

République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi Ouzou
Faculté de génie Electrique et Informatique
Département Informatique



Mémoire de fin d'étude
En vue d'obtention du diplôme Master
Domaine : Mathématique et Informatique
Filière : Informatique
Spécialité : Réseaux, Mobilité et Systèmes Embarqués

Thème
**« Conception et réalisation d'un système de contrôle pour une
prothèse myoélectrique »**

Présenté Par :
LAIDAOUI Salah

Proposé :
Mr HEMDANI.
Dirigé par :
Mr LAGHROUCHE.

Soutenu le 29/11/2020, devant le jury composé de :
Mr. M. DAOUI Président
Mr. M. LAGHROUCHE Promoteur
Mr. M.LAHDIR Examineur

Promotion 2019/2020

Remerciements :

Je veux exprimer par ces quelques lignes mes remerciements et ma gratitude envers tous ceux en qui, par leur présence, leur soutien, leur disponibilité et leurs conseils, j'ai trouvé courage afin d'accomplir ce modeste projet.

Les plus grandes leçons ne sont pas tirées d'un livre mais d'un enseignant tel que vous. Merci Monsieur C.HAMDANI d'avoir pris le temps de m'aider au cours de mon cursus et de m'avoir accompagné dans la maîtrise de mes connaissances.

Je tiens également à remercier monsieur M.LEGHROUCHE qui m'a fait l'honneur d'être mon encadreur, je voudrai lui témoigner ma gratitude pour sa patience et son soutien qui m'a été précieux afin de mener mon travail à bon port.

Merci.

Je remercie ensuite les membres du jury, Monsieur M.DAOUI et Monsieur M.LEHDIR, qui m'ont fait l'honneur de bien vouloir étudier avec attention mon travail.

Je voudrai exprimer également ma profonde gratitude à tous les enseignants du département informatique spécialement les enseignants de la spécialité RMSE.

Dédicace :

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,

A ma chère mère,

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.

Je vous remercie pour le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

A la mémoire de ma grande mère,

Ce travail est dédié à ma grande mère, qui m'a toujours poussé et motivé dans mes études.

J'espère que, du monde qui est sein maintenant, elle apprécie cet humble geste comme preuve de reconnaissance de la part d'un petit fils qui a toujours prié pour le salut de son âme. Puisse dieu, le tout puissant, l'avoir en sa sainte miséricorde.

A MON CHER ONCLE AOMAR

Tu as été et tu seras toujours un exemple pour moi par tes qualités humaines, ta persévérance et perfectionnisme. En témoignage de brut d'années de sacrifices, sollicitudes, d'encouragement et de prières. Pourrais-tu trouver dans ce travail le fruit de toutes tes peines et de tous tes efforts.

En ce jour, j'espère réaliser l'un de tes rêves. Aucune dédicace ne saurait exprimer mes respects, ma reconnaissance et mon profond amour.

Puisse Dieu te préserver et te procurer santé et bonheur.

A MON CHER ONCLE HACENE

Mon conseiller, et ami fidèle, qui m'a assisté dans les moments difficiles et m'a pris doucement par la main pour traverser ensemble des épreuves pénibles.

Je te suis très reconnaissant, et je ne te remercierai jamais assez pour ton amabilité, ta générosité, ton aide précieuse.

A mes deux chères Tantes

Vous avez toujours été présentes pour les bons conseils.

Votre affection et votre soutien m'ont été d'un grand secours au long de ma vie.

Veillez trouver dans ce modeste travail ma reconnaissance pour tous vos efforts.

**A mes petits cousins
Mellissa, Rayene, Said, Asma, Alycia, Cerine, Malak et Islam**

En témoignage de mon affection fraternelle, de ma profonde tendresse et reconnaissance, je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protège et vous garde.

A ma chère Lilia

Merci énormément pour ton soutien plus que précieux, Merci pour ton grand cœur toutes tes qualités qui seraient trop longues à énumérer. Ma vie ne serait pas aussi magique sans ta présence et ton amour.

Sommaire :

SOMMAIRE :	1
LISTE DES FIGURES :	4
LISTE DES ABREVIATIONS :	6
INTRODUCTION GENERALE	7
CHAPITRE 1 : VUE D'ENSEMBLE DU SIGNAL SEMG	10
INTRODUCTION :	10
1.1 PHYSIOLOGIE	10
1.1.1 Les neurones :.....	10
1.1.2 Les muscles :.....	11
1.1.3 Les nerfs et les muscles sont des tissus excitables.	13
1.2 ACQUISITION DE DONNEES EMG	14
1.2.1 Les techniques d'acquisitions EMG	14
1.2.2 Muscles cibles.....	14
1.2.3 Mesure du signal sEMG à l'aide des électrodes :.....	17
1.2.4 Caractéristiques et prétraitement du signal EMG	20
CHAPITRE 2 : TRAITEMENT ET CLASSIFICATION DES SIGNAUX SEMG	23
INTRODUCTION	23
2.1 SEGMENTATION DES DONNEES:	24
2.2 EXTRACTION DES FONCTIONNALITES :	25
2.2.1 Domaine temporel :.....	25
2.2.2 Domaine fréquentiel :.....	26

2.2.3	Domaine temps-fréquence :	27
2.3	CLASSIFICATION :	27
2.3.1	Réseaux de neurones Artificiels (RNA).....	28
2.3.2	Classificateur à logique floue (FL). ^[25]	30
2.3.3	Classificateur SVM (Support Vector Machines) ^[17]	31
2.3.4	Classificateur K-plus proche voisin (KNN) ^[25]	32
CHAPITRE 3 : CONCEPTION.....		34
INTRODUCTION :.....		34
3.1	ANALYSE DU SYSTEME DE CONTROLE DE LA PROTHESE :.....	35
3.2	CONCEPTION DU RNA :.....	37
3.2.1	Collecte de données :.....	38
3.2.2	Création du dataset :	38
3.2.3	Création du RNA.....	39
3.2.4	Entraînement du RNA.....	39
3.2.5	Utilisation du réseau :.....	39
CHAPITRE 4 : REALISATION.....		40
INTRODUCTION :.....		40
4.1	ENVIRONNEMENTS DE DEVELOPPEMENT :.....	40
4.1.1	Environnements de développement matériel :.....	40
4.1.1.1	Circuit d'acquisition de signaux sEMG:.....	40
4.1.1.2	La carte à microcontrôleur:.....	41
I.	Caractéristiques ^[26] :	42
II.	Fonctionnalités ^[26] :	43
4.1.1.3	Servomoteur MG-996R :	47
4.1.1.4	Le pont diviseur de tension :.....	48
4.1.1.5	La Main INMOOV :	49
4.1.2	Environnements de développement logiciel :	50
4.1.2.1	STM32CubeIDE :	50

4.1.2.2 XCUBE.AI :	51
4.1.2.3 GOOGLE COLAB:	52
4.1.2.4 KERAS:	52
4.1.2.5 SerialPlot:	52
4.2 ASSEMBLAGE DU SYSTEME :	53
4.2 Configuration du projet :	56
4.2.1 Configuration des ADCs :	57
4.2.2 Configuration de l'usart:	60
4.2.3 Structuration/ collecte des données:	61
4.2.4 Création du dataset:	62
4.2.5 Création du RNA:	63
4.2.6 Implémentation du RNA sur μ C:	65
4.2.7 Configuration des servomoteurs:	67
CONCLUSION GENERALE	70
BIBLIOGRAPHIE :	72

Liste des figures :

Figure 1: anatomie d'un neurone.....	11
Figure 2 : anatomie d'un muscle squelettique.....	12
Figure 3 : Jonction neuromusculaire.....	12
Figure 4 : représentation de différentes unités motrices.....	13
Figure 5 : Muscles de l'avant-bras (couche superficielle) : vue postérieure ^[6]	15
Figure 6: Muscles de l'avant-bras (couche superficielle) : vue antérieure ^[6]	16
Figure 7 : électrode sèche.....	18
Figure 8 : électrode gélifiée.....	18
Figure 9 : représentation schématique de l'emplacement d'une électrode ^[16]	19
Figure 10: positionnement des électrodes.....	19
Figure 11 : amplification différentielle. ^[16]	21
Figure 12: schéma général du fonctionnement des systèmes de contrôle de prothèses bioniques...	23
Figure 13 : segmentation disjointe ^[17]	24
Figure 14 : Segmentation superposée. ^[17]	24
Figure 15 : Structure d'un neurone artificiel.....	28
Figure 16 : Structure d'un MLP.....	29
Figure 17 : Représentation fonctionnelle d'un MLP.....	30
Figure 18 : Schéma fonctionnelle d'un classificateur a logique floue ^[25]	31
Figure 19 : structure d'un classificateur SVM.....	31
Figure 20 : Schéma résumant le chapitre.....	33
Figure 21 : vue schématique du système.....	34
Figure 22 : représentation schématique de l'évolution du signal EMG dans notre système.	35
Figure 23 : les gestes choisis dans cette étude.	36
Figure 24: structure simple d'un MLP.....	37
Figure 25 : représentation schématique du MLP développé dans cette étude.....	37
Figure 26 : collecte de données.....	38
Figure 27 : Processus d'entraînement d'un RNA.....	39
Figure 28 : circuit d'acquisition de signaux sEMG.....	41
Figure 29 : Composants de la carte STM32F4 DISCOVERY.....	43
Figure 30 : Schéma d'un servomoteur.	48
Figure 31: servomoteur.....	48
Figure 32 : schéma du pont diviseur de tension.	48
Figure 33: le prototype de mains robotisé utilisé dans notre projet.....	50
Figure 34 : interfaces du logiciel.....	53
Figure 35 : schéma représentant les branchements effectués.....	53
Figure 36 : schéma électrique du système.....	54
Figure 37 : mise en œuvre.....	55
Figure 38 : les étapes de la réalisation de notre système.....	56
Figure 39 : Configuration des ADCs -1-.....	57
Figure 40 : Configuration des Horloges.....	58
Figure 41 : Configuration du DMA -1-.....	58

Figure 42 : Configuration du DMA -2-	59
Figure 43 : Configuration des ADCs -2-	59
Figure 44 : Configuration d l'USART.	60
Figure 45 : Initialisation du DMA.	61
Figure 46 : Fonction de structuration de données -1-	61
Figure 47 : Fonction de structuration de données-2-	61
Figure 48 : Affichage des données a l'écran.	62
Figure 49 : Une Partie du dataset.	62
Figure 50 : Importation du Dataset.	63
Figure 51 : Ensemble d'entrainement/Ensemble de tests.	63
Figure 52 : Définition du modèle.	64
Figure 53 : Précision du modèle.	64
Figure 54 : Exportation du Modèle.	65
Figure 55 : Configuration du package d'intelligence artificielle.	65
Figure 56 : Importation du modèle keras.	66
Figure 57 : la fonction MX_X_CUBE_AI_Process.	66
Figure 58 : Représentation schématiques de la longueur d'impulsion des servomoteurs	67
Figure 59 : Configuration des timers.	68
Figure 60 : Initialisation du Timer.	68
Figure 61 : exemple basique pour faire tourner un servomoteur.	68
Figure 62 : fonction d'actionnement de la main robotique.	69

Liste des abréviations :

EMG : électromyographie.

sEMG : électromyographie de surface.

SNC : Système nerveux centrale.

SNP : Système nerveux périphérique.

SNR : Rapport signal bruit (signal to noise ratio).

IEMG : EMG intégrée.

MAV : Valeur absolue moyenne.

MMAV1 : Valeur absolue moyenne modifié.

RMS : moyenne quadratique (root mean square).

VAR : variance.

SSI : Intégrale carrée simple.

HEMG : Histogramme EMG.

FMD : Médiane de fréquence.

FMN : Moyenne de fréquence.

MFMD : Médiane de fréquence modifiée.

PSD : densité spectrale de puissance.

RNA : Réseau de neurone artificiel.

SVM : machine à vecteur de support (support vector machine).

NN : neural network.

SIF : système d'inférence floue.

HMM : modèle de Markov caché.

MLP : perceptron multicouches (Multi layers perceptron).

FL : Logique floue (fuzzy logic).

KNN : Methode des k plus proches voisins (K Nearest Neighbors).

Introduction générale

L'avancement remarquable des progrès de la recherche médicale et l'amélioration continue des services de santé publique ou liés au secteur privé couplés aux produits de technologie d'assistance, comme les fauteuils roulant, dispositif d'écoute pour mal entendant et les technologies portables ont apporté une contribution considérable à la réalisation de l'aspiration universelle à vivre longtemps et en bonne santé.

Ainsi, les prothèses qui sont des dispositifs destinés à remplacer une partie du corps, de pallier une défaillance ou de traiter une maladie, sont des exemples d'aides techniques largement utilisées dans les traitements traumatologiques et orthopédiques.

L'ampleur des demandes en implants prothétiques est directement liée aux amputations diverses dus aux causes traumatiques, pathologiques ou congénitales. Ces amputations inéluctables lorsque la vie d'une personne est menacée par une tumeur, une infection ou par le syndrome du pied diabétique, constituent un problème de santé publique majeur. La tendance actuelle pour aider ces amputés, entre autres à être indépendants, est la réalisation d'implants très sophistiqués notamment les prothèses, et l'amélioration des techniques d'implantation.

Maurice le Blanc a estimé en 2008 à près de 10 millions le nombre de la population mondiale des amputés dont 30%, soit 3 millions, sont liées aux membres supérieurs [2]. Aux Etats-Unis, une étude sur la prévalence de la perte de membres inférieurs et supérieurs [1] a estimé, en 2005, le nombre de personnes qui ont perdu un membre à 1.6 millions et a projeté que ce nombre va être multiplié par deux pour atteindre 3.6 millions en 2050. Le rapport a conclu qu'environ un Américain sur 190 vit avec la perte d'un membre. La coalition des amputés, organisation nationale américaine de lutte contre la perte d'un membre, a affichée sur son site web qu'environ 185 000 amputations se produisent aux Etats-Unis chaque année.

Le membre supérieur est un outil indispensable à l'humain pour évoluer dans son environnement. Sa perte a de nombreuses conséquences différentes non seulement en termes physiques mais aussi sociaux, économiques et psychologiques. Afin de minimiser ces conséquences et d'aider l'amputé à s'adapter à la vie normale, des mains et des poignets artificiels (prothèses) sont utilisés pour effectuer des activités quotidiennes telles que s'habiller, écrire et saisir différents objets.

Il existe deux grandes catégories de prothèses, les prothèses actives et les passives. Les prothèses actives sont des prothèses dont les mobilités peuvent être commandées. On distingue deux sous-catégories, les actives mécaniques et les actives électriques. Les prothèses passives ne possèdent pas la fonction de préhension mais ont une belle apparence du point de vue cosmétique.

La technologie prothétique développée jusqu'à présent à un sérieux compromis entre la précision du contrôle et le coût. La majorité des prothèses disponibles sur le marché, sont en dehors de la portée du simple citoyen, nous pouvons citer la prothèse Dynamic Arm BeBionic développée par le géant allemand Otto Bock, coûte dans les environs de 70 000 \$ [3]; ou encore Luke arm développée par Mobius Bionics qui coûte 100 000 \$ [3].

L'objectif principal de cette étude est de développer un système de contrôle, basé sur les signaux EMG, capable de détecter et d'interpréter différents mouvements de la main de l'utilisateur pour faire fonctionner une main bionique imprimable. Pour ce faire, le programme doit remplir les aspects suivants:

1. acquisition et segmentation des données (signaux EMG).
2. Identifier le mouvement spécifique sous les données EMG traitées.
3. Actionnement en temps réel de la main robotique à cinq doigts.
4. Intégration des différents éléments: main robotique, circuit EMG et système de contrôle.
5. Faible coût.

Ce mémoire, s'articule sur les quatre chapitres suivant :

Chapitre 1 : Vu d'ensemble du Signal sEMG : Dans ce chapitre, nous parlerons d'une manière générale sur la théorie EMG, nous aborderons la physiologie et l'acquisition du signal.

Chapitre 2 : Traitement et classification des signaux sEMG : Ce chapitre est consacré aux différentes techniques de traitement des signaux sEMG, mais aussi aux techniques de classification de ces derniers.

Chapitre 3 : Conception : le chapitre 3 vise à illustrer concrètement les différentes étapes de conception et les démarches suivies pour l'interprétation des mouvements d'une main en partant d'un signal sEMG brute.

Chapitre 4 : Réalisation : Ce dernier chapitre représente l'intronisation de ce travail, il a pour objectif l'implémentation de notre application, mais aussi la mise en œuvre du système de contrôle myoélectrique. Nous y verrons donc les différents outils utilisés, Un aperçu du code ainsi que les configurations réaliser dans ce travail.

Chapitre 1 : Vue d'ensemble du signal sEMG.

Introduction :

Le comportement de l'être humain dépend de l'activité organisée et complexe des neurones. ^[4] Tous les mouvements volontaire ou non sont causés par ces cellules nerveuses. Dans ce qui va suivre nous aurons une vue d'ensemble sur la physiologie et quelques définitions utiles.

1.1 Physiologie

1.1.1 Les neurones :

- **Définition :** Les neurones sont des cellules nerveuses que l'on trouve dans le système nerveux central (SNC) et le système nerveux périphérique (SNP). Bien qu'ils varient considérablement par leur forme et leur taille, les neurones sont composés en général d'un corps cellulaire, de dendrites, et d'un axone. ^[5]

- **Les différents types de neurones ^[5] :**
 - Les neurones sensoriels : conduisent les influx nerveux jusqu'au SNC.
 - Sensoriels somatiques : conduisent les influx qui proviennent des récepteurs de la peau, des os, des muscles et des articulations.
 - Sensoriels viscéraux : conduisent les influx qui proviennent des viscères.

 - Les neurones moteurs : conduisent les influx en provenance du SNC.
 - Somatiques moteurs : innervent les muscles squelettiques.
 - Viscéraux moteurs (autonomes moteurs) : innervent le muscle cardiaque, les muscles lisses et les glandes.

 - Les neurones d'association (inter neurones) : conduisent les influx des neurones sensoriels aux neurones moteurs.

La figure suivante, est une représentation schématique de la structure générale d'un neurone.

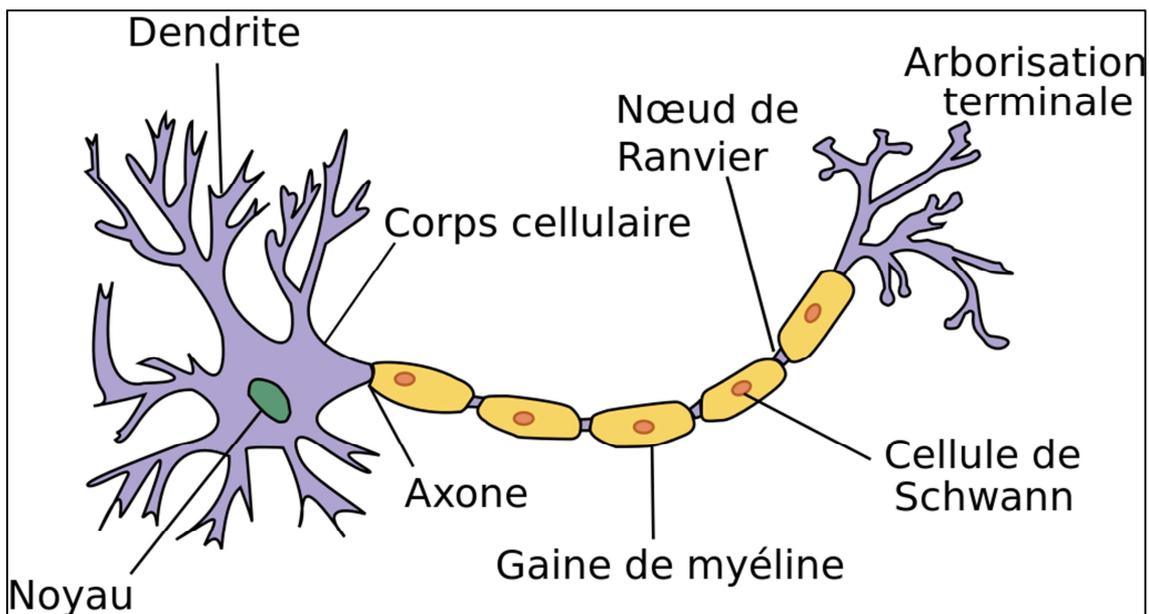


Figure 1: anatomie d'un neurone

Les muscles forment l'ensemble de tissus le plus important de l'organisme et représente environ la moitié du poids du corps. ^[4] C'est en déplaçant des composants intracellulaires spécialisés que les muscles produisent de la force et se raccourcissent, c'est-à-dire se contractent. Grâce à leur capacité de se contracter, des groupements de cellules musculaires peuvent produire mouvement et travail. ^[4]

1.1.2 Les muscles :

Il existe trois types de tissu musculaire : lisse, cardiaque et squelettique. Chaque type est caractérisé par une structure, une fonction et une localisation différente. Les fonctions du muscle sont :

- **Le mouvement.** Les mouvements du corps tels que la marche, la respiration, la parole ainsi que ceux qui sont associés à la digestion et aux flux liquidiens.
- **La production de la chaleur.**
- **La posture et le soutien du corps.**

Dans ce qui va suivre nous allons, nous concentrer sur les muscles squelettiques.

La structure microscopique du muscle squelettique.

Les cellules musculaires qui ressemblent à de petits filaments sont appelées fibres musculaires. Chaque fibre squelettique est une cellule striée multinucléée contenant un grand nombre de myofibrilles cylindriques qui s'étendent sur toute la longueur de la cellule. [5]

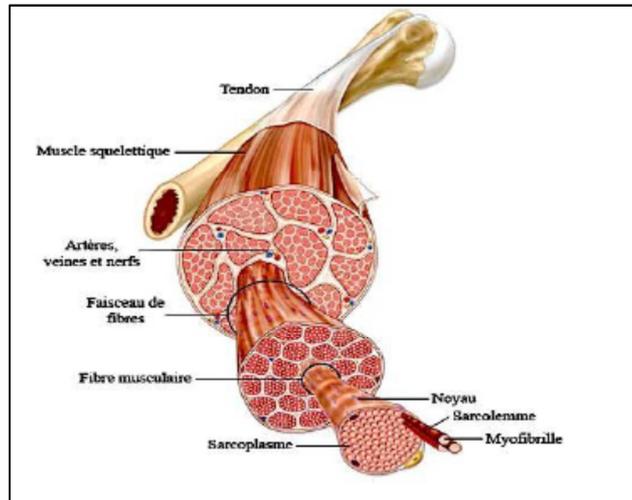


Figure 2 : anatomie d'un muscle squelettique

La jonction neuromusculaire.

La stimulation d'un neurone provoque la contraction du muscle squelettique. L'espace compris entre la terminaison axonale d'un neurone moteur et la fibre musculaire est appelé jonction neuromusculaire. [5]

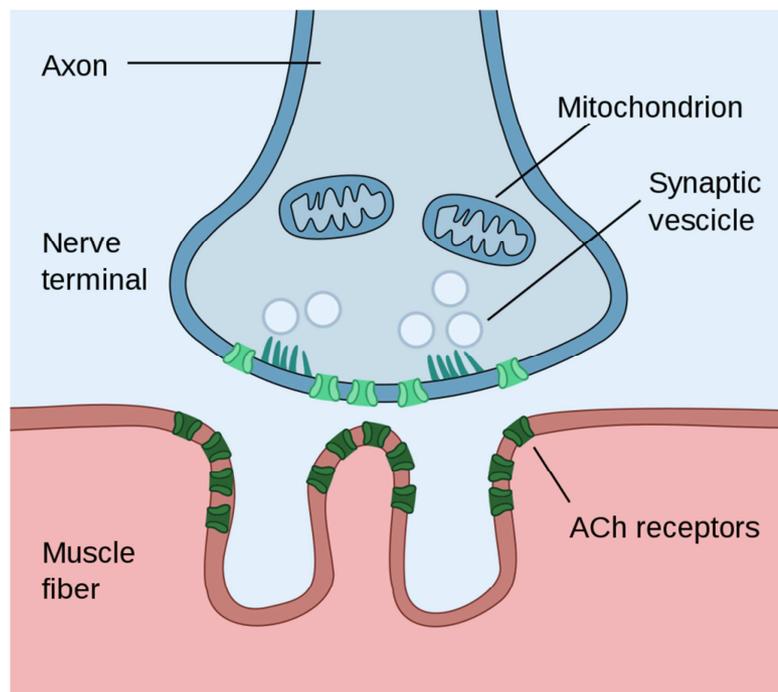


Figure 3 : Jonction neuromusculaire

L'unité motrice.

L'ensemble formé par les ramifications d'un unique neurone moteur et par les fibres des muscles squelettiques qu'elles innervent, est appelé une unité motrice. La contraction d'un muscle squelettique met en jeu plusieurs unités motrices. Des mouvements précis et hautement coordonnés nécessitent peu d'unités motrices. ^[5]

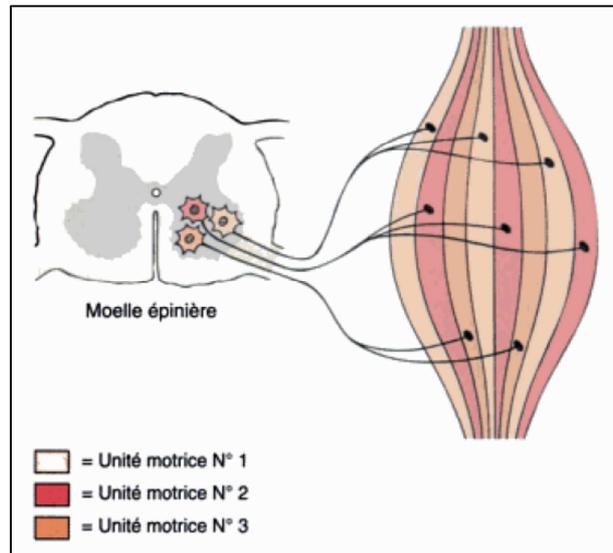


Figure 4 : représentation de différentes unités motrices

1.1.3 Les nerfs et les muscles sont des tissus excitables.

Les nerfs et les muscles sont dits excitables car ils sont capables de produire des signaux électriques quand ils sont stimulés de façon appropriée. Les cellules nerveuses qui portent le nom de neurones, utilisent ces signaux électriques pour recevoir, traiter, créer et transmettre des messages. Dans les cellules musculaires ces signaux déclenchent la contraction. ^[4]

Si un humain veut contracter un muscle, les neurones du cerveau envoient des signaux à travers une chaîne de neurones vers le neurone moteur. Le neurone moteur dépolarise la fibre musculaire, ce qui entraîne la contraction de la fibre musculaire. La contraction de toutes les fibres entraîne la contraction du muscle ^[4].

1.2 Acquisition de données EMG

EMG signifie électromyographie. C'est l'étude des signaux électriques musculaires. L'EMG est parfois appelé activité myoélectrique. Le tissu musculaire conduit des potentiels électriques similaires à ceux des nerfs et le nom donné à ces signaux électriques est le potentiel d'action musculaire^[7].

1.2.1 Les techniques d'acquisitions EMG

Il existe en générale deux manière d'acquérir les signaux EMG^[8]

- de manière invasive en insérant une électrode aiguille à travers la peau directement dans le muscle. Cette technique est un outil clinique standard utilisé principalement à des fins de diagnostic, car elle fournit une description localisée à haute résolution de l'activité électrique du muscle, bien que relativement douloureuse pour le patient

- de manière non invasive en plaçant une électrode de surface sur la peau recouvrant le muscle.

Les étapes mentionnées dans cette étude sont décrites en se concentrant sur les techniques d'acquisition EMG du muscle de l'avant-bras non invasives.

1.2.2 Muscles cibles

Les muscles de l'avant-bras sont responsables des mouvements du coude et de la main^[9]

Ce travail s'est concentré sur certains muscles superficiels de l'avant-bras car leur accessibilité permet des enregistrements non invasifs en plaçant des électrodes sur la peau recouvrant chaque muscle cible.

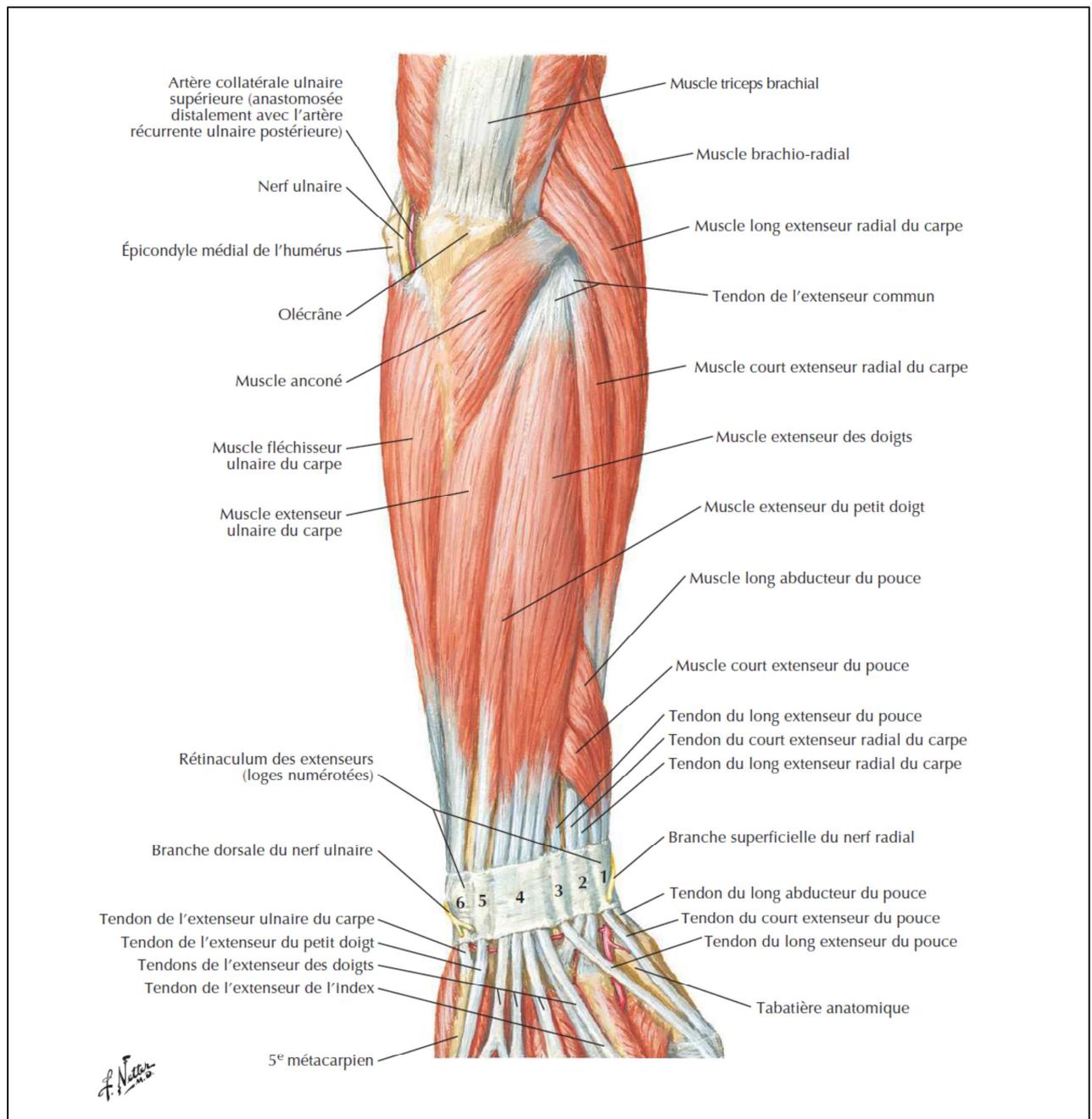


Figure 5 : Muscles de l'avant-bras (couche superficielle) : vue postérieure [6]

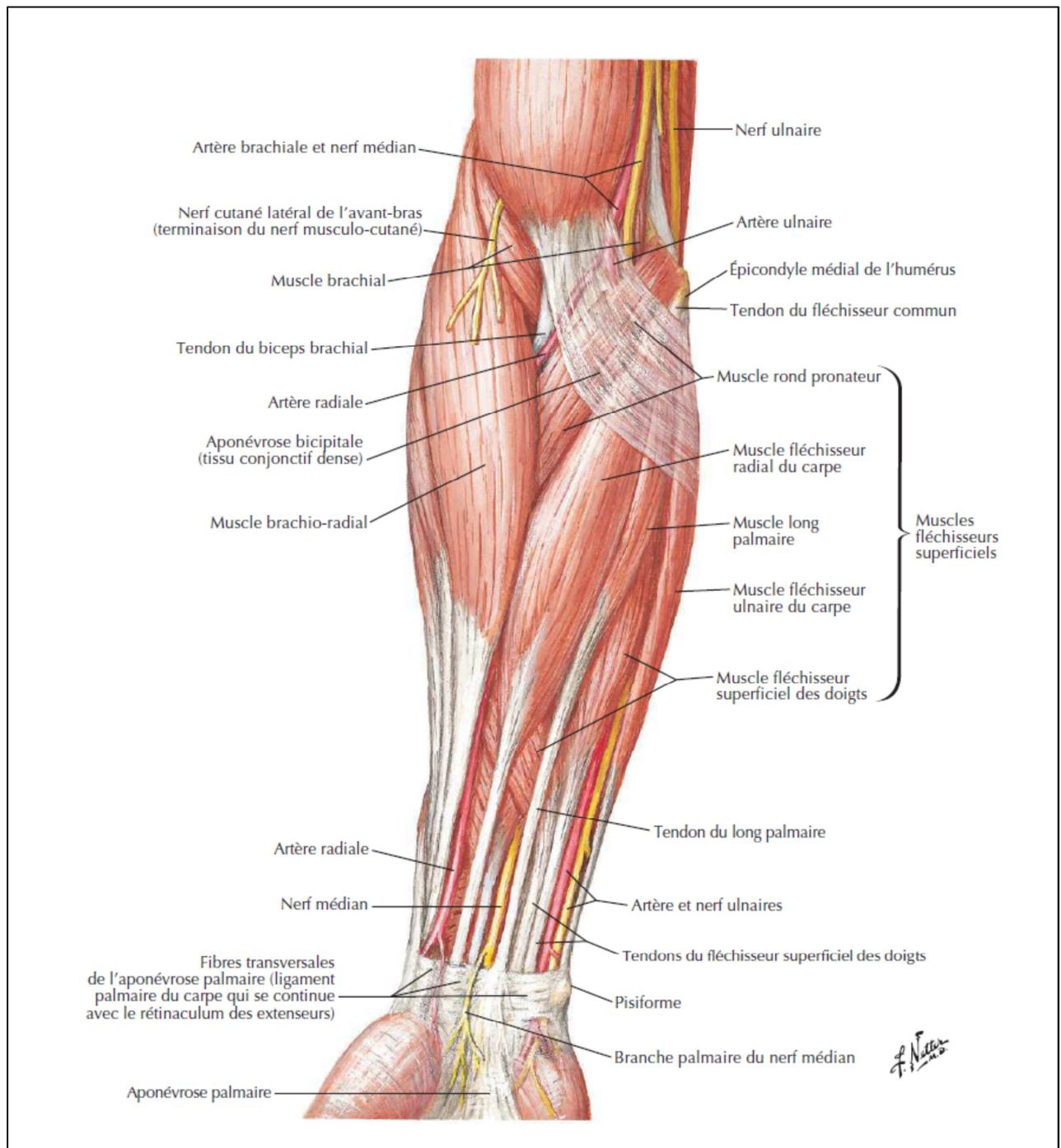


Figure 6: Muscles de l'avant-bras (couche superficielle) : vue antérieure [6]

Dans cette études nous allons nous concentrer sur :

- Les muscles fléchisseurs superficiels des doigts.
- Muscle fléchisseur ulnaire du carpe

1.2.3 Mesure du signal sEMG à l'aide des électrodes :

Les électrodes sont des transducteurs qui convertissent le courant ionique généré par la contraction des muscles en courant électrique. Selon le caractère invasif, il existe deux types d'électrodes: de surface ou intramusculaires (électrodes à fil fin ou aiguille) ^[10].

- **Les électrodes à aiguille** : Elles touchent directement la fibre musculaire. Elles permettent d'obtenir le signal émis par une seule fibre mais pressentent l'inconvénient d'abimer la peau et le tissu musculaire.
- **Les électrodes de surface** Elles sont fixées sur la peau au niveau du muscle cible. Elles fournissent le signal émis par un ensemble de fibres musculaires et non plus d'une seule comme précédemment. Parmi les électrodes de surface, on distingue :
 - les électrodes humides à base de gel Ag/AgCl, à usage unique et a durée de vie limitée.
 - les électrodes sèches, réutilisables et a plus grande durée de vie. Leur impédance électrique est haute et instable et la transpiration déforme légèrement le signal. ^[11]

1.2.3.1 Préparation de la peau

Les électrodes de surface nécessitent une préparation spéciale de la peau pour créer un contact stable avec une faible impédance cutanée. Pour les peaux sèches, l'impédance varie de quelques kilos à des mégohms en raison des artefacts de mouvement, de la sueur ou du pourcentage de graisse corporelle. Ces artefacts déforment le signal, les dispositions suivantes doivent donc être prises en compte avant de placer les électrodes ^[10]:

1. Épilation: cela Améliore l'adhérence des électrodes.

2. Nettoyage de la peau: les cellules mortes de la peau, la saleté et la sueur qui pourraient être présentes sur le site d'enregistrement augmentent l'impédance. Pour les éliminer, un abrasif spécial, du papier de verre ou de l'alcool peut être appliqué sur la surface. Les dommages doivent être évités lorsque cela est fait correctement, la peau reçoit généralement une couleur rouge clair.

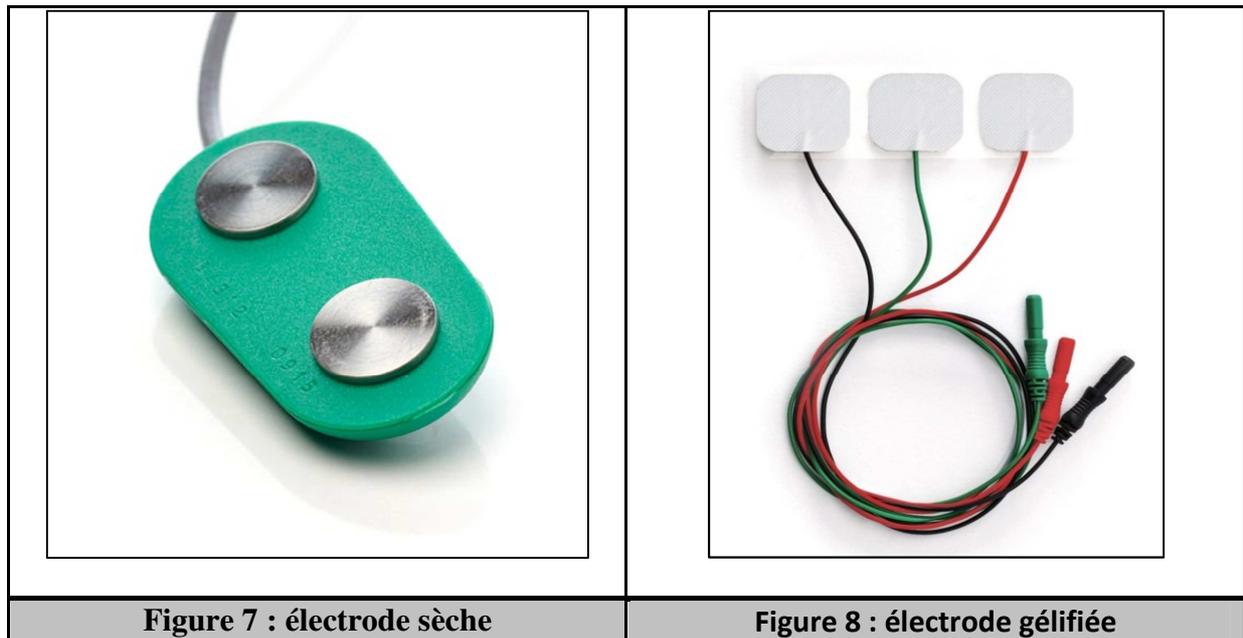
1.2.3.2 Types d'électrodes de surface

Deux types d'électrodes de surface sont couramment utilisés ^[13]

- Électrodes gélifiées utilisant un gel électrolytique comme interface chimique entre la peau et la partie métallique de l'électrode ce qui améliore leurs performances et augmente la conductivité. Le matériau le plus courant pour la partie métallique de l'électrode est l'argent et le chlorure d'argent (Ag-AgCl). Leurs principaux avantages sont une faible impédance, une bonne adhérence et un faible coût. Ce sont généralement des électrodes jetables.

- Électrodes sèches en contact direct avec la peau. Ce type d'électrodes ne nécessite pas d'interface gel à la jonction peau-électrode. Le matériau préféré est l'or (Au), malgré son coût élevé, en raison de leur très faible impédance. Ils sont très résistants et peuvent être réutilisés.

Dans cette étude, les électrodes qui ont été utilisées sont des électrodes de surface EMG Ag-AgCl.



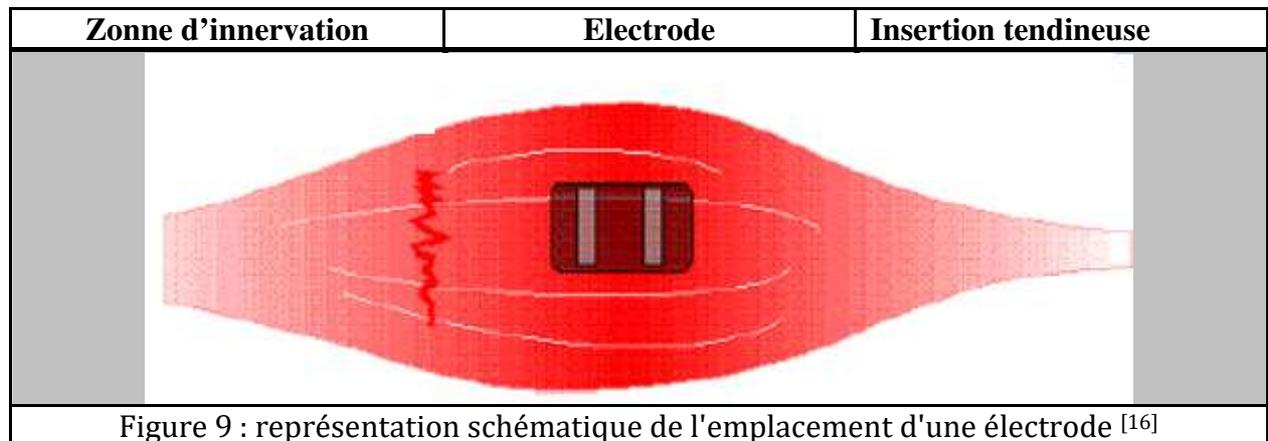
1.2.3.3 Placement des électrodes ^[16]

Les électrodes doivent être placées dans le Centre du muscle cible, le long de la ligne médiane longitudinale du muscle. Cet agencement maximise le nombre de fibres musculaires pouvant être enregistrées avec les électrodes.

Il est important d'éviter les insertions tendineuses car les fibres musculaires deviennent plus minces et plus proches, ce qui réduit l'amplitude du signal et les rend plus sensibles à la diaphonie ou aux interférences de l'activité musculaire non ciblée. Les points moteurs ou les zones d'innervation doivent également être évités. Ces zones ont la densité neuronale la plus élevée, donc une activité électrique minimale provoque une contraction perceptible sur la surface musculaire, altérant la stabilité.

Une électrode de référence supplémentaire doit être située sur un tissu électriquement neutre, comme une proéminence osseuse, à proximité de la zone cible. Dans ce cas, deux électrodes

Ag-AgCl ont été placées sur le dessus du centre des muscles cibles mentionnés pour mesurer le signal différentiel, et une électrode de référence, dans le coude.



La figure précédente montre la position des électrodes afin de mesurer le signal sEMG de manière optimale.

Dans notre travail les électrodes seront placées comme le montre la figure suivante :



Figure 10: positionnement des électrodes.

1.2.4 Caractéristiques et prétraitement du signal EMG

1.2.4.1 Définitions :

- **Un signal** : Un signal est la représentation physique de l'information qu'il transporte de sa source à son destinataire. Il sert de vecteur à une information. Il constitue la manifestation physique d'une grandeur mesurable (courant, tension, force, température, pression, etc.).
- **Le bruit** : le bruit est défini comme tout phénomène perturbateur gênant la perception ou l'interprétation d'un signal, par analogie avec les nuisances acoustiques (interférence, bruit de fond, etc.).
- **L'amplification** : L'amplification consiste en augmentation de la tension et/ou l'intensité d'un signal électrique. Dans le signal obtenu directement avec les électrodes donne des signaux d'amplitude extrêmement faible environ 10 μV à 3 mV [rap final].
- **Le filtrage** : Un filtre est un circuit électronique qui réalise une opération de traitement du signal. Il atténue certaines composantes d'un signal et en laisse passer d'autres. Il existe plusieurs types de filtres, dont les plus connus sont :

Filtre passe-bas : Il ne laisse passer que les fréquences au-dessous de sa fréquence de coupure. C'est un atténuateur d'aiguës pour un signal audio.

Filtre passe-haut : Il ne laisse passer que les fréquences au-dessus d'une fréquence déterminée, appelée "fréquence de coupure". Il atténue les autres (les basses fréquences).

Filtre passe-bande : Il ne laisse passer qu'une certaine bande de fréquences (et atténue tout ce qui est au-dessus ou en-dessous). Il est très utilisé dans les récepteurs radio, tv... pour isoler le signal que l'on désire capter.

- **L'échantillonnage** : L'échantillonnage consiste à prélever à des instants précis, le plus souvent équidistants, les valeurs instantanées d'un signal. Le signal analogique $s(t)$, continu dans le temps, est alors représenté par un ensemble de valeurs discrètes.
- **Le théorème de SHANNON** : montre que la reconstitution correcte d'un signal nécessite que la fréquence d'échantillonnage f_e soit au moins deux fois plus grande que la plus grande des fréquences f_M du spectre du signal : $f_e > 2f_M$

1.2.4.2 Caractéristiques du signal EMG :

- L'amplitude du signal EMG varie d'une valeur allant de $10 \mu V$ à $3 mV$. Une aussi grande plage de valeur s'explique par le fait que plusieurs paramètres sont à prendre en considération soit : le positionnement de l'électrode, la préparation préalable de la peau et aussi la grosseur du muscle étudié. ^[15]
- les fréquences importantes à conserver dans le signal du muscle vont de 20 à $500 Hz$. ^[15]

1.2.4.3 Prétraitements des signaux EMG ^[16]:

La fidélité du signal EMG détecté par l'électrode influence tout traitement ultérieur du signal. Par conséquent, il est important d'utiliser un circuit d'acquisition qui fournit une distorsion minimale et un rapport signal sur bruit le plus élevé. Les caractéristiques suivantes sont importantes pour répondre à cette exigence.

- a. Amplification différentielle. L'idée de l'amplification différentielle est d'enregistrer les signaux EMG sur deux sites proches, de soustraire les deux signaux et d'amplifier la différence. Le bruit de la ligne électrique ou la diaphonie ne peuvent pas être filtrés car ils contribuent à la plage d'énergie dominante du signal EMG. Cependant, l'amplification différentielle annule ce bruit commun tout en amplifiant les changements dans les muscles superficiels. La figure ci-dessous illustre l'amplification différentielle.

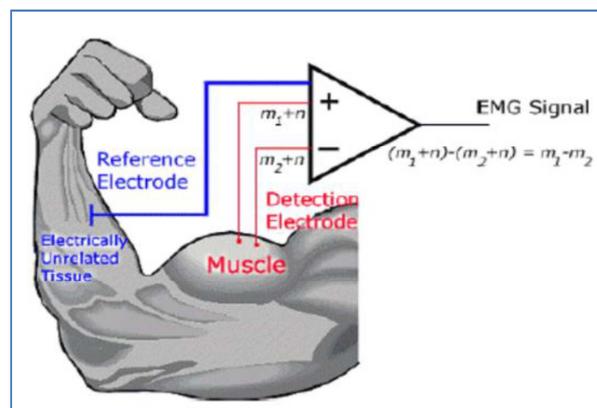


Figure 11 : amplification différentielle. ^[16]

- b. Impédance d'entrée élevée de l'amplificateur différentiel. L'impédance d'entrée du préamplificateur doit être au moins 10 fois supérieure à l'impédance électrode-peau pour éviter l'atténuation et la distorsion du signal. Les impédances des deux entrées doivent être équilibrées.

- c. filtrage passe-bande. Le rapport signal / bruit (SNR) peut être augmenté en filtrant la plage 20-500 Hz.
- d. Problèmes de sécurité électrique. Comme le sujet est directement connecté au circuit via les électrodes, la défaillance de tout composant électrique peut produire un courant nocif traversant la peau et pénétrant dans les tissus. Pour éviter cela et garantir la sécurité de l'utilisateur, le sujet doit être isolé des connexions associées à la source d'alimentation via des isolateurs optiques ou des transformateurs. Cependant, le risque de courant dommageable est réduit pour les circuits alimentés à basse tension (3-15 V).

Cette étape de prétraitement vise à conserver un maximum d'informations, à réduire le bruit et à éviter toute distorsion du signal. Après ce processus, le signal EMG peut être traité pour extraire les caractéristiques pertinentes pour la reconnaissance de formes.

Chapitre 2 : Traitement et classification des signaux sEMG.

Introduction

Les progrès de la robotique et des prothèses ont permis de développer des membres artificiels plus complexes avec plusieurs degrés de liberté. Dans le passé, l'estimation de base de l'amplitude et du taux de changement du signal EMG était suffisante pour contrôler une fonction. Cependant, les nouveaux appareils nécessitent des systèmes de contrôle plus complexes qui visent à fournir ^[14] :

- La précision : Il est essentiel de reproduire fidèlement les intentions de l'utilisateur. Pour l'améliorer, plus d'informations doivent être extraites du signal EMG en augmentant le nombre de canaux de l'enregistrement et en créant un ensemble de fonctionnalités qui tire le meilleur parti des données obtenues.
- L'interface intuitive : Le problème principal du faible taux d'acceptation des systèmes de contrôle myoélectrique est la différence entre les commandes de mouvement instinctif du sujet et les connaissances requises pour effectuer réellement cette action. La solution consiste soit à former l'utilisateur à acquérir les compétences de contrôle nécessaires, soit à développer des interfaces utilisateur intelligentes capables d'apprendre les modèles d'activation musculaire lors de conditions de fonctionnement changeantes.
- La réponse courte durée : La réponse temporelle du système de contrôle est limitée aux contraintes en temps réel, de sorte que l'utilisateur ne perçoit pas de retard lors de la manipulation. Pour ce faire, le traitement des signaux d'entrée en segments courts pour une analyse plus approfondie est l'approche la plus efficace.

La figure 10 illustre le schéma général du fonctionnement des systèmes de contrôle de prothèses de main bioniques, qui est divisé en: segmentation des données, extraction de caractéristiques, classification et contrôleur. Chaque étape sera discutée dans les sous-sections suivantes.



Figure 12: schéma général du fonctionnement des systèmes de contrôle de prothèses bioniques

2.1 Segmentation des données:

La segmentation des données est le processus de division des signaux d'entrée en tranches de temps pour un traitement ultérieur. Il existe plusieurs approches de segmentation basées sur la longueur de la fenêtre, les caractéristiques du signal dans la fenêtre ou l'état du signal.

Pour le contrôle des membres prothétiques, le temps de réponse doit être inférieur à 300 ms pour atteindre les contraintes temps réel ^[17].

Il existe deux principales techniques de fenêtrage: la segmentation adjacente ou disjointe et la segmentation superposée. Les segments du fenêtrage disjoints ont longueur prédéfinie, tandis que dans le fenêtrage superposé, les segments glissent les uns sur les autres. ^[17]

Les figures suivantes illustrent les deux principales techniques de fenêtrage :

A)

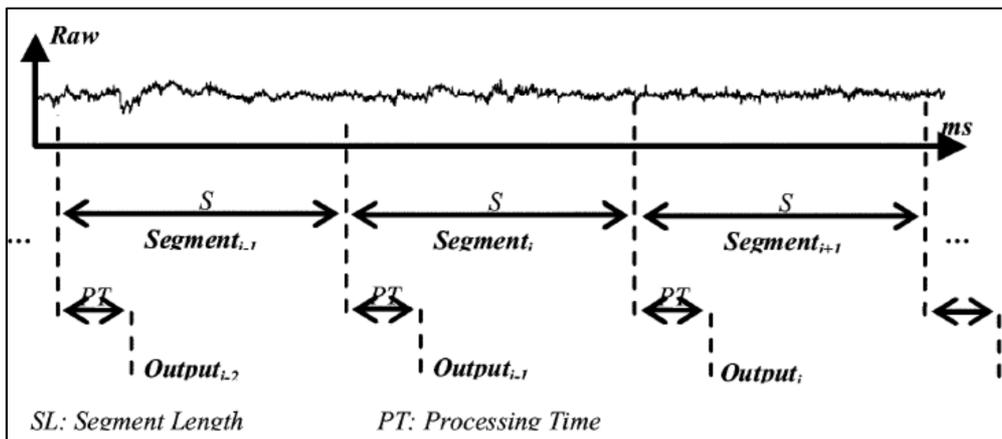


Figure 13 : segmentation disjointe ^[17]

B)

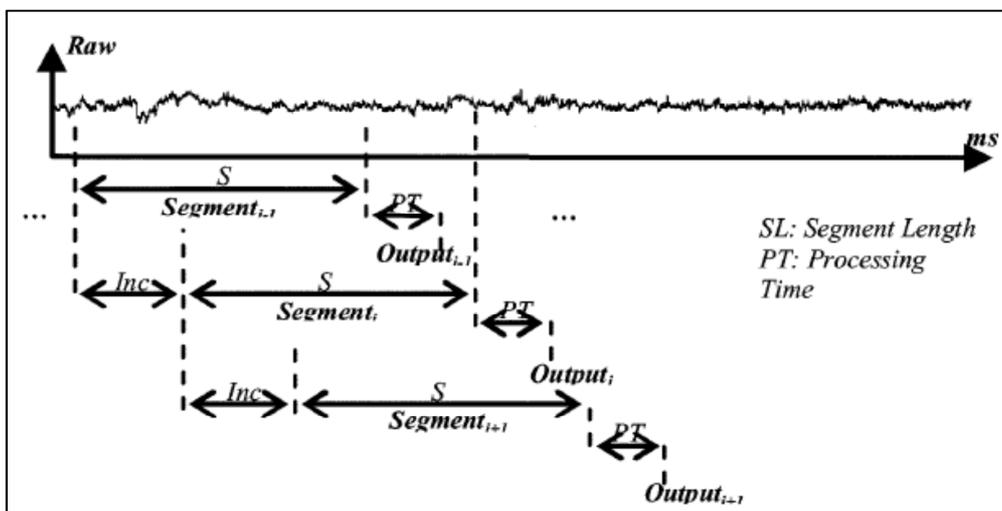


Figure 14 : Segmentation superposée. ^[17]

2.2 Extraction des fonctionnalités :

2.2.1 Domaine temporel :

Les fonctionnalités du domaine temporel sont les plus populaires dans la reconnaissance de formes EMG, car elles sont faciles et rapides à calculer, car elles ne nécessitent pas de transformation. Les caractéristiques du domaine temporel sont calculées sur la base de l'amplitude du signal et les valeurs résultantes donnent une mesure de l'amplitude, de la fréquence et de la durée de la forme d'onde dans certains paramètres limités. [18]

1. EMG intégré (IEMG) :

Ce paramètre est trouvé en calculant la somme des valeurs absolues des signaux EMG. Il peut être traité comme un estimateur de puissance de signal. Il est défini comme ^[19]

$$IEMG_k = \sum_{i=1}^N |x_i|$$

Où x_i est la valeur de chaque partie du segment k , et N est la longueur du segment.

2. Valeur absolue moyenne (MAV) :

Estime la valeur absolue moyenne d'un signal, en ajoutant la valeur absolue de toutes les valeurs x_i dans un segment k et en la divisant par la longueur du segment N ^[20].

$$MAV_k = \frac{1}{N} \sum_{i=1}^N |x_i|$$

3. Moyenne de la valeur absolue modifiée 1 (MMAV1) : c'est une extension de MAV ^[21].

$$MMAV1_k = \frac{1}{N} \sum_{i=1}^N \omega_i |x_i|$$

$$\omega(i) = \begin{cases} 1, & 0.25N \leq i \leq 0.75N \\ 0.5, & \text{sinon} \end{cases}$$

4. Il s'agit de la moyenne quadratique de l'EMG sur un intervalle de temps. En d'autres termes, le RMS permet de quantifier les amplitudes du signal EMG sur un intervalle de temps donné, calculé sur des intervalles courts. ^[21]

$$RMS_k = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$

5. Variance (VAR) [20]:

$$VAR_k = \frac{1}{N} \sum_{i=1}^N (x_i - x)^2$$

Où x est la valeur moyenne du segment k .

6. Intégrale carrée simple (SSI). Utilise l'énergie du signal EMG comme fonction [21] :

$$SSI_k = \sum_{i=1}^N (|x_i^2|)$$

7. Histogramme de l'EMG (HEMG). Divise les éléments du signal EMG en b segments équidistants et renvoie le nombre d'éléments dans chaque segment [21].

2.2.2 Domaine fréquentiel :

Les caractéristiques du domaine fréquentiel sont basées sur la densité spectrale de puissance (PSD) estimée du signal et sont calculées par périodogramme ou des méthodes paramétriques, mais ces caractéristiques par rapport aux caractéristiques du domaine temporel nécessitent plus de calculs et de temps pour être calculées. [18]

1. Coefficients auto-régressifs (AR). Décrit chaque échantillon de signal EMG comme une combinaison linéaire d'échantillons précédents plus un terme d'erreur de bruit blanc. Les coefficients AR sont utilisés comme caractéristiques dans la reconnaissance de formes EMG [21].

$$x_k = - \sum_{i=1}^N a_i x_{k-i} + e_k$$

Où a_i sont les coefficients AR, e_k est le bruit blanc ou la séquence d'erreur, et N est l'ordre du modèle AR.

2. Médiane de fréquence (FMD). La médiane de fréquence divise la densité spectrale de puissance en deux parties égales. Elle est donnée par [18]:

$$x_k = \frac{1}{2} \sum_{i=1}^M PSD_i$$

où M est la longueur de la densité spectrale de puissance, et PSD_i est la $i^{\text{ème}}$ ligne de la densité spectrale de puissance.

3. Moyenne de fréquence (FMN). La moyenne de fréquence est donnée par ^[18]:

$$F_{MN} = \frac{\sum_{i=1}^M f_i PSD_i}{\sum_{i=1}^M PSD_i}$$

où M est la longueur de la densité spectrale de puissance, $f_i = (i * \text{taux d'échantillonnage}) / (2 * M)$ et PSD_i est la $i^{\text{ème}}$ ligne de la densité spectrale de puissance.

4. Médiane de fréquence modifiée (MFMD). C'est la fréquence auquel le spectre est divisé en deux régions avec une amplitude égale. Il est donné par ^[21]:

$$MFMD = \frac{1}{2} \sum_{j=1}^M A_j$$

Où A_j est le spectre d'amplitude EMG à la case de fréquence j .

2.2.3 Domaine temps-fréquence :

Ces caractéristiques décrivent comment la puissance du spectre varie dans le temps. La transformée de Fourier à court temps et la transformée en ondelettes sont les plus utilisées. ^[21]

2.3 Classification :

Une fois que les données acquises ont été correctement traitées pour améliorer les informations pertinentes tout en minimisant les informations redondantes, l'étape suivante de la reconnaissance de formes est la classification. Le classificateur idéal doit répondre aux exigences suivantes ^[22]:

- Être capable de gérer les différents modèles de manière optimale (classification efficace des nouveaux modèles).
- Empêcher le sur-ajustement (modélisation du bruit ou des erreurs aléatoires)
- Classification rapide pour répondre aux contraintes en temps réel

Il existe plusieurs techniques pour catégoriser le vecteur de caractéristiques réduit: le classificateur bayésien (BC), les réseaux de neurones (NN), les systèmes d'inférence floue (FIS), les machines à vecteurs de support (SVM) et les modèles de Markov cachés (HMM). Les sous-sections suivantes décrivent en détail Quelques procédure de classification.

2.3.1 Réseaux de neurones Artificiels (RNA)

Les réseaux de neurones artificiels s'inspirent du fonctionnement d'un neurone biologique (notamment sa capacité à collecter, traiter et diffuser des potentiels d'action). Les RNA sont composés de neurones artificiels, unités de traitement de l'information et d'un ensemble d'entrées et de poids.

Dans la première étape du traitement des neurones artificiels, le réseau est calculé par la somme arithmétique des entrées et des poids, comme le montre la figure 15, où n est le nombre total d'unités d'entrée, x_i est l'entrée d'une unité et w_i , son poids associé. L'étape suivante évalue le réseau obtenu avec une fonction d'activation (f_{act}). Cette fonction agit comme une porte logique qui est activée (valeur de sortie proche de +1) ou désactivée (valeur de sortie proche de 0) en fonction de la valeur nette d'entrée. Cette activation doit être non linéaire, les fonctions d'activation proposées: seuil linéaire, pas, pas arbitraire, exponentiel ou sigmoïde [23].

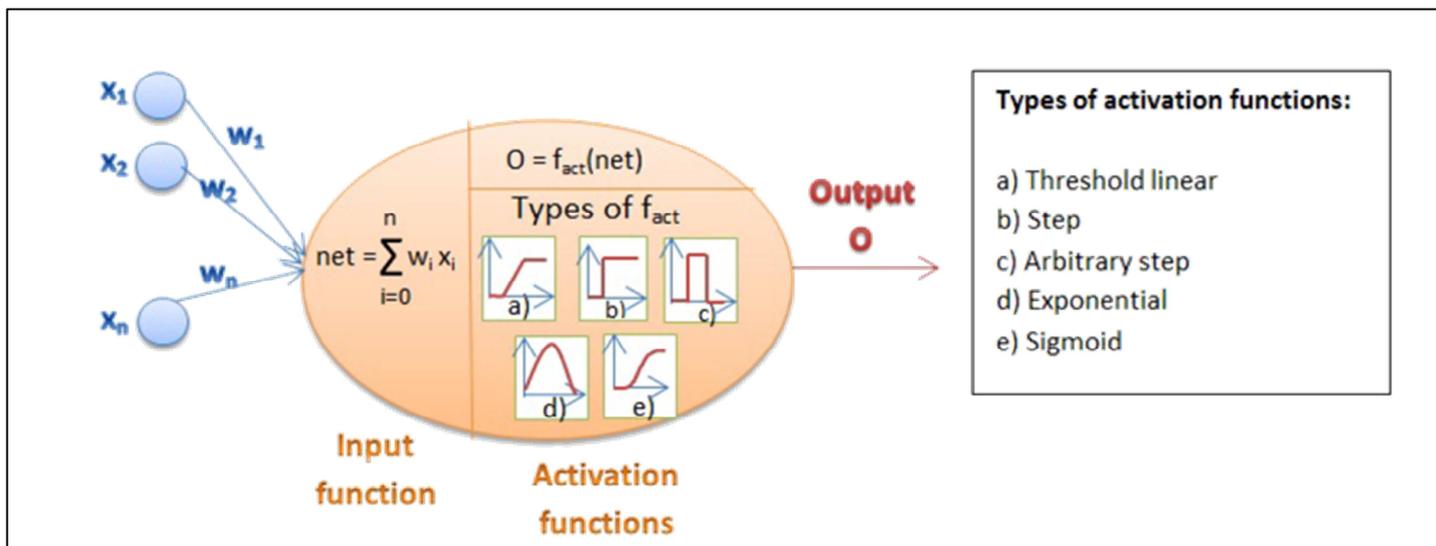


Figure 15 : Structure d'un neurone artificiel

Le RNA le plus courant est le Perceptron multicouche (MLP) formé par plusieurs unités d'entrée, plusieurs couches cachées (généralement une ou deux) et un niveau d'unités de sortie, comme représenté sur la figure 16. Chaque unité d'entrée est connectée à chaque unité de la couche cachée avec un poids associé. De même, cette couche cachée est connectée aux neurones artificiels de la couche cachée ou de sortie suivante. Cette disposition correspond à un réseau entièrement connecté, mais il existe également des réseaux partiellement connectés.

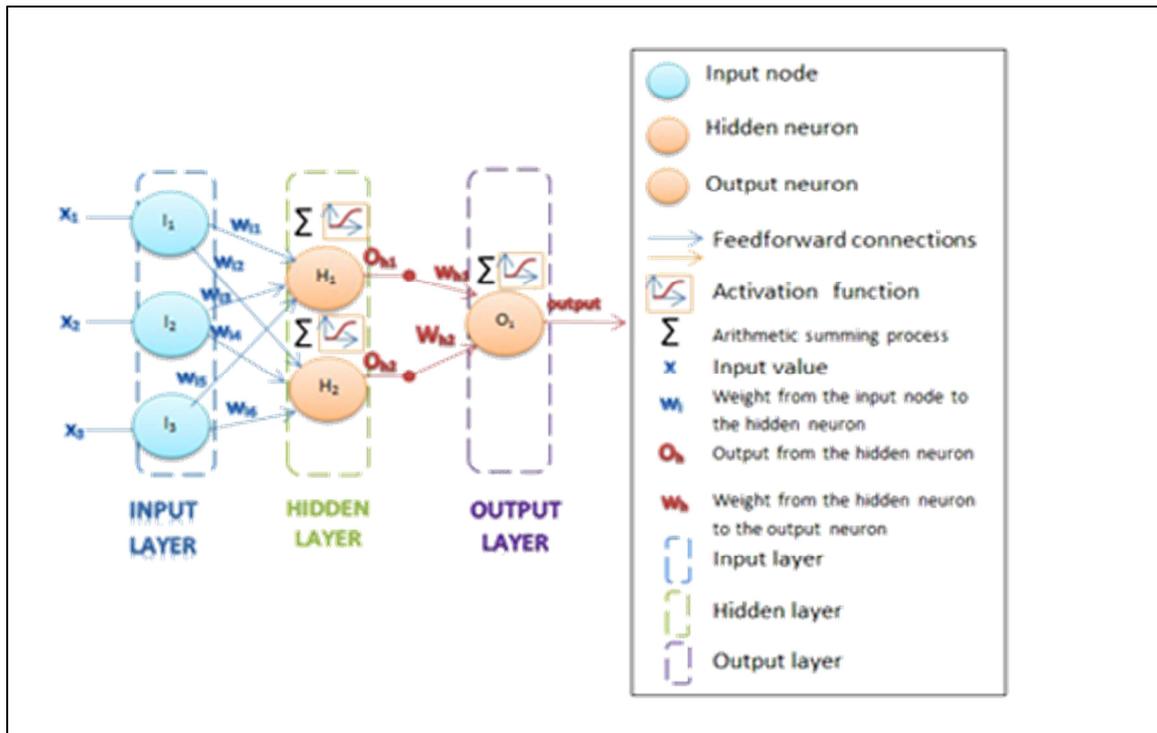


Figure 16 : Structure d'un MLP

Les avantages d'un RNA reposent sur des compétences d'apprentissage en intelligence artificielle, en étant capable de ^{[24], [23]}:

- Apprendre à la fois les relations linéaires et non linéaires directement à partir des données d'entrée sans modèle mathématique antérieur.
- S'adapter aux conditions changeantes.
- Traiter les ensembles de données corrompus ou partiels.

Ce flux de travail implique une étape de formation pour ajuster les poids du réseau. Ceci introduit le concept de réseaux de neurones à rétro propagation (BPNN) dans lesquels chaque unité ou neurone reçoit un feedback de la couche suivante pour adapter leurs poids et améliorer les performances de classification; voir la figure 17. Les unités peuvent même avoir des auto-connexions. Cette récurrence fournit des informations sur les événements passés fonctionnant comme une mémoire à court terme.

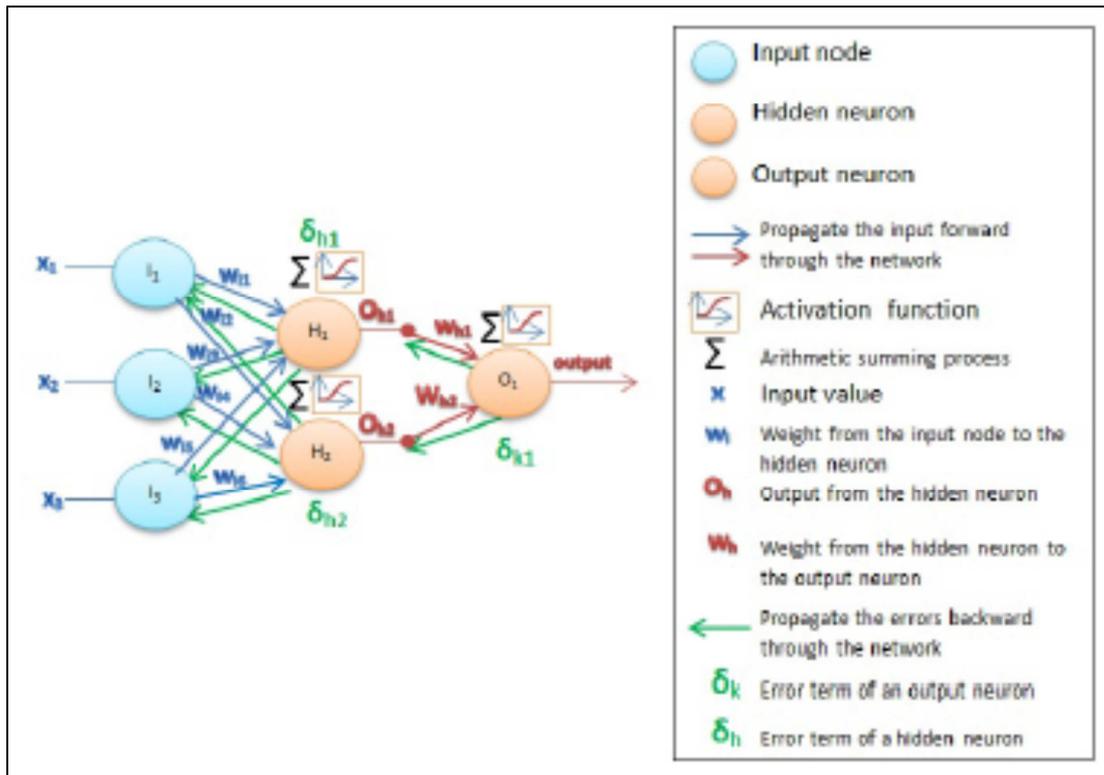


Figure 17 : Représentation fonctionnelle d'un MLP

2.3.2 Classificateur à logique floue (FL). ^[25]

La logique floue présente plusieurs avantages dans le traitement du bio-signal car elle est capable de traiter des données incertaines, imprécises et même contradictoires; détecter des modèles cachés complexes et permettre l'intégration d'expérience et d'informations empiriques dans le système pour améliorer les performances de classification. Les systèmes d'inférence floue (SIF) sont d'excellents simulateurs du processus décisionnel humain.

La logique floue fonctionne avec des ensembles flous ou des séries de variables linguistiques qui modélisent l'incertitude des problèmes (c'est-à-dire qualitativement: faible, moyenne, élevée; quantitativement: peu, plusieurs, beaucoup, etc.). Par conséquent, les entrées numériques doivent être transformées en ces valeurs floues par une fonction d'appartenance, dont les valeurs vont de l'intervalle continu [0, 1] suivant une forme d'onde trapézoïdale, triangulaire ou gaussienne. Ce processus est appelé fuzzification. Une fois dans le domaine flou, la classification est effectuée selon un ensemble prédéfini de règles floues IF ALORS. La sortie de cette étape est une autre variable floue qui doit à nouveau être convertie en une quantité nette (non floue) par défuzzification. Ainsi, le SIF comprend une interface de fuzzification, une base de données de fonctions d'appartenance, une base de règles floues, une unité de prise de décision et une interface de défuzzification. Voir la figure 18.

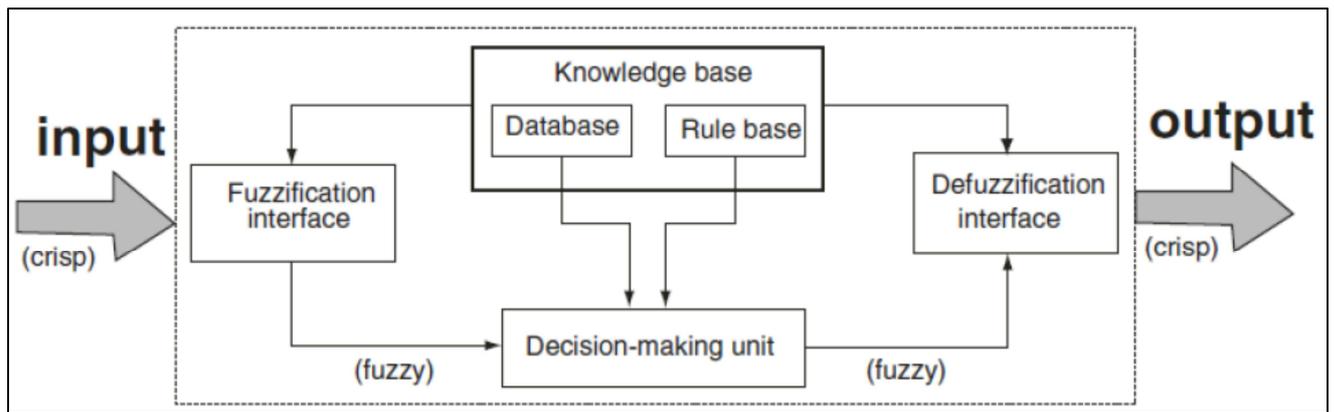


Figure 18 : Schéma fonctionnelle d'un classificateur a logique floue ^[25]

2.3.3 Classificateur SVM (Support Vector Machines) ^[17]

Support Vector Machine (SVM) est une approche de noyau qui construit un hyperplan de séparation optimal qui maximise la marge possible entre les points de k classes différentes. La marge est la distance perpendiculaire de l'hyperplan à un échantillon de classe. Par conséquent, cet hyperplan peut être défini comme une combinaison linéaire des échantillons informatifs des données d'apprentissage ou des vecteurs de support qui constituent la frontière de la classe, comme illustré à la figure 19. Les données d'apprentissage sont cartographiées ou classées par une fonction de noyau non linéaire comme une fonction de base linéaire, polynomiale, sigmoïde ou radiale.

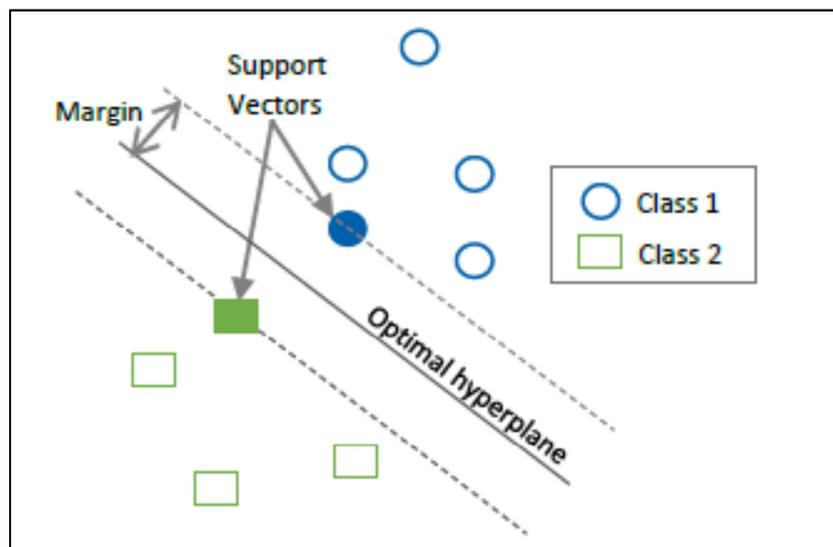


Figure 19 : structure d'un classificateur SVM

À l'origine, SVM est un classificateur binaire qui divise les données en deux classes. Cependant, il peut être implémenté pour les problèmes multi classes via: un contre tous (OAA) ou un contre un (OAO). L'approche OAA utilise k classificateurs binaires pour évaluer l'ensemble d'apprentissage d'une classe par rapport aux autres. Cette méthode est simple, relativement rapide et précise. D'autre part, l'OAO utilise $k(k-1)$ SVM pour tester toutes les paires possibles de combinaisons de classes. À la fin, le résultat final est la classe qui obtient les contributions les plus élevées de l'ensemble des tests. Cette procédure a l'avantage de récupérer une probabilité pour chaque classe, ce qui permet une meilleure généralisation et une meilleure certitude.

2.3.4 Classificateur K-plus proche voisin (KNN) ^[25]

La base de cette approche est que chaque échantillon est classé en fonction de la classe de leurs k voisins les plus proches (KNN). Cette technique est un processus à deux volets qui identifie d'abord les voisins les plus proches, puis classe l'entrée par vote majoritaire ou par vote pondéré par la distance. La dernière méthode donne plus de pertinence aux voisins les plus proches en utilisant l'exponentielle inverse ou négative de la distance euclidienne comme poids. Le principal avantage de KNN est donc sa facilité de mise en œuvre.

Cependant, KNN a besoin de données d'entraînement pendant l'exécution pour comparer les échantillons de test, et de grands ensembles de données d'entraînement peuvent diminuer les performances en temps réel. En effet, il est assez vulnérable à des caractéristiques non pertinentes ou redondantes, mais la réduction de la dimensionnalité et la pondération des voisins ont résolu efficacement ce problème.

Pour conclure, la figure 20 résume le contenu de ce chapitre, montrant les différentes étapes impliquées dans la reconnaissance de formes d'un système de contrôle myoélectrique, et les techniques associées pour chaque étape. La présente thèse a été développée en tenant compte de ces bases théoriques pour classer les mouvements de l'utilisateur afin d'actionner une main robotique. Les méthodes employées sont en gras sur la figure 20.

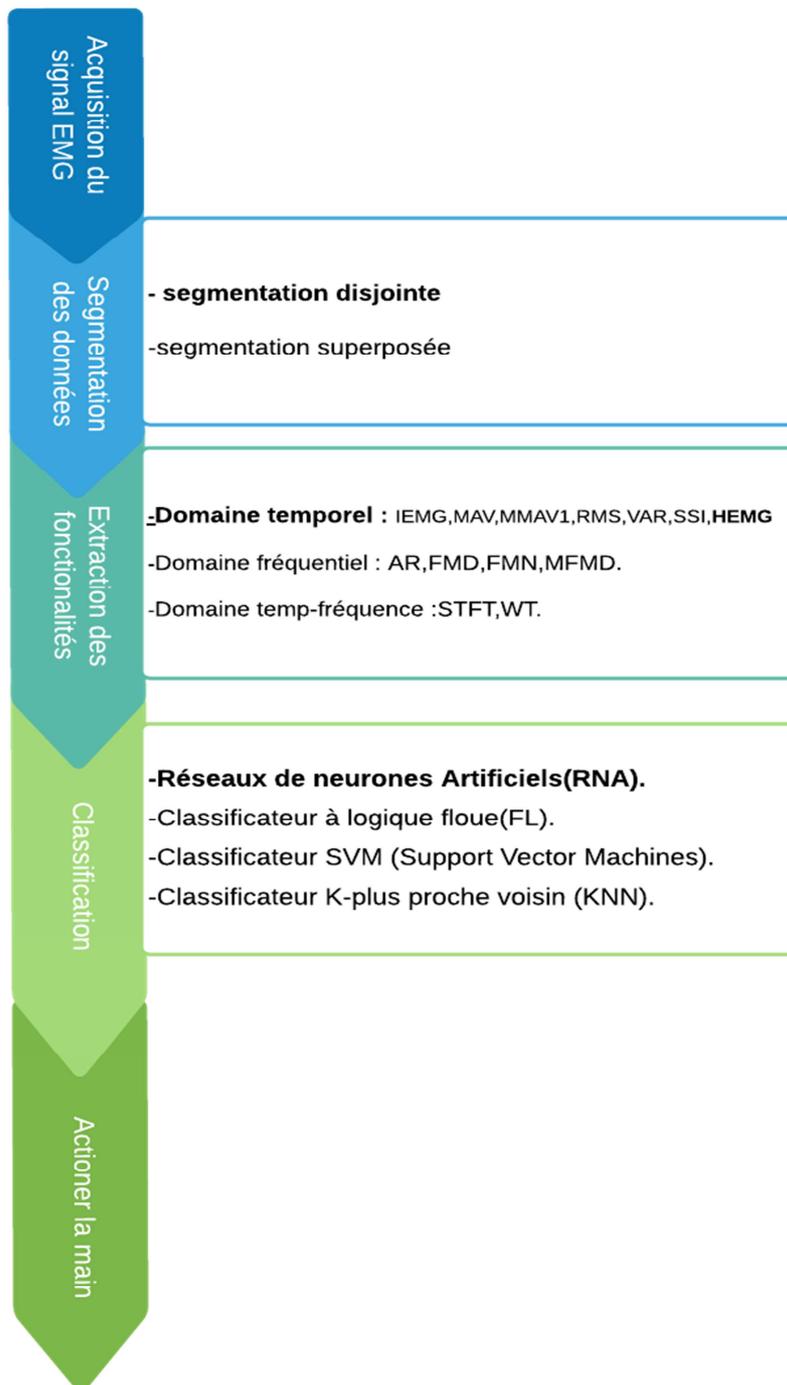


Figure 20 : Schéma résumant le chapitre.

Chapitre 3 : Conception.

Introduction :

L'acquisition du signal sEMG est une étape principale dans le système de contrôle de prothèse myoélectrique, vu que l'intégrité de ce signal détermine le degré de précision de notre prothèse.

Tout fois c'est une tâche peu facile, étant donnée les variations de l'emplacement des électrodes, de l'état de la peau et des interférences matérielles.

Notre projet consiste en la réalisation d'un système capable de collecter les signaux sEMG brutes, de les structurer sous forme de données, de classifier ces données et enfin d'actionner la main robotisée. La figure suivante nous donne une idée générale du système recherché.

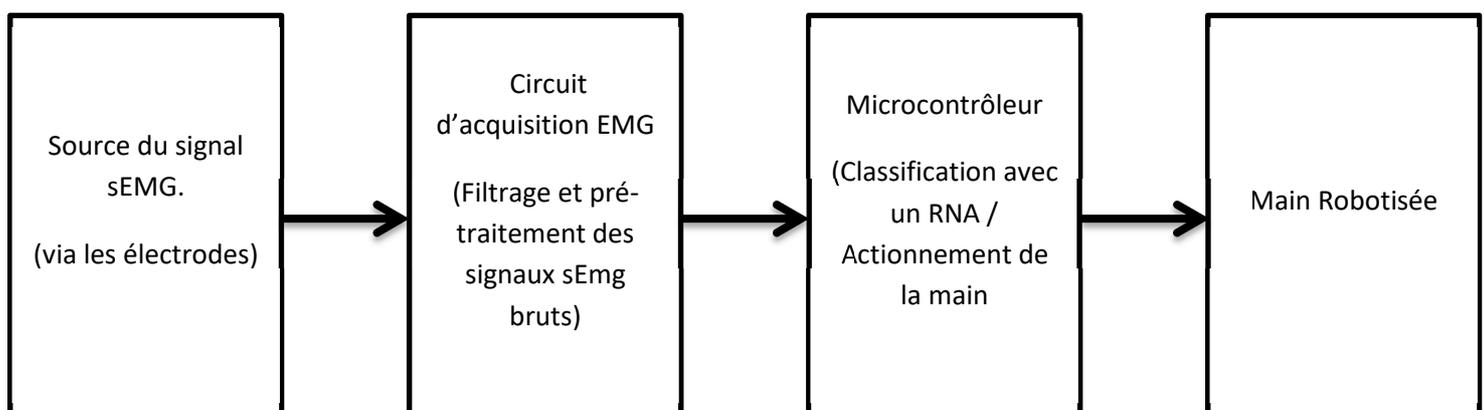


Figure 21 : vue schématique du système.

Dans ce chapitre nous verrons d'une façon plus détaillée les évolutions du signal sEMG de sa source jusqu'à l'actionnement de la main, l'architecture proposée pour ce système, la démarche et les fonctionnalités utilisées pour la mise en œuvre de notre projet. Nous expliquerons promptement toutes les étapes qui composent le schéma ci-dessous.

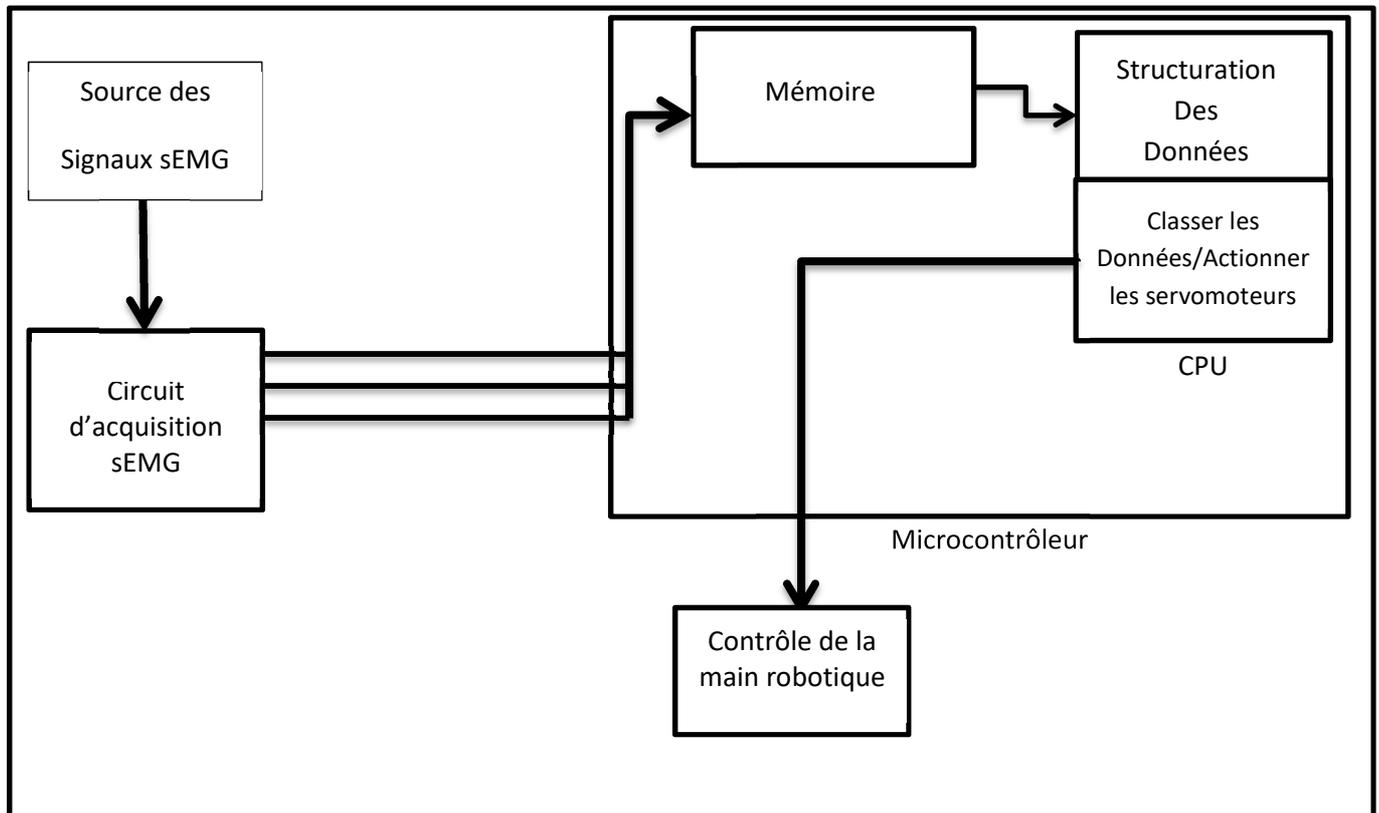


Figure 22 : représentation schématique de l'évolution du signal EMG dans notre système.

3.1 Analyse du système de contrôle de la prothèse :

Les signaux mesurés avec le circuit d'acquisition sEMG doivent être envoyés au microcontrôleur pour effectuer le traitement. Par conséquent, un ADC est nécessaire pour convertir les signaux analogiques en signaux numériques qui peuvent être gérés par le microcontrôleur.

Un ADC à haute résolution permet de numériser avec précision de petits signaux analogiques, comme l'EMG mesuré. La fréquence d'échantillonnage est ajustée pour respecter le théorème de Nyquist, qui stipule que la fréquence d'échantillonnage doit être au moins deux fois supérieure à la limite de fréquence supérieure du signal d'origine pour être correctement récupérée après la numérisation. Dans ce cas, la fréquence d'échantillonnage doit être d'au moins 1 kHz.

Vue le nombre important d'échantillons l'utilisation du CPU pour lire la valeur de l'ADC n'est pas une solution optimale.

Le DMA est un excellent outil à utiliser avec l'ADC lorsqu'on a besoin de transférer beaucoup d'échantillons à la mémoire en continu. Il peut être utilisé pour l'échantillonnage audio, un oscilloscope personnalisé, ou dans notre cas pour l'échantillonnage du signal sEMG. Une fois les données correctement transférées dans la mémoire, il faut les structurer. Puisque nous avons opté dans le chapitre précédent pour l'Histogramme de l'EMG comme méthode d'extraction de caractéristiques, nous obtiendrons des données structurées sous la forme suivante :

$(I1C1, I1C2, I1C3, I1C2, I2C2, I3C2, I1C3, I2C3, I3C3)$

Avec $IxCx$ pour : Intervalle n° X, Canal n° X.

Après la structuration des données, vient l'étape de la classification. Notre choix s'est porté sur les réseaux de neurones artificiels. Dans notre cas le RNA en questions recevra en entrée les données structurées, et nous obtiendrons en sortie un vecteur de 6 valeurs chaque index du vecteur correspond à un geste de la main et chaque valeur du vecteur correspond à la probabilité qu'il s'agit du geste en question. La figure suivante illustre les différents gestes choisis dans ce projet.

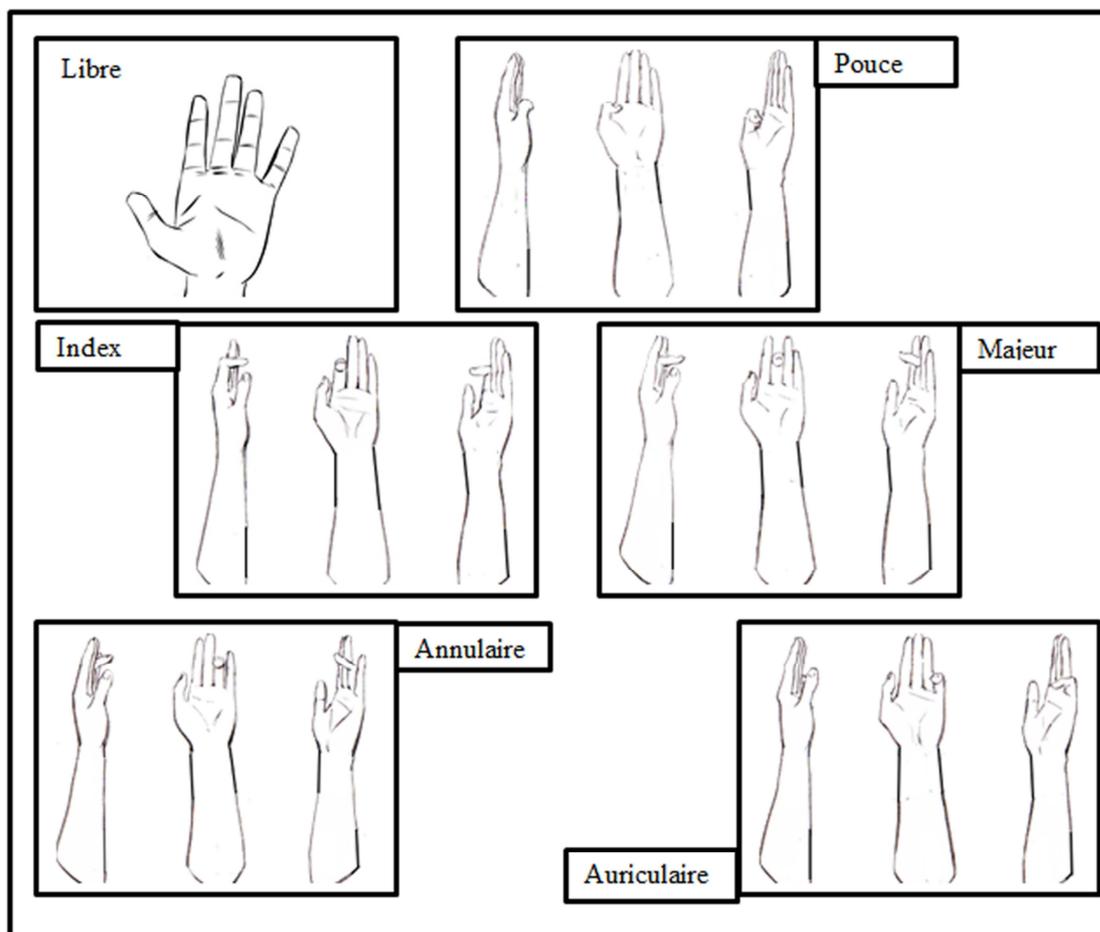


Figure 23 : les gestes choisis dans cette étude.

Selon le résultat obtenu une combinaison d'actionneurs est choisie, pour faire bouger la main robotique de façon adéquate.

3.2 Conception du RNA :

La structure générale d'un MLP est comme suit :

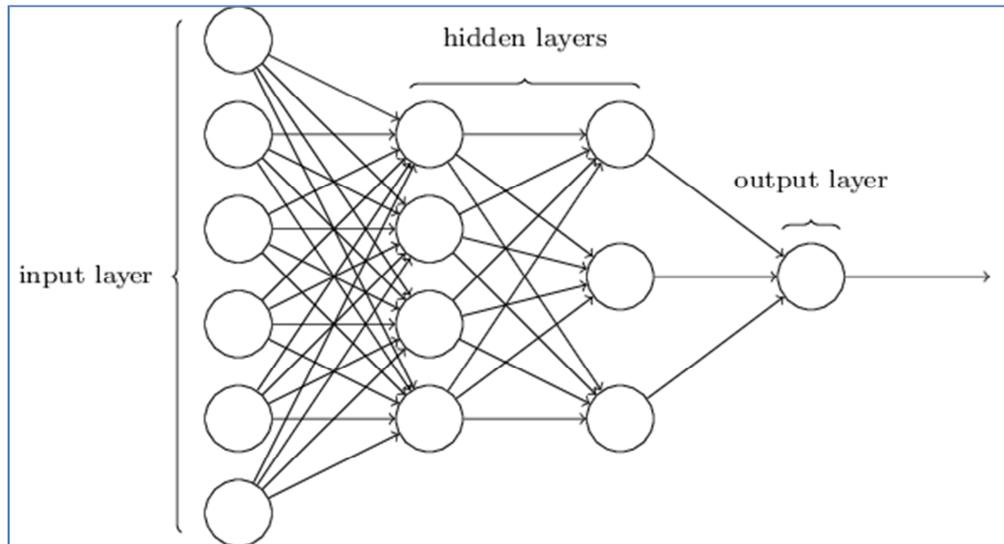


Figure 24: structure simple d'un MLP

Dans cette Etudes, Notre MLP est composé de :

- Une couche d'entrée (inputs à 9 nœuds).
- De 5 couche cachées séquentielles comportant respectivement : 60, 120, 80, 70, 60 nœuds.
- Une couche de sortie de 6 nœuds.

La figure ci-dessous est une représentation schématique de notre MLP :

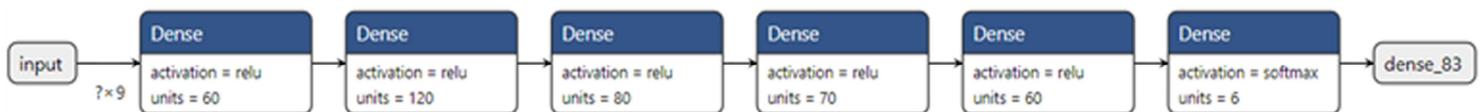


Figure 25 : représentation schématique du MLP développé dans cette étude

Le flux de travail de la génération RNA comprend:

1. Collecte de données
2. Création du dataset (ensemble de données).
3. Création du réseau
4. Formation en réseau
5. Utilisation du réseau

3.2.1 Collecte de données :

Tous d'abord on commence par la collecte de données qui se fait en suivant ce schéma :

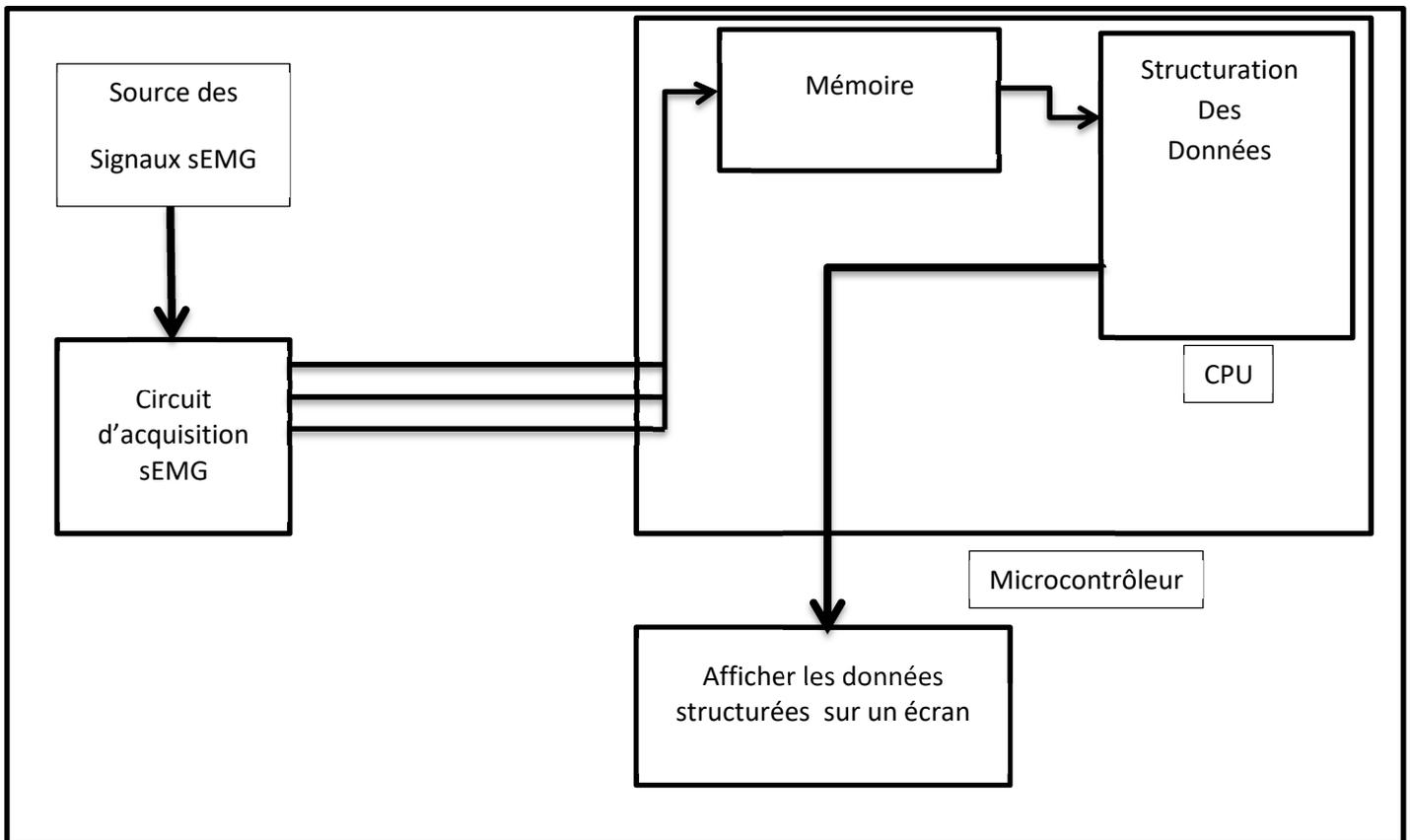


Figure 26 : collecte de données

3.2.2 Création du dataset :

Une fois les données collectées, nous allons construire le dataset, c'est une étape relativement simple qui consiste à étiqueter chaque enregistrement. Autrement dit ajouter un champ qui définit la classe de l'enregistrement. Comme la montre la figure suivant :

$(I1C1, I1C2, I1C3, I1C2, I2C2, I3C2, I1C3, I2C3, I3C3, \text{etiquette})$

Avec $IxCx$ pour : Intervalle n° X, Canal n° X.

Exemples :

(0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0-libre)
(0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 26.00, 6.00, 1-pouce)
(0.00, 0.00, 0.00, 0.00, 0.00, 22.00, 26.00, 29.00, 0.00, 2-index)

NB : Le rôle du chiffre ajouté au début de l'étiquette est d'assurer un ordre croissant.

3.2.3 Création du RNA

Les réseaux neuronaux se composent de :

- Couche d'entrée.
- Une quantité arbitraire de couches cachées
- Une couche de sortie,
- Un ensemble de poids et de biais entre chaque couche

Un choix de fonction d'activation pour chaque couche caché . Dans cette étude, nous utiliserons une fonction d'activations relues et softmax. Les poids et les biais sont initialisés aléatoirement, il faut donc les ajuster correctement.

3.2.4 Entraînement du RNA

Naturellement, les bonnes valeurs pour les poids et les biais déterminent la force des prédictions. Le processus d'ajustement des poids et des biais des données d'entrée est connu sous le nom de formation du Réseau neuronal.

Chaque itération du processus de formation se compose des étapes suivantes :

- Calcul de la sortie prévue, connue sous le nom de feedforward.
- Mise à jour des poids et des biais, connus sous le nom de rétropropagation.

Le graphique séquentiel ci-dessous illustre le processus.

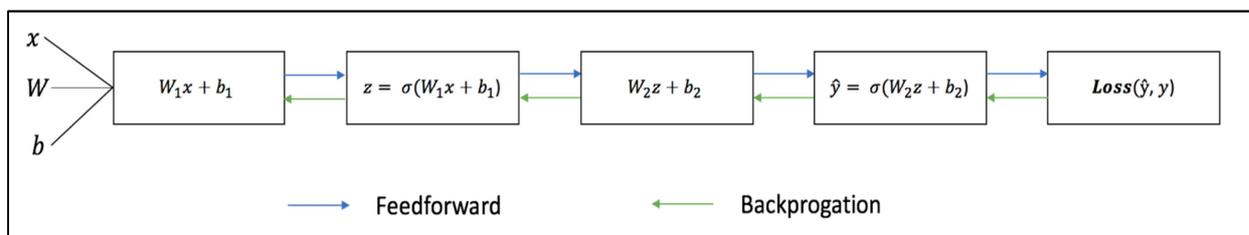


Figure 27 : Processus d'entraînement d'un RNA

Notre objectif à l'entraînement est de trouver le meilleur ensemble de poids et de biais qui minimise la perte.

3.2.5 Utilisation du réseau :

Après avoir créé et entraîné le réseau de neurones artificiels, on exporte le modèle pour l'exploitation dans notre système de contrôle.

Chapitre 4 : Réalisation.

Introduction :

Le but du système de contrôle myoélectrique est d'utiliser le nombre minimum possible de composants pour collecter avec précision le signal EMG .Ce chapitre décrit et explique dans une première partie les avantages des composants (software et hardware) utilisés pour acquérir les signaux EMG, en se concentrant sur le circuit électronique d'acquisition EMG, le microcontrôleur, le pont diviseur de tension, la main INMOOV et les différents logiciels. La seconde partie représente l'intronisation de ce travail, elle a pour objectif l'implémentation de notre application, mais aussi la mise en œuvre du système de contrôle myoélectrique. Nous y verrons donc Un aperçu du code, les configurations réaliser dans ce travail, les résultats des tests effectuées et enfin nous citerons des perspectives.

4.1 Environnements de développement :

4.1.1 Environnements de développement matériel :

4.1.1.1 Circuit d'acquisition de signaux sEMG:

Le circuit d'acquisition EMG utilisé a été développé par Schoen-Scott Technology LLC qui se spécialise dans le développement de produits biomédicaux.

Bien qu'il existe de nombreux facteurs influençant la mesure des signaux sEMG, tels que les facteurs physiologiques et non physiologiques, le système de détection joue un rôle important dans l'assurance de bonnes données de mesure qualitatives. Certains des facteurs qui doivent être pris en compte avec les systèmes de détection, sont :

- Contact électrode-peau (impédance, bruit).
- Filtrage spatial pour la détection du signal.
- Distance inter-électrode.
- Taille et forme des électrodes.
- Inclinaison du système de détection par rapport à l'orientation des fibres musculaires.
- Emplacement des électrodes sur le muscle.

Il ne s'agit pas d'une liste complète de tous les facteurs qui influencent les signaux sEMG, mais fournit un aperçu de la complexité que l'on pourrait rencontrer lors de la collecte et le traitement des signaux sEMG.

La ZTEMG-1300 est un circuit d'acquisition sEMG à trois (3) canaux très compacte. Chaque canal possède son propre potentiomètre de polarisation réglable pour éviter la saturation du capteur. Les Caractéristiques de ce capteur sont:

1. Spécialement conçu pour l'interfaçage avec des microcontrôleurs.
2. Tension de polarisation réglable pour chaque canal

3. Les trois canaux sEMG comportent chacun un connecteur de 3,5 mm pour faciliter la connexion des fils conducteurs et des électrodes.
4. Compatible avec la planche à pain.
5. Tension d'alimentation: min 12V
6. Tension de sortie de puissance: + 5V et -5V
7. Tension de sortie du signal EMG: -50 a + 5V
8. Gain d'amplification global de 1 000

Le circuit ZTEMG-1300 est illustré ci-dessous :

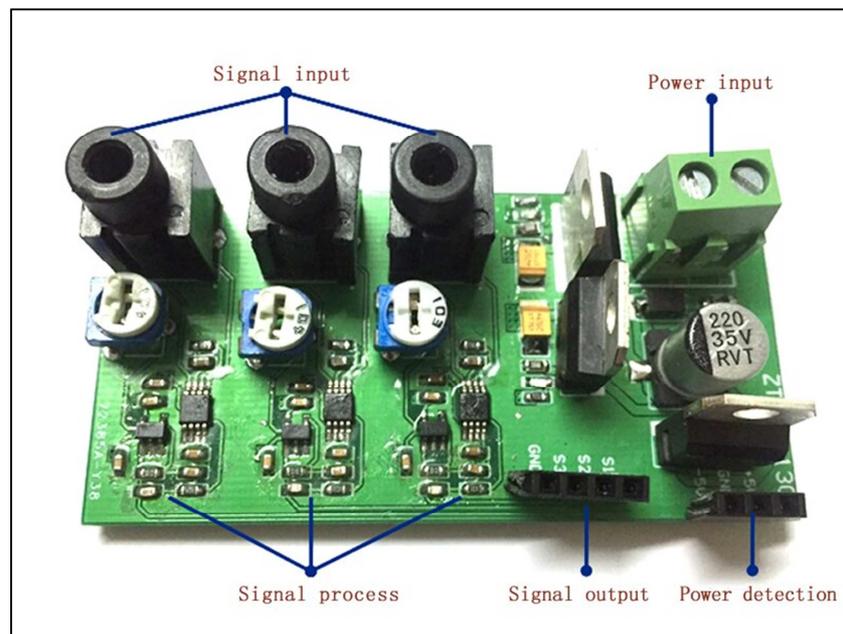


Figure 28 : circuit d'acquisition de signaux sEMG.

4.1.1.2 La carte à microcontrôleur:

STMicroelectronics est une société de développement et de réalisation de solutions destinées à un grand nombre d'applications microélectroniques, elle fournit une vaste gamme de cartes électroniques : la famille microcontrôleurs STM32FXXXX, parmi ces cartes nous avons la STM32F4 DISCOVERY développée autour du microprocesseur ARM Cortex-M4F 32 bits.

Afin de permettre la création d'un large éventail d'applications, le kit de découverte STM32F4 est fourni avec une bibliothèque logicielle complète (firmware) STM32Cube. La plateforme matérielle du kit permet d'accéder aux fonctionnalités du microcontrôleur.

La carte STM32F4 Discovery permet de développer des applications facilement en exploitant les capacités du microcontrôleur STM32F4 muni d'un processeur ARM Cortex-M4F 32 bits.

I. Caractéristiques ^[26] :

La carte STM32F4Discovery offre les caractéristiques suivantes :

- ❖ Un microcontrôleur STM32F407VGT6 avec processeur ARM Cortex-M4F 32 bits doté de :
 - Une mémoire Flash de 1 Mo.
 - Une mémoire vive de 192 Ko.
 - Une FPU (floating point unit).
 - Un module DSP (Digital Signal Processor).
- ❖ Un ST-LINK/V2 intégré.
- ❖ Alimentation de la carte.
 - Par bus USB.
 - Par alimentation externe : 3 V ou 5 V.
- ❖ Un accéléromètre à 3 axes ST MEMS LIS3DSH.
- ❖ Un capteur audio (microphone) digital omnidirectionnel ST MEMS MP45DT02.
- ❖ Un DAC audio (CIRRUS CS43L22) avec un contrôleur de haut-parleur de classe D intégré.
- ❖ Huit LEDs :
 - LD1 (rouge/verte) pour la communication USB.
 - LD2 (rouge) pour la mise sous tension 3 V.
 - LEDs Utilisateur : LD3 (orange), LD4 (verte), LD5 (rouge) et LD6 (bleue).
 - 2 LEDs USB OTG (USB On-The-Go) : LD7 (verte) VBUS, LD8 (rouge) surintensité.
 - Deux boutons-poussoirs (utilisateur et réinitialisation).
 - Interface USB OTG avec connecteur micro-AB.
 - En-têtes d'extension pour les E/S pour une connexion rapide à la carte.
 - Logiciel gratuit, complet et comprenant une variété d'exemples, constituant le package logiciel STM32CubeF4.

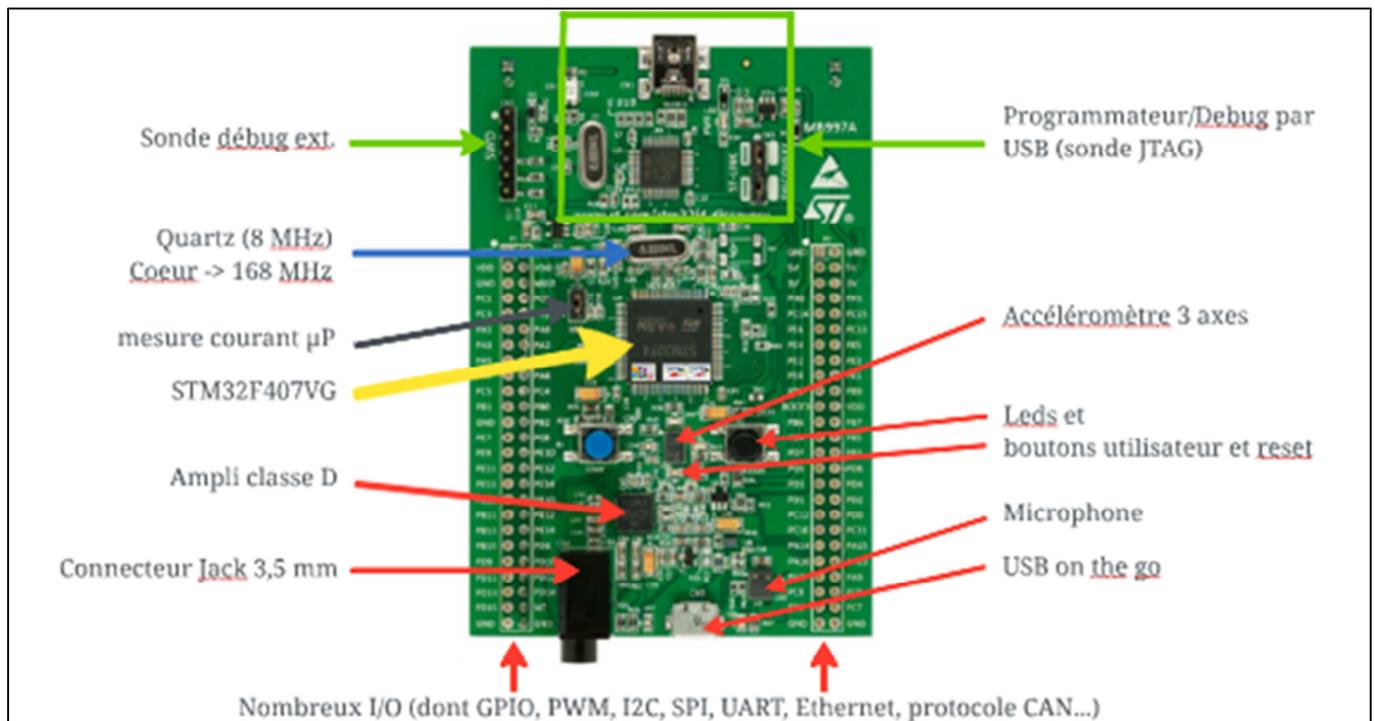


Figure 29 : Composants de la carte STM32F4 DISCOVERY.

II. Fonctionnalités ^[26] :

STM32F4-DESCOVERY présente plusieurs fonctions d'acquisitions, de traitement et de communication. Ici, nous présenterons seulement les fonctionnalités en relation avec notre projet qui sont :

1. Entrées Sorties à usage Général (GPIO) :

Un GPIO est un moyen de communication entre le microcontrôleur et les périphériques externes.

La STM32 est suffisamment munie de broches E/S à usage général, ayant typiquement 80 broches E/S bidirectionnels à un niveau de tension 3.3V. Ces broches sont disposées en cinq ports ayant chacun 16 lignes d'E/S. Ces ports sont nommés A, B, C, D et E et sont tolérant à 5V. Donc pour configurer un périphérique relié à une broche il suffit de configurer les GPIOs Correspondant en utilisant le module HAL_GPIO.

Les caractéristiques principales de la GPIO sont :

- 16 E/S sous contrôle.
- Mode de sorties : push-pull, open-drain +pull-up/down.
- Mode d'entrées : floating, pull-up/down, analog.
- Un multiplexage très flexible des broches, permet l'utilisation des broches d'E/S comme GPIO avec une simple configuration sans pour autant connaître les registres, leurs mappages en mémoires ou la manière de configuration des périphériques.
- Attribution séparée des vitesses pour les E/S.

2. *Convertisseur analogique numérique (ADC) :*

Les microcontrôleurs STM32 ont l'un des ADC les plus avancés. Ils sont basés sur la méthode d'approximations successives avec une résolution de 12 bits.

L'ADC possède jusqu'à 19 Canaux (channels) multiplexés qui lui permettent de mesurer des signaux à partir de 16 sources externes et 2 sources internes et :

- 3 entrées internes : V_REFINT, V_BAT, temp-sensor.
- Une entrée de référence analogique positive V_(REF+) : ($1.8\text{v} \leq V_{(REF+)} \leq V_{DDA}$).
- Une entrée d'alimentation analogique V_DDA ($1.8\text{v} \leq V_{DDA} \leq 3,6\text{V}$) : (pour une vitesse réduite) ou ($2,4\text{v} \leq V_{DDA} \leq 3,6\text{V}$) : (à pleine vitesse).
- Une entrée de référence analogique négative V_(REF-) : ($V_{(REF-)} = V_{SSA}$).
- Une entrée de masse analogique V_SSA.
- Un bloc de conversion analogique numérique.
- Un bloc de registre de données de 16 bits.
- Un bloc de commande d'interruption.
- Un bloc de déclenchement de la conversion.
- Un watchdog analogique qui permet de détecter si la tension d'entrée est au-delà des seuils définis par l'utilisateur.

L'ADC de la STM32F4 Discovery présente plusieurs caractéristiques :

- Résolution de l'ADC est configurable à 12, 10, 8 ou même 6 bits.
- Deux principaux modes de fonctionnement :
 - Mode indépendant (independent) : Chaque module ADC fonctionne de manière autonome.
 - Mode double (dual) : Deux unités ADC travaillant mutuellement ensemble comme s'il s'agissait d'une seule unité.
- Mode de Conversion :
 - Single : conversion simple, qui peut être une seule conversion à partir d'un canal.
 - Scan : conversion multicanale, qui peut être une seule conversion à partir de plusieurs canaux.
 - Single continuous : conversion simple et continue, qui peut être plusieurs conversions à partir d'un canal.
 - Scan continuous : conversion multicanale continues, qui peut être plusieurs conversions à partir de plusieurs canaux.
 - Discontinuous : conversion séquentielle à partir de quelques canaux dans un groupe.
- L'ADC peut être déclenché par logiciel (HAL_ADC), par matériel interne (timers) ou externe (les broches EXTI-11, EXTI-15).
- Le stockage de données est configurable avec DMA (Direct Memory Access).
- L'horloge du bloc ADC (ADCCLK) est générée à partir de l'horloge APB2, divisée par un prescaler programmable.

- Le temps de conversion total est le temps d'échantillonnage + 12 cycles d'horloge, cela permet un taux d'échantillonnage et de conversion allant jusqu'à 2,4 Msps (millions of samples per second).
- Un groupe de conversion se compose d'une séquence de conversions qui peut être effectué sur n'importe quel canal et dans n'importe quel ordre. La conversion est organisée en deux groupe :
 - Groupe régulier : il est composé de jusqu'à 16 conversions, c'est un groupe de canaux fixe régulièrement converti.
 - Groupe injecté : il est composé de jusqu'à 4 conversions, il peut interrompre temporairement une conversion d'un groupe régulier car il est plus prioritaire,

3. Accès directs à la mémoire (DMA) :

Étant donné que les valeurs converties d'un canal sont stockées dans un registre de données uniques, il est utile d'utiliser DMA pour la conversion de plusieurs canaux réguliers. Cela permet d'éviter la perte de données.

L'accès direct à la mémoire (DMA) est utilisé pour fournir un transfert de données à haute vitesse entre périphériques et mémoire et entre mémoire et mémoire. Les données peuvent être déplacées rapidement par DMA sans aucune action CPU. Cela permet de libérer des ressources CPU pour d'autres opérations.

Les principales fonctionnalités du DMA :

- Le contrôleur DMA est basé sur une architecture de bus complexe, il combine l'architecture du bus maître AHB avec FIFO pour optimiser la bande passante du système.
- Il a deux Architectures du bus AHB master (maître), une dédiée aux accès mémoire et une dédiée aux accès périphériques.
- Il a une interface de programmation AHB slave (esclave) prenant en charge uniquement les accès 32 bits.
- Le DMA a un total de 8 flux (16 flux pour les 2 DMA).
- Chaque flux du DMA a jusqu'à 8 canaux sélectionnables, selon les bits CHSEL [2 :0] dans les DMA_SxCR.
- Chaque canal est dédié à un périphérique spécifique et peut être sélectionner par logiciel. Ce qui permet à plusieurs périphériques d'utiliser le DMA au même temps.
- Chaque canal a un tampon FIFO, d'une longueur de 4 mots (4 à 32 bits), ces tampons :
 - ❖ Peuvent être utilisés pour stocker, temporairement, des données entrantes ; ensuite quand le seuil sélectionnée ($\frac{1}{4}$, $\frac{2}{3}$, $\frac{3}{4}$, ou plein) est atteint, le transfert pourra commencer (mode FIFO).
 - ❖ Peuvent également être utilisés en mode de transfert direct (mode direct).
- Mode de transfert : normal, circulaire (cas de tampons circulaires).
- Types de transfert :
 - ❖ De mémoire en mémoire (seul le 2-ème module DMA pourra être utilisé)
 - ❖ Transfert de mémoire en périphérique, et de périphérique en mémoire.

- Le module DMA est configuré pour incrémenter automatiquement l'adresse source et l'adresse destination, après chaque transfert de données.
- Les priorités entre les demandes de flux DMA sont programmables par logiciel (4 niveaux, composé de : très haut, haut, moyen, bas) ou matériel en cas d'égalité (demande 0 a priorité sur la demande 1, etc.). Ces priorités sont gérées par l'arbitre (Arbiter).
- Les tampons FIFO peuvent être configuré comme double tampon (transfert en Ping-Pong).
- Lorsque le mode FIFO est sélectionné, le transfert de donnée en rafale peut être utilisé avec une taille de rafale de 4, 8 ou 16 unités de données.

4. Les UARTs :

La communication avec l'UART se fait par l'envoi d'une série de bits (trame) avec un délai entre chaque bit.

Dans la transmission asynchrone la trame est constituée de :

- Un bit de démarrage (START).
- Séquence de 8 bits de données.
- Un bit de parité (PARITY) qui peut être utilisé pour détecter des erreurs.
- Un bit d'arrêt (STOP).

Les caractéristiques de l'UART :

- BaudRate : c'est la vitesse de transmission (Bits/sec), les valeurs communes de BaudRate sont : 2400, 9600, 19200, 38400, 57600, 115200, ...
- WordLength : c'est le nombre de bits de données transmis ou reçus dans une trame qui peut contenir jusqu'à 8 ou 9 bits. Ce nombre n'inclut pas les bits généraux transmis, tels que les bits START et STOP.
- StopBits : c'est le nombre de bits STOP transmis pour signaler la fin d'une trame (1 ou 2 bits).
- Parity : il indique le mode de parité (Aucune, paire ou impaire).
- Mode : il indique si le mode est en Emission Tx, Réception Rx, ou peut fonctionner à la fois en émission et en réception.
- HwFlowCtl : il indique si le mode de contrôle de flux matériel RS232 est :
 - Désactivé.
 - Request To Send (RTS) est activé.
 - Clear To Send (CTS) est activé.
 - RTS et CTS sont activés.
- OverSampling : c'est la technique d'échantillonnage d'un signal avec une fréquence d'échantillonnage supérieur à la fréquence du signal. Peut prendre 8 ou 16

5. Les Timers :

Un timer est un compteur électronique avec une fréquence de comptage qui est une fraction de son horloge source. Sa résolution qui dépend principalement de son architecture et contient par conséquent l'amplitude des valeurs sur lesquels il peut compter.

L'entrée de l'horloge du timer est précédée par un préscaler (diviseur), son rôle est de ralentir la fréquence de comptage du timer en opérant une première division de la fréquence de l'horloge.

Les Catégories de timers dans un μC STM33F4 :

- Les timers de base :

Ce sont des timers à 16 bits utilisés comme générateurs de base de temps et ils n'ont pas de broches de d'E/S. Les timers de base peuvent également être utilisés comme "maîtres" pour les autres timers (utilisés comme esclaves). Ils représentent la forme la plus simple de timer dans les μCs STM32.

- Les timers à usage général

Ce sont des timers à 16/32 bits, fournissant les fonctionnalités classiques qu'un timer d'un μC moderne devrait implémenter, Ils sont aussi utilisés comme générateur de base de temps. Ils sont utilisés dans n'importe quelle application pour :

- Une comparaison de sortie (synchronisation et génération de délai).
- Une capture d'entrée (pour mesurer la fréquence d'un signal externe).
- La génération d'un signal PWM.

Les timers de cette catégorie fournissent quatre canaux d'E/S programmables excepté pour les deux sous-catégories suivantes :

- Les timers à 1 canal.
- Les timers à 2 canaux.

- Les timers avancées

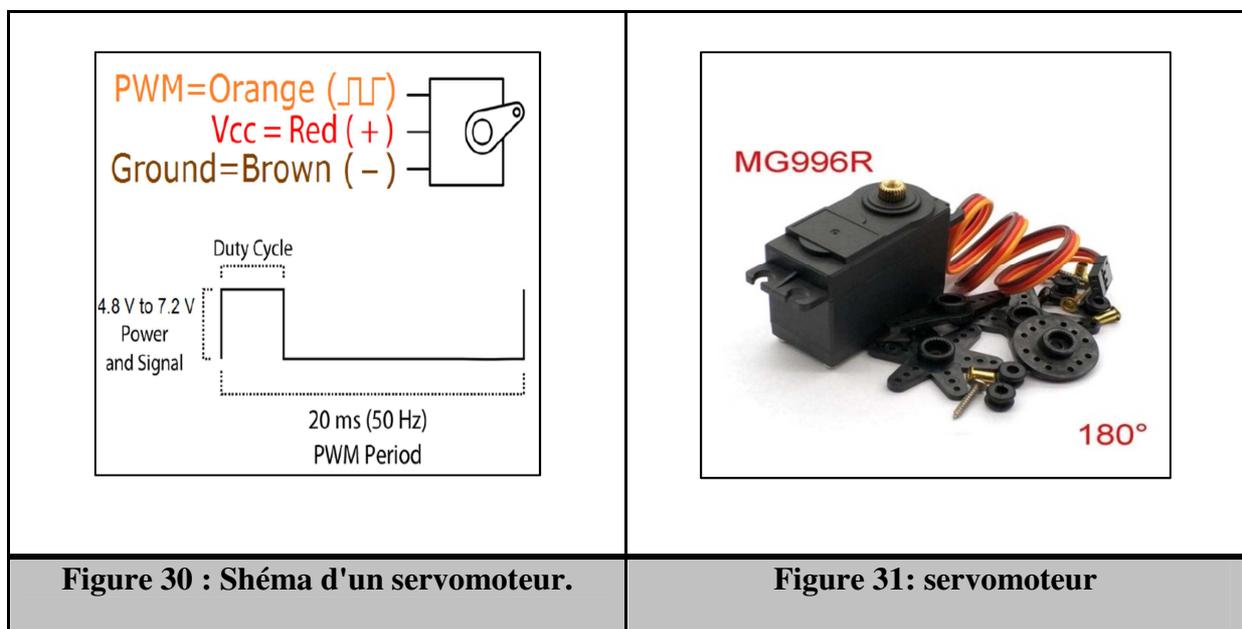
Ce sont des timers à 16 bits et sont les plus complets d'un μC STM32. En plus des fonctionnalités d'un timer à usage général, ils incluent plusieurs fonctionnalités liées à la commande de moteurs et aux applications de conversion de puissance numérique...

4.1.1.3 Servomoteur MG-996R :

Le servomoteur MG996R est doté d'un engrenage en métal résultant en un couple de calage extrêmement élevé de 10 kg dans un boîtier minuscule. Ce servomoteur standard à couple élevé peut tourner d'environ 180 degrés (90 dans chaque direction).

Caractéristiques :

- Poids: 55 g
- Dimensions: 40,7 x 19,7 x 42,9 mm environ.
- Couple de décrochage: 9,4 kgf · cm (4,8 V), 11 kgf · cm (6 V)
- Vitesse de fonctionnement: 0,17 s / 60 ° (4,8 V), 0,14 s / 60 ° (6 V)
- Tension de fonctionnement: 4,8 V à 7,2 V
- Courant de fonctionnement 500 mA –900 mA (6 V)
- Courant de décrochage 2,5 A (6 V)
- Largeur de bande morte: 5 μs
- Conception à double roulement à billes stable et antichoc
- Plage de température: 0 °C - 55 °C



4.1.1.4 Le pont diviseur de tension :

Le pont diviseur de tension est un montage électrique simple. Il permet de déterminer une tension proportionnellement à une autre tension. Ce type de montage est utilisé couramment pour créer une tension de référence dans un circuit électrique.

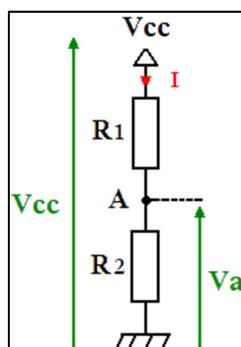


Figure 32 : schéma du pont diviseur de tension.

La plus simple représentation du diviseur de tension consiste à placer deux résistances électriques en série (voir le schéma ci-dessus). Ce type d'association de résistances étant omniprésent dans les montages électriques, le pont diviseur en devient une des notions fondamentales en électronique.

Il est possible de calculer facilement la valeur de V_A dans le montage ci-dessus, à la condition de connaître les valeurs des résistances et la valeur de la tension V_{cc} . La première formule à utiliser est celle de la loi d'Ohm qui permet de citer cette équation:

$$I = V_{cc} / (R1+R2)$$

En utilisant la loi d'Ohm une seconde fois, il est possible de déterminer l'équation suivante:

$$V_a = R2 * I$$

Dans la formule ci-dessous, il suffit de remplacer le courant (I) par sa valeur équivalente (la première équation) pour déterminer facilement l'équation de V_A :

$$V_a = R2 * (V_{cc} / (R1+R2))$$

L'ADC de la stm32f4 Discovery fonctionne avec une tension maximale de 3.3v et de l'autre côté le circuit ZTEMG-1300 génère une tension qui varie entre 0V et 5V, ce qui justifie l'utilisation du pont diviseur de tension.

- **Calculs :**

$$V_a = 1000 * (5 / (1000 + 3000)) = 3.3333333v$$

Donc $R1 = 1k\Omega$, $R2 = 2k\Omega$.

4.1.1.5 La Main INMOOV :

Au cours des deux dernières années, le développement de membres bioniques imprimés en 3D est devenu très populaire. InMoov est un projet indépendant développant une vie comme un robot humanoïde à partir de la technologie d'impression 3D. L'ensemble du projet est open source et fournit un excellent aperçu de la conception mécanique dans la production de pièces de corps robotiques imprimées en 3D ^[12].

La nature open source de ce projet permet au public d'accéder aux conceptions assistées par ordinateur et de suivre des guides étape par étape sur la façon d'imprimer et d'assembler ce système en 3D. Les doigts InMoov sont contrôlés par des tendons actionnés par des servomoteurs placés dans l'avant-bras. ^[12]



Figure 33: le prototype de mains robotisé utilisé dans notre projet

La main robotisée que nous avons utilisée est représentée dans la figure précédente.

4.1.2 Environnements de développement logiciel :

4.1.2.1 STM32CubeIDE :

Le développement d'applications basé sur la carte STM32 ne peut se faire sans passer par les étapes de configuration et de programmation, pour y parvenir l'outil de développement multi-OS STM32CubeIDE, qui fait partie de l'écosystème logiciel STM32Cube est mis à notre disposition.

STM32CubeIDE est une plate-forme de développement C / C ++ avancée, ayant un ensemble de fonctionnalités permettant la configuration des périphériques, la génération de code, la compilation de code et le débogage pour les microcontrôleurs et microprocesseurs STM32. Il est basé sur le framework Eclipse et la chaîne d'outils GCC pour le développement, et GDB pour le débogage.

Ses principales caractéristiques sont :

- Intégration des services de STM32CubeMX :
 - Microcontrôleur STM32, microprocesseur, plateforme de développement et exemple de sélection de projet.
 - Configuration des périphériques...
 - Création de projet et génération du code d'initialisation.
 - Logiciels et middleware complétés par des packages d'extension STM32Cube améliorés.
- Basé sur Eclipse avec prise en charge des modules complémentaires Eclipse, de la chaîne d'outils GNU C / C ++ pour Arm et du débogueur GDB
- Fonctionnalités de débogage supplémentaires, notamment :
 - Vues du cœur du processeur, du registre périphérique et de la mémoire.
 - Analyse du système et traçage en temps réel (SWV).
 - Outil d'analyse des défauts du processeur.
- Prise en charge des sondes de débogage ST-LINK (STMicroelectronics) et J-Link (SEGGER)
- Importer un projet depuis Atollic, TrueSTUDIO et AC6 System Workbench pour STM32 (SW4STM32).

4.1.2.2 XCUBE.AI :

C'est un package d'extension d'intelligence artificielle pour STM32Cube qui fait partie de l'écosystème STM32Cube.AI, il étend les capacités de STM32CubeMX avec la conversion automatique du réseau neuronal préformé.

Ce pack d'extension X-CUBE-AI offre également plusieurs moyens de valider les modèles de réseau neuronal à la fois sur PC et STM32, ainsi que de mesurer les performances sur les cartes STM32 sans code C.

Ses caractéristiques sont :

- Génération d'une bibliothèque optimisée STM32 à partir de modèles de réseaux de neurones pré-entraînés.
- Prise en charge native de divers frameworks de Deep Learning tels que Keras, TensorFlow Lite, Caffe, ConvNetJs et Lasagne, et support de tous les frameworks qui peuvent exporter au format standard ONNX tels que PyTorch , Microsoft Cognitive Toolkit, MATLAB et plus...
- Prend en charge la quantification 8 bits des réseaux Keras et des réseaux quantifiés TensorFlow Lite.
- Permet l'utilisation de réseaux plus importants en utilisant un stockage externe.
- Portabilité facile à travers différentes séries de microcontrôleurs STM32 grâce à l'intégration STM32Cube.
- Gratuit.

4.1.2.3 GOOGLE COLAB:

Google Colab ou Colaboratory est un service cloud gratuit, offert par Google. Il est basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique.

Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud sans aucune autre installation apart un navigateur. Il offre également ces avantages :

- Aucune configuration requise.
- Ecriture et exécution de codes python sur un navigateur.
- Accès gratuit aux GPU.
- Partage facile.

4.1.2.4 KERAS:

Keras est une API conçue pour l'apprentissage en profondeur, écrite en python et qui s'exécute sur la plate-forme d'apprentissage automatique TensorFlow. Elle a été développée dans le but de résoudre les problèmes liés à l'apprentissage automatique.

Ce framework de deep learning est le plus utilisé car il facilite la réalisation de nouvelles expériences en une vitesse élevée, il offre des API cohérentes et simples, il minimise le nombre d'actions utilisateur requises et fournit des messages d'erreur clairs et exploitables. Il contient également une documentation complète et des guides du développeur.

4.1.2.5 SerialPlot:

Il s'agit d'un logiciel basé sur Qt pour tracer les données à partir du port de série en temps réel. SerialPlot accepte 3 différents types d'entrées de données :

- flux binaire simple, prend en charge différents formats de nombre (non signé/signé - 8/16/32 bits et float)
- Données ASCII au format CSV
- format cadre personnalisé défini par l'utilisateur (octet de démarrage du cadre, taille du cadre, checksum etc.).

Il est possible prendre des instantanés de l'affichage du tracé et les stocker.



Figure 34 : interfaces du logiciel

4.2 Assemblage du système :

La première étape de la mise en œuvre de notre système est l'assemblage de toutes les parties matérielles décrites précédemment. Les deux schémas suivant montrent de manière détaillée le branchement à effectuer.

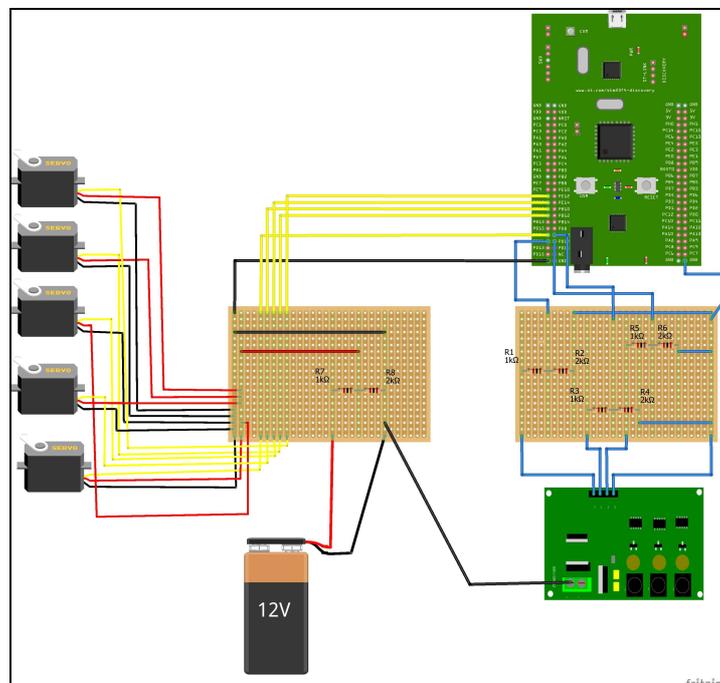


Figure 35 : schéma représentant les branchements effectués.

Après l'assemblage des composants le résultat est comme suit :

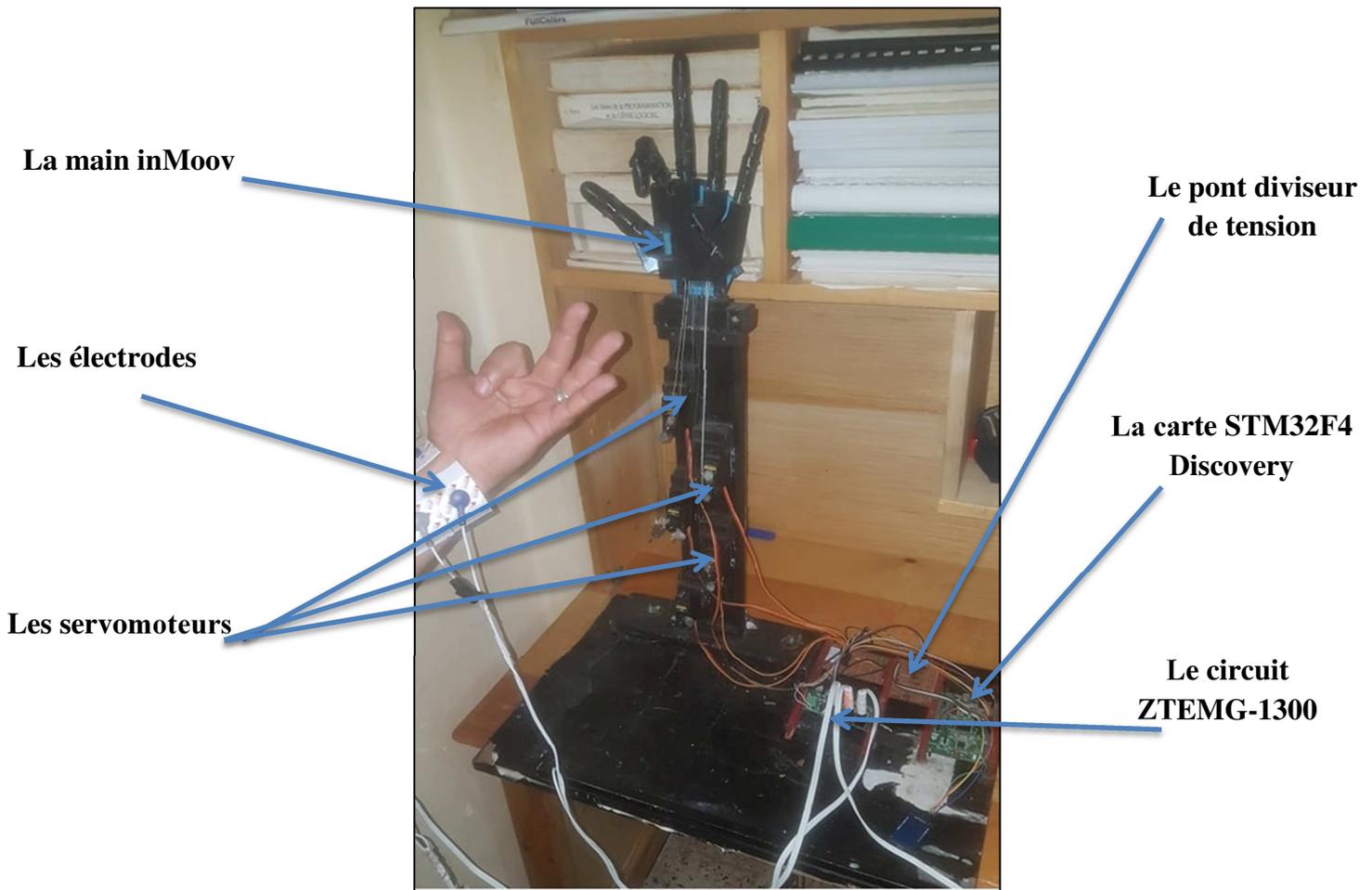


Figure 37 : mise en œuvre.

4.2 Configuration du projet :

Après avoir réalisé les branchements matériels en suivant les schémas précédents, nous devons à présent effectuer les configurations nécessaires pour le bon fonctionnement du système.

Ce schéma ci-dessous nous donne les étapes de configuration et de réalisation de notre système.

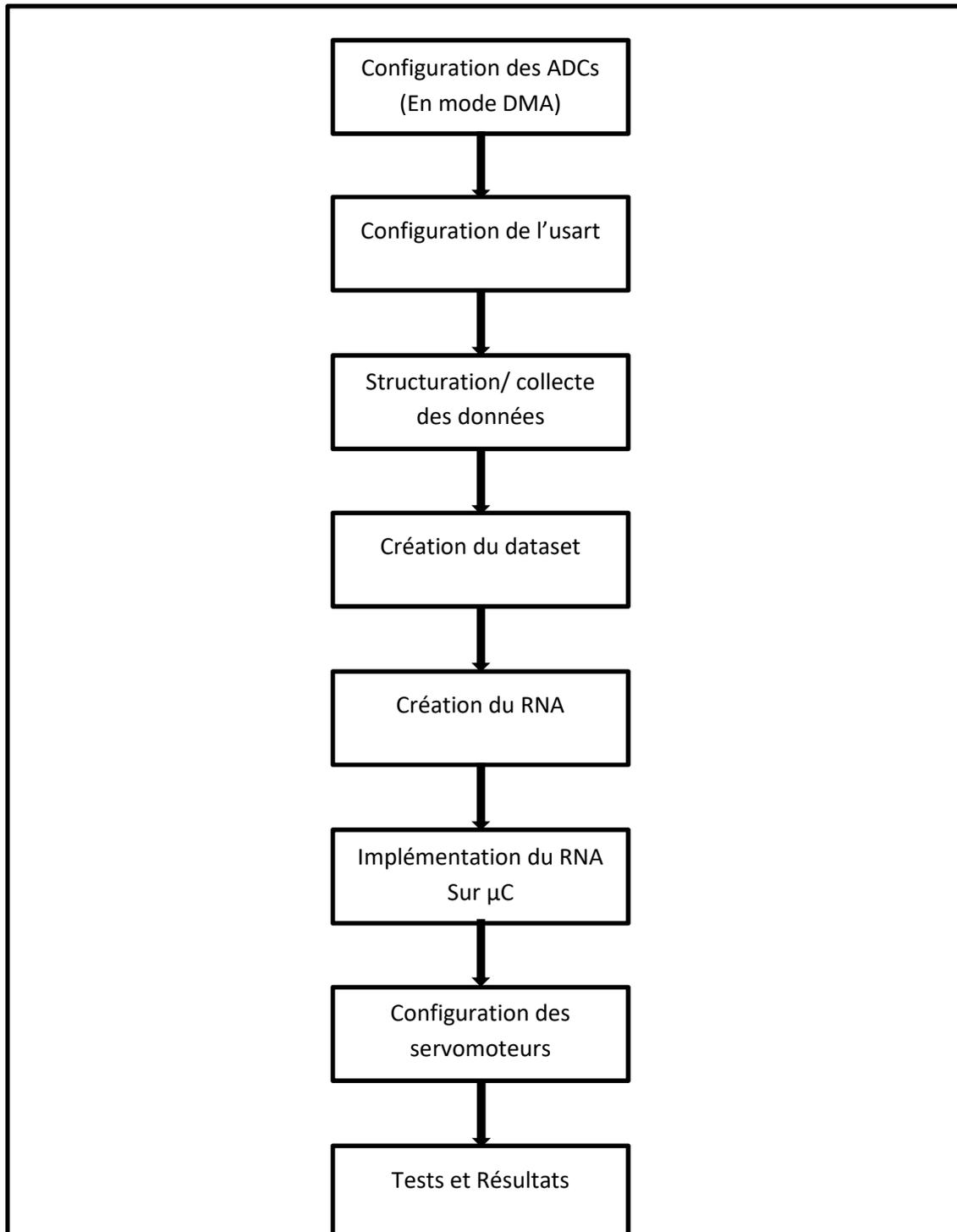


Figure 38 : les étapes de la réalisation de notre système.

Tous au long de cette partie nous expliquerons en détaille chacun des éléments composant le précédent schéma.

4.2.1 Configuration des ADCs :

Sous Catégories, on déroule le menu Analog, puis on sélectionne ADC. Dans le menu qui s'affiche (mode) on coche les cases IN0, IN1, IN2.

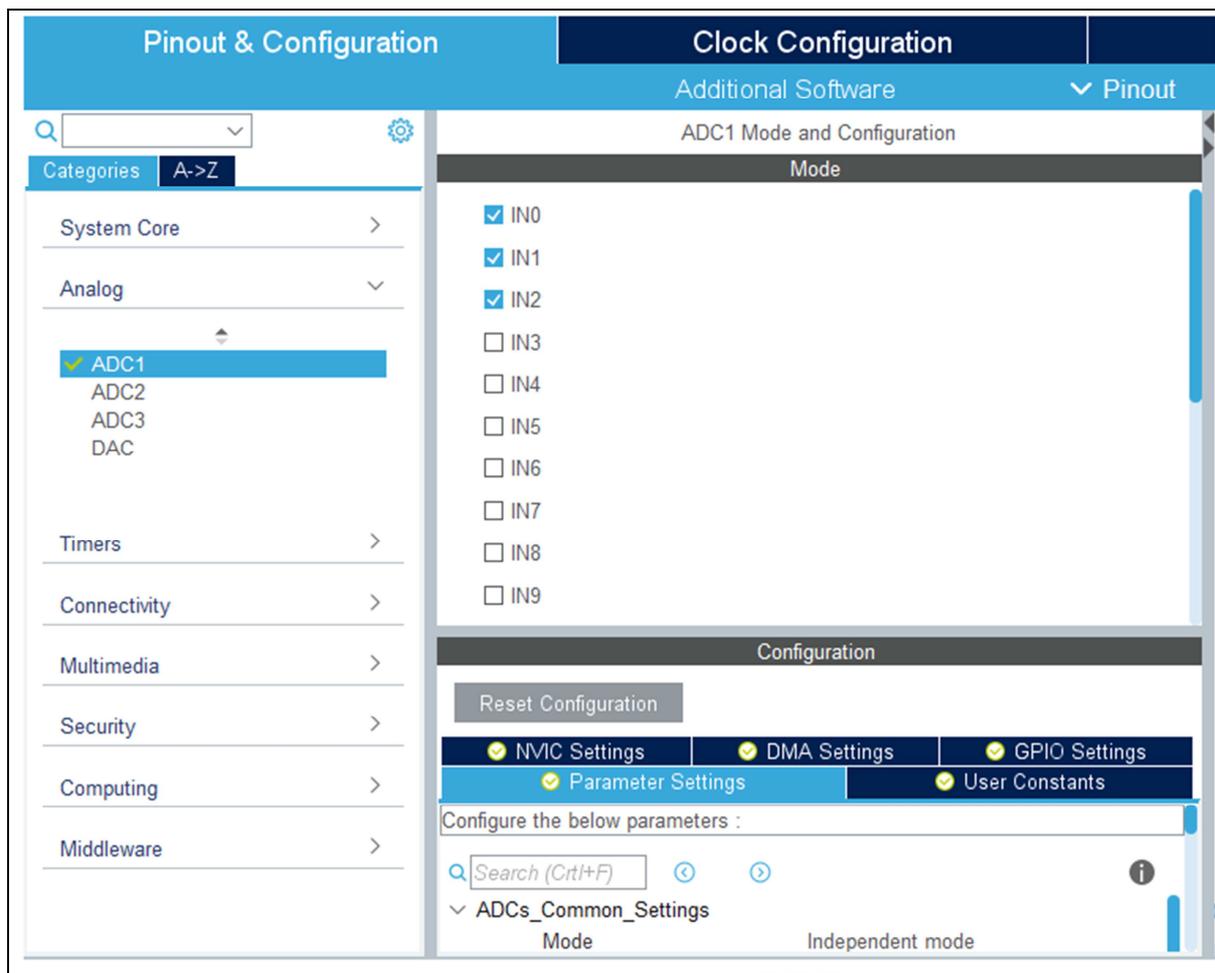


Figure 39 : Configuration des ADCs -1-

Avant d'aller plus loin nous devons configurer les horloges correctement, pour cela on sélectionne l'onglet Clock Configuration. Selon le manuelle de référence de la stm32f4 on a :

L'ADC1 est relié à APB2 Peripheral Cloks .

On la règle à 25MHz.

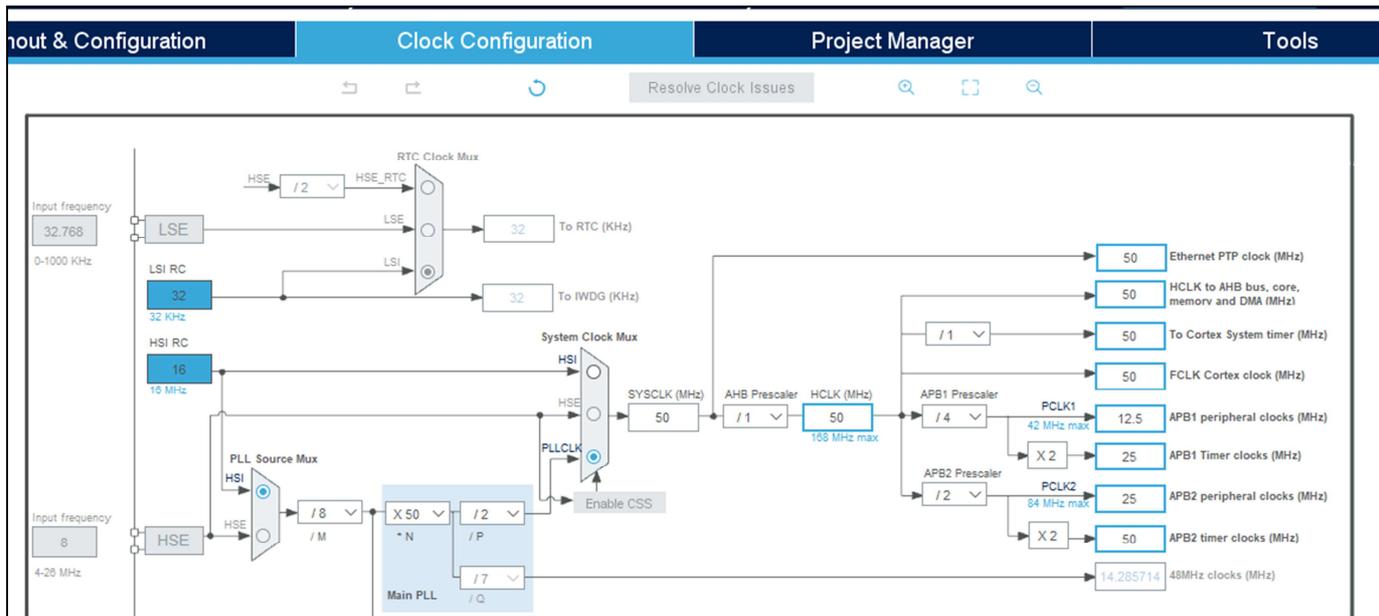


Figure 40 : Configuration des Horloges.

L'étape suivante est la configuration du DMA :

On retourne dans l'onglet Pinout & Configuration, sous catégories on déroule le menu Système core , et on sélectionne DMA. Dans le menu qui apparait on sélectionne DMA2, on clique sur le bouton Add pour créer une nouvelle requête DMA.

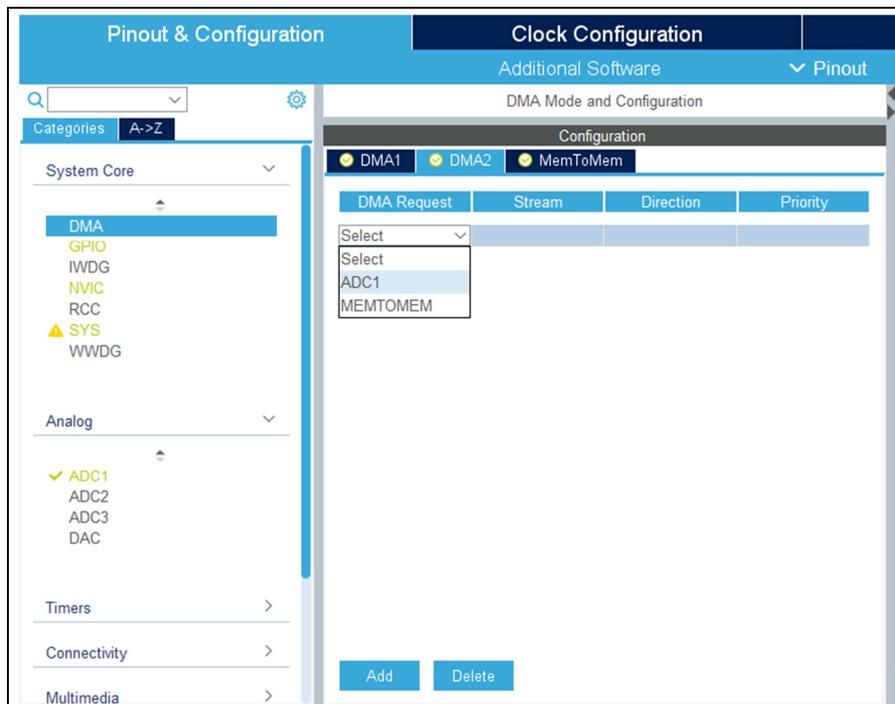


Figure 41 : Configuration du DMA -1-.

Dans DMA Request Settings on choisit le mode circulaire, et increment address : memory.

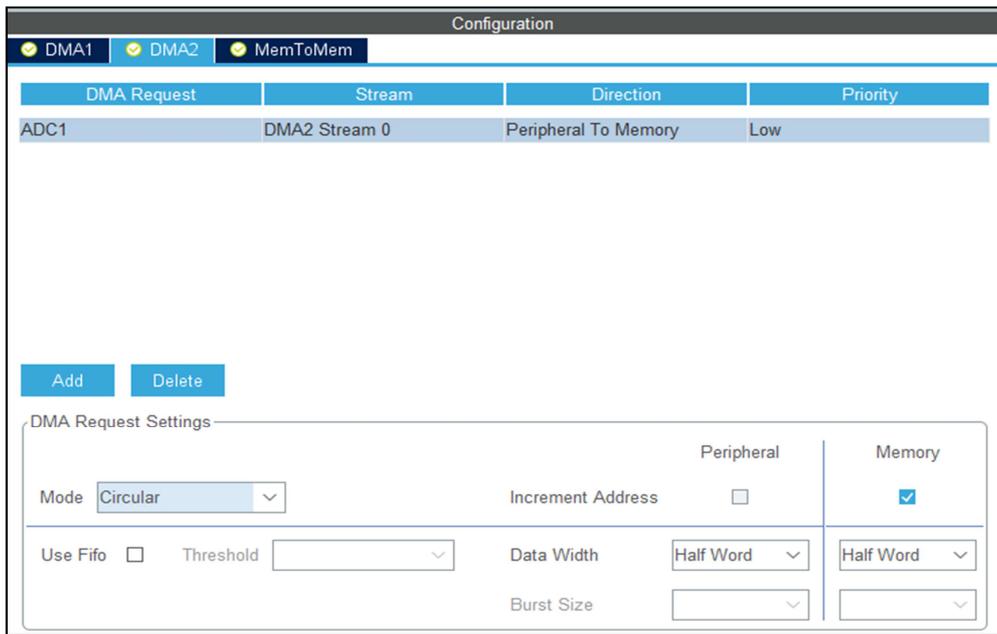


Figure 42 : Configuration du DMA -2-

Encore une fois on repasse dans le menu analog >ADC1>Configuration >Parameter Settings, et on règle les paramètres comme montrer dans la figure suivante.

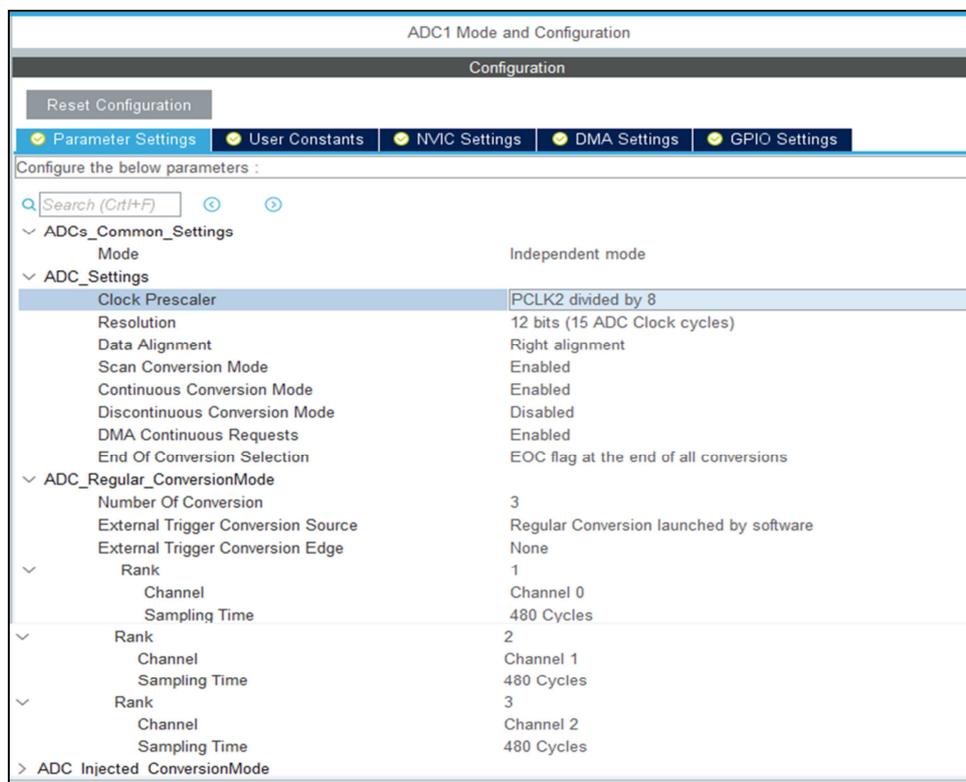


Figure 43 : Configuration des ADCs -2-

D'après les données suivantes :

- Segmentation de 250ms.
- Plage du signal EMG 20Hz-500Hz
- Théorème de Nyquist-Shannon
- Horloge de 50MHz.
- Et les deux formules suivantes :
 - $T_{conv_cycle} = \text{Simpling_time}(\text{cycle}) + 15(\text{resolution } 12\text{bits})$
 - $T_{conv_second} = T_{conv_cycle} * \text{Periode.}$

On peut facilement trouver le nombre de valeurs récupérées pendant les 250ms.

1- Calculer la fréquence :

$$25\text{MHz} / 8(\text{clock prescaler}) = 3.125\text{MHz.}$$

2- Période = $1 / 3.125\text{MHz} = 0.32\mu\text{s}$

3- $T_{conv_cycle} = 480 + 15(\text{resolution } 12\text{bits}) = 495\text{cycles}$

4- $T_{conv_second} = 495 * 0.32 = 158.4\mu\text{s}$

158.4µs pour chaque capture de chaque canal.

Donc 475.2 µs pour prélever un échantillon des trois canaux. Pour une période de 250ms (250 000µs) : 1578 valeurs pour les 3 valeurs soit 526 valeurs par canal.

4.2.2 Configuration de l'USART:

Pour la configuration de l'USART, sous Catégories, on déroule le menu Coconnectivity, on sélectionne USART2 et dans mode on met asynchrone et on laisse les valeurs par défaut.

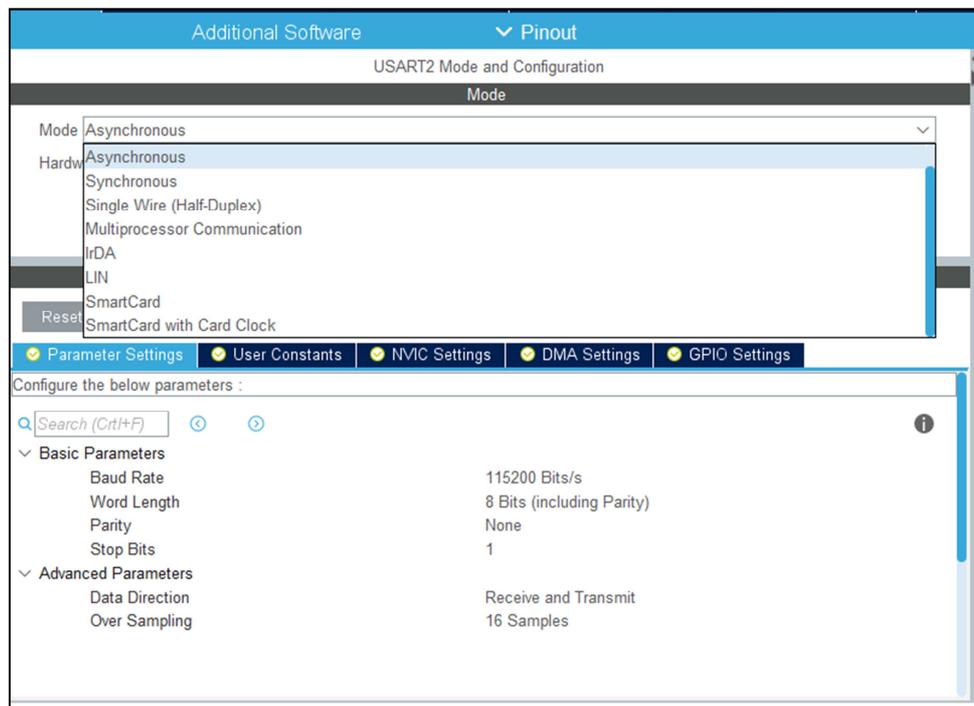


Figure 44 : Configuration de l'USART.

4.2.3 Structuration/ collecte des données:

Après l'étape de configuration vient l'étape de la structuration et la collecte de données, les figures suivantes montrent les parties de codes chargées de la collecte et la structuration de données.

```
if( HAL_ADC_Start(&hadc1) != HAL_OK)
return 0;
// -- Enables ADC DMA request
if (HAL_ADC_Start_DMA(&hadc1, (uint32_t*)ADC1ConvertedValues, adc_len) != HAL_OK)
return 0;
```

Figure 45 : Initialisation du DMA.

```
i=0;
cmt=0;

while(i!=adc_len-3){
    Ch0[cmt]=ADC1ConvertedValues[i];
    Ch1[cmt]=ADC1ConvertedValues[i+1];
    Ch2[cmt]=ADC1ConvertedValues[i+2];
```

Figure 46 : Fonction de structuration de données -1-.

```
emg_result_type exploitation_des_donnees(uint32_t Val_EMG[ch_len]){

    emg_result_type a;
    a.intrval4=0;
    a.intrval5=0;
    a.intrval6=0;
    for(int i=0;i<ch_len;i++){ if(Val_EMG[i]<1750){a.intrval4++;}
                             if(Val_EMG[i]<1750){a.intrval4++;}
                             if((Val_EMG[i]>2800)){a.intrval5++;}

    }

    return a;
```

Figure 47 : Fonction de structuration de données-2-.

```

sprintf(buffer, "%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f\n",
        inputs[0],inputs[1],inputs[2],inputs[3],inputs[4],inputs[5],inputs[6],inputs[7],inputs[8]
        );
        len=strlen(buffer);
        HAL_UART_Transmit(&huart2, buffer, len, 1000);

```

Figure 48 : Affichage des données a l'écran.

4.2.4 Création du dataset:

Une fois les données structurées et afficher sur l'écran il ne reste plus qu'à les étiqueter à l'aide d'un éditeur de texte, puis enregistrer sous format csv.

```

0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0-libre
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0-libre
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0-libre
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0-libre
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0-libre
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,26.00,6.00,1-pouce
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,48.00,14.00,1-pouce
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,45.00,14.00,1-pouce
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,48.00,14.00,1-pouce
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,43.00,14.00,1-pouce
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,52.00,11.00,1-pouce
0.00,0.00,0.00,0.00,0.00,6.00,22.00,31.00,0.00,2-index
0.00,0.00,0.00,0.00,0.00,5.00,22.00,37.00,0.00,2-index
0.00,0.00,0.00,0.00,0.00,5.00,17.00,20.00,0.00,2-index
0.00,0.00,0.00,0.00,0.00,5.00,15.00,32.00,0.00,2-index
0.00,0.00,0.00,2.00,0.00,12.00,12.00,53.00,0.00,2-index
0.00,0.00,0.00,73.00,31.00,54.00,66.00,68.00,22.00,3-majeur
0.00,0.00,0.00,73.00,31.00,54.00,66.00,69.00,22.00,3-majeur
0.00,0.00,0.00,70.00,31.00,54.00,66.00,69.00,22.00,3-majeur
0.00,0.00,0.00,36.00,21.00,34.00,40.00,33.00,15.00,3-majeur
0.00,0.00,0.00,21.00,1.00,19.00,33.00,27.00,6.00,3-majeur
0.00,0.00,0.00,131.00,83.00,35.00,60.00,98.00,24.00,4-annulaire
0.00,0.00,0.00,121.00,91.00,43.00,68.00,100.00,21.00,4-annulaire
0.00,0.00,0.00,103.00,64.00,25.00,45.00,63.00,12.00,4-annulaire
0.00,0.00,0.00,102.00,58.00,30.00,55.00,48.00,5.00,4-annulaire
0.00,0.00,0.00,106.00,54.00,25.00,45.00,41.00,0.00,4-annulaire
0.00,0.00,0.00,14.00,0.00,0.00,5.00,0.00,0.00,5-auriculaire
0.00,0.00,0.00,20.00,0.00,0.00,5.00,1.00,0.00,5-auriculaire
0.00,0.00,0.00,20.00,0.00,0.00,5.00,1.00,0.00,5-auriculaire
0.00,0.00,0.00,20.00,0.00,0.00,5.00,8.00,0.00,5-auriculaire
0.00,0.00,0.00,20.00,0.00,0.00,5.00,12.00,0.00,5-auriculaire

```

Figure 49 : Une Partie du dataset

La figure ci-dessus montre une partie du dataset.

4.2.5 Création du RNA:

L'étape suivante de la réalisation consiste à écrire le programme du RNA. On commence par importer le fichier csv.

```
import numpy as np # linear algebra
import pandas as pd
from subprocess import check_output
dataset = pd.read_csv('./emg.csv')
dataset
```

	0.00	0.00.1	0.00.2	0.00.3	0.00.4	0.00.5	0.00.6	0.00.7	0.00.8	0-libre
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0-libre
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0-libre
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0-libre
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0-libre
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0-libre
...
611	0.0	0.0	0.0	32.0	12.0	15.0	31.0	12.0	0.0	5-auraculaire
612	0.0	0.0	0.0	32.0	12.0	15.0	31.0	12.0	0.0	5-auraculaire
613	0.0	0.0	0.0	13.0	0.0	11.0	10.0	8.0	0.0	5-auraculaire
614	0.0	0.0	0.0	17.0	0.0	0.0	3.0	8.0	0.0	5-auraculaire
615	0.0	0.0	0.0	17.0	0.0	0.0	3.0	0.0	0.0	5-auraculaire

616 rows x 10 columns

Figure 50 : Importation du Dataset.

On divise le dataset en deux sous ensemble un ensemble d'entraînement et un ensemble de test.

```
import seaborn as sns
X = dataset.iloc[:,0:9].values
y = dataset.iloc[:,9].values

from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
y1 = encoder.fit_transform(y)

Y = pd.get_dummies(y1).values

from sklearn.model_selection import train_test_split
X_train,X_test, y_train,y_test = train_test_split(X,Y,test_size=0.3,random_state=0)
```

Figure 51 : Ensemble d'entraînement/Ensemble de tests.

Définition du modèle :

```
model = Sequential()

model.add(Dense(60,input_shape=(9,),activation='relu'))
model.add(Dense(120,activation='relu'))
model.add(Dense(80,activation='relu'))
model.add(Dense(70,activation='relu'))
model.add(Dense(60,activation='relu'))
model.add(Dense(6,activation='softmax'))

model.compile(Adam(lr=0.001),'categorical_crossentropy',metrics=['accuracy'])

model.fit(X_train,y_train,epochs=500)
y_pred = model.predict(X_test)

y_test_class = np.argmax(y_test,axis=1)
y_pred_class = np.argmax(y_pred,axis=1)
```

Figure 52 : Définition du modèle.

Affichage de la précision de la prédiction du modèle :

```
#Accuracy of the predicted values
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test_class,y_pred_class))
print(confusion_matrix(y_test_class,y_pred_class))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	20
1	0.97	1.00	0.99	38
2	1.00	1.00	1.00	20
3	0.92	0.96	0.94	23
4	1.00	0.95	0.97	38
5	0.96	0.96	0.96	46
accuracy			0.97	185
macro avg	0.97	0.98	0.98	185
weighted avg	0.97	0.97	0.97	185

```
[[20 0 0 0 0 0]
 [ 0 38 0 0 0 0]
 [ 0 0 20 0 0 0]
 [ 0 0 0 22 0 1]
 [ 0 0 0 1 36 1]
 [ 0 1 0 1 0 44]]
```

Figure 53 : Précision du modèle.

Comme on le voit sur la figure précédente, la précision de notre RNA a atteint 97%.

Il ne reste qu'une dernière ligne de code, celle de l'exportation du model keras.

```
model.save('emg.h5')
```

Figure 54 : Exportation du Modèle.

4.2.6 Implémentation du RNA sur μ C:

Une fois le modèle exporté il va falloir générer des librairie à partir de ce dernier pour qu'on puisse l'utiliser sur Notre μ c.

Dans l'onglet Pinout & Configuration on sélectionne Additional software, sous le menu Software component class on coche Artificielle Inteligence, on termine la configuration en cochant la case core et en choisissant application Template, dans le menu Paks .

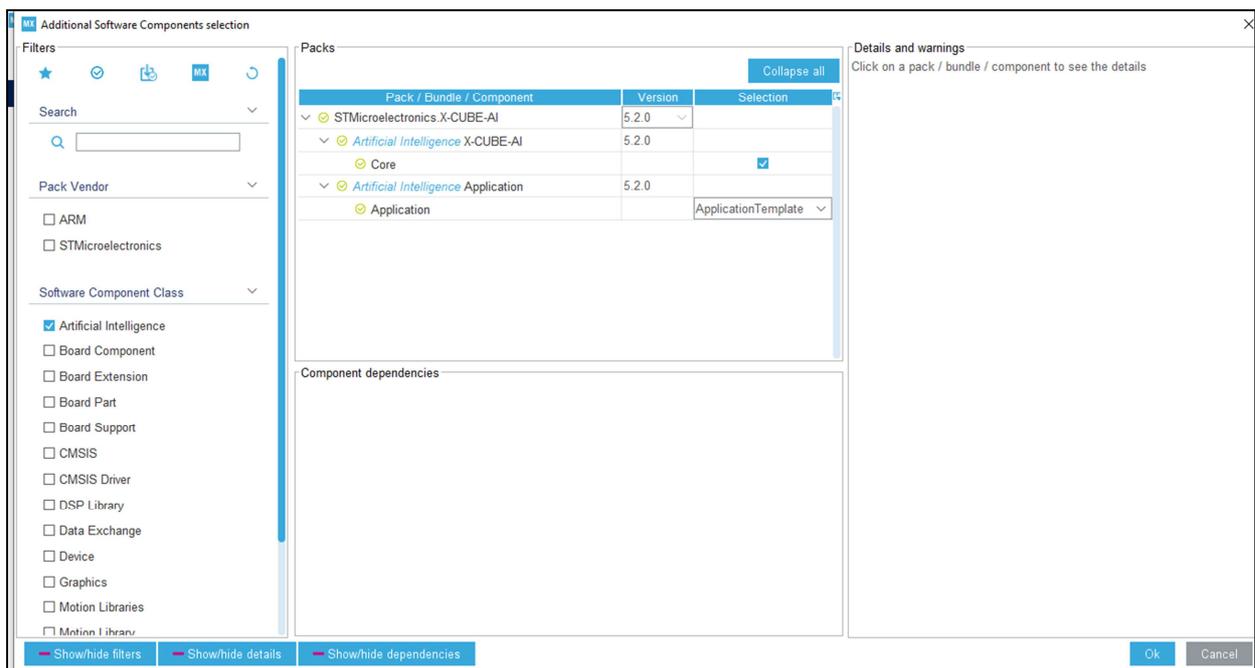


Figure 55 : Configuration du package d'intelligence artificielle.

Dans l'onglet pinout & Configuration sous categories, un nouveau choix s'offre à nous, Additional software, on clique sur STMicrocontroller X.CUBE.AI et on importe notre modèle.

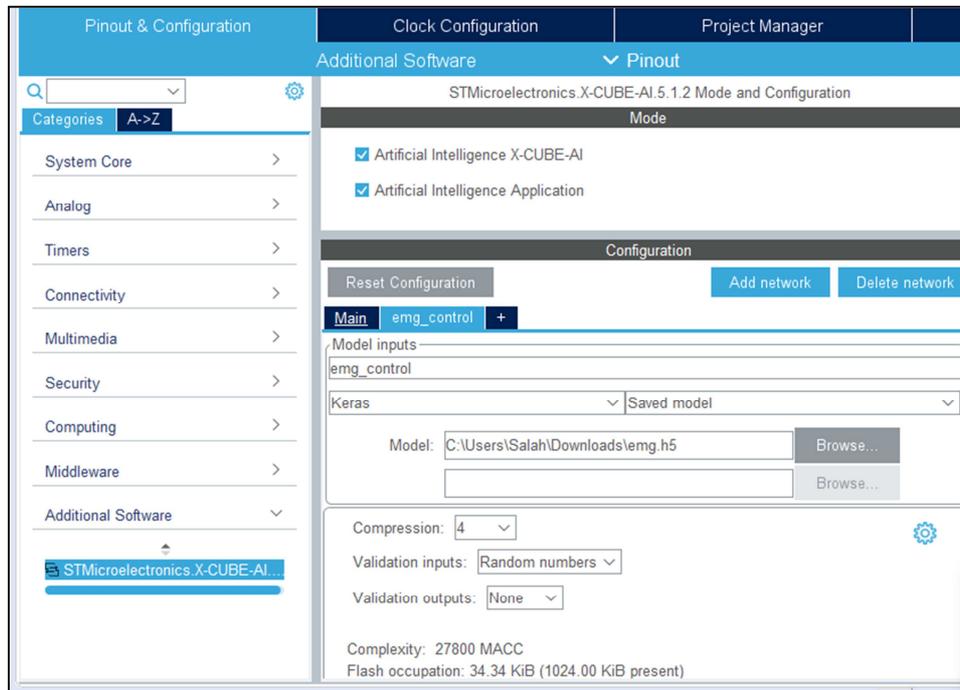


Figure 56 : Importation du modèle keras.

Pour la partie code, Nous avons quatre fonctions générées par STM32Cube.AI: aiInit (), aiRun (), MX_X_CUBE_AI_Init () et MX_X_CUBE_AI_Process (). Les fonctions MX_X_CUBE_AI_Init () et MX_X_CUBE_AI_Process () sont appelées dans le fichier main.c. Nous pouvons laisser les fonctions aiInit (), aiRun (), MX_X_CUBE_AI_Init () telles quelles. Nous devons modifier uniquement la fonction MX_X_CUBE_AI_Process ().

Voici la fonction MX_X_CUBE_AI_Process ():

```

int nb_run = 20;
int res;
char buffer[32] = {0};
AI_ALIGNED(4)
static ai_i8 in_data[AI_EMG_CONTROL_IN_1_SIZE_BYTES];

AI_ALIGNED(4)
static ai_i8 out_data[AI_EMG_CONTROL_OUT_1_SIZE_BYTES];

((ai_float *)in_data)[0] = (ai_float) (inputs[0]);
((ai_float *)in_data)[1] = (ai_float) (inputs[1]);
((ai_float *)in_data)[2] = (ai_float) (inputs[2]);
((ai_float *)in_data)[3] = (ai_float) (inputs[3]);
((ai_float *)in_data)[4] = (ai_float) (inputs[4]);
((ai_float *)in_data)[5] = (ai_float) (inputs[5]);
((ai_float *)in_data)[6] = (ai_float) (inputs[6]);
((ai_float *)in_data)[7] = (ai_float) (inputs[7]);
((ai_float *)in_data)[8] = (ai_float) (inputs[8]);
aiRun(in_data, out_data);

```

Figure 57 : la fonction MX_X_CUBE_AI_Process.

4.2.7 Configuration des servomoteurs:

Ce servomoteur peut aller de 0 à 180 degrés en fonction de la largeur de l'impulsion. Fondamentalement, on doit garder l'impulsion (+5v) élevée pour une période particulière de temps :

- 1 millisecondes et correspond à 0 degrés
- 1,5 milliseconde et correspond à 90 degrés
- 2 millisecondes et correspond à 180 degrés

La période entre deux impulsions doit être de 20ms ou nous pouvons dire que la fréquence de l'impulsion doit être de 50 Hz.

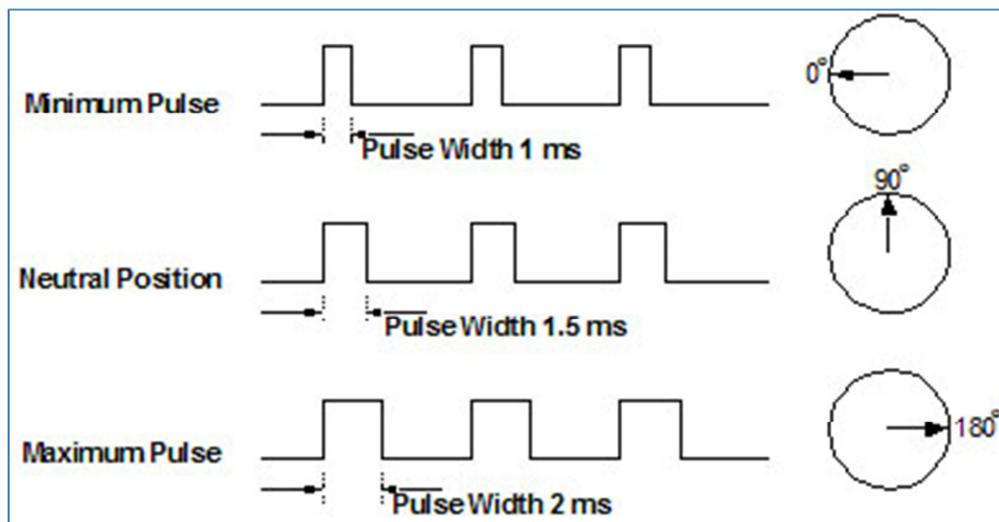


Figure 58 : Représentation schématisques de la largeur d'impulsion des servomoteurs

Le TIM3 et TIM4 ont été utilisés. Ils sont connectés au bus APB1 qui me donne horloge minuterie max jusqu'à 90 MHz. Mais le registre présalaire est seulement 16 bits donc la valeur maximale est de 65535. Dans notre cas l'horloge est réglée de telle sorte que TIM3/TIM4 obtiennent seulement 25 MHz.

Dans l'onglet configuration du sous menu TIM3 (même chose pour TIM4), on définit les paramètres comme suit :

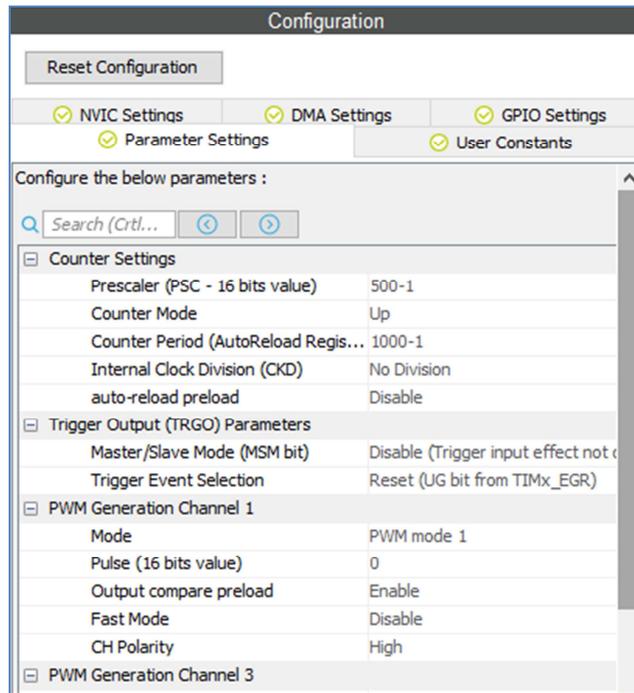


Figure 59 : Configuration des timers.

Exemple de codes :

```
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
```

Figure 60 : Initialisation du Timer.

```
while (1)
{
    htim2.Instance->CCR1 = 25; // duty cycle is .5 ms
    HAL_Delay(2000);
    htim2.Instance->CCR1 = 75; // duty cycle is 1.5 ms
    HAL_Delay(2000);
    htim2.Instance->CCR1 = 125; // duty cycle is 2.5 ms
    HAL_Delay(2000);
}
```

Figure 61 : exemple basique pour faire tourner un servomoteur.

```

        if(outputs[0]==1){
        sprintf(buffer, "libre\n");
        HAL_UART_Transmit(&huart2, buffer, strlen(buffer), 1000);
        HAL_Delay(1000);
        }

        if(outputs[1]==1){
        sprintf(buffer, "pouce\n");
        HAL_UART_Transmit(&huart2, buffer, strlen(buffer), 1000);
        servo1Fermer();
        HAL_Delay(1000);
        servo1Ouvrir();
        HAL_Delay(1000);
        }

        if(outputs[2]==1){
        sprintf(buffer, "index\n");
        HAL_UART_Transmit(&huart2, buffer, strlen(buffer), 1000);
        servo2Ouvrir();
        HAL_Delay(1000);
        servo2Fermer();
        HAL_Delay(1000);
        }

        if(outputs[3]==1){
        sprintf(buffer, "majeur\n");
        HAL_UART_Transmit(&huart2, buffer, strlen(buffer), 1000);
        servo3Ouvrir();
        HAL_Delay(1000);
        servo3Fermer();
        HAL_Delay(1000);
        }

```

Figure 62 : fonction d'actionnement de la main robotique.

Conclusion Générale

En ce qui concerne les objectifs initialement proposés, la plupart d'entre eux ont été atteints, car le système mis en œuvre est capable d'acquérir, de traiter et de classer les signaux EMG pour reproduire les mouvements de la main avec suffisamment de précision; intégrant le système d'acquisition EMG, le système de contrôle basé sur le RNA et l'actionnement manuel robotisé.

En outre, les fonctionnalités sélectionnées dans le domaine temporel ont démontré une faible charge de calcul. Le temps écoulé a été minimisé pendant ces étapes de prétraitement pour préparer les données EMG pour le RNA.

L'optimisation du RNA et de son architecture interne a été ajustée expérimentalement, en fonction du nombre de répétitions de mouvement utilisées dans la formation NN des algorithmes d'entraînement et de performance et du nombre de neurones dans la couche cachée.

Néanmoins, les tests de signaux acquis lors de différentes sessions ont montré des performances de classification faibles et aléatoires. On pense que cette augmentation des erreurs de classification est causée par la nature stochastique des signaux EMG, variant facilement d'une session à l'autre; non seulement en raison de changements physiologiques, mais également si le placement des électrodes n'est pas exactement le même ou encore si les électrodes utilisés ne sont pas les mêmes (même marque).

Par conséquent, la principale conclusion tirée du présent travail est que notre système peut en effet reconnaître des modèles de mouvement de la main basés sur des signaux EMG, et être intégré dans un modèle de système de contrôle pour l'actionnement d'une main robotique imprimable.

Toutefois nous pouvons citer quelques points afin d'améliorer ce projet. La première et la plus importante proposition est de rechercher la satisfaction des exigences en temps réel. Comme les fonctionnalités de calcul associées aux fonctions d'apprentissage. Cependant, la modification de ces paramètres peut affecter la précision de la classification et la généralisation du classificateur. Par conséquent, une analyse plus approfondie doit être menée pour évaluer clairement les paramètres cruciaux impliqués dans la performance du RNA et comment ils affectent le temps écoulé.

Le présent travail nécessite l'acquisition du signal EMG et la formation du RNA pour chaque utilisateur, ce qui implique le placement de neuf électrodes de surface dans les emplacements précis. Cette disposition est non seulement fastidieuse pour l'utilisateur mais peut également produire des changements significatifs dans les signaux enregistrés, réduisant considérablement la capacité de généralisation du classificateur. Par conséquent, le développement d'un système d'enregistrement unifié qui intègre les neuf électrodes et s'adapte parfaitement au sujet, de sorte que les électrodes soient placées correctement et facilement à chaque session, pourrait être tout à fait avantageux pour le système.

Bibliographie :

- [1] Ziegler-Graham K, MacKenzie EJ, Ephraim PL, Travison TG, Brookmeyer R. Estimating the prevalence of limb loss in the United States: 2005 to 2050. *Arch Phys Med Rehabil*. 2008 Mar;89(3):422-9. doi: 10.1016/j.apmr.2007.11.005.
- [2] Maurice LeBlanc, MSME, CP "Give Hope - Give a Hand" - The LN-4 Prosthetic Hand
- [3] Mathieu Ramanarivo, 2018, CONCEPTION ET FABRICATION D'UNE PROTHÈSE MYOÉLECTRIQUE BIOFIDÈLE POUR AMPUTÉS HUMÉRAUX.
- [4] L. Sherwood and A. Lockhart, *Physiologie humaine*. Paris: De Boeck université, 2010.
- [5] anatomie et physiologie humaine, PP1 and WORLD ASSOCIATION of PLANETARIAN HEALTH.
- [6] F. Netter and J. Richer, *Atlas d'anatomie humaine*, 6th ed. 2015.
- [7] Alvaro ALTAMIRANO ALTAMIRANO. (2017). <<EMG Pattern Prediction for Upper Limb Movements Based on Wavelet and Hilbert-Huang Transform>>. Doctorat de l'Université de Lorraine.
- [8] L. Sornmo and P. Laguna. Bioelectrical signal processing in cardiac and neurological applications. Elsevier. Academic Press, 2005.
- [9] R. Barański and A. Kozupa, "Hand Grip-EMG Muscle Response", *Acta Physica Polonica A*, vol. 125, no. 4, pp. A-7-A-10, 2014.
- [10] P. Konrad, *ABC EMG*. Gliwice: Technomex Spółka z o.o., 2007.
- [11] Stephane Huet, Classification de signaux Myoélectrique, janvier 2004.
- [120] M. Elsayed Hussein, 3D Printed Myoelectric Prosthetic Arm. 2014.
- [13] Day, Scott. (2002). Important factors in surface EMG measurement. Bortec Biomedical Ltd Publishers. 1-17.
- [14] Englehart, K., and B. Hudgins. "A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control." *Biomedical Engineering, IEEE Transactions on* 50.7 (July 2003): 848–854.
- [15] J. Coudé, Conception d'un électromyogramme. Chicoutimi: Université du Québec à Chicoutimi, Module d'ingénierie, 2010.
- [16] Luca, Carlo J De. *Surface Electromyography: Detection and Recording*. N.p.: DelSys Incorporated, 2002.

- [17] Oskoei, Mohammadreza & Hu, Huosheng. (2008). Oskoei, M.A.: Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb. *IEEE Transactions on Biomedical Engineering* 55(8), 1956-1965. *IEEE transactions on bio-medical engineering*. 55. 1956-65. 10.1109/TBME.2008.919734.
- [18] M.A. Oskoei and H. Hu. GA-based feature subset selection for myoelectric classification. In *Proc. Int. Conf. Robot. Biomimetics*, pages 1465–1470. Citeseer, 2006.
- [19] H.P. Huang and C.Y. Chen. Development of a myoelectric discrimination system for a multi-degree prosthetic hand. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 3, pages 2392–2397. IEEE, 1999.
- [20] K. Englehart and B. Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, 50(7):848–854, 2003.
- [21] A. Phinyomark, C. Limsakul, and P. Phukpattaranont. A novel feature extraction for robust EMG pattern recognition. *Journal of Computing*, 1(1):71–80, 2009.
- [22] Oskoei, Mohammadreza Asghari, and Huosheng Hu. “Myoelectric Control systems—A Survey.” *Elsevier* 2.4 (Oct. 2007): 275–294.
- [23] Bu, N., M. Okamoto, and T. Tsuji. “A Hybrid Motion Classification Approach for EMG-Based Human–Robot Interfaces Using Bayesian and Neural Networks.” *IEEE Transactions on Robotics* 25.3 (June 2009): 502–511.
- [24] Zecca, M., et al. “Control of Multifunctional Prosthetic Hands by Processing the Electromyographic Signal.” *Critical Reviews™ in Biomedical Engineering* 30.4-6 (2002): 459–485.
- [25] Chan, A.D.C., and K.B. Englehart. “Continuous Myoelectric Control for Powered Prostheses Using Hidden Markov Models.” *IEEE Transactions on Biomedical Engineering* 52.1 (Jan. 2005): 121–124.
- [26] Discovery kit with STM32F407VG MCU User Manual
www.st.com/resource/en/user_manual/dm00039084.pdf