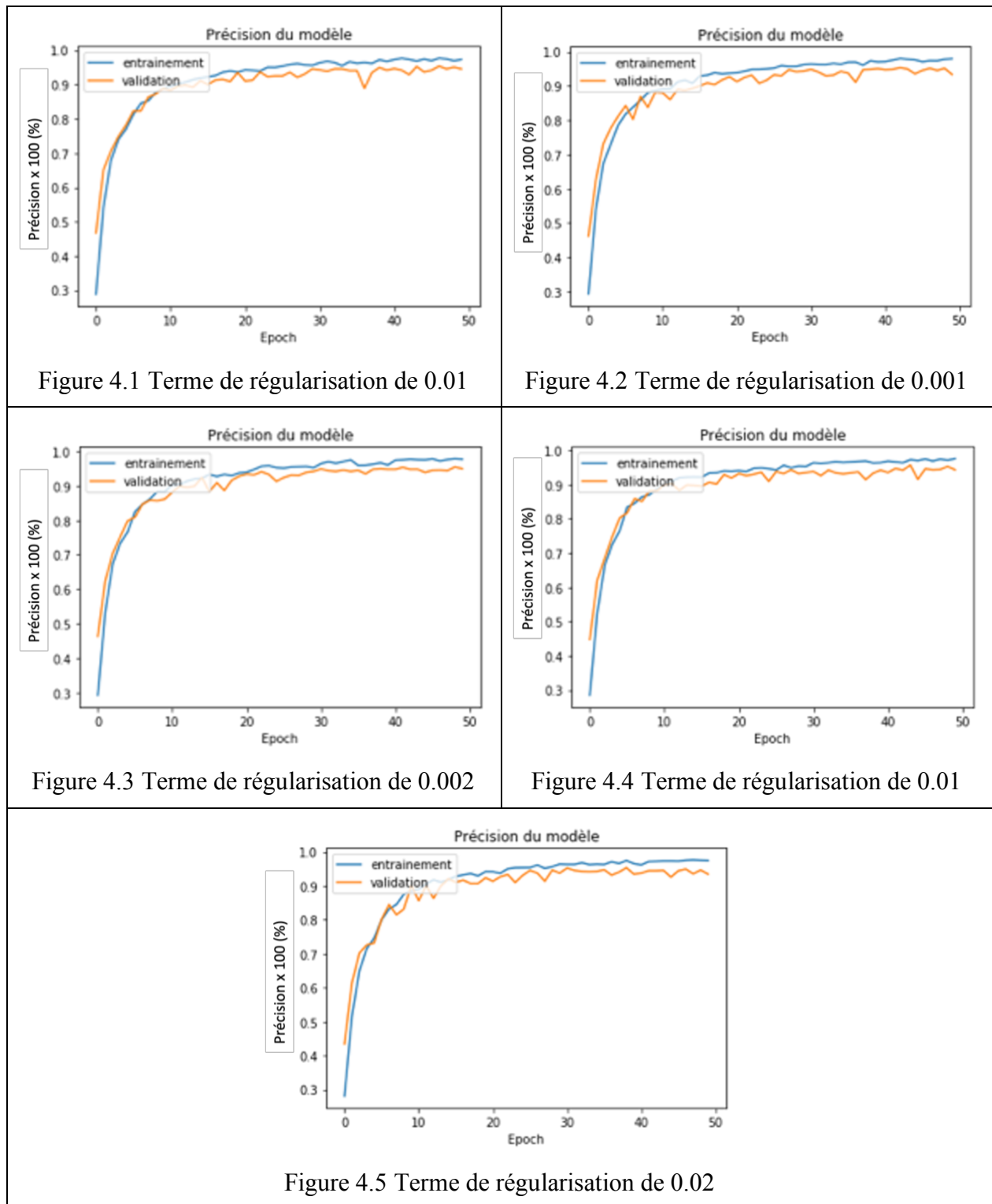


Tableau 5 Nombre d'échantillons par lot

