

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

Université de Saida Dr Tahar Moulay

Faculté de Technologie

Département d'électronique



Mémoire de Fin d'Etude

Pour Obtenir le Diplôme du Master

Spécialité : Génie Biomédical

Option: Instrumentation Biomédical

THEME

**Classification des signaux EEG par réseau neuronal
et régression logistique pour commande bio-robot.**

Présenté par : -Mahdi Nourel Houda

- Allam Halima

Devant les jury composé de :

M^R Arbouche OmarPrésident

M^R Mansouri Boualem.....Examineur

M^R Benallou Yassine.....Encadreur

M^R Yahi Youcef..... Co-encadreur

Année Universitaire: 2019-2020

REMERCIEMENTS

Avant d'entamer ce projet de fin d'étude, nous remercions d'abord "ALLAH" le tout puissant de nous avoir donné la volonté pour accomplir ce modeste travail.

Nous tenons à exprimer nos vifs remerciements notre encadreur Monsieur "Benallou Yassine" nous avoir encadré durant notre projet de fin d'études, sa patience et le nombreux conseil qu'il nous a prodigué.

Nous adressons toute notre gratitude et manifester notre très sincère reconnaissance à "Mr Yahi" pour l'attention qu'il a porté à notre travail. Nous avons découvert grâce à lui le monde de la recherche dans les meilleures conditions.

Nous tenons à remercier Mr "Arbouche Omar" d'avoir accepté de présider les jury.

Nos remerciements à Mr "Manssour" d'avoir accepté d'examiner ce travail.

Nous les remercions pour le temps et de l'attention qu'ils ont en fin, à tous ceux qui nous ont aidés de près ou de loin pour la réalisation de projet de fin d'étude.

Dédicace :

A L' aide de dieu tout puissant on a pu achever ce modeste travail que Je dédie :

A mes chers parents que le dieu les protège, pour leurs encouragements,

leur soutien moral, spirituel et leur tolérance durant toutes mes années d'études.

A mes sœurs Nabila , Imene.

A mes frères AbdelHAFFID, Mohamed Amine.

A ma chère tante, Najat.

A toutes la famille Mahdi grands et petits. .

Comme je dédie ce mémoire à mes amis très chers Fatima, Imene .

A mon binôme Halima.

*A tous ceux qui m'ont aidé de loin ou de près, Du profond de mon cœur, je dédie ce travail à
tous ceux qui me sont chers.*

NOURELHONDA

Dédicace

Ce projet fin d'étude est dédié à :

A ma mère "Messaoudi fatima" : Ton amour et ta disponibilité infinis me guident et me soutiennent dans ma vie quotidienne. Je souhaite que ce travail traduise le résultat de tes prières et de tes bénédictions de tous les jours. Je voudrais qu'il soit le témoignage de mon amour, et de ma profonde reconnaissance. Je prie pour que tu vives le plus longtemps possible pour voir grandir ta fille.

A mon père, "Allam kadda" : Je dédie cette thèse en guise de récompense, de tes peines, de tes privations et de ta patience dans mon éducation. J'ai encore en mémoire cette peur que tu éprouvais à la veille de tous mes examens de fin d'année. Et puisque je ne pourrais jamais te rendre ce que tu m'as donné, je voudrais que cette thèse soit le symbole de ta confiance. Puisse tu avoir une longévité heureuse et bénie.

Mon cœur, ma vie et mon bonheur : ma grand mère « mebaraka » et mon grand père « zendari ».

A mon frère et mes sœurs : Abd el Kader, Fatiha, Mokhtaria et son mari Krime, Imen, Mebaraka, Zineb, Khadra. Je souhaite que cette œuvre consolide d'avantage notre fraternité.

♥ *Ange de la famille : "SOUHANE ARIJ" ♥*

A mon soutien moral et source de joie et de bonheur, mon fiancé « Ouaggaad youcef » pour l'encouragement et l'aide qu'il m'a toujours accordé.

A mes oncles et tantes (Aicha, Fatiha, Setti), à mes cousins et cousines : Je renouvelle ma gratitude pour les encouragements et la solidarité.

A toutes mes amies : Tawes, Aicha R, Houda, Amina, Saida et Abd elkhalek, Je formule le vœu que notre amitié soit sans usure. Je vous réitère mon indéfectible attachement et ma reconnaissance.

A la famille ALLAM: Je n'ai pas de mot pour exprimer mes sentiments. Vous m'avez acceptée comme votre enfant et je vous en suis reconnaissant.

Halima

TABLES DES MATIERES	
Remerciement	i
Dédicace	ii
Tables Des Matieres	A
Listes Des Figures	D
Liste des tableaux	H
Résumée	I
Introduction générale	01
Chapitre 1 : Anatomie et physiologie de corps humain	
1-1 Introduction	04
1-2 anatomie et physiologie de corps humain	04
1-2-1 Organisation du système nerveux	04
1-2-2 anatomie cérébrale	04
1-2-3 Le système nerveux général	06
1-2-3-1 Organisation système nerveux	06
1-2-3-2 le système nerveux somatique	09
1-2-3-3 Systèmes sensoriels et moteurs (système nerveux périphérique)	10
1-2-3-4 Le système nerveux périphérique (SNP)	10
1-2-3-5 Le système d'action musculaire (la motricité musculaire)	11
1-2-3-6Le fonctionnement de La motricité musculaire	12
1-2-4les types de muscles	13
1-2-4-1 les muscles viscéral (muscle lisses)	13
1-2-4-2 le muscle strié cardiaque	13
1-2-4-3 les muscles squelettiques et les muscles latéral	14
1-2-4-4 la composition et organisation de muscle	14
1-2-4-4-1 les cellules de muscles squelettiques	14
1-2-4-4-2 l'anatomie de muscle squelettique	15
1-2-4-4-3 les fibres musculaires squelettique	17
1-2-4-4-4 le sarcomère	18
1-2-4-5 La jonction musculaire	19
1-2-5 Potentiel de membrane	21
1-2-5-1Membrane de cellules électriques actives	22

1-2-5-2 L'état électrique de la membrane cellulaire	26
1-2-5-3 Le potentiel d'action	27
1-2-5-4 Propagation de potentiel d'action	30
1-3 la conclusion générale	31
Chapitre 2 :l'instrumentation de détection et traitement de signal physiologique	
2-1 Introduction	33
2-2 Contexte et motivation	33
2-3 Système BCI	35
2-4 Le matériel	36
a) EEG (EMOTIVEPOC)	36
b) Carte Arduino	38
c) Le module Xbee	40
2-5 Sélection du logiciel	40
a) Matlab	40
b) Importateur EPOC Simulink EEG	40
2-6 Optimisation et mise en œuvre de la machine à vecteur de support	41
2-7 Mathématique du classificateur SVM	42
a) Concept de base de SVM	42
b) Marge et optimisation	43
c) Fonctions du noyau et classification non linéaire	45
2-8 Bibliothèque LibSVM et optimisation des paramètres	45
a) Bibliothèque LibSVM	45
b) Optimisation des paramètres C et γ	46
2-9 Le Fonctionnement d'entrée de SVM pour un système BCI basé sur EEG	47
2-10 Programmes auto-développés utiles pour le système BCI basé sur EEG	49
2-10-1 Programme de collecte de données dans le domaine temporel EEG	49
2-10-2 Programme de conversion des données EEG du domaine temporel vers le domaine fréquentiel	52
2-10-3 Programme de création de modèle SVM	54
a) Analyse d'ensemencement	55
b) Faire un modèle de prédiction	62

2-11- Programmes de contrôle du robot	64
a) Mode d'entraînement	65
2-12-Analyse des données EEG brutes et des fonctionnalités en tant qu'entrées SVM	72
2-12-1-Règle d'enregistrement des données EEG	72
2-12-2-Méthode d'enregistrement des données EEG	72
2-12-3-Taille des fonctionnalités pour l'entrée SVM	73
Chapitre3 : Résultats et interprétations	
3-1-Introduction	76
3-2-Résultats des simulations des modèles SVM	76
3-2-1-Classification des données des distributions normales	76
3-2-2-Classification de 2 données directionnelles	78
3-2-3-Classification de 3 données directionnelles	80
3-2-4-Classification de 4 données directionnelles	82
3-2-5-Classification de 2 données alphabétiques	84
3-2-6-Classification de 4 données directionnelles et 2 données alphabétiques	85
3-3-Implémentation et test du système de contrôle du robot	86
3-3-1-Schéma fonctionnel du mouvement du robot et système de contrôle d'affichage alphabétique	86
3-3-2-Configuration de Xbee	89
3-3-3-Configuration du mode série	90
3-3-4-Implémentation du contrôle du robot	91
a) Côté émetteur	91
b) Côté récepteur	92
3-4-conclusion générale	93

Listes des figures :

Chapitre 1 : Anatomie et physiologie de corps humain

Figure 1 : système nerveux autonome [1]

Figure 2 : L'image descriptive d'interconnexion de system nerveux [2]

Figure 3 : l'image descriptive de l'intérieur de nerf CNS [3]

Figure 4 : le fonctionnement de system nerveux périphérique [4]

Figure 5 : transmission de signal E/S cérébrale [5]

Figure 6: cerveau et colonne vertébrale du système nerveux central [6]

Figure 7 :Muscle viscéral(muscle lisse)[7]

Figure 8 : Les trois couches de tissu conjonctif [8]

Figure 9 :Fibre musculaire.[9]

Figure 10 :Le sarcomère, la région d'une ligne Z à la suivante, est l'unité fonctionnelle d'une fibre musculaire squelettique.[10]

Figure 11 : Plaque d'extrémité et innervation du moteur.[11]

Figure 12 : Protéines membranaires et transmembranaires.[12]

Figure 13 : Canaux à porte de ligand.[13]

Figure 14 : Canaux voltage-dépendants.[14]

Figure 15 : Canaux de fuite..[15]

Figure 16 : Mesure de la charge à travers une membrane avec un voltmètre.[16]

Figure 17 : Graphique du potentiel d'action.[17]

Figure 18 : Étapes d'un potentiel d'action. [18]

Chapitre 2 :l'instrumentation de captation et traitement de signal physiologique

Figure 1 :EEG EMOTIV EPOC

Figure 2 : 10-10 Système et emplacement d'électrodes EEG

Figure 3 : SDK du panneau de commande EEG EPOC

Figure 4: ArduinoUno R3 board

Figure 5: Antenne Xbee

Figure 6 :Serveur de signaux simulink EPOC

Figure 7 :Un modèle simulink simple pour la collecte de données EEG

Figure 8 :Un hyperplan simple divisant un ensemble de données

Figure 9 :Le programme demande à l'utilisateur de saisir des informations générales

Figure 10 :Le programme demande à l'utilisateur de saisir les noms des classes

Figure 11 : Le programme demande à l'utilisateur d'utiliser l'EPOC et de réfléchir à la classe spécifique

Figure 12 :Le programme vérifie le sentiment de l'utilisateur quant à la qualité des données

Figure 13 : Données EEG du domaine temporel dans le répertoire de travail

Figure 14 : Insertion du nombre de classes à convertir

Figure 15 :Insertion du nom des dossiers de données du domaine temporel

Figure 16 : Choix du type de données à convertir en

Figure 17 : De nouveaux répertoires sont créés

Figure 18 :Le programme demande à l'utilisateur de choisir entre 2 options

Figure 19 : Le nom du fichier texte

Figure 20 : Insérez la borne inférieure de la sortie

Figure 21 : L'exemple d'un fichier de sortie

Figure 22 : Entrez le nombre de dossiers, la taille des données et le nombre de couches

Figure 23 : la sortie couches par couches

Figure 24 : Entrez la plage de canaux

Figure 25 : Plage de fréquences d'entrée

Figure 26 : Processus de formation SVM

Figure 27 : Calque = 1 dans la boîte de dialogue de gauche, Calque = 4 dans la boîte de dialogue de droite

Figure 28 : Choix entre 3 options

Figure 29 : Saisie des meilleurs C et gamma

Figure 30 : Enregistrement du modèle final dans un emplacement

Figure 31 : Choix du mode d'exécution du programme

Figure 32 : Choix d'un modèle

Figure 33 : Saisie d'informations pour le mode pratique du modèle

Figure 34 : type de rétroaction d'entrée

Figure 35 : collecte de la référence

Figure 36 : Collecte des données de référence du coefficient de corrélation (à gauche) -Référence des données de reprise (à droite)

Figure 37 : Informer l'utilisateur que les données EEG avec un résultat de détection correct sont prises

Figure 38 : Commencer la formation pour une classe

Figure 39 :Commencez à prendre un morceau de données

Figure 40 : Reprendre les données

Figure 41 : Résultat final pour le mode d'entraînement b) Mode temps réel

Figure 42 : Indiquant le type de signal envoyé au robot

Chapitre 3 : Résultat et interprétation

Figure 1 :Système de contrôle du mouvement du robot et de l'affichage alphabétique

Figure 2 : Configuration des Récepteurs et émetteurs du module Xbee

Figure 3 :La connexion de l'émetteur Arduino et Xbee et LCD

Figure 4 : la voiture robotique émetteur

Liste des tableaux :

Chapitre 3 : Résultat et interprétation

Tableau 1 : Résultat de la classification des données de la distribution normale

Tableau 2 : Résultat de la classification gauche vs droite avec une plage de fréquences de 4 à 30 Hz

Tableau 3 : Performances individuelles de 2 directions gauche vs droite 4-30Hz

Tableau 4 : Résultat de classification gauche vs droite vs avant avec plage de fréquences 4-30 Hz

Tableau 5 : Performances individuelles de 3 directions gauche vs droite vs avant 4-30Hz

Tableau 6 : Résultat de classification gauche vs droite vs avant vs arrière avec plage de fréquences 4-30 Hz
Tableau 7 Performances individuelles de 4 directions gauche vs droite vs avant vs arrière 4-30Hz

Tableau 7 : Performances individuelles de 4 directions gauche vs droite vs avant vs arrière 4-30Hz

Tableau 8 : Résultat de classification A vs B avec plage de fréquences 4-30 Hz

Tableau 9 : Performances individuelles de 2 lettres A vs B 4-30Hz

Tableau 10 : Résultat de la classification des données directionnelles et alphabétiques pour la première graine de 0 à 64 Hz

Tableau 11 : Séquences de signaux pour différents caractères

Résumée :

Notre mémoire de fin d'étude présente la conception et l'expérimentation d'un système capable de détecter la pensée humaine comme les directions de conduite et les lettres en utilisant les signaux d'ondes cérébrales détectés par l'électroencéphalogramme (EEG) et un algorithme d'apprentissage automatique appelé machine à vecteurs du support (support vector machine SVM). Le projet est motivé et peut donner des solutions à des maladies comme la maladie de la sclérose latérale amyotrophique (SLA), qui fait que les patients perdent sérieusement la mobilité et les capacités d'expression. Le système développé dans ce mémoire comporte trois étapes principales. Tout d'abord, avec le casque EPOC d'EmotivCompany, ou l'utilisateur peut enregistrer les signaux EEG lorsqu'il pense à une direction ou à une lettre et d'enregistrer également les données dans un ordinateur sans fil. Ensuite, une grande quantité de données EEG transportant les informations de différentes instructions et lettres d'utilisateur qui forment la manière exhaustive du modèle de classification SVM.

Enfin, le modèle SVM bien formé sera utilisé pour détecter toute nouvelle réflexion sur les directions et les lettres de l'utilisateur. Les résultats de détection du modèle SVM peuvent être transmis sans fil à une voiture robotique comportant un écran LCD construit avec des microcontrôleurs arduino pour contrôler ses mouvements ainsi que l'affichage graphique sur LCD. L'une des grandes applications potentielles du système développé est de créer un système avancé de fauteuil roulant à contrôle cérébral avec écran LCD pour aider les patients atteints de SLA dans leur mobilité et leurs communications quotidiennes.

Introduction général

Introduction général :

Introduction générale :

Notre cerveau est constitué de plusieurs milliards des cellules qui échangent en permanence des informations, ces échanges provoquent l'émission des courants électriques faibles (les ondes cérébrales) que l'on peut capter grâce à l'électroencéphalogramme. L'EEG est un examen médical simple et sans danger, il enregistre certains aspects de l'activité cérébrale. Pour procéder à un EEG, on place à divers endroits sur le cuir chevelu un appareil qui mesure la différence de potentiel entre divers lieux du cortex, et on obtient des tracés appelés les ondes cérébrales.

Les signaux EEG peuvent être utilisés efficacement pour diverses applications telles que la reconnaissance des émotions, les interfaces cerveau-ordinateur (BCI) etc. L'une des applications les plus importantes d'analyse des signaux EEG est son utilisation en neurosciences pour diagnostiquer les maladies et troubles cérébraux. La crise d'épilepsie est l'un des troubles neurologiques les plus courants dans le monde. Sa détection est généralement effectuée par les médecins à l'aide d'un balayage visuel des signaux EEG, ce qui prend du temps et peut être inexact.

L'interface cerveau-ordinateur (BCI) est en train de devenir un domaine essentiel de l'ingénierie biomédicale et de la technologie qui utilise les technologies EEG pour fournir la technologie des dispositifs d'assistance (AT) aux humains. Par conséquent, cet article présente les résultats de l'analyse des signaux EEG de divers états cognitifs humains pour extraire les caractéristiques EEG appropriées qui peuvent être utilisées pour contrôler les appareils BCI qui peuvent être utilisés par des personnes handicapées, paralysées aussi elle est utilisable par les neurologues pour faciliter l'interprétation et le diagnostic des différentes maladies cérébrales. Les caractéristiques d'EEG en termes de densité spectrale de puissance, d'écart type et d'entropie sont sélectionnées et étudiées à partir de quelques exercices mentaux comme résoudre rapidement les mathématiques et se détendre (ne rien faire ou bien la position la plus stable majeure). La machine à vecteur de support (SVM) et le classificateur K-Nearest Neighbours (k-NN) qu'elle étés

Introduction général :

utiliser dans le projet de Yahi et ALL en 2019 . Parmi toutes ces caractéristiques, la meilleure précision a été obtenue par la densité spectrale de puissance. Les précisions de cette fonction sont respectivement de 93%, avec WT, SVM et K-NN. Enfin, l'algorithme de traduction sera construit à l'aide de fonctions EEG sélectionnées et classées pour développer notre système BCI .

Chapitre I

Chapitre 1 : Anatomie et physiologie de corps humain**1-1 Introduction :**

L'anatomie et la physiologie sont les disciplines complémentaires de la biologie, la science qui étudie les organes vivants. L'anatomie humaine étudie la structure du corps et les relations entre les structures du corps. La physiologie humaine s'intéresse au fonctionnement des différentes parties du corps. De manière générale, c'est la structure qui détermine la fonction.

1-2 anatomie et physiologie de corps humain :**1-2-1 Organisation du système nerveux :**

Le système nerveux commande les muscles, contrôle le fonctionnement de tous les organes et renseigne sur le monde extérieur via les informations sensorielles. Il est le siège des facultés intellectuelles. Du point de vue anatomique, on distingue : le système nerveux central (SNC) ou névraxe et le système nerveux périphérique (SNP).

1-2-2 anatomie cérébrale:

Anatomie du cerveau Vue d'ensemble Le cerveau est un organe étonnant de trois parties qui contrôle toutes les fonctions du corps, interprète les informations du monde extérieur et représente l'essence de l'esprit et de l'âme. L'intelligence, la créativité, les émotions et la mémoire sont quelques-unes des nombreuses choses régies par le cerveau. Protégé à l'intérieur du crâne, le cerveau est composé du cerveau, du cervelet et du tronc cérébral. Le cerveau reçoit des informations à travers nos cinq sens: la vue, l'odorat, le toucher, le goût et l'audition, souvent plusieurs à la fois. Il assemble les messages d'une manière qui a du sens pour nous et peut stocker cette information dans notre mémoire. Le cerveau contrôle nos pensées, notre mémoire et notre parole, les mouvements des bras et des jambes et la fonction de nombreux organes de notre corps. Le système nerveux central (SNC) est composé du cerveau et de la moelle épinière. Le système nerveux périphérique (SNP) est composé de

nerfs spinaux se ramifiant de la moelle épinière et de nerfs crâniens allant du cerveau. Le cerveau est composé du Cerebrum, du cervelet et du tronc cérébral.

-Cerebrum: est la plus grande partie du cerveau et se compose d'hémisphères droit et gauche. Il remplit des fonctions plus élevées, telles qu'interprétation du toucher, de la vision et de l'audition, ainsi que de la parole, du raisonnement, des émotions, de l'apprentissage et du contrôle précis du mouvement.

-Cervelet: est situé sous le cerveau. Sa fonction est de coordonner les mouvements musculaires, de maintenir la posture et d'équilibrer.

-Tronc cérébral: sert de centre de relais reliant le cerveau et le cervelet à la moelle épinière. Il exécute de nombreuses fonctions automatiques telles que la respiration, la fréquence cardiaque, la température corporelle, les cycles de réveil et de sommeil, la digestion, les éternuements, la toux, les vomissements et la déglutition.

-Cerveau droit - cerveau gauche :Le cerveau est divisé en deux moitiés: les hémisphères droit et gauche . Ils sont reliés par un faisceau de fibres appelé corpus callosum qui transmet les messages d'un côté à l'autre. Chaque hémisphère contrôle le côté opposé du corps. Si un accident vasculaire cérébral se produit du côté droit du cerveau, votre bras ou jambe gauche peut être faible ou paralysé.

Toutes les fonctions des hémisphères ne sont pas partagées. En général, l'hémisphère gauche contrôle la parole, la compréhension, l'arithmétique et l'écriture. L'hémisphère droit contrôle la créativité, la capacité spatiale, les compétences artistiques et musicales. L'hémisphère gauche est le langage dominant chez environ 92% des personnes.

1-2-3 Le système nerveux général:

1-2-3-1 Organisation système nerveux :

Les nerfs contiennent des neurones du système nerveux autonome ou nerveux (ANS), qui relie le système nerveux central (SNC) aux viscères.

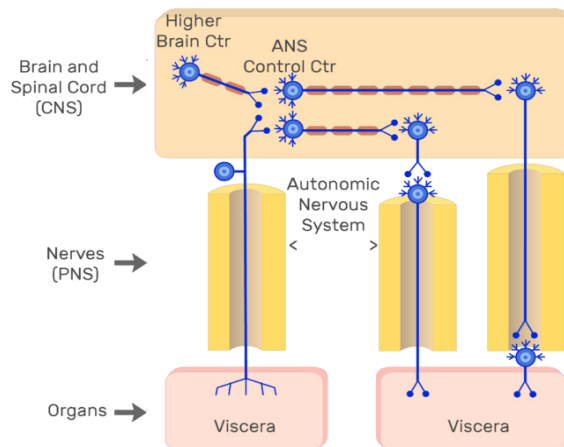


Figure I-1 : système nerveux autonome [1]

Principaux organes et divisions du système nerveux :

-Le système nerveux est divisé de façon anatomique et fonctionnelle entre le système nerveux central et le système nerveux périphérique.

-Le système nerveux central (SNC) constitue le cerveau descriptif de l'image et le cordon rachidien, qui servent de centre d'évaluation et de contrôle du corps.

Ces organes vitaux sont entourés et protégés par les os du crâne et de la colonne vertébrale. Le système nerveux périphérique (SNP) est constitué d'un réseau de descripteurs d'image qui s'étendent dans tout le corps.

Les nerfs se connectent au système nerveux central par le biais de 12 paires de nerfs crâniens avec description d'image et de 31 paires de nerfs rachidiens avec description.

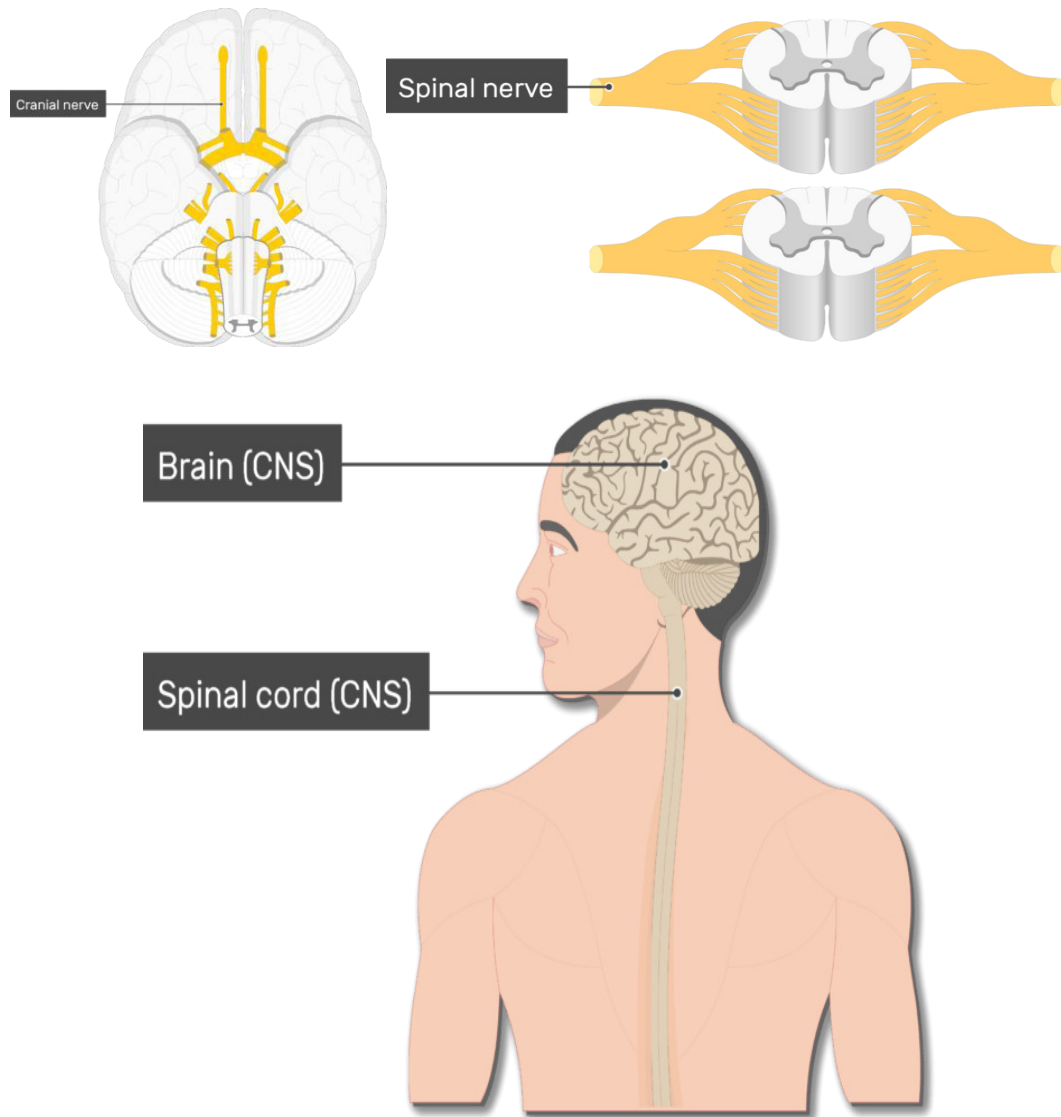


Figure I-2 :L'image descriptive d'interconnexion de system nerveux [2]

À l'intérieur des nerfs sont représentés les processus de description d'image de nombreuses cellules nerveuses (ou neurones).

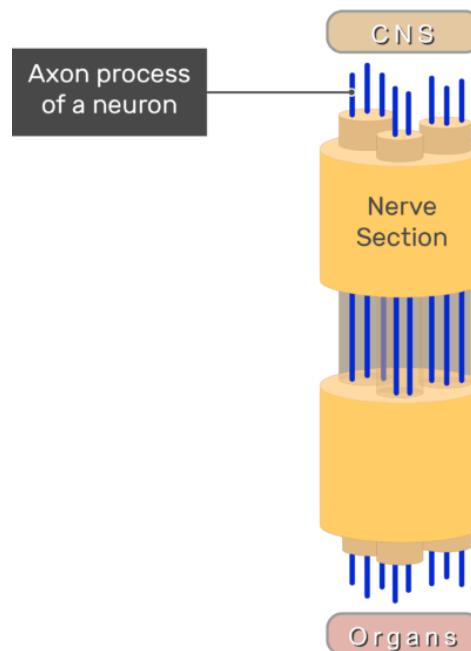


Figure I-3 : l'image descriptive de l'intérieur de nerf CNS [3]

Les axones filiformes parcourent la longueur des nerfs et sont regroupés par tissu conjonctif en faisceaux cylindriques appelés faisceaux.

Les axones du neurone moteur transmettent les impulsions du système nerveux central aux organes périphériques. La plupart des nerfs sont mélangés avec une description d'image, ce qui signifie qu'ils contiennent à la fois des neurones moteurs et sensoriels. Sur la base de leurs fonctions, les neurones des nerfs divisent le système nerveux périphérique en systèmes nerveux somatique et autonome.

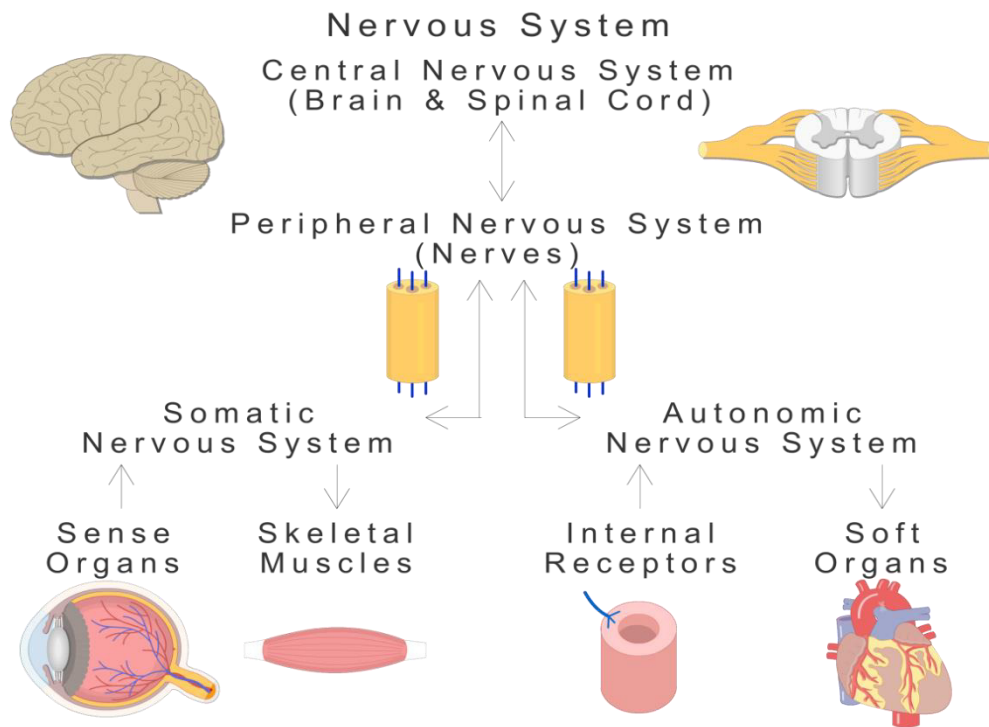


Figure I-4 : le fonctionnement de system nerveux périphérique [4]

Les neurones somatiques portent des signaux qui permettent au SNC de surveiller consciemment l'environnement externe et de contrôler les mouvements du corps. En revanche, les neurones autonomes transmettent des signaux permettant au système nerveux central de réguler inconsciemment l'environnement interne du corps et de contrôler l'activité des organes mous (viscères).

1-2-3-2 le système nerveux somatique :

Les signaux de douleur sont transmis au cerveau par une voie impliquant deux voies du système nerveux (NS). La première est une voie NS descriptive dans la figure ..et la seconde une voie NS centrale.

Lorsque la description de la peau est induite par un objet pointu, comme une punaise, les dommages résultants sont détectés par les récepteurs de la douleur (ou nocicepteurs).

Des impulsions électrochimiques (ou potentiels d'action) sont générées dans les neurones sensoriels somatiques.

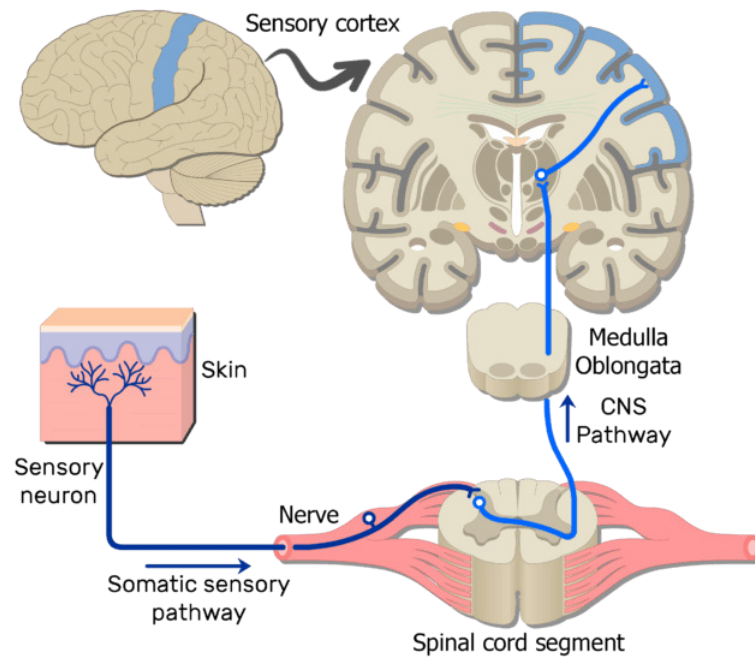


Figure I-5 : transmission de signal E/S cérébrale [5]

Alors ce système baser sur des biocapteurs (sensoriel part)

1-2-3-3 Systèmes sensoriels et moteurs (système nerveux périphérique) :

Le système nerveux périphérique (SNP) c'est le système responsable de potentiels d'action entre le système nerveux central et les organes périphériques.

1-2-3-4 Le système nerveux périphérique (SNP) :

En fonction de la direction dans laquelle, ils conduisent les potentiels d'action et même les neurones divisent le SNP en divisions descriptives (afférentes).

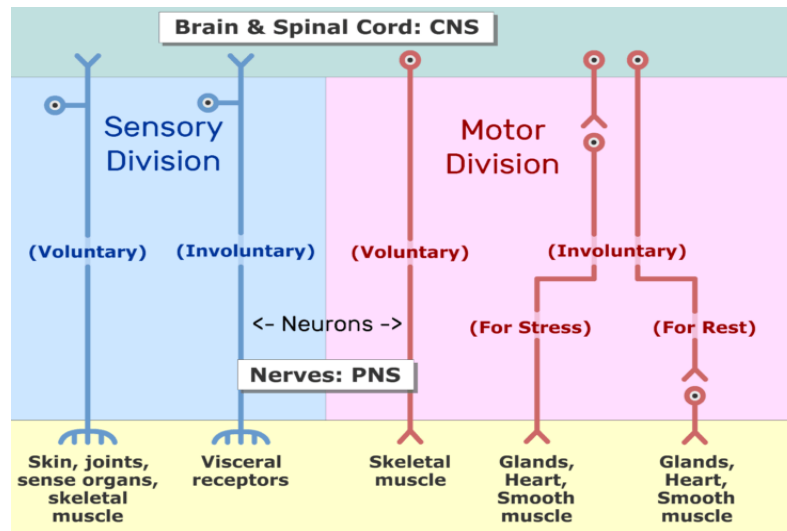


Figure I-6: cerveau et colonne vertébrale du système nerveux central [6]

Les neurones de la division sensorielle conduisent les potentiels d'action vers le SNC. Les deux divisions ont des branches qui permettent au système nerveux central de surveiller et de contrôler volontairement (consciemment) ou involontairement (inconsciemment) certains tissus et organes.

Alor comment contrôler volontairement les muscles squelettiques ?

1-2-3-5 Le système d'action musculaire (la motricité musculaire) :

Le système d'action basé sur 6 parties essentielles sont :

1-Cortex de mouvement primaire

2-Capsule interne

3-motoneurone supérieur

4-Pyramide médullaire

5-Tractus corticaux-spinal de la moelle épinière

6- motoneurone inférieur (qui baser sur 3 parties essentielles)

–corne ventral

–Nerf spinal-Effecteurs muscle squelettique

1-2-3-6 Le fonctionnement de La motricité musculaire :

On appelle un muscle fléchisseur c'est le muscle qui commande l'articulation par un mouvement rapproche les deux bras du membre (exemple : l'opposition à une extension de biceps rapproche l'avons bras vers le bras.) Le système musculaire est responsable du mouvement du corps humain. Environ 700 muscles nommés sont fixés aux os du système squelettique et représentent environ la moitié du poids corporel d'une personne. Chacun de ces muscles est un organe distinct constitué de tissu musculaire squelettique, de vaisseaux sanguins, de tendons et de nerfs. Les tissus musculaires se trouvent également à l'intérieur du cœur, des organes digestifs et des vaisseaux sanguins. Dans ces organes, les muscles servent à déplacer des substances dans tout le corps .Il existe trois types de tissus musculaires: viscérale (lisse), cardiaque et squelettique.

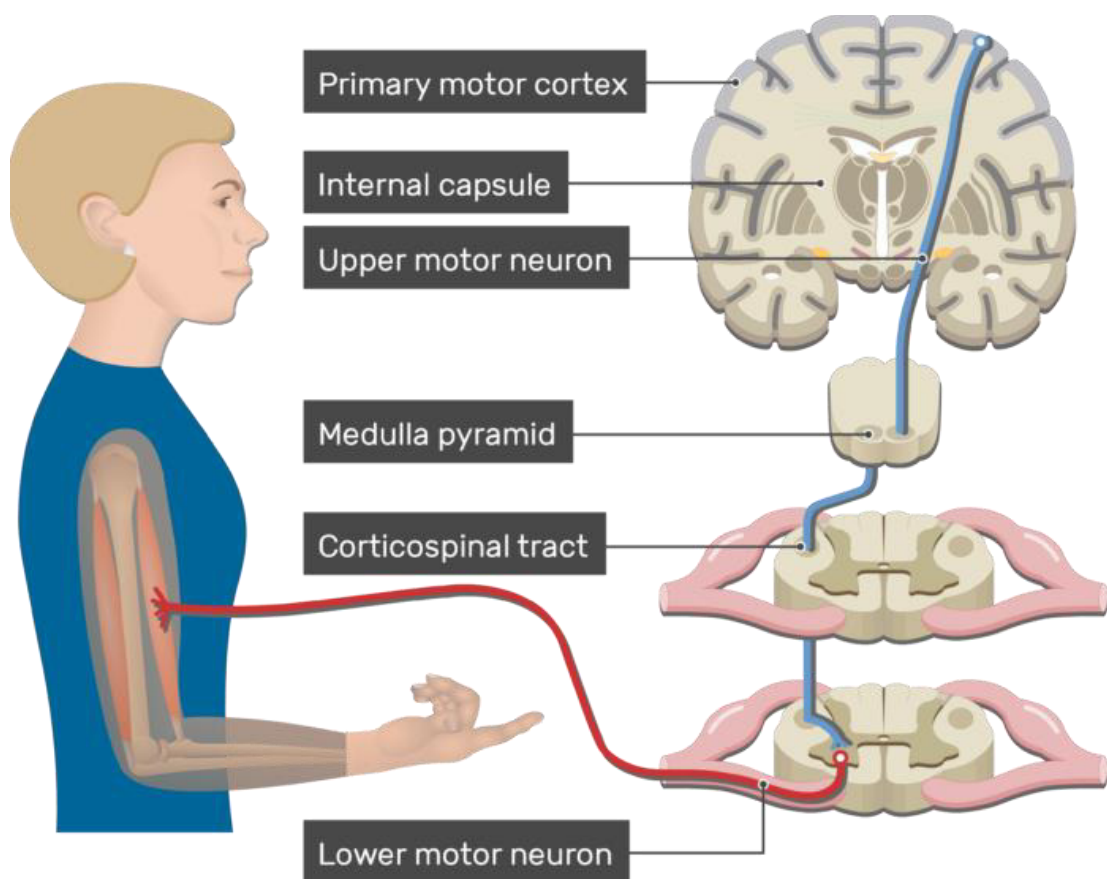


Figure I-7 : Muscle viscéral (muscle lisse)[7] .

Le muscle viscéral se trouve à l'intérieur d'organes tels que l'estomac, les intestins et les vaisseaux sanguins. Le muscle viscéral, le plus faible de tous

les tissus musculaires, force les organes à se contracter pour faire circuler des substances dans l'organe. Parce que le muscle viscéral est contrôlé par la partie inconsciente du cerveau, on l'appelle muscle involontaire - il ne peut pas être directement contrôlé par l'esprit conscient. Le terme «muscle lisse» est souvent utilisé pour décrire un muscle viscéral car il a un aspect très lisse et uniforme lorsqu'il est examiné au microscope. Le fonctionnement de ce type de muscle involontairement soit la contraction ou bien le relâchement musculaire.

1-2-4 les types de muscles :

1-2-4-1 les muscles viscéral (muscle lisses)

Le muscle lisse contient des cellules en forme de fuseau de 50 à 250 µm de long et de 5 à 10 µm de diamètre. Ces cellules possèdent un seul noyau central. Les filaments épais (myosine) et minces (actine) entourent le noyau et la majeure partie du cytoplasme.

1-2-4-2 le muscle strié cardiaque :

Présent uniquement dans le cœur, le muscle cardiaque est responsable du pompage du sang dans tout le corps. Le tissu musculaire cardiaque ne peut pas être contrôlé consciemment, il s'agit donc d'un muscle involontaire. Alors que les hormones et les signaux du cerveau ajustent le taux de contraction, le muscle cardiaque se stimule pour se contracter. Le stimulateur cardiaque naturel du cœur est constitué de tissu musculaire cardiaque qui stimule la contraction d'autres cellules du muscle cardiaque. En raison de son auto-stimulation, le muscle cardiaque est considéré comme étant auto-rythmique ou intrinsèquement contrôlé.

Les cellules du tissu musculaire cardiaque sont striées, c'est-à-dire qu'elles semblent présenter des bandes claires et sombres lorsqu'elles sont examinées au microscope optique. La disposition des fibres de protéines à l'intérieur des cellules provoque ces bandes claires et sombres. Les annulations indiquent qu'une cellule musculaire est très forte, contrairement aux muscles viscéraux.

Les cellules du muscle cardiaque sont des cellules en forme de X ou de Y ramifiées étroitement liées entre elles par des jonctions spéciales appelées disques intercalés. Les disques intercalés sont constitués de projections en

forme de doigts provenant de deux cellules voisines qui s'emboîtent et créent un lien fort entre les cellules. La structure ramifiée et les disques intercalés permettent aux cellules musculaires de résister aux pressions artérielles élevées et à la pression exercée par le pompage du sang tout au long de la vie. Ces caractéristiques aident également à diffuser rapidement les signaux électrochimiques d'une cellule à l'autre, de sorte que le cœur puisse battre comme une unité.

1-2-4-3 les muscles squelettiques et les muscles latéral :

Le muscle squelettique : est le seul tissu musculaire volontaire dans le corps humain - il est contrôlé consciemment. Chaque action physique qu'une personne accomplit consciemment (par exemple, parler, marcher ou écrire) nécessite un muscle squelettique. Le muscle squelettique a pour fonction de se contracter pour rapprocher les parties du corps de l'os auquel le muscle est attaché. La plupart des muscles squelettiques sont attachés à deux os situés de part et d'autre d'une articulation. Le muscle sert donc à rapprocher les parties de ces os.

Les cellules des muscles squelettiques : se forment lorsque de nombreuses cellules pro-génitrices plus petites s'assemblent pour former de longues fibres droites multi-nucléées. Striées comme le muscle cardiaque, ces fibres musculaires squelettiques sont très fortes. Le muscle squelettique tire son nom du fait que ces muscles se connectent toujours au squelette à au moins un endroit.

1-2-4-4 la composition et organisation de muscle :

1-2-4-4-1 les cellules de muscles squelettiques :

Chaque cellule musculaire composant le muscle strié squelettique contient au minimum deux et jusqu'à 500 noyaux : ce sont des grandes cellules (ou fibres) musculaires multi-nucléées. Les autres constituants du muscle sont des nerfs, des vaisseaux sanguins et du tissu conjonctif. Parmi les muscles striés squelettiques les plus connus, on peut citer les biceps, les quadriceps ou les abdominaux.

1-2-4-4-2 l'anatomie de muscle squelettique :

La plupart des muscles squelettiques sont attachés à deux os par les tendons. Les tendons sont des bandes denses de tissu conjonctif régulier et dense dont les fibres de collagène attachent fermement les muscles aux os. Les tendons sont soumis à un stress extrême lorsque les muscles les tirent; ils sont donc très forts et sont tissés dans les revêtements des muscles et des os.

Les muscles se déplacent en raccourcissant leur longueur, en tirant sur les tendons et en rapprochant les os. L'un des os est tiré vers l'autre os, qui reste immobile. La place sur l'os stationnaire qui est relié au muscle par des tendons s'appelle l'origine. La place sur l'os en mouvement qui est connecté au muscle via les tendons s'appelle l'insertion. Le ventre du muscle est la partie charnue du muscle située entre les tendons qui assure la contraction proprement dite.

La caractéristique la plus connue du muscle squelettique est sa capacité à se contracter et à provoquer des mouvements. Les muscles squelettiques agissent non seulement pour produire un mouvement, mais également pour arrêter le mouvement, par exemple en résistant à la gravité pour maintenir la posture. De petits ajustements constants des muscles squelettiques sont nécessaires pour maintenir un corps droit ou équilibré dans n'importe quelle position. Les muscles empêchent également les mouvements excessifs des os et des articulations, préservant la stabilité du squelette et empêchant les dommages ou la déformation de la structure du squelette. Les articulations peuvent être désalignées ou disloquées entièrement en tirant sur les os associés; les muscles travaillent pour maintenir les articulations stables. Les muscles squelettiques sont situés dans tout le corps aux ouvertures des voies internes pour contrôler le mouvement de diverses substances. Ces muscles permettent aux fonctions telles que la déglutition, la miction et la défécation d'être sous contrôle volontaire. Les muscles squelettiques protègent également les organes internes (en particulier les organes abdominaux et pelviens) en agissant comme une barrière externe ou un bouclier contre les traumatismes externes et en soutenant le poids des organes. Les muscles squelettiques contribuent au maintien de l'homéostasie dans le corps en générant de la chaleur. La contraction musculaire nécessite de l'énergie et, lorsque l'ATP est décomposé, de la chaleur est produite. Cette chaleur est très perceptible pendant l'exercice, lorsqu'un mouvement musculaire soutenu provoque une

élévation de la température corporelle et en cas de froid extrême, lorsque des frissons produisent des contractions aléatoires du muscle squelettique pour générer de la chaleur. Chaque muscle squelettique est un organe composé de divers tissus intégrés. Ces tissus comprennent les fibres du muscle squelettique, les vaisseaux sanguins, les fibres nerveuses et le tissu conjonctif. Chaque muscle squelettique comporte trois couches de tissu conjonctif (appelé «mysia») qui l'enferment et structurent le muscle dans son ensemble, tout en compartimentant les fibres musculaires dans le muscle (Figure I-8). Chaque muscle est enveloppé dans une gaine de tissu conjonctif dense et irrégulier appelé épimysium, qui permet à un muscle de se contracter et de se déplacer de manière puissante tout en maintenant son intégrité structurelle. L'épimysium sépare également le muscle des autres tissus et organes de la région, permettant ainsi au muscle de se déplacer indépendamment.

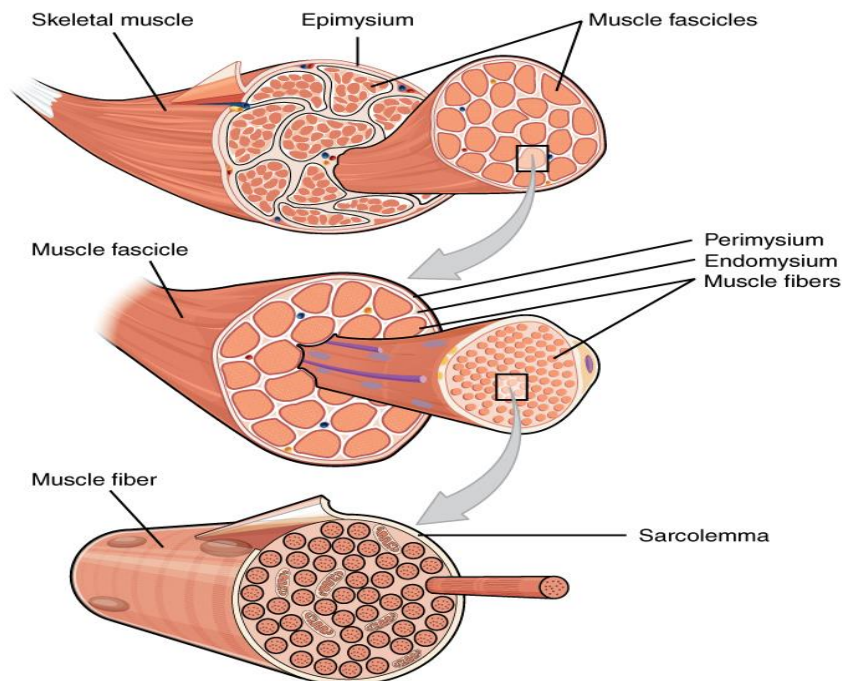


Figure I-8. Les trois couches de tissu conjonctif [8]

1-2-4-4-3 les fibres musculaires squelettique :

Comme les cellules des muscles squelettiques sont longues et cylindriques, elles sont communément appelées fibres musculaires. Les fibres musculaires squelettiques peuvent être assez grosses pour les cellules humaines, avec des diamètres allant jusqu'à 100 μm et des longueurs allant jusqu'à 30 cm dans le Sartorius de la jambe. Au début du développement, les myoblastes embryonnaires, chacun avec son propre noyau, fusionnent avec des centaines d'autres myoblastes pour former les fibres musculaires squelettiques multinucléées. Les noyaux multiples désignent des copies multiples de gènes, permettant la production de grandes quantités de protéines et d'enzymes nécessaires à la contraction musculaire. Une autre terminologie associée aux fibres musculaires est enracinée dans le sarco grec qui signifie «chair». La membrane plasmique des fibres musculaires est appelée sarcolemme, le cytoplasme est appelé sarcoplasme et le réticulum endoplasmique lisse et spécialisé, qui stocke, libère, et récupère les ions calcium (Ca^{++}) est appelé réticulum sarcoplasmique (SR) (Figure I-9). Comme on le décrira bientôt, l'unité fonctionnelle d'une fibre musculaire squelettique est le sarcomère, un arrangement hautement organisé d'actine (myofilaments contractiles) et de myosine (filament épais), ainsi que d'autres protéines de soutien.

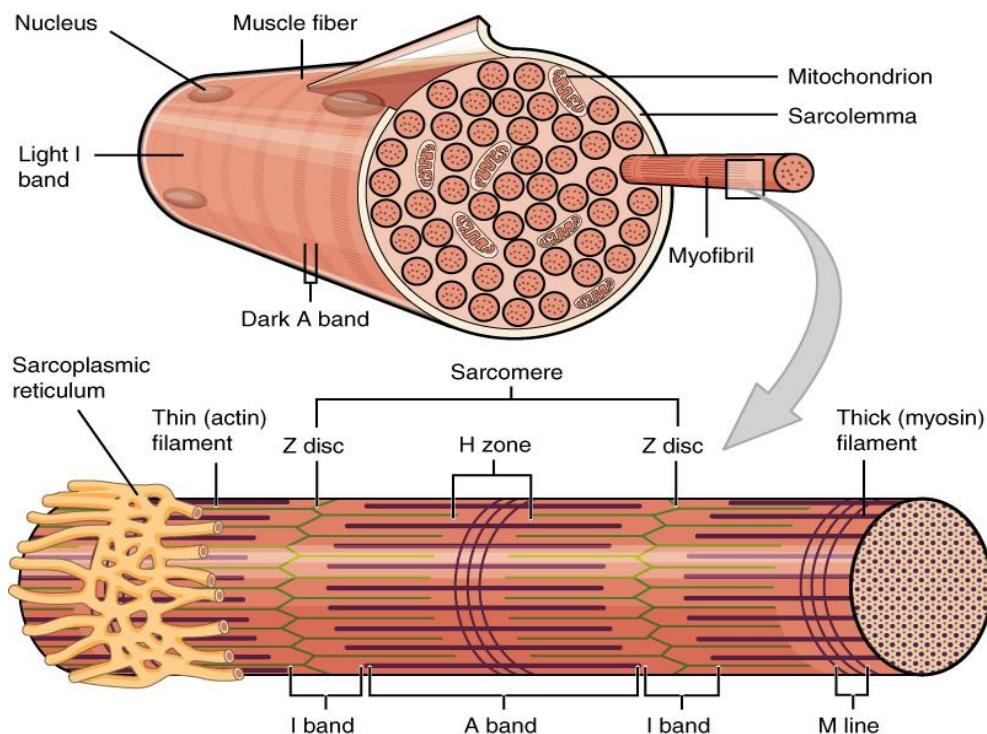


Figure I-9. Fibre musculaire.[9]

1-2-4-4-4 le sarcomère :

L'aspect strié des fibres musculaires squelettiques est dû à la disposition séquentielle des myofilaments d'actine et de myosine dans un ordre séquentiel d'une extrémité à l'autre de la fibre musculaire. Chaque paquet de ces microfilaments et de leurs protéines régulatrices, la troponine et la tropomyosine (ainsi que d'autres protéines) est appelé sarcomère.

Le sarcomère est l'unité fonctionnelle de la fibre musculaire. Le sarcomère lui-même est contenu dans la myofibrille qui recouvre toute la longueur de la fibre musculaire et se fixe au sarcolemme à son extrémité. À mesure que les myofibrilles se contractent, l'ensemble de la cellule musculaire se contracte. Les myofibrilles n'ayant qu'un diamètre d'environ $1,2 \mu\text{m}$, on en trouve des centaines à des milliers (chacune contenant des milliers de sarcomères) à l'intérieur d'une fibre musculaire. Chaque sarcomère a une longueur approximative de $2 \mu\text{m}$ et présente une disposition en forme de cylindre tridimensionnel. Il est entouré de structures appelées disques Z (également appelées lignes Z, car les images sont bidimensionnelles), sur lesquelles sont ancrés les myofilaments d'actine (Figure I-3) Du fait que l'actine et son complexe troponine-tropomyosine (faisant saillie des disques Z vers le centre

du sarcomère) forment des brins plus minces que la myosine, on parle alors de filament fin du sarcomère. De même, parce que les brins de myosine et leurs têtes multiples (faisant saillie du centre du sarcomère vers les disques Z, mais pas trop loin) ont plus de masse et sont plus épais, on les appelle le filament épais du sarcomère.

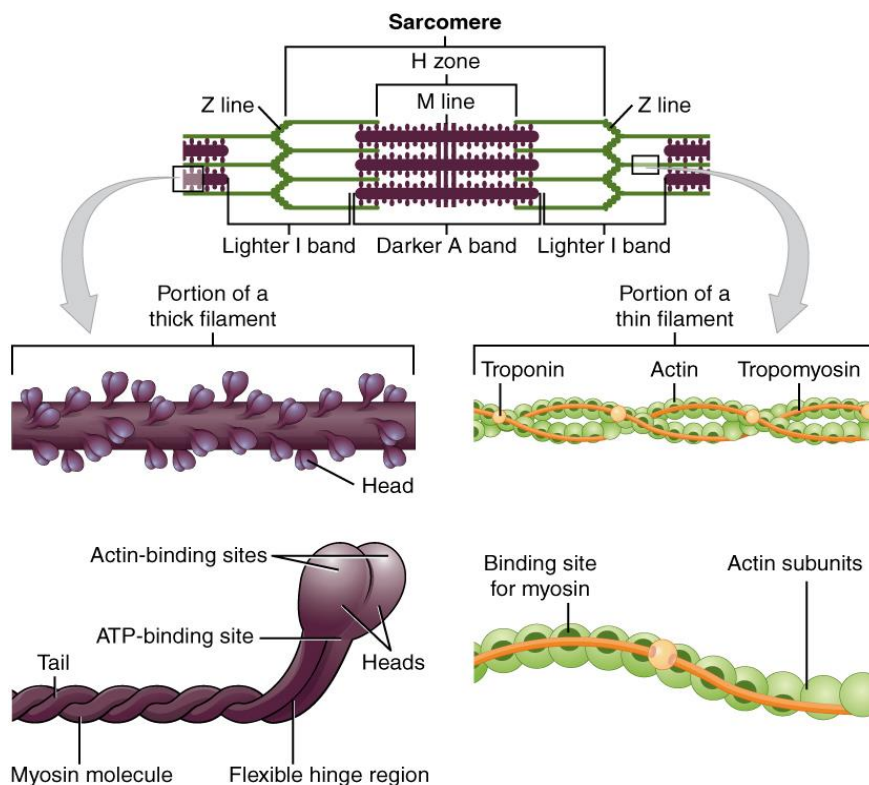


Figure I-10. Le sarcomère, la région d'une ligne Z à la suivante, est l'unité fonctionnelle d'une fibre musculaire squelettique.[10]

1-2-4-5 La jonction musculaire :

Une autre spécialisation du muscle squelettique est le site où le terminal d'un neurone moteur rencontre la fibre musculaire, appelée jonction neuromusculaire (NMJ). C'est là que la fibre musculaire répond d'abord à la signalisation du neurone moteur. Chaque fibre de muscle squelettique dans chaque muscle squelettique est innervée par un neurone moteur au NMJ. Les signaux d'excitation du neurone sont le seul moyen d'activer de manière fonctionnelle la contraction de la fibre.

Toutes les cellules vivantes ont des potentiels membranaires, ou des gradients électriques à travers leurs membranes. L'intérieur de la membrane est généralement compris entre -60 et -90 mV, par rapport à l'extérieur. C'est ce que l'on appelle le potentiel de membrane d'une cellule. Les neurones et les cellules musculaires peuvent utiliser leurs potentiels membranaires pour générer des signaux électriques. Ils le font en contrôlant le mouvement des particules chargées, appelées ions, à travers leurs membranes pour créer des courants électriques. Ceci est réalisé en ouvrant et en fermant des protéines spécialisées dans la membrane, appelées canaux ioniques. Bien que les courants générés par les ions se déplaçant à travers ces protéines de canal soient très faibles, ils forment la base de la signalisation neurale et de la contraction musculaire.

Les neurones et les cellules musculaires squelettiques sont électriquement excitables, ce qui signifie qu'ils sont capables de générer des potentiels d'action. Un potentiel d'action est un type spécial de signal électrique qui peut se déplacer le long d'une membrane cellulaire sous forme d'onde. Cela permet à un signal d'être transmis rapidement et fidèlement sur de longues distances.

Bien que le terme couplage excitation-contraction déroute ou effraie certains étudiants, il se résume à ceci: pour qu'une fibre musculaire squelettique se contracte, sa membrane doit d'abord être «excitée», autrement dit, elle doit être stimulée pour déclencher un potentiel d'action. Le potentiel d'action des fibres musculaires, qui parcourt le sarcolemme sous forme d'onde, est «couplé» à la contraction réelle par la libération d'ions calcium (Ca^{++}) provenant du SR. Une fois libéré, le Ca^{++} interagit avec les protéines protectrices, les obligeant à se déplacer de façon à ce que les sites de liaison à l'actine soient disponibles pour être attachés par les têtes de myosine. La myosine tire ensuite les filaments d'actine vers le centre, raccourcissant la fibre musculaire.

-Dans le muscle squelettique, cette séquence commence par des signaux provenant de la division motrice somatique du système nerveux. En d'autres termes, l'étape «d'excitation» dans les muscles squelettiques est toujours déclenchée par une signalisation du système nerveux (Figure I-11).

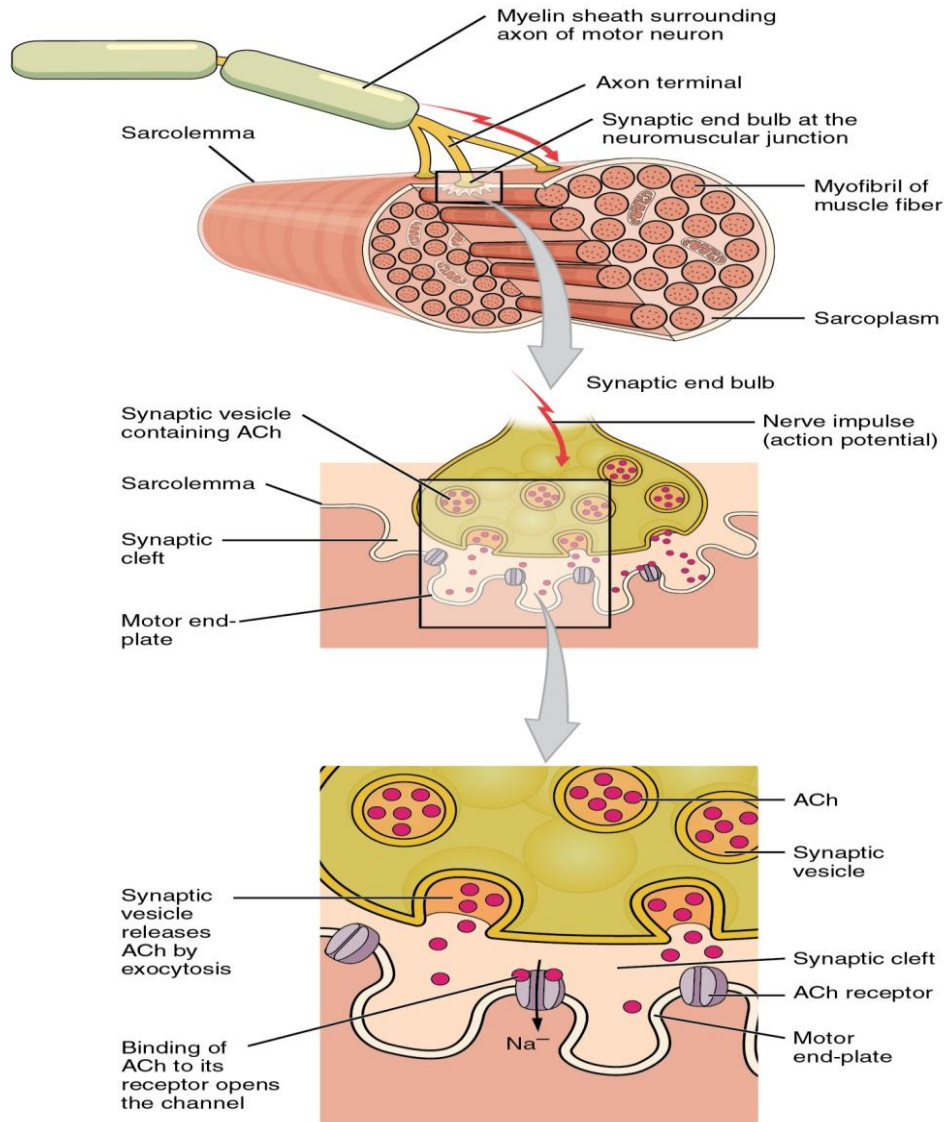


Figure I-11. Plaque d'extrémité et innervation du moteur.[11]

1-2-5 Potentiel de membrane :

Les fonctions du système nerveux - sensation, intégration et réponse - dépendent des fonctions des neurones sous-jacents à ces voies. Pour comprendre comment les neurones sont capables de communiquer, il est nécessaire de décrire le rôle d'une membrane excitable dans la génération de ces signaux. La base de cette communication est le potentiel d'action, qui montre comment les changements dans la membrane peuvent constituer un signal. Examiner la manière dont ces signaux fonctionnent dans des circonstances plus variables implique de jeter un regard sur les potentiels gradués, qui seront abordés dans la section suivante.

1-2-5-1 Membrane de cellules électriques actives :

La plupart des cellules du corps utilisent des particules chargées, des ions, pour constituer une charge à travers la membrane cellulaire. Auparavant, il a été démontré que cela faisait partie du fonctionnement des cellules musculaires. La contraction des muscles squelettiques, basée sur le couplage excitation-contraction, nécessite l'intervention d'un neurone. Les deux cellules utilisent la membrane cellulaire pour réguler le mouvement des ions entre le fluide extracellulaire et le cytosol. Comme vous l'avez appris dans le chapitre sur les cellules, la membrane cellulaire est principalement responsable de la régulation de ce qui peut traverser la membrane et de ce qui ne reste que d'un côté. La membrane cellulaire est une bicouche phospholipidique, de sorte que seules les substances qui peuvent passer directement à travers le noyau hydrophobe peuvent diffuser à travers l'aide. Les particules chargées, qui sont par définition hydrophiles, ne peuvent pas traverser la membrane cellulaire sans assistance (Figure I-12). Les protéines transmembranaires, en particulier les protéines de canal, rendent cela possible. Plusieurs canaux de transport passifs, ainsi que des pompes de transport actif, sont nécessaires pour générer un potentiel transmembranaire et un potentiel d'action. La protéine porteuse appelée pompe à sodium / potassium, qui déplace les ions sodium (Na^+) d'une cellule et les ions potassium (K^+) dans une cellule, régule ainsi la concentration en ions des deux côtés de la membrane cellulaire.

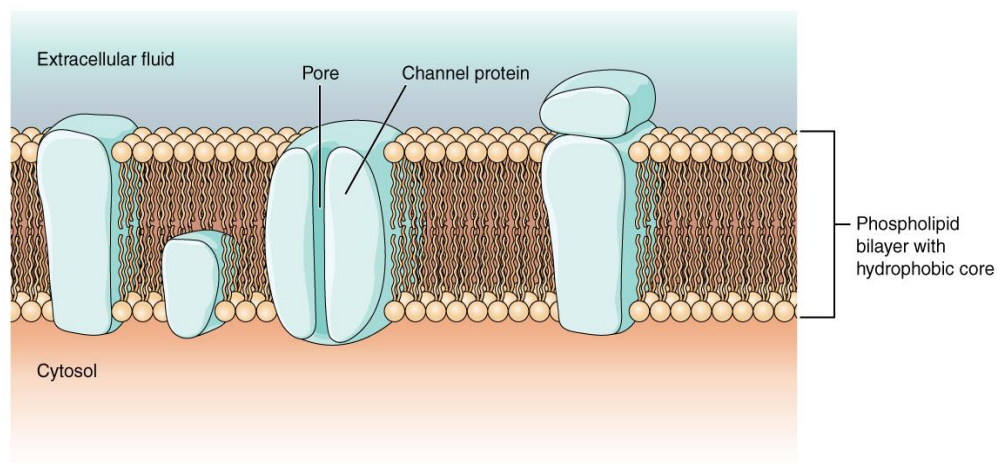


Figure I-12. Protéines membranaires et transmembranaires.[12]

La pompe à sodium / potassium nécessite de l'énergie sous forme d'adénosine triphosphate (ATP), elle est donc également appelée ATPase. Comme cela a été expliqué dans le chapitre sur les cellules, la concentration de Na^+ est plus élevée à l'extérieur de la cellule qu'à l'intérieur, et la concentration de K^+ est plus élevée à l'intérieur de la cellule qu'à l'extérieur. Cela signifie que cette pompe déplace les ions contre les gradients de concentration pour le sodium et le potassium, raison pour laquelle elle nécessite de l'énergie. En fait, la pompe maintient essentiellement ces gradients de concentration.

Les canaux ioniques sont des pores permettant à des particules chargées spécifiques de traverser la membrane en réponse à un gradient de concentration existant. Les protéines sont capables de recouvrir la membrane cellulaire, y compris son noyau hydrophobe, et peuvent interagir avec la charge en ions en raison des propriétés variées des acides aminés présents dans des domaines ou des régions spécifiques du canal protéique. Les acides aminés hydrophobes se trouvent dans les domaines qui se rattachent aux queues hydrocarbonées des phospholipides. Les acides aminés hydrophiles sont exposés aux environnements liquides du liquide extracellulaire et du cytosol. De plus, les ions vont interagir avec les acides aminés hydrophiles, qui seront sélectifs pour la charge de l'ion. Les canaux pour les cations (ions positifs) auront des chaînes latérales chargées négativement dans le pore. Les canaux pour les anions (ions négatifs) auront des chaînes latérales chargées positivement dans le pore. Cela s'appelle une exclusion électrochimique, ce qui signifie que le pore du canal est spécifique à la charge.

Les canaux ioniques peuvent également être spécifiés par le diamètre du pore. La distance entre les acides aminés sera spécifique pour le diamètre de l'ion lorsqu'il se dissocie des molécules d'eau qui l'entourent. En raison des molécules d'eau environnantes, les pores plus larges ne sont pas idéaux pour les ions plus petits car les molécules d'eau interagiront, par des liaisons hydrogène, plus facilement que les chaînes latérales d'acides aminés. Ceci est appelé exclusion de taille. Certains canaux ioniques sont sélectifs pour la charge mais pas nécessairement pour la taille, et sont donc appelés canaux non spécifiques. Ces canaux non spécifiques permettent aux cations, en

particulier Na^+ , K^+ et Ca^{2+} , de traverser la membrane, mais excluent les anions.

Un canal lié à un ligand s'ouvre car une molécule de signalisation, un ligand, se lie à la région extracellulaire du canal. Ce type de canal est également appelé récepteur ionotrope car lorsque le ligand, appelé neurotransmetteur dans le système nerveux, se lie à la protéine, les ions traversent la membrane en modifiant sa charge (Figure I-13).

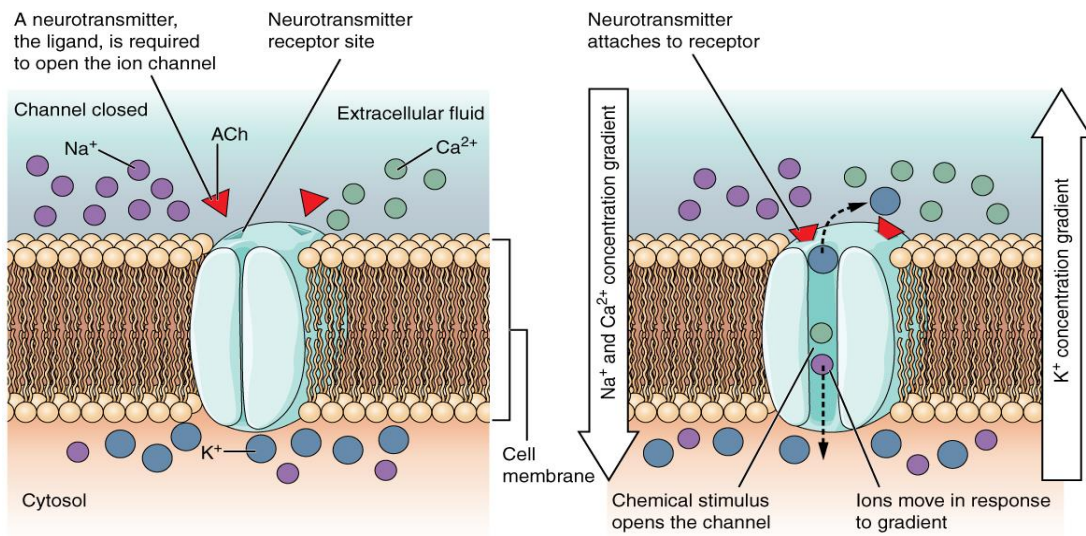


Figure I-13. Canaux à porte de ligand.[13]

Un canal déclenché mécaniquement s'ouvre en raison d'une déformation physique de la membrane cellulaire. De nombreux canaux associés au sens du toucher. (somatosensation) sont contrôlés mécaniquement. Par exemple, lors de l'application d'une pression sur la peau, ces canaux s'ouvrent et permettent aux ions de pénétrer dans la cellule. Un canal similaire à ce type de canal serait le canal qui s'ouvre sur la base des changements de température, comme lors du test de l'eau dans la douche (Figure I-14).

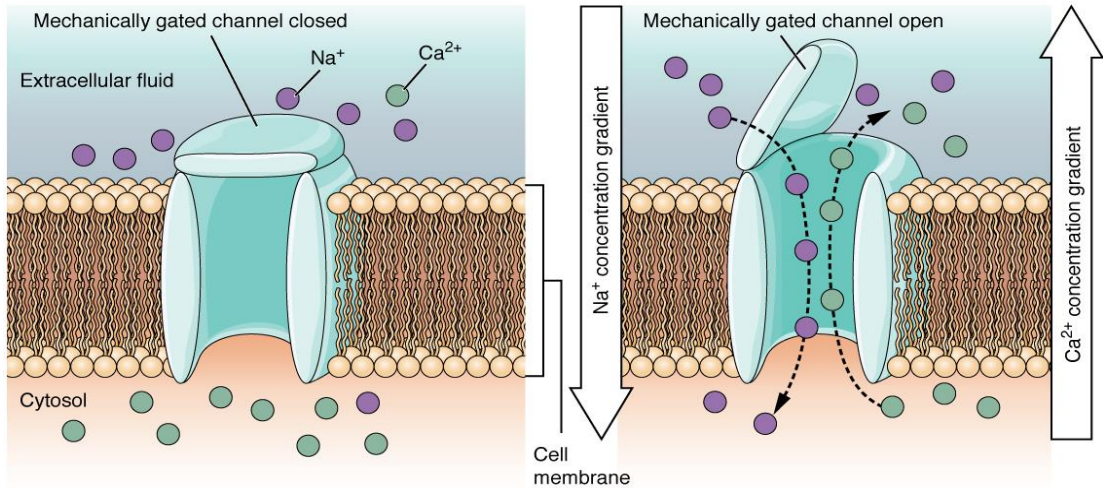


Figure I-14. Canaux voltage-dépendants.[14]

Un canal de fuite est déclenché de manière aléatoire, ce qui signifie qu'il s'ouvre et se ferme de manière aléatoire, d'où la référence à une fuite. Il n'y a aucun événement réel qui ouvre le canal; au lieu de cela, il a un taux intrinsèque de commutation entre les états ouvert et fermé. Les canaux de fuite contribuent à la tension transmembranaire au repos de la membrane excitable (figure I-15).

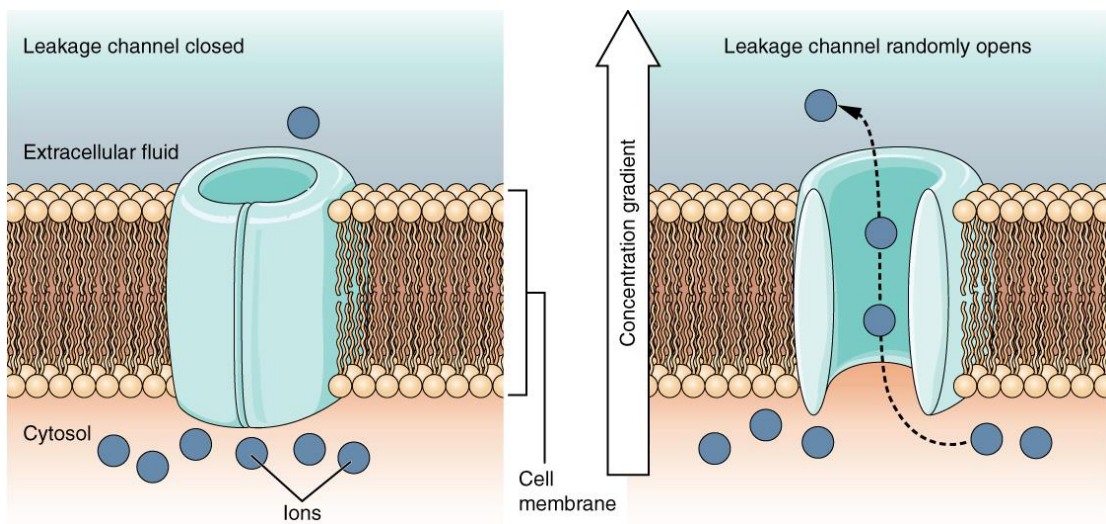


Figure I-15. Canaux de fuite..[15]

1-2-5-2 L'état électrique de la membrane cellulaire :

L'état électrique de la membrane cellulaire peut avoir plusieurs variations. Ce sont toutes des variations du potentiel de membrane. Un potentiel est une distribution de charge à travers la membrane cellulaire, mesurée en millivolts (mV). La norme consiste à comparer l'intérieur de la cellule par rapport à l'extérieur, de sorte que le potentiel de membrane est une valeur représentant la charge du côté intracellulaire de la membrane sur la base de l'extérieur étant nul, relativement parlant (Figure I-16).

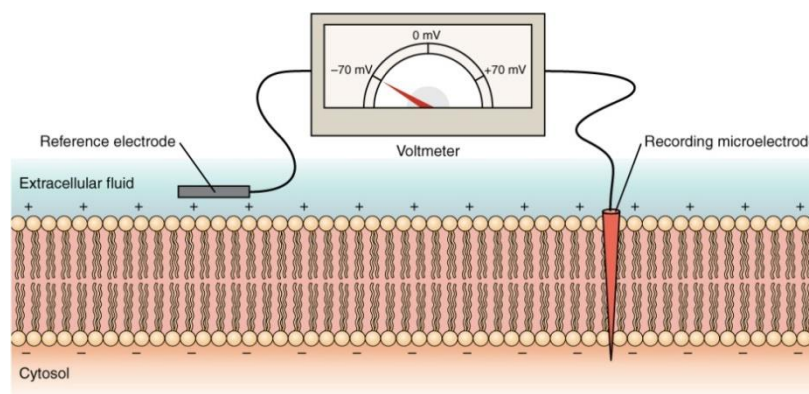


Figure I-16. Mesure de la charge à travers une membrane avec un voltmètre.[16]

La concentration en ions dans les fluides extracellulaires et intracellulaires est largement équilibrée, avec une charge neutre nette. Cependant, une légère différence de charge se produit directement à la surface de la membrane, à la fois à l'intérieur et à l'extérieur. C'est la différence dans cette région très limitée qui a tout le pouvoir des neurones (et des cellules musculaires) de générer des signaux électriques, y compris des potentiels d'action.

Avant de pouvoir décrire ces signaux électriques, il faut expliquer l'état de repos de la membrane. Lorsque la cellule est au repos et que les canaux ioniques sont fermés (à l'exception des canaux de fuite qui s'ouvrent de manière aléatoire), les ions sont répartis sur la membrane de manière très prévisible. La concentration de Na^+ à l'extérieur de la cellule est 10 fois supérieure à la concentration à l'intérieur. En outre, la concentration de K^+ à l'intérieur de la cellule est supérieure à celle à l'extérieur. Le cytosol contient

une concentration élevée d'anions, sous forme d'ions phosphate et de protéines chargées négativement. Les gros anions sont un composant de la membrane cellulaire interne, y compris les phospholipides spécialisés et les protéines associées au feuillet interne de la membrane (le terme feuillet est utilisé pour désigner un côté de la membrane bicouche lipidique). La charge négative est localisée dans les gros anions.

Avec les ions distribués à travers la membrane à ces concentrations, la différence de charge est mesurée à -70 mV, valeur décrite comme le potentiel de membrane au repos. La valeur exacte mesurée pour le potentiel de membrane au repos varie d'une cellule à l'autre, mais -70 mV est le plus souvent utilisé comme valeur. Cette tension serait en réalité beaucoup plus basse, sauf pour les contributions de certaines protéines importantes dans la membrane. Les canaux de fuite permettent au Na^+ de se déplacer lentement dans la cellule ou au K^+ de le quitter lentement, et la pompe Na^+ / K^+ les restaure. Cela peut sembler être un gaspillage d'énergie, mais chacun joue un rôle dans le maintien du potentiel de la membrane.

1-2-5-3 Le potentiel d'action :

Le potentiel de membrane au repos décrit l'état stable de la cellule, qui est un processus dynamique équilibré par une fuite d'ions et un pompage d'ions. Sans aucune influence extérieure, cela ne changera pas. Pour que le signal électrique commence, le potentiel de la membrane doit changer.

Cela commence par une ouverture de canal pour Na^+ dans la membrane. Comme la concentration de Na^+ est 10 fois plus élevée à l'extérieur de la cellule qu'à l'intérieur de la cellule, les ions se précipiteront dans la cellule en grande partie en raison du gradient de concentration. Le sodium étant un ion chargé positivement, il modifiera la tension relative immédiatement à l'intérieur de la cellule par rapport à l'extérieur. Le potentiel de repos est l'état de la membrane à une tension de -70 mV, de sorte que le cation sodium qui pénètre dans la cellule la rend moins négative. C'est ce qu'on appelle la dépolarisation, ce qui signifie que le potentiel de membrane se rapproche de zéro.

Le gradient de concentration en Na^+ est si fort qu'il continuera à pénétrer dans la cellule même après que le potentiel de la membrane soit devenu nul, de sorte que la tension immédiatement autour du pore commence à devenir positive. Le gradient électrique joue également un rôle, car les protéines négatives situées sous la membrane attirent l'ion sodium. Le potentiel de membrane atteindra +30 mV d'ici à l'introduction du sodium dans la cellule. Lorsque le potentiel de la membrane atteint +30 mV, d'autres canaux voltage-dépendants s'ouvrent dans la membrane. Ces canaux sont spécifiques à l'ion potassium. Un gradient de concentration agit également sur K^+ . Lorsque K^+ commence à quitter la cellule en prenant une charge positive, le potentiel de membrane commence à revenir vers sa tension de repos. C'est ce qu'on appelle la repolarisation, ce qui signifie que la tension de la membrane revient à la valeur de -70 mV du potentiel de la membrane au repos.

La repolarisation ramène le potentiel de membrane à la valeur de -70 mV qui indique le potentiel de repos, mais dépasse en réalité cette valeur. Les ions potassium atteignent l'équilibre lorsque la tension membranaire est inférieure à -70 mV. Une période d'hyperpolarisation se produit alors que les canaux K^+ sont ouverts. La fermeture de ces canaux K^+ est légèrement retardée, ce qui explique ce court dépassement.

Ce qui a été décrit ici est le potentiel d'action, qui est présenté sous forme de graphique de la tension en fonction du temps sur la figure I-17. Il s'agit du signal électrique généré par le tissu nerveux pour la communication. La variation de la tension membranaire de -70 mV au repos à +30 mV en fin de dépolarisation correspond à une variation de 100 mV. Cela peut aussi être écrit comme un changement de 0,1 V. Pour mettre cette valeur en perspective, pensez à une batterie. Une pile AA que vous pourriez trouver dans une télécommande de télévision a une tension de 1,5 V, ou une pile de 9 V (la pile rectangulaire avec deux bornes à une extrémité) correspond, de toute évidence, à 9 V. Le changement dans le potentiel d'action est un ou deux ordres de grandeur inférieur à la charge de ces batteries. En fait, le potentiel de membrane peut être décrit comme une batterie. Une charge est stockée à travers la membrane qui peut être libérée dans les bonnes conditions. Une pile de votre télécommande a enregistré une charge qui est «libérée» lorsque vous appuyez sur un bouton.

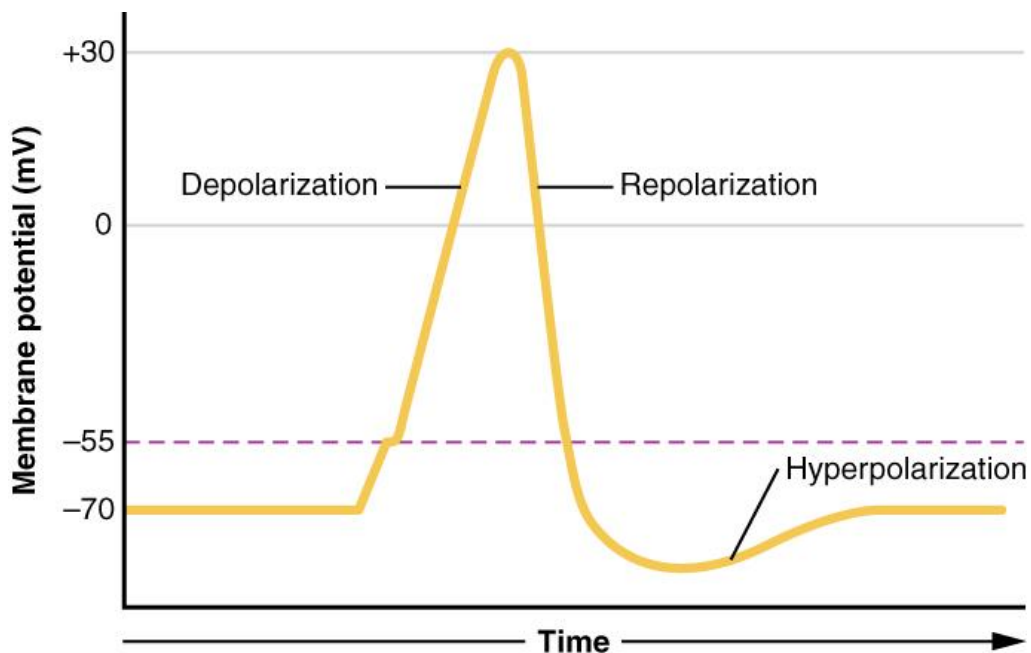


Figure I-17. Graphique du potentiel d'action.[17]

Un troisième type de canal qui joue un rôle important dans la dépolarisation du potentiel d'action est le canal Na^+ voltage-dépendant. Les canaux qui commencent à dépolariser la membrane en raison d'un stimulus aident la cellule à se dépolariser de -70 mV à -55 mV. Une fois que la membrane a atteint cette tension, les canaux de Na^+ voltage-dépendants s'ouvrent. C'est ce qu'on appelle le seuil. Toute dépolarisation qui ne modifie pas le potentiel de membrane à -55 mV ou plus n'atteindra pas le seuil et n'entraînera donc pas de potentiel d'action. De plus, tout stimulus qui dépolarise la membrane à -55 mV ou plus entraînera l'ouverture d'un grand nombre de canaux et un potentiel d'action sera initié.

En raison de ce seuil, le potentiel d'action peut être assimilé à un événement numérique: il se produit ou non. Si le seuil n'est pas atteint, aucun potentiel d'action ne se produit. Si la dépolarisation atteint -55 mV, le potentiel d'action se poursuit jusqu'à +30 mV, valeur à laquelle K^+ provoque la répolarisation, y compris le dépassement hyperpolarisant.

De plus, ces changements sont les mêmes pour chaque potentiel d'action, ce qui signifie qu'une fois le seuil atteint, la même chose se produit. Un stimulus plus fort, qui pourrait dépolariser la membrane bien au-delà du seuil, ne créera

pas un potentiel d'action «plus grand». Les potentiels d'action sont «tout ou rien». Soit la membrane atteint le seuil et tout se passe comme décrit ci-dessus, soit la membrane n'atteint pas le seuil et rien d'autre ne se produit. Tous les potentiels d'action culminent à la même tension (+30 mV), de sorte qu'un potentiel d'action n'est pas plus grand qu'un autre. Des stimuli plus forts vont initier plusieurs potentiels d'action plus rapidement, mais les signaux individuels ne sont pas plus grands. Ainsi, par exemple, vous ne ressentirez pas de sensation de douleur plus intense, ni de contraction musculaire plus forte, en raison de la taille du potentiel d'action car ils ne sont pas de tailles différentes.

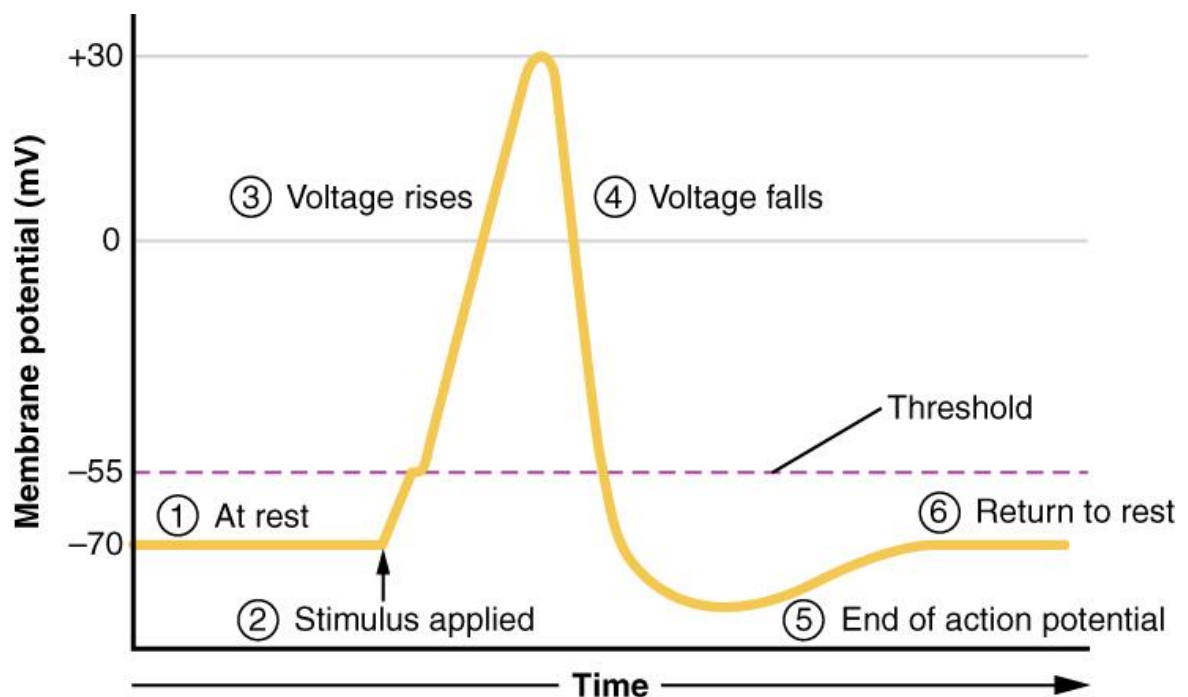


Figure I-18. Étapes d'un potentiel d'action. [18]

1-2-5-4 Propagation de potentiel d'action :

Le potentiel d'action est initié au début de l'axone, à ce qu'on appelle le segment initial. Il existe une forte densité de canaux Na^+ voltage-dépendants de sorte qu'une dépolarisation rapide puisse avoir lieu ici. En descendant le long de l'axone, le potentiel d'action se propage car davantage de canaux Na^+ dépendants du potentiel sont ouverts à mesure que la dépolarisation se propage. Cette propagation est due au fait que le Na^+ pénètre dans le canal et

se déplace le long de la membrane cellulaire. Lorsque le Na^+ se déplace ou coule sur une courte distance le long de la membrane cellulaire, sa charge positive dépolarise un peu plus la membrane cellulaire. À mesure que cette dépolarisation se propage, de nouveaux canaux Na^+ dépendants de la tension s'ouvrent et davantage d'ions se précipitent dans la cellule, étalant un peu plus loin la dépolarisation.

Alors la technique biomédicale permet de modéliser le mécanisme de contraction musculaire sous forme d'un circuit électronique pour simplifier l'étude de l'action neuro-musculaire.

1-3 la conclusion générale :

L'EEG peut déterminer les changements de potentiel électrique au niveau du cuir chevelu. Cela peut être utile pour diagnostiquer des troubles cérébraux. L'EEG mesure les fluctuations de tension induites par le courant ionique dans les cellules nerveuses du cerveau. Dans les contextes cliniques, chacun a un rôle dans le traitement de l'information et c'est ce qui nous permet de créer une réponse. De cela, l'EEG indique l'enregistrement de l'activité électrique automatique du cerveau sur une période de temps.

Chapitre

II

Chapitre 2 :l'instrumentation de détection et traitement de signal physiologique

2-1 Introduction :

Notre cerveau est constitué de plusieurs milliards des cellules qui échangent en permanence des informations, ces échanges provoquent l'émission des courants électriques faibles (les ondes cérébrales) que l'on peut capter grâce à l'électro-encéphalo-gramme. L'EEG est un examen médical simple et sans danger, il enregistre certain aspect de l'activité cérébral. Pour procéder à un EEG, on place à divers endroits relies un appareil qui mesure la différence de potentiel entre divers lieux du cortex, et on obtient des tracées appelé les ondes cérébral.

2-2 Contexte et motivation :

Un signal EEG mesure la somme des activités électriques de milliers, voire de millions de neurones qui ont la même orientation spatiale dans le cerveau humain. Il peut être acquis via un petit disque électrique plat (capteur d'électrode) assis directement sur un emplacement du cuir chevelu [1] et est normalement représenté comme une forme d'onde en microvolt. Les populations neuronales dans différents emplacements du cerveau peuvent générer des signaux EEG avec différentes propriétés statistiques lorsque les êtres humains pensent à des tâches différentes. Par conséquent, les signaux EEG multicanaux peuvent avoir des schémas différents en termes de tâches. À partir de ce principe, la reconnaissance de formes EEG grâce à des algorithmes d'apprentissage automatique sera en mesure d'identifier les pensées des êtres humains. Par exemple, lorsque les êtres humains planifient deux directions de conduite possibles qui sont en avant et en arrière, les données EEG générées peuvent être classées en utilisant l'algorithme de la machine à vecteur de support (SVM) [2 & 3] en deux catégories associées aux directions. De tels résultats de classification EEG avec une grande précision pourront pour fournir une méthode et une technologie pour le contrôle de mouvement mobile. Lorsque les êtres humains imaginent ou pensent les mots anglais lettre par lettre à l'esprit, les modèles EEG peuvent varier avec les lettres et, par conséquent, les données EEG portant les informations de la lettre peuvent même être classées en différentes catégories par l'algorithme SVM [1 & 2]. Les résultats de classification précis seront en mesure de

fournir une nouvelle façon de communiquer. En outre, pour les applications liées au cerveau humain, la technologie d'acquisition d'EEG peut être coûteuse, sûre, portable et sans fil par rapport à d'autres technologies telles que l'imagerie par résonance magnétique fonctionnelle et la chirurgie invasive d'implantation d'électrodes dans le cerveau pour les pointes neurales.

La sclérose latérale amyotrophique (SLA) est une maladie chronique évolutive des motoneurons qui fait perdre aux patients la mobilité et les capacités d'expression [4]. Cette maladie survient lorsque des cellules nerveuses spécifiques du cerveau et de la moelle épinière qui contrôlent les mouvements volontaires dégèrent progressivement. La perte de ces motoneurons affaiblit et dépérit les muscles sous leur contrôle, entraînant une paralysie et des difficultés d'élocution [5]. Par conséquent, les patients SLA peuvent ne pas être capables de bouger ou de parler normalement. La SLA affecte environ 8 000 Algériens et 120 nouveaux cas chaque année. Malheureusement, la cause de ce processus pathologique est inconnue, il n'y a pas de remède pour la SLA, et les patients peuvent mourir dans les 5 ans suivant le diagnostic de cette maladie.

Motivée par la maladie de la SLA et les propriétés des signaux EEG du cerveau humain, notre recherche consiste à développer un système prototype basé sur l'EEG qui peut surveiller l'activité du cerveau humain, détecter la pensée de l'homme comme les directions de déplacement et les lettres, puis contrôler le mouvement de la voiture robotique. Dans notre travail pour le développement du système, nous utilisons le package Emotiv EPOC, un ordinateur personnel, les logiciels (Matlab, Python et Arduino) et des microcontrôleurs (package Arduino). Un tel système peut également être appelé système d'interface cerveau-ordinateur (BCI). Sur la base du système développé, une application telle qu'un fauteuil roulant de contrôle cérébral à base d'EEG avancé avec écran LCD peut être développée pour améliorer la qualité de levage des patients SLA avec une meilleure mobilité et communications.

2-3 Système BCI :

Système BCI basé sur EEG pour le contrôle de la voiture robotique et de l'affichage alphabétique Dans notre travaille, nous nous concentrons sur la création d'un système BCI pour contrôler le mouvement de la voiture robotique via les signaux EEG uniquement. Le principal défi du développement d'un tel système est que les données provenant du cerveau humain sont normalement extrêmement bruyantes. Par conséquent, nous devons utiliser des techniques spéciales ou compliquées de traitement du signal et d'analyse des données pour reconnaître la pensée humaine avec une grande précision. Par exemple, nous utilisons SVM, un algorithme d'apprentissage machine non linéaire sophistiqué et populaire, pour classer les modèles EEG en différentes classes pour décoder la pensée humaine.

L'idée principale de notre système BCI basé sur la classification d'EEG. Tout d'abord, un utilisateur enregistre les signaux EEG lorsqu'il pense à une direction ou à une lettre via un casque EPOC, et enregistre également ces données de domaine temporel dans un ordinateur personnel (PC) sans fil et automatiquement. Ensuite, les données EEG du domaine temporel sont converties en densité de spectre de puissance (PSD) dans le domaine fréquentiel en tant qu'entrée de l'algorithme SVM. La sortie de l'algorithme SVM serait la classe correspondante de l'entrée. Dans notre recherche, la sortie serait une direction ou une lettre. Avant que le modèle SVM puisse être utilisé pour détecter toute nouvelle réflexion sur les directions et les lettres de l'utilisateur, ses paramètres doivent être optimisés. Pour ce faire, une grande quantité de données d'apprentissage qui sont des paires d'entrées et de sorties connues sont générées par l'utilisateur. Dans notre projet, le modèle SVM est largement optimisé grâce à la bibliothèque libSVM [6]. Après cela, le modèle SVM bien formé peut être utilisé pour classer tout nouvel essai EEG dans une direction ou une lettre. Le résultat de la classification du modèle SVM bien formé est ensuite envoyé au robot via la carte Arduino, Python et Xbee pour contrôler à distance la direction de conduite. Lorsqu'un essai EEG est entièrement traité, un autre essai EEG est converti en PSD dans le domaine fréquentiel. Ensuite, ce PSD sera entré dans le modèle SVM bien formé. Ensuite, la sortie SVM sera utilisée pour déplacer le robot ou contrôler l'affichage LCD en conséquence.

Pour tester les performances du système BCI conçu, nous avons enregistré 1000 essais EEG, de 5 secondes chacun dans une seule catégorie, c'est la catégorie directionnelle. Dans la catégorie directionnelle, nous nous sommes concentrés sur 4 directions: gauche, droite, avant et arrière. Nous avons pris 200 données pour chaque direction. Ensuite, nous avons créé la catégorie alphabétique, nous nous sommes concentrés sur 2 lettres: A et B.

Enfin, nous avons pris 100 données pour chaque lettre. Ensuite, une partie de ces essais EEG avec la direction ou les lettres correspondantes a été utilisée pour former un modèle de classification SVM pour chaque catégorie. Les autres parties des essais EEG ont été utilisées pour montrer les performances du système BCI développé.

2-4 Le matériel :

a) EEG (EMOTIVEPOC) :

Le casque EMOTIV EPOC, comme le montre la figure 1 de la société Emotive [7], est capable d'enregistrer les signaux EEG du cerveau humain. Cet appareil est choisi pour deux raisons principales. Tout d'abord, Emotive EPOC est un appareil bien connu sur le marché et la recherche déposée et il est donc très sûr pour nous de faire des expériences. Deuxièmement, c'est un appareil à très faible coût par rapport aux autres casques EEG sur le marché et, par conséquent, il rend notre système développé à faible coût. Le dispositif EPOC fournit 14 canaux plus 2 références avec une fréquence d'échantillonnage de 128 données par seconde. Les 14 canaux ont les noms suivants: 1-AF3, 2-F7, 3-F3, 4-FC5, 5-T7, 6-P7, 7-O1, 8-O2, 9-P8, 10-T8, 11-FC6, 12-F4, 13-F8, 14-AF4. Différents canaux ont des caractéristiques différentes, car ils correspondent à différentes zones du cerveau. Les noms des canaux peuvent refléter leurs zones cérébrales.

En fait, F, P, O et T représentent le lobe frontal, le lobe pariétal, le lobe occipital et le lobe temporal, de manière réceptive. Cependant, C ne signifie pas le lobe central car il n'y a pas du tout de lobe central. Au lieu de cela, C signifie la zone centrale du cerveau. Les nombres impairs dans les noms représentent l'hémisphère gauche et les nombres pairs représentent l'hémisphère droit. Veuillez-vous reporter au système de placement des électrodes EEG 10-10 et aux hémisphères cérébraux gauche et droit comme indiqué sur la figure 2 pour les canaux et les zones cérébrales spécifiques.



Figure 1 :EEG EMOTIV EPOC

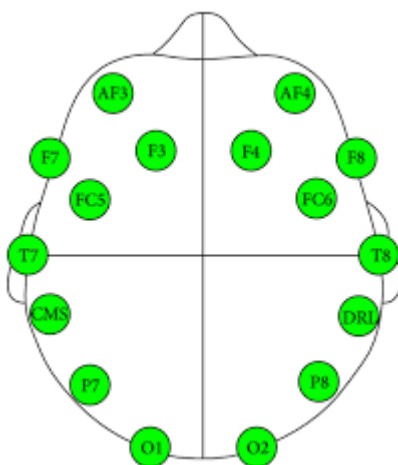


Figure 2 10-10 Système et emplacement d'électrodes EEG ,

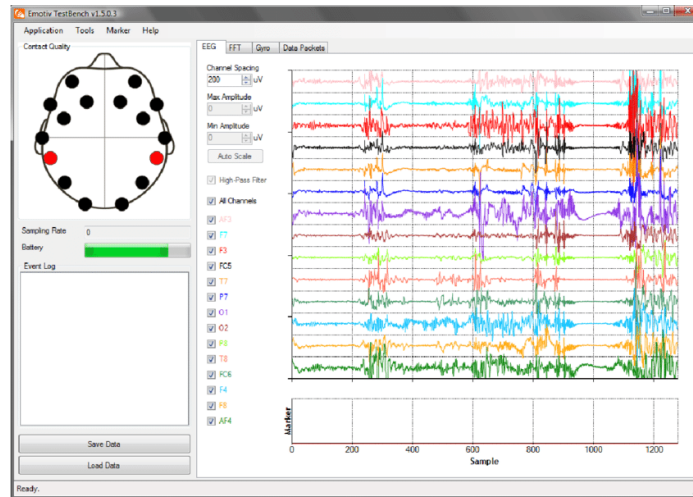


Figure 3 SDK du panneau de commande EEG EPOC

b) Carte Arduino :

Dans ce projet, la carte ArduinoUno comme le montre la figure 4 est la carte de commande principale pour la voiture robotique et le contrôleur de robot sans fil. Cette carte possède 14 broches d'entrée / sortie numériques avec 6 entrées analogiques. Il dispose également d'un cristal de quartz 16 Mhz, d'une connexion USB, d'une prise d'alimentation et d'un bouton de réinitialisation. Pour ce projet, deux cartes Arduino seront nécessaires pour la communication sans fil entre un ordinateur personnel (PC) et la voiture robotique. La première carte est connectée directement à l'ordinateur via un câble USB avec le logiciel BCI pour créer une interface Matlab-Arduino.

Tous les résultats calculés à partir des programmes Matlab seront envoyés dans ce premier tableau. La première planchesera utilisé pour contrôler un appareil appelé Xbee qui peut envoyer des signaux sans fil à la deuxième carte pour contrôler la voiture robotique.

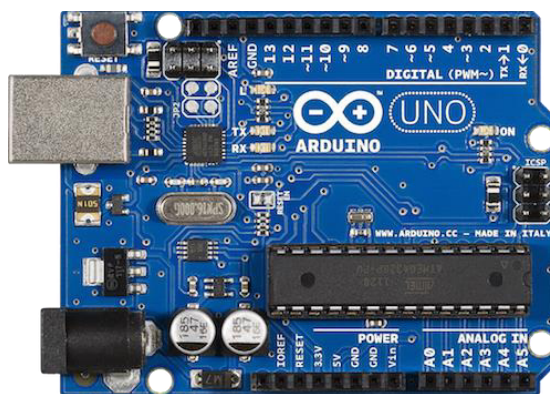


Figure 4 ArduinoUno R3 board

c) Le module Xbee:

La carte Xbee comme le montre la figure 5 est une simple antenne utilisée dans de nombreuses applications électroniques. Dans ce projet, cette carte servira de principal canal de communication entre le BCI et la voiture robot.



Figure 5 Antenne Xbee

2-5 Sélection du logiciel :

Pour construire le système BCI efficace pour ce projet, certains logiciels et packages uniques sont choisis. Nous avons utiliser EEGLAB, PHYSIONET qu'il sont des database (plateforme) sur matlab.

a) Matlab :

Matlab est un logiciel informatique multi-paradigmes très utilisé dans le domaine de la recherche. L'interfaçage avec le programme Matlabs peut être construit sur la base d'autres langages tels que C, C ++ et java. C'est également un logiciel open source qui permet aux utilisateurs d'ajouter de nombreux packages et fonctions. Pour ce projet, il y a deux raisons principales de choisir Matlab. Tout d'abord, il prend en charge le package de la bibliothèque Libsvm qui est une boîte à outils très efficace pour optimiser et implémenter le modèle SVM. Deuxièmement, bien que Matlab soit un langage informatique numérique, il contient de nombreux packages qui permettent à l'utilisateur de contrôler le matériel et le microcontrôleur tels que Arduino. Grâce à cet avantage, nous pourrions utiliser la fenêtre de commande Matlab pour contrôler et envoyer des signaux à une carte Arduino.

b) Importateur EPOC Simulink EEG :

Dans la recherche de maîtrise de Qing Gou [8], pour collecter les données, Qing a utilisé un logiciel d'Emotiv appelé «Emotiv Control Panel». Ce logiciel développé à partir d'Emotiv permet aux utilisateurs de collecter les données EEG brutes de l'appareil EPOC, puis d'enregistrer les données au format CVS utilisé dans Microsoft Excel. Si nous utilisons Emotiv Control Panel dans ce projet pour la collecte de données EEG, nous devons convertir manuellement les données EEG au format Matlab (.mat) pour exécuter l'algorithme SVM. Nous n'avons donc pas pu développer de système automatique. Au lieu de cela, nous avons utilisé «EPOC Simulink EEG Importer» qui est un package qui permet à un utilisateur de collecter automatiquement les données EEG via Simulink dans l'environnement Matlab. Ce package est également très facile à utiliser. Tout d'abord, un fichier exe est exécuté afin de connecter le casque avec le Simulink comme le montre la figure 6. Une fois la connexion établie, un fichier Simulink est appelé via un code Matlab simple et collecte de nouvelles données à chaque fois comme le montre la figure 7 .



Figure 6 Serveur de signaux simulink EPOC

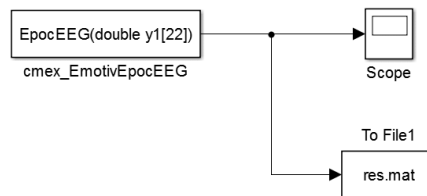


Figure 7 Un modèle simulink simple pour la collecte de données EEG

2-6 Optimisation et mise en œuvre de la machine à vecteur de support :

La machine à vecteurs de support ou SVM est un ensemble de méthodes d'apprentissage automatique qui sont utilisées pour classer différentes caractéristiques de données, ou aussi la régression et la détection des valeurs aberrantes [2]. Contrairement à d'autres méthodes d'apprentissage automatique telles que l'arbre du modèle logistique (LMT) ou le classificateur quadratique [9 & 10], SVM est une technique de classification non linéaire et non paramétrique basée sur le noyau qui a montré de bonnes applications de classification en analyse médicale, en reconnaissance de caractères ou en charge électrique prévisions et autres domaines de classification. SVM montre de très bons résultats sur la classification des données avec une grande dimensionnalité et il peut prendre en charge la «malédiction de la dimensionnalité» [11]. Dans [11], l'auteur a souligné que «la malédiction de la

dimensionnalité est un phénomène bien connu mais pas entièrement compris. Trop de données, en termes de nombre de variables d'entrée, n'est pas toujours une bonne chose». De plus, pour une seule classe, une classe inconnue ou des classes significativement déséquilibrées, le problème de classification devient très compliqué. Dans cette thèse, bien que nous n'ayons pas affaire à une classification non équilibrée ou à une classe, notre problème vient de la crédibilité des données. Comme les données sont de type EEG, le problème de classification est normalement de grande dimension et compliqué. Par conséquent, la méthode SVM peut être la solution optimale pour le problème lié aux données EEG.

De nombreuses recherches ont été menées pour les applications de SVM dans l'analyse de données EEG. Par exemple, dans [12], une expérience est mise en place pour distinguer l'imagination du mouvement de la main droite et de la main gauche. Pour mener cette expérience, on a demandé à trois sujets d'imaginer les mouvements gauche et droit pendant 1,5 seconde avec un intervalle de 10 secondes. Il y a eu 160 données de mouvements manuels enregistrées. Ces données ont ensuite été converties en densité de spectre de puissance (PSD) et des bandes de fréquences de 9 à 14 Hz sont utilisées pour la classification. Les résultats ont montré que 80% de la précision des tests peuvent être atteints. Cela a montré la possibilité d'utiliser SVM comme classificateur pour EEG [12]

Les données. Cependant, du point de vue statistique, la petite taille de l'ensemble de formation, comme un ensemble de 160 données, peut ne pas être très convaincante pour tirer une conclusion. Par conséquent, dans ce projet, nous allons créer un grand nombre de données de formation.

2-7 Mathématique du classificateur SVM :

a) Concept de base de SVM

Nous utilisons un classificateur SVM linéaire de deux classes pour montrer son concept de base. Un classificateur SVM à deux classes nécessite un ensemble de données d'apprentissage de l'entrée x et de la sortie y pour optimiser les paramètres dans le modèle SVM. Cet ensemble doit être une matrice X et Y pour:

$$D = \{(x_i) | x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}\}_{i=1}^{m_i} \quad (\text{II-1.1})$$

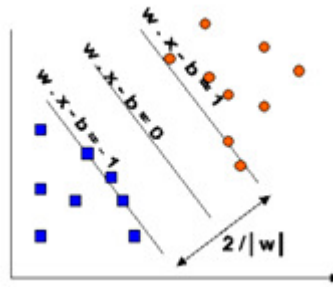


Figure 8 Un hyperplan simple divisant un ensemble de données

Où x_i est un vecteur ligne représentant une valeur de l'entrée mappée sur une étiquette ou une valeur de la sortie $y_i = 1$ ou -1 . Une hyperPlan peut être écrite comme un ensemble de points x vérifiant que:

$$\{x: (x) = w^T x + b = 0\} \quad (\text{II-1.2})$$

Cet hyperplan sépare l'ensemble de données dans deux régions différentes. La région supérieure recouvre l'ensemble positif tandis que la région inférieure recouvre l'ensemble négatif [13] comme le montre la figure 8.

b) Marge et optimisation

Il existe de nombreux hyperplans pouvant être dessinés pour séparer un ensemble d'entrées d'apprentissage x_i . Pour décider de la solution optimale et de l'hyperplan unique, certaines optimisations sont nécessaires. Comme le montre la figure 8, la distance entre deux hyperplans $w \cdot x - b = 1$ et $w \cdot x - b = -1$ est $2/|w|$. nous voulons maximiser cette distance en minimisant simplement la valeur de $|w|$ ou minimiser mathématiquement commodément $1/2||w||^2$ alors on trouve :

$$\min_w 1/2||w||^2 \quad (\text{II-1.3})$$

$$\text{conditionnée par: } w \cdot x_i - b \geq 1 \text{ ou } w \cdot x_i - b \leq -1 \quad (\text{II-1.4})$$

$$\text{ou } y_i (w \cdot x_i + b) \geq 1 \quad (\text{II-1.5})$$

Ce problème d'optimisation peut être résolu par le point de selle de la fonction de Lagrange:

$$Lp = Lw,, = 1/2\|w\|^2 \sum_{i=1}^m \lambda_i [y_i (w \cdot x_i + b) - 1] \quad (\text{II-1.6})$$

Pour résoudre le problème des extrema, nous prenons la différenciation partielle de L par rapport à w et b. Tout d'abord pour w, nous avons

$$d/d_w Lw,, = w - \sum_{i=1}^m \lambda_i y_i x_i = 0 \quad (\text{II-1.7})$$

ce qui nous donne: $w = \sum_{i=1}^m \lambda_i y_i x_i \quad (\text{II-1.8})$

Ensuite, prendre la différenciation partielle par rapport à b conduit à:

$$d/db Lw,, = \sum_{i=1}^m \lambda_i y_i \quad (\text{II-1.9})$$

Brancher (II-1.8) et (II-1.9) dans (II-1.6) nous donne la solution extrema:

$$\max \lambda = \sum_{i=1}^m \lambda_i - 1/2 \sum_{i=1}^m \sum_{j=1}^m \lambda_i \lambda_j y_i y_j x_i x_j \quad (\text{II-1.10})$$

où: $0 \leq \lambda_i \leq C \text{ for } i = 1, 2, \dots, m. \quad (\text{II-1.11})$

le choix de la valeur du facteur de pénalité C est important comme indiqué dans [14]. S'il est trop grand, nous avons une pénalité élevée pour aucun point séparable et nous pouvons stocker de nombreux vecteurs de support pour provoquer le problème de sur-ajustement. Si le facteur de pénalité C est trop petit, nous pouvons avoir le problème de sous-ajustement.

Lorsque le λ optimisé est résolu, nous pouvons obtenir la fonction de décision comme suit:

$$(x) = \text{sgn}(\sum_{i=1}^m \lambda_i y_i x_i + b) \quad (\text{II-1.12})$$

En utilisant (II-1.12), nous pouvons mapper n'importe quelle valeur de x en 1 ou -1.

c) Fonctions du noyau et classification non linéaire :

Pour certains problèmes complexes, les données d'apprentissage ne peuvent pas être simplement classées par un classificateur SVM linéaire. Afin de résoudre ces problèmes, l'entrée x peut être mappée dans un espace de dimension supérieure par la fonction de noyau Φ . Ensuite, par SVM, nous trouverons comment sélectionner un hyperplan linéaire avec la marge maximale dans ce nouvel espace dimensionnel. Il existe de nombreuses fonctions de noyau populaires.

Le noyau RBF sera utilisé dans ce mémoire pour les raisons principales suivantes:

1. Les échantillons seront cartographiés dans un espace de dimension supérieure;
2. Il est moins coûteux en calcul que le noyau polynomial car moins les paramètres doivent être trouvés
3. C'est le meilleur noyau qui peut faire face au problème visé.

2-8 Bibliothèque LibSVM et optimisation des paramètres :

a) Bibliothèque LibSVM

LIBSVM est une bibliothèque pour les machines à vecteurs de support (SVM). Cette bibliothèque a été développée depuis l'an 2000. Les principaux objectifs sont d'aider les utilisateurs à appliquer facilement SVM à leurs applications. Cette section présente brièvement l'utilisation de LibSVM pour résoudre les problèmes SVM.

LibSVM fournit les deux fonctions les plus importantes de l'environnement Matlab nommées «svmtrain» et «svmpredict». Tout d'abord, nous présentons la fonction svmtrain. Voici une manière typique d'utiliser cette fonction:

```
model = svmtrain(y_train, x_train, Parameters)
```

Cette fonction nous aide à générer un modèle de classification, compte tenu des données de formation x_train et y_train et des paramètres, où x_train est une matrice $m \times n$ et y_train est un vecteur $m \times 1$ avec chaque composant

correspondant à un vecteur ligne de x_{train} . Le paramètre a sa propre forme comme suit:

Paramètres = ['-c' C_value '-g' gamma_value '-w1' w1_val '-w2' w2_value '-b 0'];

C_value et gamma_value représentent les valeurs des paramètres C et γ que nous choisissons.

w_1 et w_2 sont le poids des classes. Habituellement, s'il y a 2 classes et que le nombre de données dans chaque classe est le même, w_1 et w_2 sont égaux à 0,5. Dans cette thèse, nous considérons uniquement le cas où chaque classe a le même nombre de données d'apprentissage, c'est-à-dire que l'ensemble de données est équilibré. En tant que tel, les valeurs de poids sont toutes égales les unes aux autres en utilisant la règle $w_i = 1/n$

for ($i = 1, \dots, n$) où n est le nombre de toutes les valeurs possibles de y ou le nombre de classes. '-b' est l'estimation de probabilité la valeur par défaut est 0, ce qui signifie que nous ne sortons aucune probabilité de chaque classe dans cette thèse. Maintenant, nous allons introduire la fonction `svmpredict`. Voici une manière typique d'utiliser cette fonction:

```
[group_train, acc_train, val_train] = svmpredict(y, x, model, '-b 0');
```

Dans cette fonction, le «modèle» a été obtenu à partir de la fonction `svmtrain`. Cette fonction est utilisée pour calculer à quelle classe appartient un vecteur ligne de la matrice x . Si les vraies valeurs de la sortie y sont disponibles, cette fonction peut également fournir le résultat de précision par le nombre de lignes correctement classées dans la matrice x divisé par le nombre total de données.

b) Optimisation des paramètres C et γ

Comme discuté à la fois dans l'article du Dr Lin [6] et dans la thèse de Qing [8], pour avoir une précision de classification plus élevée, nous devons optimiser les paramètres C et γ avant d'optimiser λ dans le modèle SVM.

Il serait coûteux en calcul si nous voulons trouver les C et γ optimaux. Pour équilibrer la quantité de calcul et une précision de classification plus élevée, nous utiliserons la méthode de recherche par grille cinq fois pour les meilleurs

C et γ dans une plage. La recherche dans la grille commence par un ensemble grossier:

Pour C : $\{2^{-5}, 2^{-5+h}, \dots, 2^{-5+20h}\}$

Pour γ : $\{2^{-10}, 2^{-10+h}, \dots, 2^{-10+15h}\}$

Où h est la taille du pas et est égal à 1 au début. À chaque fois de la recherche, une procédure de validation croisée de cinq fois est utilisée pour évaluer une paire de C et γ . Dans la validation croisée, le total des données d'entraînement est divisé en cinq volets. Quatre plis sont utilisés pour former le modèle tandis que le pli restant est utilisé pour tester le modèle sous la paire donnée de C et γ .

Par conséquent, nous avons cinq précisions de test et nous en prenons la moyenne. Pour optimiser les paramètres et réduire les coûts de calcul, une valeur limite de la précision de la formation est fixée à 80%.

Si la précision d'entraînement d'une paire de C et γ est inférieure à cette valeur, cette paire ne peut pas être la meilleure et nous rejetons simplement cette paire. À chaque recherche, la meilleure paire de C et γ avec la précision de test moyenne la plus élevée est choisie. Après une recherche, la taille de pas h est divisée par deux et ainsi une recherche de grille plus fine à la prochaine fois peut être effectuée en dix pas autour de C et γ obtenus à partir de la recherche précédente. Enfin, la meilleure paire de C et γ est trouvée après cinq recherches.

2-9 Le Fonctionnement d'entrée de SVM pour un système BCI basé sur EEG :

Dans notre système BCI basé sur EEG développé, la caractéristique ou l'entrée entrant directement dans le modèle SVM n'est pas des signaux EEG dans le domaine temporel. Au lieu de cela, nous convertissons ces signaux en densité de spectre de puissance (PSD) dans le domaine fréquentiel comme caractéristique de classification. L'algorithme typique utilisé pour cette conversion du domaine temporel au domaine fréquentiel est la transformée de Fourier rapide (FFT).

FFT est un algorithme pour calculer la transformée de Fourier discrète (DFT) [16]. Il peut réduire l'ordre des calculs de $2N^2$ à $2N \log_2(N)$, où N est le nombre de calculs nécessaires.

La fonction ci-dessous montre la DFT:

$$X(k) = \sum_{n=1}^N x[n] \exp(-j\omega kn) \quad (\text{II-2.1})$$

$$\text{Telle que } \omega_k = \frac{2\pi}{N} K \quad (\text{II-2.2})$$

PSD représente la puissance du signal du domaine temporel à différentes fréquences. L'équation pour calculer la PSD d'un signal temporel discret est:

$$\begin{aligned} P(k) &= ((\Delta t)^2/T) \left| \sum_{n=1}^N x[n] \exp(-j\omega kn) \right|^2 \\ &= (\Delta t/N) \left| \sum_{n=1}^N x[n] \exp(-j\omega kn) \right|^2 \\ &= (1/F_s * N) \left| \sum_{n=1}^N x[n] \exp(-j\omega kn) \right|^2 \\ &= (1/F_s * N) |X(k)|^2 \quad (\text{II-2.3}) \end{aligned}$$

$$\text{Où } [n] = (n * \Delta t), T = N * \Delta t, F_s = 1/\Delta t$$

L'équation (II-2.3) sera utilisée dans le codage Matlab pour le calcul FFT. La fenêtre rectangulaire est utilisée dans le calcul FFT ci-dessus. Cependant, pour éviter la fuite de puissance dans le calcul, la fenêtre de hamming sera utilisée à la place dans notre recherche, qui a pour fonction:

$$h(n) = 0.54 - 0.46 \cos(2\pi(n/N)) \quad (\text{II-3})$$

2-10 Programmes auto-développés utiles pour le système BCI basé sur EEG :

2-10-1 Programme de collecte de données dans le domaine temporel EEG :

Dans notre recherche, un programme est créé pour collecter automatiquement des données de domaine temporel EEG de haute qualité d'un casque EPOC vers un PC au format compatible avec Matlab. Tout d'abord, les données EEG brutes

est tiré du casque EPOC. Une fois le casque EPOC connecté au serveur de signal EPOC Simulink. Un programme Matlab «TakingData.m» est exécuté pour démarrer le processus de collecte de données. Selon l'endroit où se trouve ce programme, il peut automatiquement changer le répertoire de travail dans le répertoire du dossier qui le contient. Tout d'abord, l'utilisateur sera invité à saisir des informations générales sur le type de données. Les informations comprennent: le nombre d'éléments de formation, le nombre de types de données et le temps en secondes pour chaque type de données. Le code appellera ensuite le modèle Simulink comme illustré à la figure 7 et exécutera le modèle Simulink pour le même temps de formation entré par l'utilisateur comme illustré à la figure 9. Le temps de formation pour une donnée sera d'une seconde dans l'exemple de cette figure. Selon le nombre de classes, l'utilisateur sera invité à saisir des noms différents pour les classes (par exemple, gauche, droite), comme illustré à la figure 10. Si les dossiers portant les mêmes noms que les noms de ces classes n'existent pas dans le répertoire de travail, de nouveaux dossiers seront créés.

Le programme demandera alors à l'utilisateur de se préparer à prendre les données de manière aléatoire. Par exemple, s'il existe deux classes de lettres A et B, le programme demandera à l'utilisateur de penser au hasard pour A et B et affichera une boîte de dialogue de message comme le montre la figure 11. Une fois que l'utilisateur a pensé à une classe pour un à une heure précise, le programme vérifiera le sentiment de l'utilisateur quant à la qualité des données en utilisant la boîte de dialogue de questions comme illustré à la figure 12. Si la réponse est «Oui», le programme collectera les données spécifiques et les placera dans un dossier spécifique. Cependant, si la réponse est «Non», le programme demandera à l'utilisateur de reprendre les données

spécifiques en appelant la boîte de message illustrée à la figure 11. Les données correspondant au nom du dossier seront nommées dans un ordre numérique (par exemple «Time_dataLeft0» si le nom de la classe est 'Left'). En comptant le nombre de fichiers dans chaque dossier à chaque fois, le programme gardera une trace de l'ordre numérique. Même lorsque l'utilisateur ferme le programme pour se reposer, la prochaine fois qu'il ou elle veut prendre de nouvelles données, le programme reprendra l'ordre numérique du nom des nouvelles données. Cela évitera le chevauchement, ce qui entraînera la perte de données. Un exemple des données prises pour les lettres A et B dans leurs répertoires de travail est illustré à la figure 13.

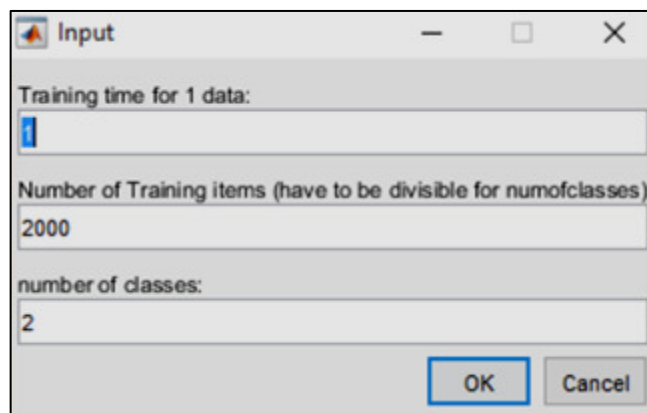


Figure 9 Le programme demande à l'utilisateur de saisir des informations générales

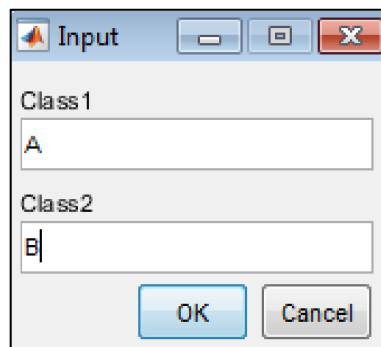


Figure 10 Le programme demande à l'utilisateur de saisir les noms des classes

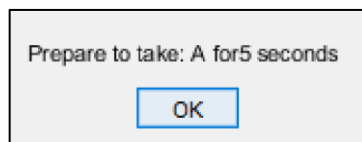


Figure 11 Le programme demande à l'utilisateur d'utiliser l'EPOC et de réfléchir à la classe spécifique

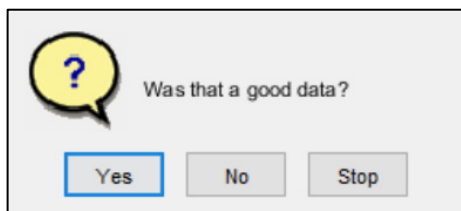


Figure 12 Le programme vérifie le sentiment de l'utilisateur quant à la qualité des données

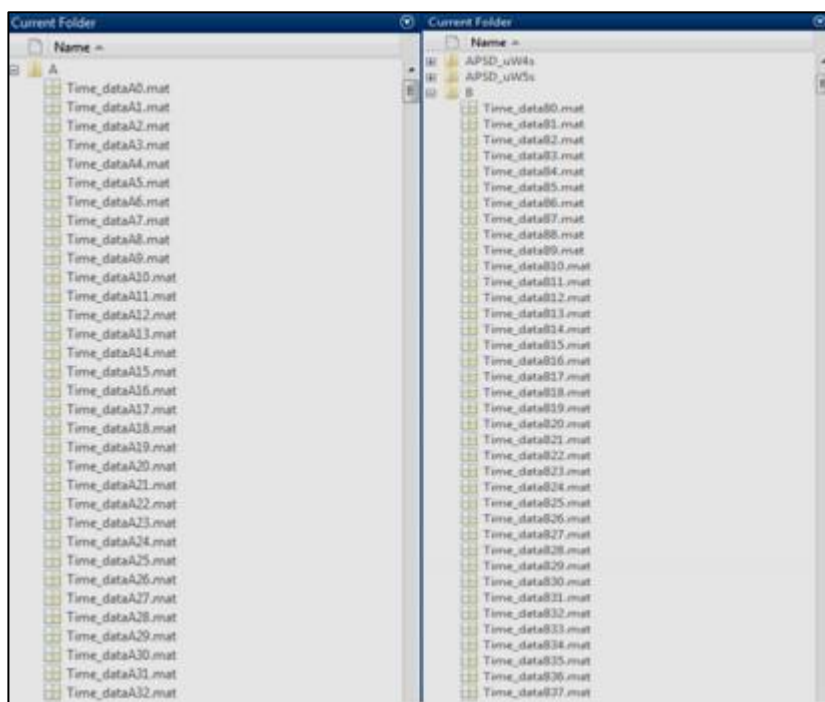


Figure 13 Données EEG du domaine temporel dans le répertoire de travail

2-10-2 Programme de conversion des données EEG du domaine temporel vers le domaine fréquentiel :

Comme discuté dans la dernière partie, dans cette thèse, les données EEG seront converties au format PSD.. Afin de faciliter le travail de conversion des données cérébrales, le programme Matlab «CovertData.m» est développé. Le but du code est de convertir toutes les données du domaine temporel précédemment enregistrées dans le répertoire de travail en

PSD dans le domaine fréquentiel. Tout d'abord, l'utilisateur doit placer ce code Matlab dans le même répertoire où se trouvent les données du domaine temporel. Lorsque le code est en cours d'exécution, une boîte de dialogue de saisie apparaîtra et demandera à l'utilisateur de saisir le nombre de classes comme indiqué dans la figure 14. Ensuite, l'utilisateur doit saisir le nom des dossiers dans lesquels les données temporelles sont contenues, comme indiqué dans la figure 15. Ici, "A" et "B" seront saisis ici à titre d'exemple. Notez que «A» et «B» représentent deux dossiers qui contiennent les données EEG du domaine temporel pour les lettres A et B et se trouvent dans le même répertoire de travail. Si ces dossiers ne se trouvent pas dans le répertoire de travail, le programme affichera des erreurs et se fermera.

Une autre boîte de dialogue apparaîtra et expliquera à l'utilisateur ce qu'il faut faire ensuite. puis, appuyez simplement sur «OK» pour continuer. Une autre boîte de dialogue d'entrée apparaîtra comme illustré à la figure 16. Ici, si l'utilisateur modifie la valeur de l'entrée de 0 à 1, cela signifie que l'utilisateur souhaite convertir les données du domaine temporel en ce type de données spécifique. Deux types de choix de conversion de données sont disponibles: «Données PSD en Db» et «Données PSD en uW» où les données PSD en Db sont fondamentalement $10\log_{10}$ de données PSD en uW. Cette boîte de dialogue d'entrée sert aux mêmes fins que la case à cocher, qui n'est pas incluse dans les modèles de boîte de dialogue de base de Matlab. Pour ce projet, puisque toutes les données durent 5 secondes, ce programme ne servira qu'à ces données. Une fois le type des données converties choisi, le programme démarre.

Tout d'abord, il créera les nouveaux répertoires comme illustré à la figure 17. Par exemple, si «A» et «B» sont le nom des dossiers contenant des données de domaine temporel et que l'utilisateur souhaite les convertir en

PSD données dans uW, il y aura 10 nouveaux répertoires créés APSD_uW1s-APSD_uW5s et BPSD_uW1s-BPSD_uW5s. C'est là que se situent les données PSD de 1s-5s pour chaque donnée de domaine temporel. Cela se fera en bouclant 5 fois pour créer cinq fenêtres de blocage différentes et en coupant différentes informations à partir d'une seule donnée temporelle. Pour être plus précis, les données PSD 1s sont converties à partir de données de domaine temporel 0-1s, les données PSD 2s sont converties à partir de données de domaine temporel 0-2s jusqu'à ce que les données PSD 5s soient converties à partir de données EEG de domaine temporel 0-5s. Avant de convertir les données, le programme gardera une trace du nombre de données déjà présentes dans chaque répertoire. Cela aidera le programme à éviter de calculer les données PSD existantes et à gagner ainsi beaucoup de temps. Le processus de conversion des données prend normalement environ 15 minutes pour 1000 essais de données dans le domaine temporel.

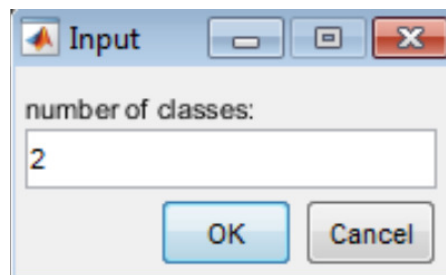


Figure 14 Insertion du nombre de classes à convertir

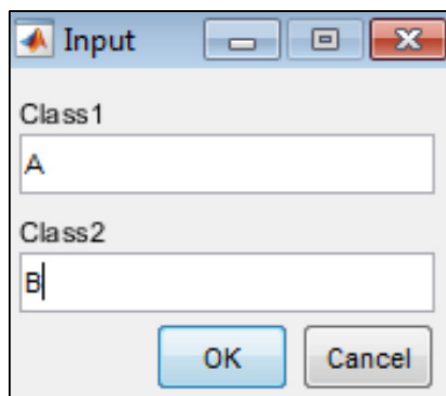


Figure 15 Insertion du nom des dossiers de données du domaine temporel

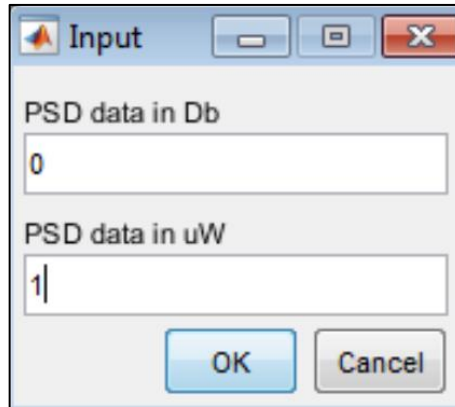


Figure 16 Choix du type de données à convertir en dossier suivent .

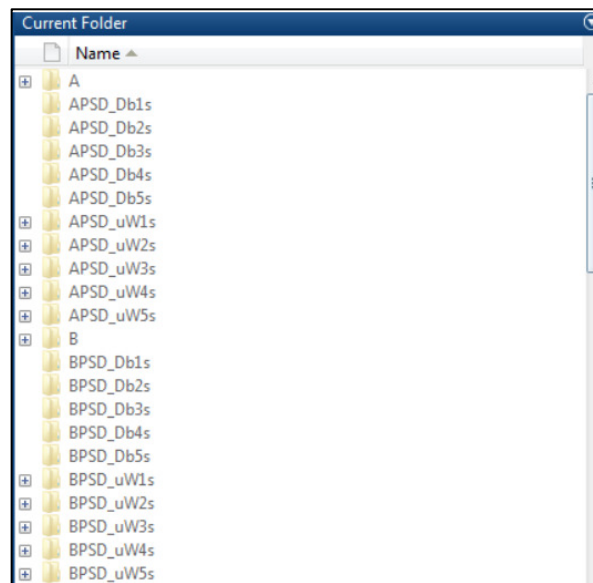


Figure 17 De nouveaux répertoires sont créés

2-10-3 Programme de création de modèle SVM :

Dans ce projet, le programme le plus critique est celui qui peut résoudre les problèmes de classification multi-classes. Nous présenterons un tel programme dans cette section. Ce programme est nommé «svm_train_n_categories.m» qui est présenté. Premièrement, puisque dans notre projet, chaque classe

a la même quantité de données, c'est-à-dire que l'ensemble de données est équilibré, notre programme est développé pour être capable de faire une classification en plusieurs classes pour ce cas. Le programme a deux fonctions principales. La première fonction est de former le modèle SVM en entraînant des données et de sortir le résultat de la formation dans un fichier texte. La

deuxième fonction consiste à créer le modèle SVM bien formé et à l'enregistrer au format .mat. Ce modèle bien formé peut être utilisé plus tard pour contrôler la voiture robot. Lorsque l'utilisateur exécute ce programme, la boîte de dialogue ci-dessous s'affiche. Le premier choix «analyse d'ensemencement» est pour la formation SVM, tandis que le deuxième choix «MakingPrediction model» est utilisé pour générer le modèle SVM bien formé.

Ensuite, nous décrivons ces deux choix

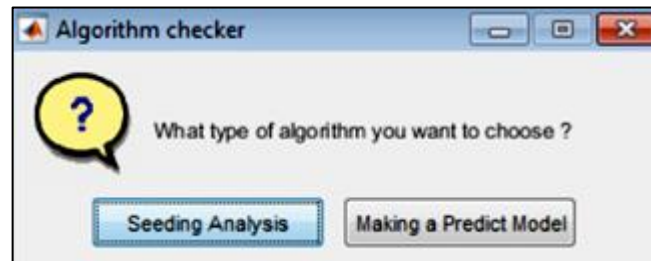


Figure 18 Le programme demande à l'utilisateur de choisir entre 2 options

a) Analyse d'ensemencement :

Si l'utilisateur choisit «Analyse d'amorçage» dans la boîte de dialogue, la nouvelle boîte de dialogue apparaîtra, vous demandant le nom du fichier texte (Figure 19). Ce fichier texte est situé dans le même répertoire de travail où se trouve le programme et le résultat de la formation SVM est stocké. Par défaut, le fichier texte sera généré automatiquement en fonction de la date et de l'heure de la simulation; les utilisateurs peuvent changer le nom s'ils le souhaitent. Lorsque le bouton "Ok" est cliqué, le fichier texte sera généré dans le programme vous demandera alors des informations avant de lancer le processus de formation.

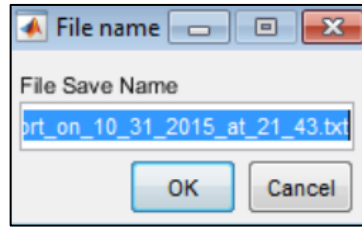


Figure 19 Le nom du fichier texte

Tout d'abord, le programme demande la limite de précision des tests que l'utilisateur souhaite voir (figure 20). Si la précision de test d'une paire de paires C et γ est inférieure à cette limite, elle n'apparaîtra pas dans le fichier texte (figure 21). Le fichier texte de sortie illustré à la figure 21 comporte 6 colonnes.

De droite à gauche se trouvent les valeurs de la graine, du meilleur paramètre C , du meilleur paramètre gamma, de la précision de l'entraînement, de la précision des tests et enfin du nombre de vecteurs de support. Dans cet exemple de sortie, seules quelques graines ayant une précision de test supérieure à 50% seront affichées. Tout résultat inférieur ou égal à 50% sera ignoré.

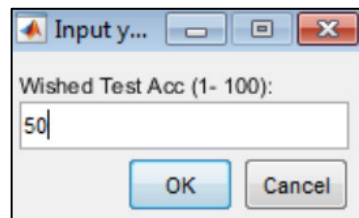


Figure 20 Insérez la borne inférieure de la sortie

1	2	3	4	5	6	7	8
Seed	Best_C	Best_gamma	Train_acc	Test_acc	Total_SV		
4	0.73841	10.3747	0.90156	0.55625	631		
10	1512.27	0.1197	0.82969	0.55	522		
13	0.56939	7.025	0.80469	0.58125	631		
15	2	2.3784	0.775	0.55	607		
26	0.64842	9.1103	0.86719	0.58125	628		
27	0.54525	7.6608	0.80781	0.55	628		
32	0.67713	5.1874	0.79219	0.55	614		
37	0.56939	7.336	0.78906	0.63125	634		
44	0.5946	10.3747	0.875	0.5625	634		
45	0.4788	17.4481	0.71094	0.55625	639		
57	0.52214	13.4543	0.85	0.56875	637		
62	0.5	17.4481	0.74687	0.55625	639		
74	0.52214	10.3747	0.83594	0.5875	635		
75	469.5061	0.4585	0.9125	0.55625	511		
76	0.5946	5.6569	0.79531	0.55	622		
80	0.52214	15.3217	0.875	0.55	638		

Figure 21 L'exemple d'un fichier de sortie

Ensuite, le programme demandera le nombre de classes pour la formation. L'utilisateur doit saisir les informations comme suit: nombre de dossiers (classes), taille des données (nombre de données dans chaque dossier) et nombre de couches (figure 22). Par défaut, le nombre de couches sera 1, ce qui signifie que toutes les données seront utilisées pour la formation. Lorsque le nombre de couches est modifié, l'ensemble de données est divisé en différents groupes. Par exemple, si la taille des données est de 1000 et le nombre de couches est de 10, alors pour chaque couche, il y aura 100 données uniquement. Ces 10 couches seront utilisées pour former le modèle SVM séparément, afin que l'utilisateur puisse comparer la qualité des données dans chaque couche. C'est important car en réalité, il est difficile de savoir si les données cérébrales prises sont bonnes pour l'entraînement ou non. L'utilisateur ne connaît la qualité de ses données qu'une fois la formation terminée, ce qui peut prendre beaucoup de temps. Par exemple, cela peut prendre au moins 4 heures pour former le modèle SVM avec les données Gauche contre Droite, qui sont des données de haute qualité. Si la qualité des données est faible, un résultat peut prendre environ 1 jour de fonctionnement. En exécutant l'algorithme SVM couches par couches, l'utilisateur peut avoir une meilleure idée des données et supprimer une section de données de mauvaise qualité si nécessaire. La figure 23 montre le fichier de sortie si l'utilisateur choisit le nombre de couches différent de 1. Comme le montre la figure 23, les première et deuxième couches ont toutes deux une précision de test similaire de 75 à 85%, ce qui signifie que ce sont de bonnes données. Normalement, le temps d'exécution pour obtenir ce fichier de sortie est

beaucoup plus court (1-2 heures) par rapport au temps d'exécution pour la taille de données complète (1000 données dans ce cas).

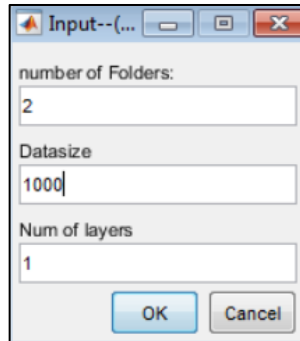


Figure 22 Entrez le nombre de dossiers, la taille des données et le nombre de couches

1	Seed	Best_C	Best_Sigma	Train_acc	Test_acc
2	Result of 50 data 1-2-3-4-5-10-11-12-13-14channel 4-14Hz 0-49 ID Left Right				
3	2	3.221	0.27263	0.9875	0.75
4	16	1.3543	0.54525	0.975	0.75
5	50	1.1892	0.14235	0.8	0.75
6	130	4.7568	0.2973	0.975	0.75
7	141	0.03125	0.25	0.85	0.8
8	160	2.4837	0.19278	0.9125	0.75
9	177	2.181	0.26107	0.9625	0.85
10	Result of 50 data 1-2-3-4-5-10-11-12-13-14channel 4-14Hz 50-99 ID Left Right				
11	1	4.362	0.22925	0.9625	0.75
12	14	5.6569	0.31046	0.9875	0.8
13	29	90.5097	0.022097	0.95	0.75
14	61	8.7241	0.096388	0.9625	0.75
15	111	9.5137	0.17678	0.975	0.8

Figure 23 la sortie couches par couches

Une fois que l'utilisateur a cliqué sur «OK» pour continuer, le programme affichera la même boîte de dialogue que celle illustrée à la figure 15. L'utilisateur devra saisir le nom de l'ensemble de données de formation qui est des données au format PSD dans notre projet. Le programme demandera alors les choix de canaux (Figure 24). Les numéros de 1 à 14 sont 14 canaux physiques sur le casque EPOC. Dans les processus de saisie et de conversion des données, les données EEG brutes d'EPOC sont utilisées comme entrée. En tant que tel, les informations pour les 14 canaux sont toujours contenues dans les données PSD mais dans le domaine fréquentiel. Pour choisir différents canaux pour l'ensemble de données de formation, l'utilisateur doit placer le tiret «-» entre le nombre, le programme convertira cette chaîne en nombres entiers et supprimera les canaux sélectionnés pour la formation SVM. Les entrées doivent être comprises entre 1 et 14 uniquement; si l'entrée utilisateur

est hors de portée, le programme signalera une erreur et demandera à nouveau la sélection du canal. Enfin, le programme demandera alors la gamme de fréquences. Dans ce projet, puisque l'EPOC avec 128 fréquences d'échantillonnage est utilisé, seules 64 fréquences de 0 à 64 Hz seront analysées. La boîte de dialogue de la figure 25 demande à l'utilisateur d'entrer les fréquences de démarrage et d'arrêt pour former l'ensemble de données d'entraînement. Les entrées doivent être comprises entre 0 et 64 Hz, si elles sont hors limites, le programme signalera à nouveau les erreurs et demandera de nouvelles entrées. Les deux options pour les canaux et la gamme de fréquences donnent à l'utilisateur une flexibilité dans la recherche. L'utilisateur peut comparer la précision des tests ou sélectionner la plage de fréquences et les canaux optimaux. Appuyer sur «OK» pour continuer, le programme créera l'ensemble de données d'entraînement et commencera l'entraînement. Le processus de formation est illustré dans le diagramme ci-dessous (figure 26).

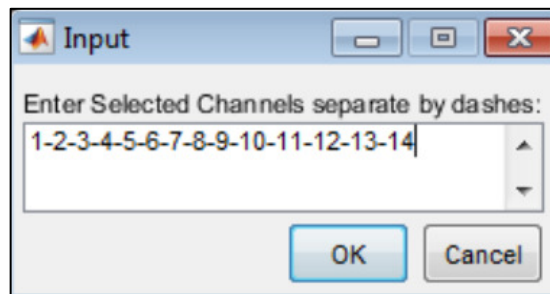


Figure 24 Entrez la plage de canaux

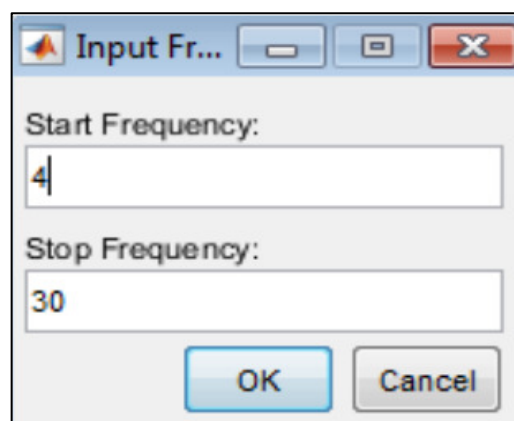


Figure 25 Plage de fréquences d'entrée

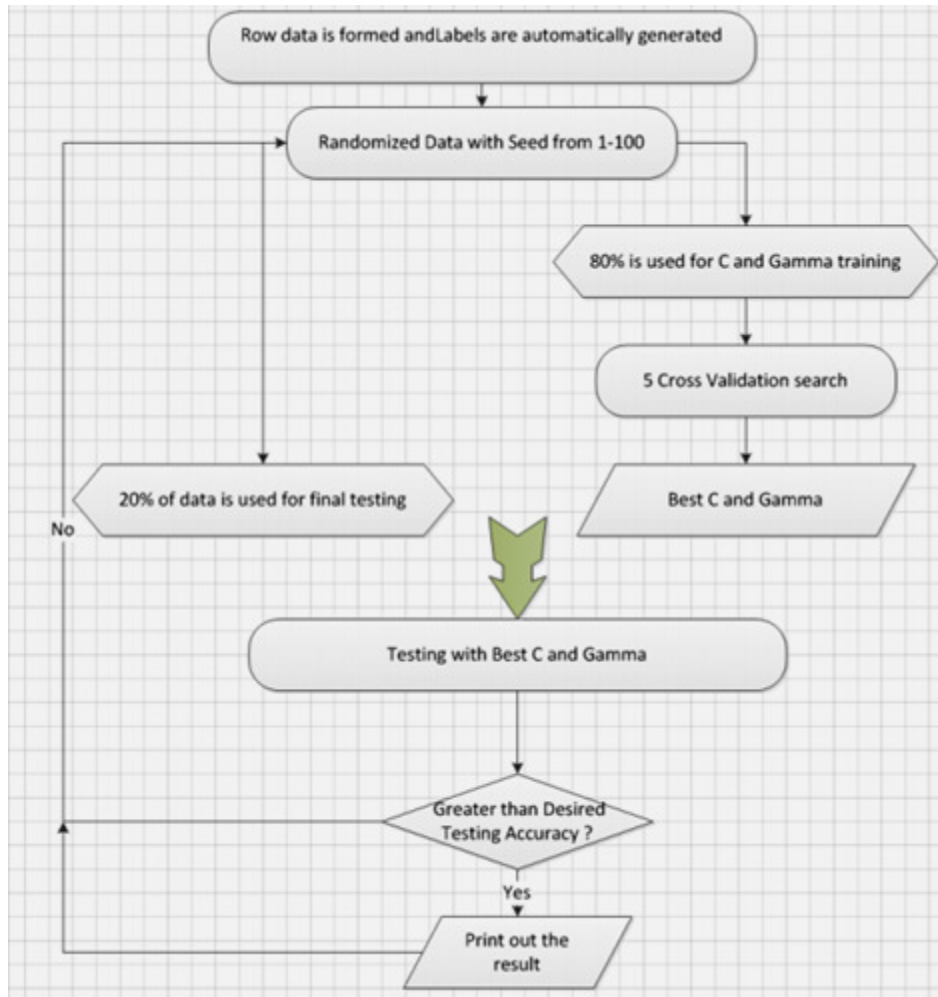


Figure 26 Processus de formation SVM

Les données PSD sont enregistrées sous la forme d'une matrice $m \times n$ où les lignes m est le nombre de canaux et les colonnes n est le nombre de fréquences discrètes de 0 à 64Hz. chaque entrée dans les données d'apprentissage doit être dans un vecteur ligne. Ce vecteur sera formé comme un vecteur de ligne $1 \times (m \times n)$ en plaçant chaque ligne de la matrice l'une à côté de l'autre.

Chacune de ces lignes de données correspondra à une étiquette. Cette étiquette est générée automatiquement en fonction du nombre de classes. Les étiquettes dépendront de l'ordre des entrées, la première entrée est toujours mappée à 1, puis l'étiquette augmente de 1 pour les classes suivantes. Par exemple, comme le montre la figure 15, la classe A sera mappée sur l'étiquette 1 et la classe B est mappée sur l'étiquette 2. S'il y a C, D et E, elles sont mappées sur les étiquettes 3, 4 et 5, respectivement. Les données sont ensuite randomisées en utilisant la valeur des graines de 1 à 100 (qui peut être

modifiée dans le code). La raison pour laquelle les graines sont utilisées est la suivante: supposons que nous avons un ensemble de données de 1000 données, où seulement 80% de ces données (800) sont choisies pour la formation SVM. Pour calculer tous les cas possibles pour former ces 800 données sur l'ensemble des 1000 données, nous utilisons l'équation de combinaison:

$$\binom{1000}{800} = \frac{1000!}{800! * 200!}$$

où les deux mille! et 800! sont trop gros pour être calculés dans Matlab, et le résultat peut également être un grand nombre. Par conséquent, il est pratiquement impossible de vérifier tous ces cas et nous devons randomiser les données plusieurs fois pour obtenir le meilleur résultat de tous les essais. L'utilisation de semences dans Matlab nous avons contrôler la randomisation et, à ce titre, peut réformer l'ensemble de données d'apprentissage pour une semence donnée. Un problème ici est que la fonction «svmtrain» nécessite une chaîne comme paramètres. Cette chaîne est écrite différemment selon le nombre de classes. Pour ce faire, nous devons normalement modifier manuellement cette seule ligne de code, ce qui réduit la flexibilité du programme. La solution est de laisser

le programme écrit lui-même ces lignes de code. Ces lignes spéciales du code sont écrites dans trois autres fichiers Matlab nommés writeFPara.m, writePara.m et writeweight.m. Ces fichiers sont modifiés plusieurs fois au cours du training en fonction du nombre de classes, de la valeur C et de la valeur gamma. Une fois les meilleurs paramètres C et γ trouvés, le programme utilisera ces paramètres pour l'ensemble de données de test final (20% des données). Si la précision du test est supérieure à la précision de test souhaitée (Figure 20), ce résultat est imprimé dans le fichier texte. Le programme passera à la graine suivante et recommencera toutes les étapes ci-dessus. La formation sera terminée une fois toutes les graines utilisées. Pratiquement, pour certains types de données, le temps de formation d'une graine peut être des heures ou des jours. En tant que tel, l'utilisateur doit contrôler cela manuellement en modifiant les paramètres My Seed.

b) comment créer un modèle de prédiction :

Si l'utilisateur choisit l'instruction «Créer un modèle de prévision» dans la boîte de dialogue de la figure 18, la même boîte de dialogue que celle illustrée à la figure 24 apparaît. Le programme demande alors les dossiers où se trouvent les données (figure 15). Ensuite, le programme demande la graine utilisée pour chaque couche. Si l'entrée de valeur de couche est 1, une seule graine est nécessaire. Cependant, si la valeur de la couche est un nombre n , supérieur à 1, l'utilisateur est invité à saisir n graines (figure 27). Cette graine est utilisée pour reconstruire l'ensemble de données de formation comme discuté dans la section précédente. S'il s'agit d'une analyse multicouche, les meilleures données de chaque couche sont extraites pour former l'ensemble d'apprentissage. Il s'agit d'une fonction supplémentaire implémentée mais pas nécessairement utilisée pour notre.

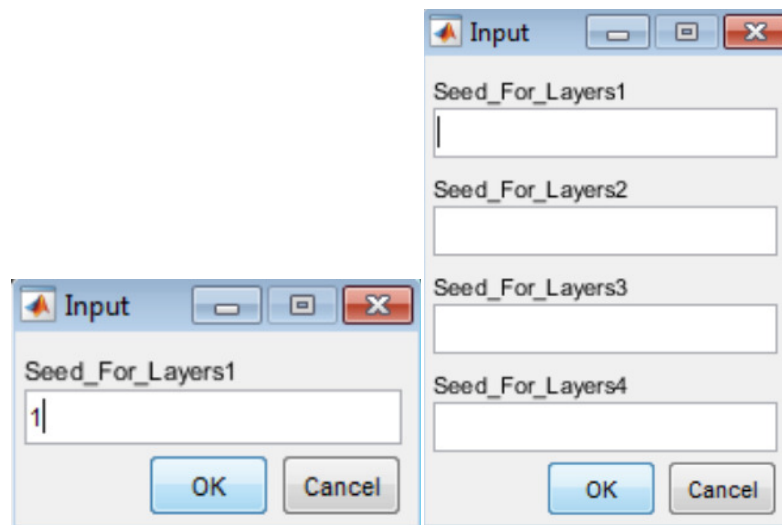


Figure 27 seed = 1 dans la boîte de dialogue de gauche, seed = 4 dans la boîte de dialogue de droite

Ensuite, les mêmes boîtes de dialogue des figures 24 et 25 s'afficheront. L'utilisateur saisira les mêmes informations qu'il ou elle utilise pour le cas d'analyse d'amorçage pour réformer la structure des données de formation.

Cela prendra 1 à 2 minutes pour la reconstruction des données, selon la taille des données. Le programme affichera alors trois options possibles (figure 28). La première option optimiseur C est utilisée lorsque l'utilisateur connaît la meilleure valeur de gamma et recherche uniquement C. À l'inverse, l'optimiseur gamma est utilisé lorsque l'utilisateur connaît le meilleur résultat pour C et recherche gamma. Encore une fois, ces 2 options sont construites dans ce programme mais non utilisées pour notre travail. La dernière option «Créer un modèle» demande alors à l'utilisateur de saisir les valeurs des meilleurs paramètres C et gamma (Figure 29). Ce sont les valeurs indiquées dans les colonnes 2 et 3 de la sortie du fichier texte (figure 21). Cliquez sur "OK" pour continuer, le programme commencera à recréer le modèle SVM bien formé qui a la même formation et la même précision de test que celle montrée dans le fichier texte. Il faudrait plusieurs secondes à une minute pour que le modèle soit recréé. Enfin, le programme demande à l'utilisateur d'enregistrer le modèle dans un répertoire souhaité (figure 30). Le répertoire par défaut sera le dossier Paramètres situé dans le même répertoire de travail.

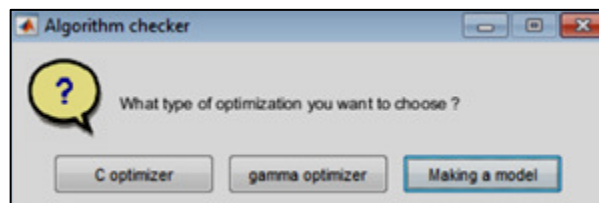


Figure 28 Choix entre 3 options

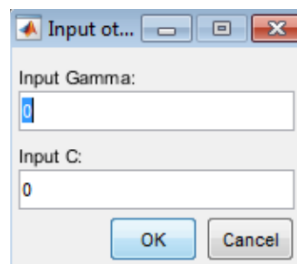


Figure 29 Saisie des meilleurs C et gamma

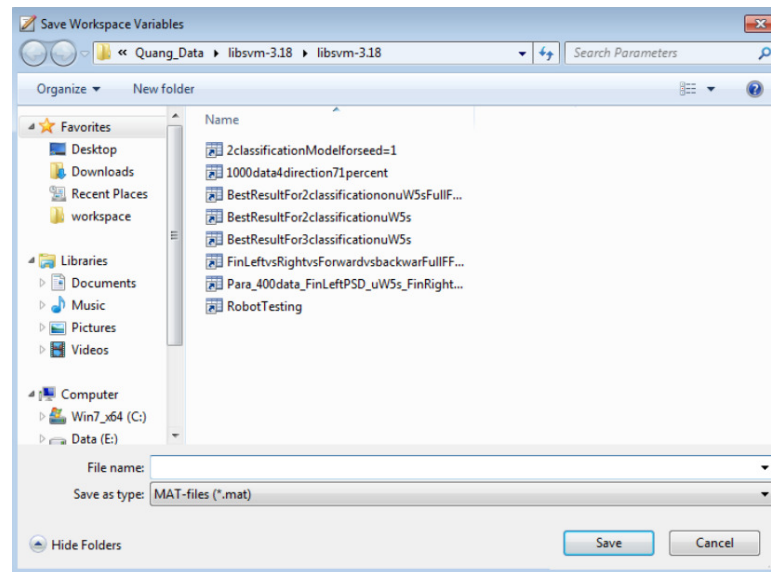


Figure 30 Enregistrement du modèle final dans un emplacement

Programmes de contrôle du robot :

Dans cette section, nous allons montrer le programme "RobotControl.m" pour établir l'interface entre l'ordinateur et le robot. Pour ce faire, nous devons utiliser une partie de Matlabpackages traitant de la communication série ainsi que du contrôle Arduino. Contrairement à d'autres types de données, la création d'un système en temps réel pour l'EEG est difficile car il n'y a pas de retour entre l'utilisateur et le système. pour la reconnaissance faciale, les utilisateurs peuvent avoir les retours (images webcam en temps réel) du système et en même temps les résultats des classifications. Ainsi, les utilisateurs peuvent contrôler leur émotion faciale pour obtenir le bon résultat de classement. Ce n'est pas le cas pour le signal EEG. Puisqu'il n'y a aucun moyen de surveiller les états du cerveau à en même temps qu'il contrôle le robot, ce modèle n'est qu'un contrôleur en boucle ouverte sans rétroaction.

En tant que t'il est très difficile de contrôler le robot pour se déplacer dans la direction souhaitée ou de contrôler l'écran LCD pour afficher la lettre souhaitée. Bien que la précision moyenne de la classification soit relativement élevée, il existe un décalage entre les performances individuelles de chaque classe. Par exemple, le robot préfère toujours se diriger vers la direction avec les performances individuelles les plus élevées. Pour résoudre cela, nous avons deux méthodes. La première méthode consiste à demander à

l'utilisateur de penser d'abord pendant environ 20 secondes avant d'exécuter le robot. Utiliser cette information comme référence, si la puissance moyenne d'une pensée est moindre que la moyenne de cette valeur de référence, le robot s'arrête. Cela élimine une partie de la classification erronée de certaines données qui ne sont pas liées à des catégories directionnelles ou alphabétiques et augmentera ainsi le nombre de suppositions correctes. La deuxième méthode utilise le coefficient de corrélation pour éliminer les mauvaises données EEG. Selon les modes choisis par l'utilisateur, le programme calcule le coefficient de corrélation avant ou après la sortie du modèle. Le programme a deux modes: le mode d'entraînement et le mode de fonctionnement en temps réel (figure 31).



Figure 31 Choix du mode d'exécution du programme

a) Mode d'entraînement :

Dans ce mode, le programme aide l'utilisateur à se familiariser avec le modèle SVM. Premièrement, le programme ouvrira automatiquement le panneau de configuration Epoc et l'importateur EEG Simulink qui se trouvent normalement dans les fichiers programme de l'ordinateur. Une fois que l'utilisateur a mis le casque et que tous les capteurs sont en bon état (tous verts), le programme demande à l'utilisateur de choisir le modèle SVM bien formé qui est enregistré dans le dossier Paramètres (figure 32). le même dossier mentionné dans la figure 30 ci-dessus. La boîte de dialogue de saisie (Figure 33) demande des informations générales avant le début de la formation (training). Les trois premières entrées doivent être identiques aux entrées lors de la création du modèle. Par exemple, un modèle devinant 2 directions avec 1000 données de 5 secondes chacune est montré dans l'image ci-dessous. Les points d'entrée représentent le moins d'essais que le programme exécutera pour chaque classe. Une fois que l'utilisateur a cliqué

sur «OK», une boîte de dialogue, comme le montre la figure 16 demande à l'utilisateur de fournir les noms du dossier de données PSD.

La raison pour laquelle nous devons prendre ces informations en temps réel est que dans le processus de formation, la fonction `libsvm` nécessite que toutes les données soient normalisées dans la plage comprise entre -1 et 1. Par conséquent, une seule entrée par elle-même n'a pas ce qui signifie la détection en temps réel. Pour résoudre ce problème, nous devons rappeler les dossiers de formation pour calculer les données normalisées. s'il y a un total de N données dans l'ensemble d'apprentissage, les nouvelles données seront alors ajoutées au groupe. La normalisation des données $N + 1$ est calculée et seul le résultat des données en temps réel est sélectionné. En fait, ce calcul réduire considérablement le temps de réponse du système puisque les données normalisées sont recalculées chaque fois que les nouvelles données sont entrées. Pour résoudre ce problème, nous calculons uniquement les données normalisées pour N données et enregistrons les valeurs importantes. Ces valeurs incluent le tableau avec la valeur minimale et la plage (max-min). La nouvelle normalisation des données uniques est calculée comme suit:

$$\text{Le temps réel normalisé} = \frac{\text{tems réel} - \text{min}}{\text{Max} - \text{min}} \quad (\text{III-1})$$

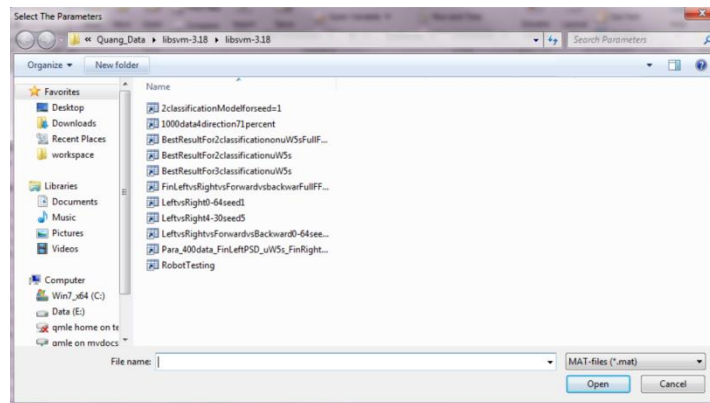


Figure 32 Choix d'un modèle

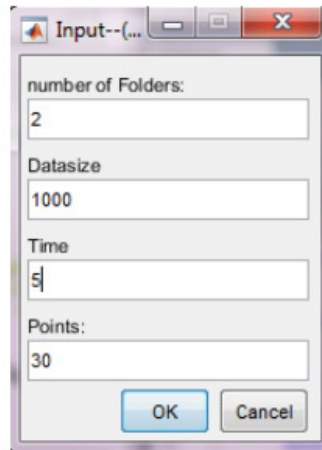


Figure 33 Saisie d'informations pour le mode pratique du modèle

Une fois les dossiers de données choisis, l'utilisateur configure la taille des données identique aux données d'entraînement du modèle. Cela comprend le choix de la plage de fréquences (figure 25) et de la plage de canaux (figure 24). Une fois les informations prises, l'utilisateur doit choisir le modèle Simulink pour prendre les données.

Cela varie en fonction de la version de Matlab utilisée. Comme le montre la figure 34, «EmotivEpoEEG_testmodel.mdl» est utilisé. Ensuite, le programme demande à l'utilisateur de choisir le type de rétroaction (figure 35). Il y a 2 options disponibles dans lesquelles la première calcule uniquement le coefficient de corrélation et la seconde calcule à la fois le coefficient de corrélation et la puissance de référence.

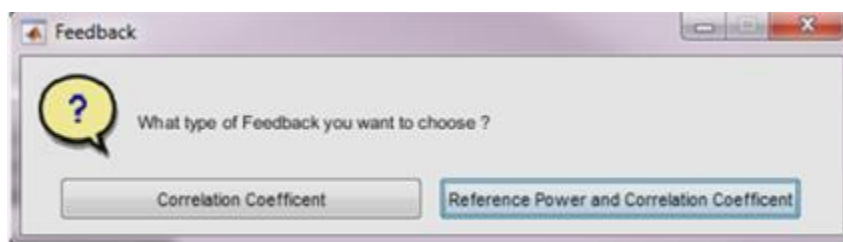


Figure 34 types de rétroaction d'entrée

Si la puissance de référence est choisie, le programme demande à l'utilisateur de se vider l'esprit pendant 20 secondes et de prendre ces données

(figure 36). Une fois les données de référence prises, elles sont converties en PSD et la valeur moyenne de cette PSD est également calculée. Ensuite, le programme demande à l'utilisateur de prendre les données de référence pour le calcul du coefficient de corrélation pour chaque classe. Par exemple, s'il y a 2 classes à gauche et à droite, 2 références sont demandées. En prenant les données de référence, le programme utilise le modèle et demande à l'utilisateur de penser une pensée spécifique pendant 5 secondes (figure 37). Si le résultat de détection du modèle SVM est correct, le programme l'utilise comme référence pour cette classe; sinon, le programme demande à l'utilisateur de réessayer jusqu'à ce qu'un essai EEG avec le résultat de détection correct soit obtenu (Figure 38).

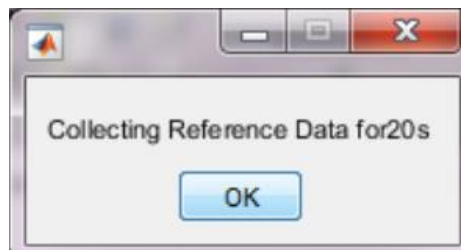


Figure 35 collecte de la référence

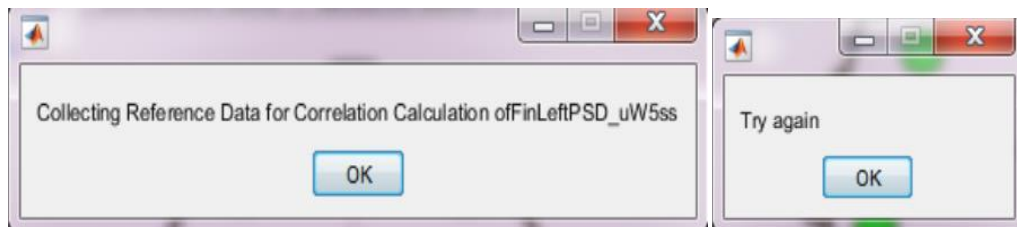


Figure 36 Collecte des données de référence du coefficient de corrélation (à gauche) -Référence des données de reprise (à droite)

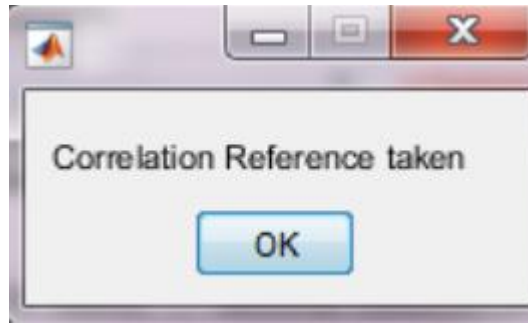


Figure 37 Informer l'utilisateur que les données EEG avec un résultat de détection correct sont prises

Le programme informe l'utilisateur de commencer la pratique. En utilisant les références de corrélation prises précédemment, le programme calcule le coefficient de corrélation. Si le résultat est inférieur à 85% ou que la puissance moyenne des données d'entrée est inférieure à la moyenne de référence (sans penser à rien), le programme demande à l'utilisateur de reprendre les données (figure 41). Si les deux tests sont réussis, le modèle classe les données en réponse correcte ou incorrecte (figure 41). Si c'est la même chose avec le cas de test (gauche ou droite), un point est ajouté au total des points. Sinon, le programme continue de fonctionner.

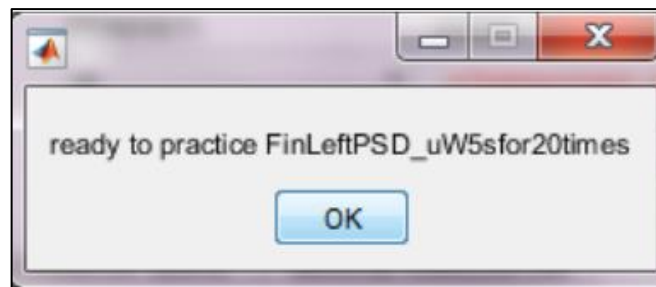


Figure 38 Commencer la formation pour une classe

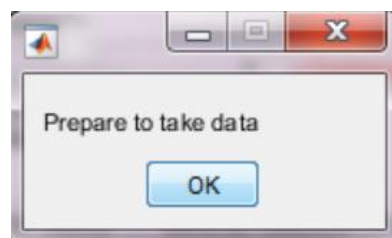


Figure 39 Commencez à prendre un morceau de données

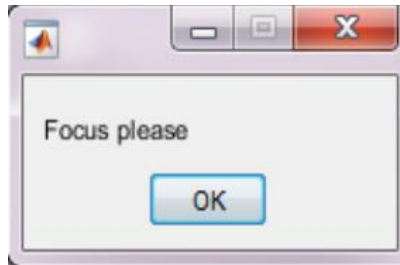


Figure 40 Reprendre les données

Les boîtes de dialogue des figures 40 et 41 continueront de s'afficher jusqu'à ce que l'utilisateur obtienne tous les points qu'il indique dans les entrées. Le nombre total d'essais est suivi afin que la précision finale puisse être calculée (figure 42). Dans ce mode, les résultats ne sont pas encore illustrés sur la voiture robot. Lorsque l'utilisateur se sent suffisamment en confiance avec le résultat, il peut relancer le programme et choisir le mode en temps réel où le mouvement de la voiture robot est contrôlé.

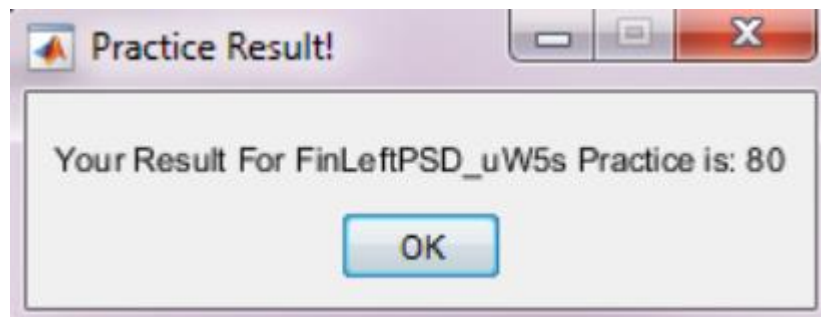


Figure 41 Résultat final pour le mode d'entraînement b) Mode temps réel

Dans ce mode, avant que la prise de données ne commence, l'utilisateur doit configurer le BCI. Le circuit de l'émetteur (carte Arduino) sera d'abord connecté à l'ordinateur, et la source d'alimentation de la voiture robot est également allumée. Ensuite, les étapes des figures 34 à 39 sont les mêmes en mode temps réel. Une fois les références prises, l'utilisateur doit indiquer le type de données qu'il souhaite envoyer au robot. -Ceci est important - par défaut, le modèle SVM directionnel classer les données PSD d'entrée en 4

entiers (1-4) et le modèle SVM alphabétique classer les données PSD d'entrée en 2 entiers (1-2). La réponse de l'utilisateur dans cette boîte de dialogue de questions aide le programme à distinguer les données de sortie de type alphabétique ou directionnel. Le programme commence à prendre les données des ondes cérébrales et informera l'utilisateur avec la boîte de dialogue de la figure 40. Les données EEG prises sont converties en PSD puis envoyées au modèle SVM bien formé pour effectuer la classification. Ensuite, selon le résultat de la classification, le programme compare le résultat de la classification avec une référence de corrélation correspondante. Si le résultat est supérieur à 85%, c'est un résultat correct qui sera envoyé au robot. Pour envoyer le signal au robot, le programme doit configurer la communication série entre Matlab et la carte Arduino. Son code est indiqué dans «run.m». L'utilisateur doit modifier manuellement la variable Comporte afin que le programme fonctionne correctement. Pour déterminer le bon Comporte, l'utilisateur doit ouvrir le «Gestionnaire de périphériques» dans la fenêtre et vérifier l'onglet de connexion USB. Cela indique à quel port le transmetteur Arduino est connecté. Une fois la connexion correctement établie, la fenêtre de commande de Matlab va désormais en mode série avec la carte Arduino. Tout ce qui est imprimé sur cette fenêtre de commande sera envoyé à la carte Arduino. Nous avons 6 signaux différents de type caractère en utilisant la première lettre de chaque classe. Ces signaux sont: 'l'- left, 'r' - right, 'f'- En avant, 'b'-En arrière, 'A'- lettre A et 'B'-lettre B. Selon le résultat du classement, ces caractères est imprimé sur la fenêtre de commande pour envoyer le signal à la carte. La carte émettrice envoie ensuite ces informations via Xbee au robot. Après chaque transmission, la connexion série est fermée entre Matlab et Arduino, ce qui rend la capacité de calcul de Matlab.

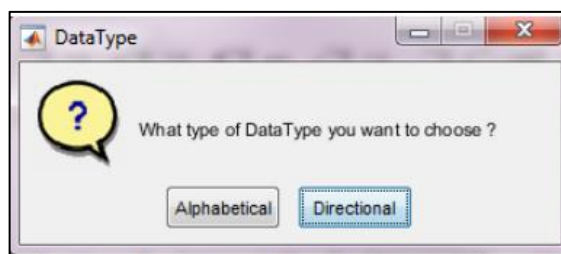


Figure 42 Indiquant le type de signal envoyé au robot

2-12-Analyse des données EEG brutes et des fonctionnalités en tant qu'entrées SVM :

2-12-1-Règle d'enregistrement des données EEG :

Afin d'avoir l'ensemble de données classifiables, l'utilisateur doit suivre une règle stricte pour l'enregistrement de toutes les données. Le rédacteur de la thèse a fait l'objet de l'expérience dans ce projet. La partie délicate de la prise de données EEG dans cette recherche est que le sujet doit imiter l'état d'un patient SLA. Il doit limiter ses mouvements corporels et essayer de rester immobile. Pour chacune des prises de données EEG directionnelles, le sujet imagine visuellement qu'il y a un objet dans la direction correspondante et réfléchir également à la signification de la direction verbalement dans sa langue. Le sujet doit fermer les yeux pour réduire les artefacts de mouvement oculaire. Pour les données alphabétiques de l'EEG, le sujet pense également visuellement à la lettre en pensant à la prononciation de la lettre.

Pour augmenter la qualité des deux types de données, après avoir pris toutes les 100 données, le sujet se repose et laisse son esprit se détendre. Au total, 6000 données EEG de 5 secondes sont prises pour 6 classes différentes. En estimation, il faudra au moins 4 heures pour compléter 1000 prises de données, ce qui donne un total de 24 heures pour 6 classes.

2-12-2-Méthode d'enregistrement des données EEG :

Pour contrôler le robot à l'aide de BCI, normalement, l'utilisateur peut penser à toutes les directions possibles à tout moment pour amener le robot vers la destination souhaitée. En tant que tel, lorsque le programme d'enregistrement des données a été développé, il a été mis en œuvre afin que l'utilisateur puisse prendre les données EEG à la fois de manière aléatoire et continue. Si l'utilisateur décide de prendre toutes les données du domaine temporel dans différentes classes en même temps, le programme créera une liste aléatoire d'étiquettes et demandera à l'utilisateur de penser différemment à chaque fois. L'utilisateur peut également décider de prendre les données en continu en choisissant de prendre les données pour une classe à chaque fois. Au début, nous pensions que prendre des données au hasard était une meilleure approche car cela peut donner à l'utilisateur la même impression que s'il contrôle le robot. Pour tester notre idée avec cette approche, 100 données EEG de 5 secondes de 2 directions (gauche et droite) ont été prises.

Les données ont ensuite été converties au format PSD et soumises à la procédure de formation et de test SVM. La meilleure précision de test était de 70% avec 92% de la précision de la formation. Ce résultat était acceptable pour un petit ensemble de données, nous avons donc décidé d'exécuter le processus de formation avec plus de données. 1000 données EEG de 5 secondes de 2 directions (gauche et droite) ont été utilisées pour la formation et les tests SVM. Le meilleur résultat n'a cependant été que de 55% pour les tests et d'environ 90% pour la formation. Le libsvm a utilisé 1280 vecteurs de support sur 1280 vecteurs d'apprentissage pour créer le modèle SVM, ce qui signifie qu'un sur-ajustement (également connu sous le nom de malédiction de dimensionnalité) s'est produit.

Ceci est normalement dû au fait que la taille des données (la dimension de chaque vecteur d'apprentissage) est trop grande, et le classificateur libsvm essaiera de trouver le modèle optimal de manière exhaustive, ce qui conduit alors au problème de sur ajustement. Pour résoudre ce problème, nous essayons de réduire la taille des données en réduisant la plage de canaux. Nous n'avons choisi que 8 canaux pour la dernière formation et voyons si les résultats du classement pourraient s'améliorer. Cependant, le résultat n'est finalement que de 60% pour les tests et d'environ 91% pour la formation. Le nombre de vecteurs de support pour cette training est d'environ 200, ce qui était encore un nombre élevé. À ce stade, nous avons décidé de redémarrer le processus d'enregistrement des données, 100 données EEG de 5 secondes des directions gauche et droite ont été prises en continu. Après le premier processus de formation, la précision des tests et la précision de la formation étaient supérieures à 90%, ce qui est très bon. Nous avons continué à prendre 1000 données de 5 secondes de chaque classe de la même manière. Pour les 8 premières graines, les résultats sont présentés dans le tableau 2 du chapitre 6. Le meilleur résultat a montré que la précision du test peut atteindre 83%. Le nombre de vecteurs de support correspondant au meilleur résultat n'est que de 912 sur 1280 données d'entraînement. C'est un très bon résultat qui montre que le modèle est bien formé et de bons modèles pour les tests en temps réel.

2-12-3-Taille des fonctionnalités pour l'entrée SVM :

La caractéristique en tant qu'entrée du modèle SVM dans notre recherche est la PSD dans le domaine fréquentiel. La taille de cette fonction est déterminée par les fréquences et les canaux sélectionnés. Les programmes

Matlab présentés dans la partie précédente donnent la flexibilité de choisir la gamme de fréquences ainsi que la gamme de canaux pour les données PSD. Le casque EPOC utilisé dans ce projet a un taux d'échantillonnage de 128 échantillons par seconde. En tant que tel, après avoir été converti dans le domaine fréquentiel, il existe 64 fréquences différentes allant de 0 à 64 Hz. Cette plage peut être réduite en 5 bandes différentes: bande Delta (1-3 Hz), bande Thêta (4-7 Hz), bande Alpha (8-12 Hz), bande Bêta (13-30 Hz) et bande Gamma (30-100 Hz) [17]. Certains chercheurs n'ont choisi que des bandes de fréquences spécifiques pour effectuer la classification. Par exemple, dans «La discrimination basée sur l'EEG entre l'imagination du mouvement de la main droite et la main gauche» [12], les auteurs ont découvert que différentes composantes de fréquence en bandes alpha (8-12 Hz) et bêta (13-30Hz) fournissaient la meilleure discrimination entre la gauche et l'imagination du mouvement de la main droite. Les résultats du classement peuvent atteindre environ 80%. Un autre article a également souligné l'activité EEG sur la bande thêta (4-7 Hz) dans la navigation dans un labyrinthe virtuel [18]. Cette recherche montre le rôle important des oscillateurs thêta dans la navigation spatiale humaine. Compte tenu de l'analyse alphabétique et linguistique, dans [19] une analyse large de différentes bandes de fréquences EEG peut montrer différentes approches de l'analyse linguistique. Plus précisément, les bandes Delta (1-4 Hz), Theta (4-7 Hz) et Alpha (8-12 Hz) montrent un effet respectueux sur les perceptions auditives et visuelles du langage. Les noms ou verbes sont ensuite comparés dans deux bandes Alpha (8-12Hz) et Beta (13-30 Hz) pour voir leurs performances. Ce résultat n'est pas très utile dans ce cas puisque seules les données alphabétiques (A ou B) sont analysées dans cet article.

Cependant, ces articles ont souligné une bonne bande de fréquences (de Theta à Beta) et peuvent aider à réduire la taille des données PSD concernées dans cette recherche. Nous avons décidé d'organiser 2 formations différentes. Dans la première formation, nous avons utilisé uniquement la plage de fréquences de 4 à 30 Hz et dans la seconde, nous avons utilisé toute la plage de fréquences possible de 0 à 64 Hz. Nous avons essayé d'exécuter la classification en utilisant toutes les fréquences (0 à 64 Hz). Cela a en fait augmenté les résultats des tests d'environ 10% dans chaque classification. Cependant, le gros inconvénient était que le temps de formation du modèle SVM est passé de 30 heures à 96 heures pour chaque classification (selon le type de données et le nombre de classes).

Chapitre

III

Chapitre3 :Résultats et interprétations**3-1-Introduction :**

Dans le troisième chapitre, nous avons analysé les résultats et les performances de la classification à un seul critère du signal EEG afin de déterminer la précision du classificateur et d'utiliser toutes les données. La classification qu'elle a été utilisée en utilisant deux classes différentes correspondant aux zones de saisie. Les résultats sont transférés vers la carte programmable ARDUINO pour faire la commande finale. Chaque colonne de tableau résultante représente la classe réelle, tandis que chaque ligne montre les pourcentages de classification réels pour chaque classe, ayant ainsi la classification correcte toujours sur la diagonale.

3-2-Résultats des simulations des modèles SVM :

Dans ce chapitre, nous résumerons les résultats des modèles SVM pour différents problèmes de classification. Nous allons tout d'abord donner le résultat d'un problème de classification très simple pour vérifier l'exactitude du programme SVM car il est au cœur de notre système développé (l'interface cerveau ordinateur,BCI).D'autre part ,nous montrerons les résultats de la classification des signaux EEG directionnels et alphabétiques. Enfin, nous présenterons quelques résultats de classification en utilisant toute la gamme de fréquences.

3-2-1-Classification des données des distributions normales :

Étant donné que le programme SVM «svm_train_n_categorie.m» est extrêmement critique dans notre système BCI développé, avant d'utiliser le programme SVM pour classer les données EEG réelles, nous avons créé un cas pour vérifier si le programme fonctionne correctement. Pour ce faire, nous utilisons une matrice 14×64 qui sera remplie des données de distribution normale aléatoire. Nous contrôlons la moyenne et l'écart type des données afin que les données soient classifiables. Pour chaque valeur moyenne de 1 à 4, cent matrices sont créées à l'aide de l'exemple (example_brain_data.m). Ce petit programme permettra à l'utilisateur de créer les données randomisées à distribution normale. L'utilisateur peut contrôler l'écart type et les valeurs

moyennes en modifiant les entrées illustrées . Ces données seront numérotées de 1 à 100 et poussées dans des dossiers portant les mêmes noms (1, 2, 3, 4). En utilisant le "svm_train_n_catefories.m", ces données dans les 4 dossiers seront classées. Si la précision de la formation et la précision des tests du résultat de la classification sont élevées, cela signifie que le programme créé est capable de classer correctement les données. Le tableau 1 ci-dessous montre les résultats de la classification pour les 4 différentes données randomisées à distribution normale. Les résultats sont proches de 100% ce qui prouve que ce programme fonctionne correctement.

Tableau 1 Résultat de la classification des données de la distribution normale

Seed	Best_C	Best_gamma	Train_acc	Test_acc	Total_n SV
1	0.312 5	0.0009765 6	97.188	100	320
2	29.34 41	0.0009765 6	100	100	316
3	0.031 25	0.016317	98.125	100	320
4	19.86 97	0.0009765 6	100	96.25	320
5	25.76 78	0.0009765 6	100	92.5	317
6	1.414 2	0.038808	100	92.5	320
7	0.031 25	0.0040792	98.125	96.25	320
8	0.031 25	0.0009756	97.188	100	320

3-2-2-Classification de 2 données directionnelles :

Les résultats de la classification pour 2 directions (gauche et droite) sont présentés dans le tableau 2. Le meilleur apparaît à la semence numéro 5 où la précision de l'entraînement est de 98,188% et la précision du test est de 83%. En utilisant la méthode d'apprentissage de la figure 26, 1280 données (200 gauche, 200 droite) ont été utilisées comme ensemble de données d'apprentissage et 400 données ont été utilisées pour les tests finaux, où le meilleur modèle n'a utilisé que 912 données sur 1280 échantillons d'apprentissage comme vecteurs de support. . Pour ce classement bidirectionnel, la formation pour chaque graine a pris au moins 6 heures. Le temps de formation varie en fonction du nombre de vecteurs de support nécessaires. En général, la classification avec le moins de vecteurs supports prendra moins de temps, mais cela ne garantit pas toujours les solutions optimales. Le programme est capable d'exécuter plus de graines. Cependant, seuls les résultats des 8 premières graines sont obtenus car cela prend beaucoup de temps et de coût de calcul (exécuter plusieurs fenêtres Matlab en même temps). Le tableau 2 donne la précision globale des tests à la cinquième colonne. Il s'agit du nombre de données EEG classées correctement divisé par le nombre total de données de test. Le tableau 3 montre la précision des tests de la classification de chaque direction. Par exemple, la deuxième colonne intitulée Gauche est le nombre de données de direction gauche classées correctement divisé par le nombre total de données de direction gauche. Pour la plupart des semences, les précisions de test individuelles ne sont pas équilibrées. Le modèle SVM peut détecter une direction avec plus de succès que l'autre. La graine numéro 3 présente la performance la plus équilibrée avec un écart de seulement 0,5% entre la classification gauche et droite. Cependant, il existe un compromis de 0,75% dans la précision globale des tests par rapport au résultat déséquilibré de la semence 5.

Tableau 2 Résultat de la classification gauche vs droite avec une plage de fréquences de 4 à 30 Hz

Seed	Best_C	Best_ga mma	Train _acc	Test_ acc	Total_ nSV
1	29.3441	0.08838 8	98.31 3		855
2	449.6006	0.01013 2	94.93 7	82.75	675
3	6.1688	0.13053	95.75	79.25	964
4	6.7272	0.125	96.43 7	82.25	934
5	14.6721	0.11463	98.18 8	79.5	912
6	122.5732	0.02116	94.68 8	83	744
7	18.2206	0.125	98.93 8	80.5	919
8	14.6721	0.1197	98.18 80	82.5	912

Tableau 3 Performances individuelles de 2 directions gauche vs droite 4-30Hz

Seesd	Left	Right	Test_accurcy
1	84.5	81	82.75
2	80.5	78	79.25
3	82.5	82	82.258
4	75	84	79.5
5	81.5	84.5	83
6	84	81.5	82.75
7	79	82	80.5
8	80.5	84.5	82.5

3-2-3-Classification de 3 données directionnelles :

Les résultats de classification pour 3 directions (gauche, droite et avant) sont présentés dans le tableau 4. Le meilleur résultat en termes de précision des tests est de 79,333% pour la semence numéro 8 avec une précision d'entraînement de 95,208%.

Selon la méthode de formation , il y avait 600 données dans le pool de tests (200 gauche, 200 droite et 200 avant) et 1920 données dans le pool de formation. Le meilleur modèle n'utilisait que 1410 données sur 1920 comme vecteurs de support. Le temps de formation pour chaque graine était compris entre 10 heures et 12 heures. Il a donc fallu près de 4 jours pour terminer le calcul de toutes les graines. Le tableau 5 montre ensuite les précisions de test individuelles. Ce tableau montre un résultat encore plus déséquilibré par rapport aux 2 performances individuelles directionnelles. L'écart entre les meilleures et les pires performances individuelles de chaque graine variait de 4,5% (graine 4) à 12% (graine 3). La graine la plus équilibrée est la graine numéro 4. Cependant, il y a un compromis de 5% de précision globale par rapport au résultat de déséquilibre de la graine 8.

Tableau 4 Résultat de classification gauche vs droite vs avant avec une plage de fréquences 4-30 Hz

Seed	Best_C	Best gamma	Train_acc	Test_acc	Total_n SV
1	10.834	0.25	98.375	77.33	1644
2	5.6569	0.1767 8	93.042	75.833	1604
3	18.220 6	0.2013 1	98.542	76.5	1576
4	9.5137	0.1767 8	95.583	74.33	1555
5	7.336	0.2394	96.583	78	1649
6	69.792 5	0.0652 67	96.33	79	1368
7	66.833 5	0.0810 52	97.5	75.167	1369
8	32	0.0846 41	95.208	79.33	1410

Tableau 5 Performances individuelles de 3 directions gauche vs droite vs avant 4-30Hz

Seed	Left	Right	Forward	Test_acc
1	80	73.5	78.5	77.333
2	79	70.5	78	75.833
3	79	67	83.5	76.5
4	72.5	73.5	77	74.33
5	77	78.5	78.5	78
6	84	71.5	81.5	79
7	75	71	79.5	75.167
8	75.5	78.5	84	79.33

3-2-4-Classification de 4 données directionnelles :

Le tableau 6 montre les résultats de classification de 4 directions (gauche, droite, avant et arrière). Le meilleur résultat dans la précision des tests est de 72% pour la semence numéro 11 avec une précision d'entraînement de 92,75%.

Les graines 2,4,10 sont éliminées des sorties du fichier texte car ces graines ont moins de 70% du résultat du test. Selon la méthode de formation, il y avait 800 données dans le pool de tests (200 gauche, 200 droite, 200 avant et 200 arrière) et 2560 données dans le pool de formation. Le meilleur modèle n'utilisait que 2089 données sur 2560 données comme vecteurs de support. Le temps estimé pour la formation de chaque graine variait entre 1 et 1,5 jour. En tant que tel, il a fallu environ 2 semaines pour terminer tous ces calculs. Le tableau 7 montre ensuite la précision des tests individuels sur chaque classe directionnelle. Il s'agit de la classification individuelle la plus déséquilibrée de la classification directionnelle. L'écart entre les meilleures et les pires performances individuelles de chaque graine peut être aussi bas que 7,5% pour la graine 9 et aussi élevé que 19,5% pour la graine 3. La graine la plus déséquilibrée est le numéro 1 où la précision des tests n'est que de 70,875%. Il existe un compromis de précision globale de 1,125% par rapport au résultat de déséquilibre de la graine 11. En fait, pendant le fonctionnement du robot en temps réel, tout modèle de classification qui a une précision de test inférieure à 80% peut ne pas être en mesure de contrôler correctement le robot. En tant que tel, pour ce problème de classification 4 directions, nous avons étendu la plage de fréquences à 0-64 Hz.

Tableau 6 Résultat de classification gauche vs droite vs avant vs arrière avec plage de fréquences 4-30 Hz

Seed	Best_C	Best gamma	Train accurasy	Test accuracy	Total_SupportVectors
1	14.6721	0.1621	94.813	70.875	2229
3	15.3217	0.19278	96.219	71.125	2277
5	6.1688	0.19278	90.625	71.125	2319
6	9.5137	0.20131	94.188	71.75	2299
7	6.442	0.17678	90.281	71.75	2322
8	18.2206	0.10511	91.031	71.125	2155
9	107.6347	0.048194	93.031	71.875	1976
11	34.8962	0.081052	92.75	72	2089

Tableau 7 Performances individuelles de 4 directions gauche vs droite vs avant vs arrière 4-30Hz

Seed	Left	Right	Forward	Backward	Test_accuracy
1	73.5	64	76.5	65	70.875
3	70.5	62	81.5	70.5	71.125
5	69	67.5	76	72	71.125
6	72	63.5	78	73.5	71.75
7	69	64.5	78.5	75	71.75
8	68	67	80.5	69	71.125
9	69	69	76.5	73	71.875
11	73	66	79	70	72

3-2-5-Classification de 2 données alphabétiques :

Le tableau 8 montre le résultat du classement pour 2 lettres (A et B). Le meilleur résultat apparaît à la graine 1 où la précision des tests est de 78,75% et la précision d'entraînement correspondante est de 92,75%. Semblable à la classification à 2 directions de gauche par rapport à droite, cette classification a utilisé 400 données dans le pool de test final et 1280 données dans le pool de formation. Le meilleur résultat a utilisé 834 données sur 1280 pour les vecteurs de support. Le temps de formation requis pour chaque graine est un peu plus long que celui de la classification 2 directions. Il a fallu environ 8 heures pour terminer la formation pour une graine, ce qui a donné au total près de 4 jours pour que le programme se déroule. Le tableau 9 montrera ensuite la précision des tests individuels de la détection de chaque lettre. Le résultat le plus équilibré a la différence de 1% à la graine 7, tandis que le résultat le plus déséquilibré a la différence de 9,5% à la graine 5. Le meilleur modèle a un écart de 4,5% entre la précision de test de A (81%) et la précision de test de B(78,75%).

Tableau 8 Résultat de classification A vs B avec plage de fréquences 4-30 Hz

Seed	Best_C	Best_gamma	Train_accuracy	Test_accuracy	Total_SupportVectors
1	1961.17 15	0.006024 3	92.75	78.75	834
2	724.077 3	0.013721	95.437	73	871
3	112.400 1	0.046151	97.938	71.25	966
4	304.437	0.019404	93.875	75.75	917
5	24.6754	0.068157	93.875	73.75	1033
6	38.0546	0.10066	98.75	72.25	1091
7	939.012 1	0.017794	98.125	75.5	917
8	122.573 2	0.017794	88.813	74.75	958

Tableau 9 Performances individuelles de 2 lettres A vs B 4-30Hz

Seed	A	B	Test_accuracy
1	81	76.5	78.75
2	74.5	71.5	73
3	68.5	74	71.25
4	76	75.5	75.75
5	69	78.5	73.75
6	68.5	76	72.25
7	75	76	75.5
8	74	75.5	74.75

3-2-6-Classification de 4 données directionnelles et 2 données alphabétiques :

En utilisant le maximum de gamme de fréquences pour le dernier processus de formation, chaque cas de classification de 6,2 à 6,5 aura sa gamme de fréquences étendue à 0-64 Hz. La raison pour laquelle seul le résultat de la première graine est indiqué dans cette section pour la classification directionnelle et la classification alphabétique est que ces problèmes de classification nécessitent beaucoup plus de temps de formation. Par exemple, cela peut prendre environ 1 jour, 2 jours et 4 jours pour les classifications 2, 3 et 4 directionnelles. Pour la classification à 2 lettres, cela peut prendre environ 2 jours. Bien que le temps de calcul ait été augmenté, la décision des tests a été augmentée d'environ 4 à 11% dans chacun des 4 cas. Comme le montre le tableau 10, chacune des classifications a été effectuée uniquement pour la première graine. Pour les classifications directionnelles, les précisions d'essai pour les classifications directionnelles 2, 3 et 4 sont respectivement de 91,5%, 88,1667% et 81,75%. Il y a un gain de 8,75%, 10,8337% et 10,875% des classifications 2, 3 et 4 directionnelles. En regardant les valeurs des paramètres C et γ , nous pouvons voir que le programme utilise des valeurs très proches de C et γ pour ces trois cas. Cela a montré que les fonctionnalités avec la gamme de fréquences étendue sont plus classifiables. Pour la classification à 2 lettres, la précision des tests a une augmentation de 4,5%. Les paramètres C et γ pour les classifications à 2 lettres sont également plus petits que les résultats dans la gamme de fréquences 4-30 Hz. Les performances individuelles du tableau 11 montrent que ces modèles sont en fait plus équilibrés que celui de 4-30 Hz. La classification la plus

déséquilibrée est la 4 directionnelle où l'écart entre les meilleures et les pires performances est de 4,5%. Le résultat le plus équilibré pour le cas directionnel est la classification gauche contre droite où l'écart n'est que de 3%. Cela montre également que les mouvements du robot directionnel 2 sont plus faciles à contrôler par EEG que le robot directionnel 4. Après avoir élargi la gamme de fréquences, tous ces résultats de tests de classification sont supérieurs à 80%, ce qui montre de bons modèles SVM pour les tests en temps réel avec le robot.

Tableau 10 Résultat de la classification des données directionnelles et alphabétiques pour la première graine de 0 à 64 Hz

Classification	Best_C	Best_gamma	Train_acc	Test_acc	Total_nSV
L/R	3.668	0.10997	98.875	91.5	827
L/R/F	4.1771	0.096388	97.0833	88.1667	1385
L/R/F/B	3.5125	0.21022	98.4063	81.75	2227
A/B	14.05	0.05985	98.625	83.025	998

3-3-Implémentation et test du système de contrôle du robot :

3-3-1-Schéma fonctionnel du mouvement du robot et système de contrôle d'affichage alphabétique :

Pour contrôler le mouvement robotique de la voiture et l'affichage LCD alphabétique, beaucoup d'efforts sont nécessaires pour communiquer simultanément entre différentes interfaces. Tout d'abord, lorsqu'un utilisateur pense à une direction ou à une lettre, des signaux EEG à 14 canaux seront acquis par le dispositif EPOC. Grâce au panneau de contrôle Emotiv et à l'importateur Simulink, ces signaux EEG seront enregistrés au format Matlab pour être facilement utilisés dans le programme Matlab SVM. Les paramètres de l'algorithme SVM seront optimisés par un processus de formation. Dans le processus de formation SVM, des ensembles de données de formation seront nécessaires. Les ensembles de données d'apprentissage signifient des paires d'entrées et de sorties du SVM. Les signaux EEG durent 5 secondes et ils seront convertis en données PSD dans le domaine fréquentiel dans une plage

de fréquences souhaitable en tant que caractéristique ou entrée du modèle SVM. La sortie serait une direction ou une lettre en conséquence. Pour que le SVM soit bien formé, 6000 données de formation pour six classes (4 directions et 2 lettres), 1000 données pour chaque classe, ont été générées pour ce projet. Comme discuté au chapitre 3, dans le processus de formation, les méthodes de validation croisée et d'optimisation de convex seront utilisées pour optimiser les paramètres SVM. Avec tous les paramètres optimaux, le SVM sera bien construit et prêt à classer tous les nouveaux signaux EEG. Ce processus de formation SVM et les étapes connexes sont indiqués par une ligne bleue sur la figure 1.

De nouveaux signaux EEG générés par l'utilisateur lors de la réflexion sur une direction ou une lettre seront envoyés au modèle SVM construit avec les paramètres optimaux via Emotive EPOC, le panneau de commande et l'importateur Simulink comme indiqué par la ligne rouge . La sortie SVM est introduite comme une fonction de contrôle. Cette fonction convertira la sortie SVM en une commande numérique telle que 000,001,010, 011 et ainsi de suite pour représenter chaque direction et lettre. Ces commandes numériques seront envoyées sans fil au microcontrôleur Arduino à l'aide de module Xbee. Si la commande numérique reçue signifie une direction, le microcontrôleur Arduino générera une impulsion d'onde spécifique pour contrôler le servomoteur pour déplacer la voiture robotique dans les directions souhaitées. Si le signal numérique reçu signifie une lettre, le microcontrôleur contrôlera l'écran LCD pour afficher la lettre immédiatement. Toutes les étapes liées au nouveau processus de signal EEG pour contrôler le mouvement du robot et l'affichage alphabétique sont indiquées par des lignes rouges . Dans les chapitres précédents, nous nous concentrons principalement sur les étapes 1 à 5. Dans les sections suivantes de ce chapitre, nous nous concentrerons sur les étapes restantes pour discuter plus en détail de la manière de créer une communication sans fil via PC, microcontrôleurs et Xbees.

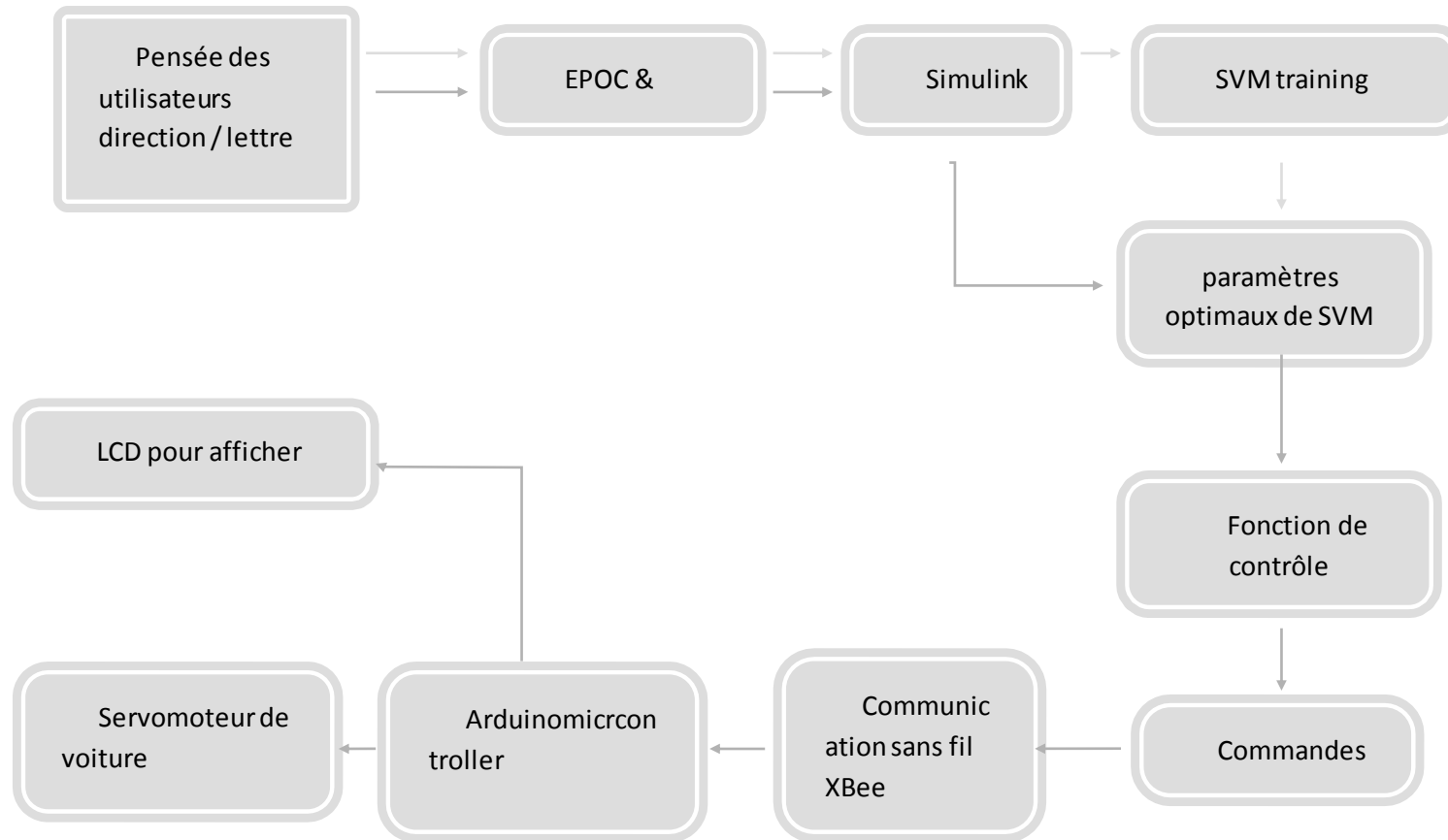


Figure 1 Système de contrôle du mouvement du robot et de l'affichage alphabétique

3-3-2-Configuration de Xbee :

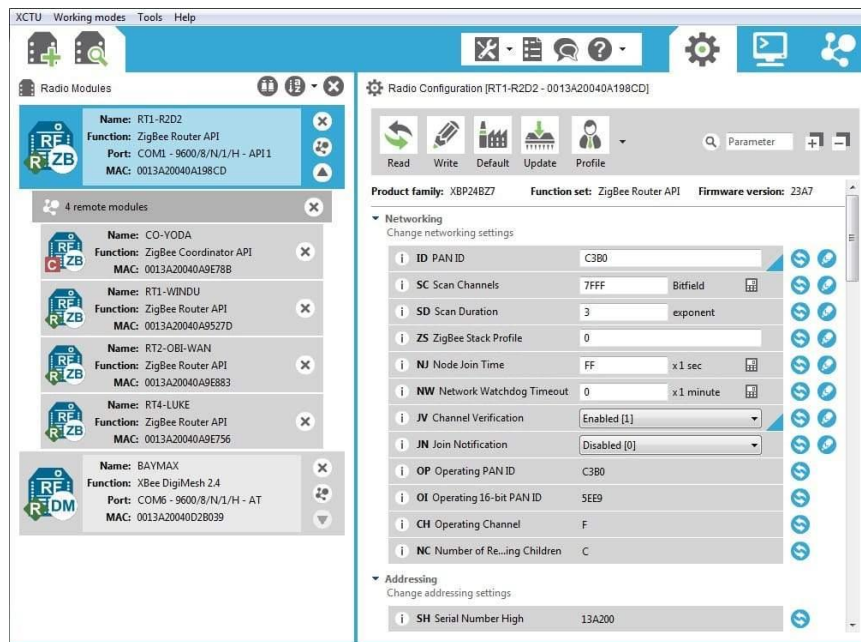


Figure 2 Configuration des Récepteurs et émetteurs du module Xbee

Dans ce projet, nous utiliserons 2 microcontrôleurs et 2 antennes Xbee pour pouvoir créer une communication sans fil entre l'ordinateur et le robot. Tout d'abord, nous devons établir la connexion entre 2 antennes Xbee en utilisant 5 xbee. Cela peut être fait en utilisant un programme X-CTU comme le montre la figure 2. Ce logiciel aidera les utilisateurs à configurer l'ID de panoramic ainsi que les canaux de plusieurs antennes Xbee. Pour configurer le Xbee, l'utilisateur doit connecter la carte Xbee à l'ordinateur à l'aide d'un cordon micro USB.

Ensuite, en choisissant l'onglet «configuration du modem», l'utilisateur peut lire, écrire ou restaurer le Xbee à sa configuration par défaut. La fenêtre sur le côté gauche de la figure 2 montre la configuration de l'émetteur, tandis que celle du côté droit est pour l'antenne du récepteur. Ces deux antennes seront connectées à deux cartes Arduino. La première carte est connectée à l'ordinateur tandis que la deuxième carte est utilisée pour contrôler le robot. Normalement, il y aura une connexion en mode série entre l'antenne Xbee et la carte Arduino. Cela peut être fait en connectant les broches Tx et Rx sur chaque carte Arduino aux broches Rx et Tx sur les Xbees. Cependant, puisqu'il

y a déjà eu une connexion en mode série entre la carte Arduino (qui contrôle Xbee) et la fenêtre de commande Matlab (ordinateur), ces broches sont toujours occupées. Nous avons décidé d'utiliser 3 broches D0, D1 et D2 sur chaque antenne Xbee pour configurer la communication. La façon dont cette communication fonctionne est que si la broche Dx sur l'émetteur est haute ou basse, les broches correspondantes sur le récepteur sera également dans le même état. Ces broches sont ensuite connectées aux broches des cartes Arduino. La carte traitera ensuite ces informations sur 3 bits et décidera de l'action à effectuer.

3-3-3-Configuration du mode série :

La configuration du mode série prend moins d'une seconde pour chaque transmission de données. Cela peut être fait en utilisant `setupSerial.m` et la fonction `Setup` dans le `transmetteur.ino` (carte Arduino rouge) illustrée à l'annexe A. La fonction `setupSerial` connectera d'abord la fenêtre de commande Matlab avec un Comport où l'Arduino est connecté. Ensuite, cette fonction décidera du débit en bauds, du nombre de bits et du bit d'arrêt pour la communication. Le débit en bauds est fixé à 9600, il y aura 8 bits de données dans chaque transmission et le bit d'arrêt est affecté à 1. Ensuite, la fonction `setupSerial` enverra un caractère simple à la carte Arduino. La carte Arduino prendra alors ces données et les réimprimera dans la fenêtre de commande. Après ce point, tous les caractères imprimés sur la fenêtre de commande Matlab seront envoyés à la carte Arduino. Cependant, comme nous avons besoin de la fenêtre de commande Matlab pour effectuer la classification des données EEG en continu, après chaque la transmission est terminée, le mode série est fermé. Le Matlab classifie ensuite les données PSD entrantes à l'aide de la bibliothèque `libsvm`. La nouvelle connexion série sera alors ouverte pour envoyer la nouvelle commande à la carte Arduino. Cette boucle ne s'arrêtera pas tant que l'utilisateur n'aura pas décidé de fermer le programme ou de désactiver l'EPOC.

3-3-4-Implémentation du contrôle du robot :

Le robot avec LCD a deux parties principales, le côté émetteur qui envoie les signaux de Fenêtre de commande Matlab, et un côté récepteur qui contrôle le robot.

a) Côté émetteur

La figure 3 montre la connexion entre Xbee et une carte Arduino. Il s'agit de la carte Arduino qui est connectée à l'ordinateur, en prenant soin des informations de la fenêtre de commande Matlab. Il y aura 5 connexions entre la carte Arduino et l'antenne Xbee. Les 2 premières broches fournissent les connexions d'alimentation ou VCC et GND. Les 3 dernières broches 2, 3 et 4 de la carte seront connectées respectivement à D0, D1 et D2 sur la carte Xbee. Il y a 7 caractères différents qui peuvent être envoyés au tableau: 's'-Stop,' l'-Left, 'r'-Right,' f'-Forward, 'b'- Backward,' A 'et' B '. La carte encodera ensuite ces caractères en séquences de 3 bits qui seront ensuite envoyées à l'antenne Xbee. Parce qu'il y a 3 bits, il y a $2^3 = 8$ séquences différentes.

Cependant, les séquences «000» et «111» sont toutes deux utilisées pour le signal d'arrêt (tableau 11)

Tableau 11 Séquences de signaux pour différents caractères

Caractère	D0	D1	D2
's'	0/1	0/1	0/1
'l'	0	0	1
'r'	0	1	0
'f'	0	1	1
'b'	1	0	0
'A'	1	0	1
'B'	1	1	0

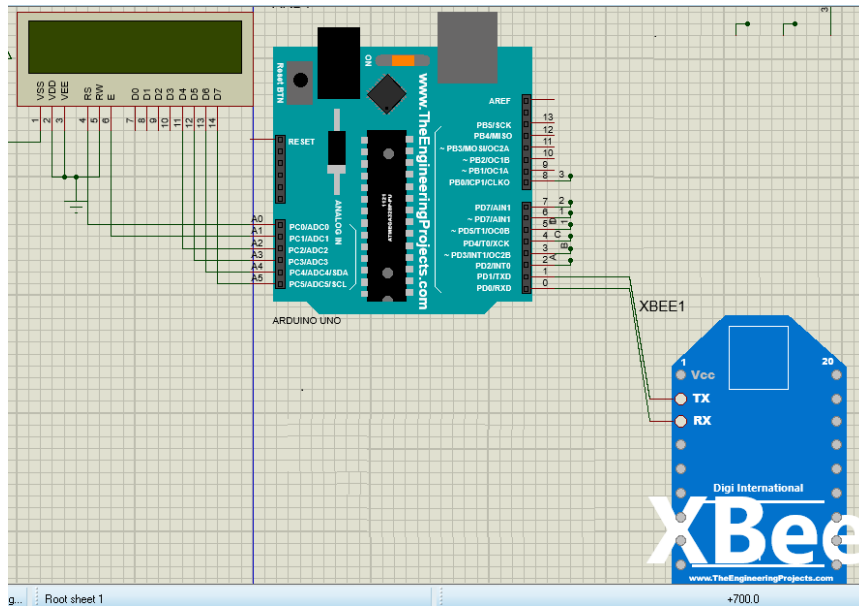


Figure 3 La connexion de l'émetteur Arduino et Xbee et LCD

b) Côté récepteur

Le robot à trois roues comme illustré à la figure 4 est contrôlé par l'autre carte Arduino. Cette carte Arduino prend en charge plus d'objectifs que celle de l'émetteur. Il y a trois tâches principales que cette carte Arduino doit accomplir. Premièrement, après avoir reçu le signal en termes de séquences de bits de données comme illustré dans le tableau 12, la carte Arduino prendra la décision de contrôler le robot. Chaque fois qu'un caractère indiquant un mouvement directionnel est envoyé, le robot se déplace dans le sens antihoraire dans le sens horaire ,en avant ou en arrière en conséquence.

Ensuite, à la fin de la transmission, une séquence d'arrêt sera envoyée pour arrêter complètement le robot, ce qui est également la deuxième tâche de la carte Arduino où différentes impulsions sont envoyées aux servomoteurs. La dernière tâche consiste à afficher les résultats sur l'écran LCD. En utilisant uniquement des séquences de 4 bits, la carte imprimera une chaîne sur l'écran LCD illustrant à la fois les mouvements directionnels et les deux lettres A, B.

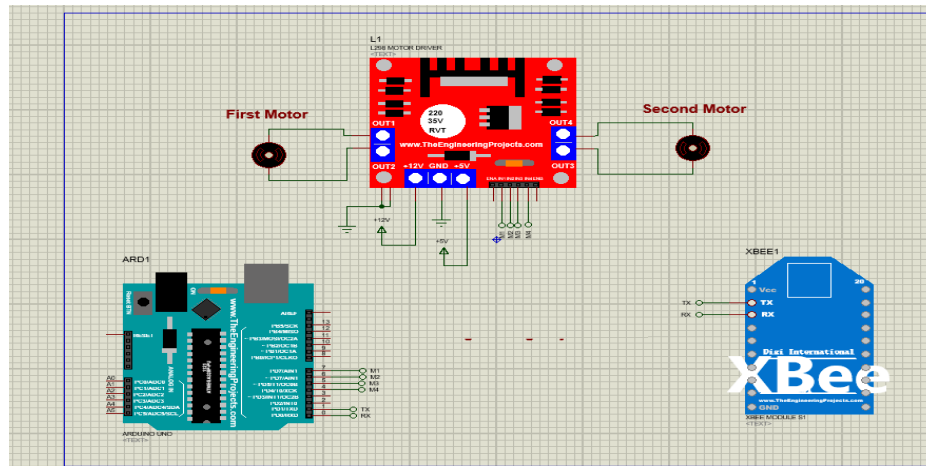


Figure 4 la voiture robotique émetteur .

3-4-conclusion générale

L'utilisation des modèles SVM avec les plus hautes précisions estimées est utilisée pour le test final. Après avoir correctement connecté le casque EPOC à l'ordinateur (tous les capteurs sont verts), le programme «RobotControl.m» a été exécuté en mode temps réel. Il y a eu trois expériences. Dans la première expérience, nous avons utilisé le classificateur bidirectionnel bien formé (tableau 10) pour détecter 25 pensées de gauche et 25 de droite. Le robot s'est déplacé dans la bonne direction avec la précision de 80% (20/25 des deux directions). Ce résultat n'a montré qu'environ 10% de décalage entre la précision des tests rapportée et la précision du contrôle du robot en temps réel. Ensuite, dans la deuxième expérience, nous avons utilisé le classificateur directionnel 4 bien formé (tableau 10) pour détecter 25 réflexions à gauche, 25 à droite, 25 en avant et 25 en arrière. Dans cette étape, il est plus facile de contrôler le robot pour se déplacer vers la gauche et la droite, nous avons obtenu 17 résultats de classification corrects sur 25 pour les deux directions. Cependant, les précisions de l'avant et de l'arrière sont faibles. Nous avons obtenu 14 résultats corrects pour le mouvement vers l'avant et 8 résultats corrects pour le mouvement vers l'arrière. La précision globale (pf) de ces tests en temps réel est de 56%, ce qui est 26% inférieur à la précision des tests indiquée dans le tableau 10. Enfin, un modèle SVM bien formé pour 2 lettres A et B a été testé. Il y avait 20 résultats corrects sur 25 pour la lettre A tandis que 15 résultats corrects sur 25 pour B. Cela a conduit à une précision globale des tests en temps réel de 70%, ce qui est 13,25% de moins que la précision des tests indiquée dans le tableau 10.

Références

Références

Chapitre 1 :

[1] Le cytoplasme et les organites cellulaires

Kolata, G. Un régime sévère ne prolonge pas la vie, du moins chez les singes. 29 août 2012;

[2] Les tissus nerveux assurent la perception et la réponse

Stern, P. Problème de discussion: se passionner pour la glie. Science . 2010

[3] Ming GL, Song H. Neurogenèse adulte dans le système nerveux central des mammifères. Annu. Rev. Neurosci. 2005; 28: 223-250.

[4] Lésion tissulaire et vieillissement

Emerson, RW. Vieillesse. Atlantique. 1862; 9 (51): 134-140.

[5] Fonctions du système tégumentaire

Académie américaine de dermatologie (États-Unis). Tatouages et piercings corporels [Internet]. Schaumburg, IL; c2013.

[6] Maladies, troubles et blessures du système tégumentaire

American Cancer Society (États-Unis). Cancer de la peau: cellules basales et squameuses [Internet]. c2013.

[7-8] Hôpital pour enfants Lucile Packard de Stanford (États-Unis).

Classification et traitement des brûlures [Internet]. Palo Alto (Californie).

[9] Mayo Clinic (États-Unis). Carcinome basocellulaire [Internet]. Scottsdale (AZ); c2012

[10] Beck, J. FYI: combien un corps humain peut-il transpirer avant de s'épuiser? Science populaire [Internet]. New York (NY);

[11] Skin Cancer Foundation (États-Unis). Faits sur le cancer de la peau [Internet]. New York (NY); c2013.

Références

- [12] Les Skull Centers for Disease Control and Prevention (États-Unis). Prévention et contrôle des blessures: traumatisme craniocérébral [Internet]. Atlanta, Géorgie.
- [13] Structure et fonction de base du système nerveux
Kramer, PD. Écouter du prozac.
- [14] Typage du sang
Croix-Rouge américaine (États-Unis). Types de sang [Internet]. c2013
- [15] Régulation homéostatique du système vasculaire
Centers for Disease Control and Prevention (États-Unis). Maîtriser la pression artérielle: l'hypertension est hors de contrôle pour trop d'Américains [Internet]. Atlanta (GA);
- [16] Transplantation et immunologie du cancer
Robinson J, Mistry K, McWilliam H, Lopez R, Parham P, Marsh SG. Recherche sur les acides nucléiques. Base de données IMGT / HLA [Internet].
- [17] Robinson J, Malik A, Parham P, Bodmer JG, Marsh SG. Antigènes tissulaires. Base de données IMGT / HLA [Internet]. 2000; 55 (3): 280-287.
- [18] Organes et structures du système respiratoire
Bizzintino J, Lee WM, Laing IA, Vang F, Pappas T, Zhang G, Martin AC, Khoo SK, Cox DW, Geelhoed GC, et al. Association entre le rhinovirus C humain et la gravité de l'asthme aigu chez les enfants. Eur Respir J [Internet]. 2010; 37 (5): 1037–1042.
- [19] Kumar V, Ramzi S, Robbins SL. Pathologie de base Robbins. 7e éd. Philadelphie (PA): Elsevier Ltd; 2005.

Références

[20] Martin RJ, Kraft M, Chu HW, Berns, EA, Cassell GH. Un lien entre l'asthme chronique et l'infection chronique. *J Allergy Clin Immunol* [Internet]. 2001, 107 (4): 595-601.

[21] La bouche, le pharynx et l'œsophage

van Loon FPL, Holmes SJ, Sirotkin B, Williams W, Cochi S, Hadler S, Lindegren ML. Rapport hebdomadaire de morbidité et de mortalité: Surveillance des oreillons - États-Unis, 1988–1993

[22] Les petits et gros intestins

American Cancer Society (États-Unis). Faits et chiffres sur le cancer: colorectal: 2011-2013

Chapitre 2 et 3 :

1. F. Lopes da Silva, «EEG: origine et mesure», Université d'Amsterdam, Amsterdam, Pays-Bas, 2010.
2. Cortes, C. ; Vapnik, V. (1995). "Réseaux de vecteurs de support". *Apprentissage automatique* 20 (3): 273.
3. Stefan Rueling, «Estimation du classificateur SVM à partir des probabilités de groupe», Schloss Birlinghoven, Allemagne, ICML, 2010.
4. Mary Lyon, R.N., M.N., «Raisons de vivre avec la SLA». ALS Association [en ligne].
5. «Définition de la sclérose latérale amyotrophique (SLA)», Medicine Net [en ligne].
6. C.-C. Chang et C.-J. Lin. LIBSVM: une bibliothèque pour les machines à vecteurs de support. *ACM Transactions on Intelligent Systems and Technology*, 2: 27: 1--27: 27, 2011.

Références

7. Emotiv Insight, «Simulink EEG Importer», Emotiv Inc [en ligne]
8. Gou, Qing «Amélioration des performances du putt de golf grâce à l'apprentissage statistique des signaux EEG». Thèse. Université de l'Arkansas, 2014
9. Niels Landwehr, Mark Hall et Eibe Frank (2003). Arbres modèles logistiques
10. Ridella S, Rovetta S, Zunino R. (1997). "Réseaux circulaires de rétropropagation pour la classification". Transactions IEEE sur les réseaux de neurones 8 (1): 84–97.
11. Evangelista, Paul, Mark Embrechts et Boleslaw Szymanski. "Apprivoiser la malédiction de la dimensionnalité dans les noyaux et la détection de nouveautés." Technologies informatiques souples appliquées: le défi de la complexité. Vol. 34. Berlin: Springer Berlin Heidelberg, 2006. 425-438.

Impression

12. G. Pfurtscheller, Ch. Neuper, D. Flotzinger, M. Pregenzer, EEG-based discrimination between imagination of right and left hand movement, Electroencephalography and Clinical Neurophysiology, Volume 103, Numéro 6, décembre 1997, Pages 642-651
13. Srivastava, Durgesh K. et Lekha Bhambhu. "CLASSIFICATION DES DONNÉES À L'AIDE D'UNE MACHINE DE VECTEUR DE SUPPORT." Journal of Theoretical and Applied Information Technology (2005 - 2009): n. pag. La toile.
14. Apaydin, Ethem, 2004. Introduction to Machine Learning. books.google.com.
15. Weisstein, Eric W. «Transformée de Fourier rapide». De MathWorld - Un Wolfram Web

Ressource.

Références

16. Bajpai, Anvita et Knal Chada. «Détection des émotions faciales en temps réel à l'aide de l'assistance

Machines vectorielles. " Journal international des sciences et applications informatiques avancées, 1 (2010): 37-40. Impression.

17. Teplan, M. «Fondamentaux de la mesure d'EEG». Évaluation de la mesure scientifique 2

18. Kahana, Michael J. Robert Sekuler, Jeremy B. Caplan, Mathew Kirschen et Joseph

Madsen. «Les oscillations thêta humaines présentent une dépendance aux tâches pendant le labyrinthe virtuel

La navigation." Letters to Nature 399 (1999): 781-84, imprimé.

19. Weiss, Sabine et Horst M. Mueller. «La contribution de la cohérence EEG à la

Enquête sur la langue. " Cerveau et langage 85 (2003) 325-43. Impression

20. Brice Rebsamen. Un fauteuil roulant contrôlé par le cerveau pour naviguer dans des environnements familiers.

Interaction homme-machine [cs.HC]. université nationale de Singapour, 2009.