UNIVERSITE DE KINSHASA

**FACULTE POLYTECHNIQUE**

DEPARTEMENT DE GENIE ELECTRIQUE ET INFORMATIQUE

Une image contenant symbole, logo, Graphique, drapeau

Description générée automatiquement

Travail de fin de cycle présenté

en vue de l’obtention du grade d’

Ingénieur Civil Informaticien

Dirigé par

**Prof. Dr. Ir. Vital KUTI LUSALA**

Co-dirigé par

**Ir. Civ. Pierre SEDI NZAKUNA**

Année académique 2021 - 2022

Par l’étudiant

**Boris IKULA NEWIS**

Gradué en Sciences Appliquées

**CONCEPTION D'UN VEHICULE ROBOTIQUE SEMI AUTONOME PILOTE VIA BCI ET INTEGRANT LA VISION PAR ORDINATEUR.**

Table des matières

[ABSTRACT I](#_Toc159978290)

[EPIGRAPHE II](#_Toc159978291)

[DEDICACE III](#_Toc159978292)

[REMERCIEMENTS IV](#_Toc159978293)

[ABREVIATION VI](#_Toc159978294)

[ILLUSTRATIONS VII](#_Toc159978295)

[AVANT-PROPOS 1](#_Toc159978296)

[INTRODUCTION GENERALE 2](#_Toc159978297)

[I. PROBLEMATIQUE 2](#_Toc159978298)

[II. INTERET DU SUJET 3](#_Toc159978299)

[III. DELIMITATION DU SUJET 3](#_Toc159978300)

[IV. OBJECTIFS 4](#_Toc159978301)

[V. PLAN DU TRAVAIL 4](#_Toc159978302)

[CHAPITRE 1 : GENERALITES SUR LE CERVEAU HUMAIN, L’ICM, LA MACHINE LEARNING ET LA CLASSIFICATION DES EEG 5](#_Toc159978303)

[1.1. INTRODUCTION 5](#_Toc159978304)

[1.2. LE CERVEAU HUMAIN 5](#_Toc159978305)

[1.3. QUID LE CERVEAU HUMAIN ? CONSTITUTION 6](#_Toc159978306)

[1.4. INTERET DU CERVEAU DANS LE PRESENT TRAVAIL 7](#_Toc159978307)

[1.5. LES INTERFACES CERVEAU-MACHINE 8](#_Toc159978308)

[1.5.1. PRINCIPE D’UNE INTERFACE CERVEAU 8](#_Toc159978309)

[1.5.2. CLASSIFICATION DES INTERFACES CERVEAU-MACHINE 9](#_Toc159978310)

[1.5.3. STRUCTURE GÉNÉRALE D’UN SYSTÈME D’INTERFACE CERVEAU-MACHINE 10](#_Toc159978311)

[1.6. LES INTERFACES CERVEAU-MACHINE À ÉLECTROENCÉPHALOGRAPHIE 10](#_Toc159978312)

[1.6.1. ACQUISITION DES SIGNAUX 10](#_Toc159978313)

[1.6.2. TYPES DE SIGNAUX ÉLECTROENCÉPHALOGRAPHIQUES ET RYTHMES CÉRÉBRAUX 12](#_Toc159978314)

[1.6.3. PRÉTRAITEMENT DES SIGNAUX 13](#_Toc159978315)

[1.6.4. EXTRACTION DES CARACTÉRISTIQUES DU SIGNAL 13](#_Toc159978316)

[1.6.5. APPLICATIONS DES INTERFACES CERVEAU-MACHINE 14](#_Toc159978317)

[1.7. NOTION DE MACHINE LEARNING 15](#_Toc159978318)

[1.8. GENERALITES SUR LA CLASSIFICATION 16](#_Toc159978319)

[1.9. PERFORMANCE D’UN CLASSIFIEUR 18](#_Toc159978320)

[1.10. CLASSIFICATION DES DONNEES EEG 19](#_Toc159978321)

[1.11. CHOIX DU CLASSIFIEUR POUR UNE ICM 20](#_Toc159978322)

[CONCLUSION 21](#_Toc159978323)

[CHAPITRE 2 : CONCEPTION DU SYSTEME 22](#_Toc159978324)

[2.1. INTRODUCTION 22](#_Toc159978325)

[2.2. OBJECTIFS 22](#_Toc159978326)

[2.3. CONCEPTION DU SYSTEME 22](#_Toc159978327)

[2.3.1. PARTIE OPERATIVE 23](#_Toc159978328)

[2.3.2. PARTIE COMMANDE 23](#_Toc159978329)

[2.4. DIAGRAMMES 24](#_Toc159978330)

[CONCLUSION 27](#_Toc159978331)

[CHAPITRE 3 : OUTILS ET METHODOLOGIES 28](#_Toc159978332)

[3.1. INTRODUCTION 28](#_Toc159978333)

[3.2. OUTILS 28](#_Toc159978334)

[3.2.1. HARDWARE 28](#_Toc159978335)

[3.2.2. SOFTWARE 34](#_Toc159978336)

[3.2.3. INVENTAIRE ENERGETIQUE DES COMPOSANTS PHYSIQUES 39](#_Toc159978337)

[3.3. METHODOLOGIES 40](#_Toc159978338)

[CONCLUSION 40](#_Toc159978339)

[CHAPITRE 4. EXPERIMENTATIONS ET RESULTATS 42](#_Toc159978340)

[4.1. CRITERIUM 42](#_Toc159978341)

[4.2. EXPERIMENTATION 42](#_Toc159978342)

[4.2.1. ENTRAINEMENT DU SUJET 42](#_Toc159978343)

[4.2.2. PILOTAGE DU ROBOT AVEC DES CAPACITES D’AUTONOMIE 44](#_Toc159978344)

[4.3. RESULTATS 47](#_Toc159978345)

[Interprétation des résultats 48](#_Toc159978346)

[4.4. FEEDBACK DES SUJETS 51](#_Toc159978347)

[4.4.1. LE PORT DU CASQUE 51](#_Toc159978348)

[4.4.2. L’APPRENTISSAGE AVEC LE CASQUE 51](#_Toc159978349)

[4.4.3. LE PILOTAGE DU ROBOT 52](#_Toc159978350)

[4.4.4. CONCLUSION 52](#_Toc159978351)

[4.5. DIFFICULTES RENCONTREES 53](#_Toc159978352)

[CONCLUSION 54](#_Toc159978353)

[CONCLUSION ET PERSPECTIVES 55](#_Toc159978354)

[BIBLIOGRAPHIE - 1 -](#_Toc159978355)

[ANNEXES - 4 -](#_Toc159978356)

[A. SCRIPTS - 4 -](#_Toc159978357)

[II. PROTOCOLE EXPERIMENTAL - 11 -](#_Toc159978358)

[L'entraînement du sujet au casque Epoc+ : - 11 -](#_Toc159978359)

[Le scénario du pilotage - 11 -](#_Toc159978360)

[III. LOGS DES SUJETS - 13 -](#_Toc159978361)

[IV. PHOTO D’OPTIMUS PRIME - 16 -](#_Toc159978362)

# ABSTRACT

Les interfaces cerveau-ordinateur (BCI) sont des systèmes qui permettent de connecter le cerveau humain à un ordinateur ou à tout autre dispositif externe pour établir une communication silencieuse et sans contact entre les humains et les machines. Ces systèmes sont particulièrement utiles pour des applications médicales, notamment pour les personnes gravement handicapées qui ont perdu toute capacité motrice. En utilisant un BCI, un utilisateur peut envoyer des commandes mentales à un dispositif robotique pour effectuer des tâches spécifiques et gagner un certain niveau d'indépendance. L'objectif de ce travail est de développer un BCI couplé à un système robotique intelligent, partiellement autonome, qui utilise la vision par ordinateur et d'autres capteurs pour faciliter l'interaction entre l'utilisateur et le dispositif robotique contrôlé. Le dispositif robotique choisi est une voiture robotique, équipée d'une caméra, d'un capteur ultrasonique, et d'un micro et d'un écouteur pour créer une télé présence. Le BCI utilisé est basée sur l'expression faciale, qui est une modalité naturelle et intuitive pour contrôler le robot. Le modèle d'apprentissage automatique utilisé pour la détection d'objets est basé sur TensorFlow Lite, qui est une version allégée et optimisée de TensorFlow, adaptée aux systèmes embarqués. Le langage de programmation utilisé est le C++, qui offre des performances élevées et une gestion fine des ressources matérielles. Le traitement de la vision par ordinateur est intégré directement sur le dispositif robotique, en utilisant OpenCV et TFLite, ce qui permet de réduire la latence et la dépendance à une connexion réseau. L'IP call est utilisé pour établir une communication audio bidirectionnelle entre l'utilisateur et le robot, ce qui permet de renforcer le sentiment de présence et d'immersion. Les résultats obtenus montrent que le système développé est capable de contrôler le robot de manière efficace et ergonomique, en utilisant les expressions faciales de l'utilisateur et en tenant compte des informations fournies par les capteurs. Le système présente également un potentiel d'application pour les personnes dans le besoin, notamment les paraplégiques, qui pourraient utiliser le robot comme un moyen de transport et d'exploration, tout en restant en contact avec leur environnement.

# EPIGRAPHE

« Les ordinateurs ne sont pas fiables, les hommes le sont encore moins. À l’origine de chaque erreur attribuée à l’ordinateur, on trouve au moins deux erreurs humaines, dont celle qui consiste à accuser l’ordinateur. »

Loi de non-fiabilité de Gibbs

# DEDICACE

Le terme « héros » n’est pas suffisant pour qualifier celui qui, après avoir vendu peau de sa chair, a passé son existence à souhaiter la venue des jours meilleurs dans votre vie, tout en restant un modèle pour vous malgré les innombrables vagues qu’apporte ce monde. A l’amour, à vous mes parents.

# REMERCIEMENTS

Louange, gloire, sagesse, action de grâce et honneur au Seigneur, qui nous a accordé les aptitudes physiques, mentales, sanitaires et intellectuelles nécessaires pour accomplir notre parcours à la Faculté Polytechnique de l’Université de Kinshasa. Nous témoignons notre reconnaissance à nos parents, IKULA ELOCK Jean et MIMBU MABWADI Patience, ainsi qu’à nos frères et sœurs KAYONGO MBANGI Pathy, IKULA NGUDI Plamedi Gabriel, très chère IKULA RIZIKI Grâce, la grande famille IKULA, MIMBU, NKEYE, YOYO, SEKO et BANGABANGA pour leur soutien indéfectible et leur encouragement constant tout au long de nos études d’Ingénieur Civil.

Notre gratitude se tourne vers notre directeur, le Professeur Vital KUTI LUSALA, pour son enthousiasme permanent, son accompagnement scientifique, la mise à disposition du Kit complet des matériels utilisés pour ce mémoire, et le temps qu’il a su gérer pour nous, malgré ses nombreuses occupations. Nous remercions très particulièrement notre co-directeur, l’ingénieur civil Pierre SEDI NZAKUNA, sans qui nous n’aurions pas été initiés à ce projet, pour sa soutenance aiguë et remarquable dans toutes les étapes franchies. Nous remercions également l’Assistant Jollam-J. LANGANGA MAYALA, le pionnier de la recherche sur les BCI à la Faculté Polytechnique de l’Université de Kinshasa, qui nous a initié aux concepts clés des BCI et nous a prêté son casque Emotiv EPOC+ pour les besoins de notre travail. Que tous les professeurs, chefs de travaux et assistants de la Faculté Polytechnique de l’Université de Kinshasa, trouvent ici l’expression de notre gratitude pour le bagage scientifique qu’ils nous ont transmis, sans lequel ce travail n’aurait pas vu le jour.

À tous ceux qui nous ont épaulés pour la réalisation de ce travail et pendant nos expériences de BCI, nous citons Nico KABONGO, Dinovic KWETE BANZA, Osée YEMBE, Pierrette MBOKASHANGA LUMU, Odilon BANINGI BAMOTANGA avec l’équipe FabLab Polytechnique, Victoria BASEMENANE ASHA. Merci du fond du cœur.

À ces personnes de marque, mon amie Pierrette LUMU MBOKASHANGA, William MUTOMBO MUTOMBO, David OMOKOKO DIOWO, Grâce GAHERERI GATOTO, Grâce BETOKO IMEKA, Grâce MALAR, Benny Daudet MPONGO NZOMBO, Aby PENGE, Jérémie BOALELA, Christian MONDO, Zin El Abidine MAHUKU, Alfred TOENGAHO, Michel HINGASE, Sacha KASAKA, Daniel KADIMA, Kevin LOBOTA, le mentor Russel MOYA, Amos MUDJENE, Kevin Jr. M’BUYAMBA, Dan TSHITAMBA, Fidèle MUTUMOSI, Eden BEYA MARCEL, Guillaume KANGAMINA, ma pote Victoria KABANGE.

À la chorale NODASA, ma deuxième famille, Rolande MAVULA, Relindis NGILA, Hervé LEBA, Dr. Miki MAKAWANI, Grâce MUBENGA, David KINTIBA, Franck KALOMBO, Dorcas BIZEA, Immaculée PANGADJANGA, Glodi MBENZA, Alphie MUBENGA, Merveille NDOMBA, Judicaëlle NZINGA.

À tous nos collègues et camarades de classe avec qui nous avons lutté ensemble jusqu’à l’achèvement de ce parcours à la Faculté Polytechnique, particulièrement à ceux de Génie Électrique & Informatique.

À mes collègues de classe en Génie Informatique avec qui nous avons partagé les joies et les peines jusqu’au bout, le CPA Caleb DUBO, Obed TSHIMANGA, Fidèle NSADI et Gilbert MBULA qui m’a inspiré l’option informatique.

À tonton Alain IKULA AZIZA, ingénieur militaire, qui m’a toujours servi de modèle scientifique depuis tout petit et m’a donné le goût de faire la Faculté Polytechnique.

À tous ceux qui ont été là pour nous, vous tous qui avez de l’estime pour cette modeste personne que nous sommes ; vous tous, que les lignes de cette page ne peuvent nommer de manière exhaustive, nous disons un grand merci, pour tout.

# ABREVIATION

BCI : Brain-Computer Interface

EEG : Electroencéphalogramme

FPS : Frames Per Seconds

ICM : Interface Cerveau Machine

IP : Internet Protocole

kNN : k Nearest Neighbors

LDA : Linear Discriminant Analysis

ML : Machine Learning

MLP : Multi Layer Perceptron

POC : Proof Of Concept

RAM : Random Access Memory

RPI : Raspberry Pi

SNR : Signal-to-Noise Ratio

SSH : Secure Shell

SVM : Support Vector Machine

TFLite : TensorFlow Lite

# ILLUSTRATIONS

[Figure 1. Le cerveau humain 7](#_Toc159978365)

[Figure 2. Logigramme Optimus prime 25](#_Toc159978366)

[Figure 3. Diagramme de séquence du système 26](#_Toc159978367)

[Figure 4. Casque Emoiv Epoc+ 29](#_Toc159978368)

[Figure 5. Equipement SunFounder Picar-V 30](#_Toc159978369)

[Figure 6. Raspberry Pi 4 31](#_Toc159978370)

[Figure 7. Webcam 32](#_Toc159978371)

[Figure 8. Baffle 33](#_Toc159978372)

[Figure 9. Ultrason 33](#_Toc159978373)

[Figure 10. Vue des électrodes dans Emotiv Launcher 35](#_Toc159978374)

[Figure 11. OpenVibe 35](#_Toc159978375)

[Figure 12. Trio C++, OpenCV et TensorflowLite 36](#_Toc159978376)

[Figure 13. Putty et K-meleon Browser 38](#_Toc159978377)

[Figure 14. OpenVibe: Les 4 classes en mode online 43](#_Toc159978378)

[Figure 15. Design de l'arborescence des classes 44](#_Toc159978379)

[Figure 16. Lancement du serveur principal listen.py 45](#_Toc159978380)

[Figure 17. CamUltrasonic prêt à être exécuté 46](#_Toc159978381)

[Figure 18. Streaming du webcam sur K-meleon browser 47](#_Toc159978382)

[Figure 19. Les fichiers .log après leur sauvegarde 50](#_Toc159978383)

[Figure 20. Optimus Prime - 16 -](#_Toc159978384)

# AVANT-PROPOS

La révolution scientifique, telle la quête du saint Graal, a engendré un éveil majeur. La technologie, arme secrète de la science, a propulsé l'humanité à travers le temps et l'espace.

Malgré sa noblesse, la cause scientifique a véritablement fait progresser notre société en résolvant divers problèmes du quotidien. Son influence s'étendant à tous les domaines de la vie, la science devient indispensable pour atteindre le summum du bien-être.

Au 21e siècle, la technologie, résultat de la symbiose entre divers domaines scientifiques, a permis le développement d'Interfaces Cerveau-Machine (ICM) à la fois ludiques, résolutives, et eschatologiques.

Ce travail démontre la puissance du BCI en tant qu'outil d'assistance dans les tâches quotidiennes des hommes, notamment pour ceux ayant perdu l'usage de certains membres. Il s'agit d'une démonstration de faisabilité, un "proof of concept", confirmant la capacité du BCI à maintenir le contrôle sur un équipement capable de s'auto-gérer dans des situations spécifiques.

Notre engagement se focalise sur la résolution de problèmes tout en favorisant l'ergonomie pour les utilisateurs équipés d'un BCI non-invasif, pilotant ainsi un robot en voie d'intelligence croissante.

Avec une confiance profonde dans le processus évolutif, nous aspirons à ce que ce projet serve de fondement pour un avenir meilleur. Nous espérons que l'utilisation des technologies ne nuit pas à l'homme dans le besoin, mais contribue plutôt à son bien-être, l'aidant à retrouver un équilibre dans un environnement sain avec un corps sain.

# INTRODUCTION GENERALE

## PROBLEMATIQUE

La vulnérabilité des personnes handicapées ne nous laisse pas indifférents, étant au courant des difficultés auxquelles ils font face dans la réalisation de telle ou telle autre tâche de leur vie de chaque jour. Ces personnes ont besoin des solutions technologiques adaptées à leur situation, qui leurs permettent de réaliser des activités quotidiennes et de communiquer avec leur entourage. Les axes actuels de la science nous ouvrent un nouveau monde où les applications de la technologie de l'ingénierie servent aussi aux avancées technologiques des biomédicaux. Parmi les technologies émergentes, les BCI offrent de nouvelles perspectives pour développer des solutions médicales basées sur la robotique.

Les Interfaces Cerveau-Ordinateur (BCI) cherchent à établir une communication sans contact entre humains et machines. Elles permettent à un utilisateur d'envoyer des commandes à un ordinateur ou à un dispositif externe en utilisant uniquement son activité cérébrale, mesurée par des capteurs électroencéphalographiques (EEG). Les BCI offrent ainsi une nouvelle modalité d'interaction humain-machine, qui peut être utile dans de nombreux domaines, notamment médical, éducatif, ludique, artistique, etc.[[1]](#footnote-1)

Nous nous focalisons sur les BCI non-invasifs, qui présentent l’avantage d’être plus accessibles, plus adaptés, notamment avec le paradigme asynchrone, et moins coûteux en termes financiers et de préservation des fonctions vitales de l’homme. Parmi les applications potentielles des BCI, l'une des plus importantes est le besoin d'assistance pour les personnes gravement handicapées, en particulier les paraplégiques. Ces personnes souffrent d'une perte totale ou partielle de la capacité motrice, ce qui les empêche de réaliser des activités quotidiennes et de communiquer avec leur entourage.

Les BCI peuvent leur offrir une alternative pour contrôler des technologies d'assistance, telles que des fauteuils roulants, des prothèses, des ordinateurs, des robots[[2]](#footnote-2). Cependant, le développement d'un BCI couplé à un système robotique intelligent pour faciliter l'interaction pose de nombreux défis, tant du point de vue technique que du point de vue de l'expérience utilisateur. Il faut en effet concevoir un système qui soit fiable, robuste, adaptatif, ergonomique, et qui respecte les besoins et les préférences de l'utilisateur. Il faut également prendre en compte les contraintes liées à l'environnement, à la connexion, à la sécurité, à l'éthique, etc.

## INTERET DU SUJET

L’intérêt du sujet réside dans la question de savoir comment utiliser les interfaces cerveau-machine (BCI) pour contrôler un système robotique intelligent, partiellement autonome, qui puisse assister les personnes handicapées dans leur mobilité et leur interaction avec leur environnement. En effet, prendre indéfiniment contrôle d’un objet s’avère une tâche épuisante au fur et à mesure que nous sommes obligés d’attribuer toute notre attention et énergie sur la chose sans perdre de vue tout ce qui peut arriver dans l’environnement où nous manifestons ce contrôle. Il est donc nécessaire de concevoir un système qui soit capable de s’adapter aux situations changeantes, de prendre des décisions autonomes, et de communiquer avec l’utilisateur.

Notre travail, intitulé CONCEPTION D'UN VEHICULE ROBOTIQUE SEMI AUTONOME PILOTE VIA BCI ET INTEGRANT LA VISION PAR ORDINATEUR, est l’étude d’un proof of concept sur les ajouts des capacités sur le robot contrôlé, comme l’a montré le travail de Pierre Sedi[[3]](#footnote-3) sur lequel nous nous basons, qui pourra être habile et apte de prendre des décisions face à des circonstances événementielles. Nous nous inspirons de son approche qui utilise l’expression faciale comme modalité de contrôle du robot, offrant une méthode naturelle et intuitive. Nous cherchons à améliorer son système en utilisant la vision par ordinateur et d’autres capteurs pour doter le robot d’une intelligence artificielle, qui lui permette de naviguer de manière semi-autonome, de détecter les obstacles, et de réagir en conséquence.

Le travail se veut être la réponse à un cahier de charge, que nous découvrirons au chapitre 2, qui nous a été donné pour ouvrir un champ d’application sur les travaux des BCI asynchrones, insistant sur la facilité et l’ergonomie de l’utilisateur.

## DELIMITATION DU SUJET

Notre travail s’inscrit dans le domaine des interfaces cerveau-machine (BCI) couplées à des systèmes robotiques intelligents, qui visent à offrir une solution d’assistance aux personnes handicapées. Nous nous intéressons plus particulièrement au smart video car, une voiture robotique contrôlée par un BCI basé sur l’expression faciale, qui utilise la vision par ordinateur et d’autres capteurs pour naviguer de manière semi-autonome[[4]](#footnote-4).

Après avoir exposé l’intérêt de notre travail, nous allons nous limiter dans l'étude expérimentale des comportements que nous attribuons au robot, sa réactivité lorsqu’il reçoit des commandes, sa fluidité dans le rendu visuel des images qu’il capte avec ses capteurs, son efficacité face à la perte de connexion avec l’utilisateur Les défis majeurs du travail se trouvent dans la gestion des ressources que nous offre le RPI 4, qui est l’élément central, la partie intelligente, le cerveau du robot.

Nous balayons les notions apprises en systèmes embarqués, traitement des signaux, réseau, analyse des circuits, programmation, génie logiciel, systèmes à multiprocesseur et machines parallèles, machine learning, operating system, computer design, électroniques, mesures électriques, interface homme machine, automatique, informatique temps réel, etc.

## OBJECTIFS

Le flux d'information enregistré à partir de la problématique, de l’intérêt, et de la délimitation du sujet, nous pousse à définir les objectifs principaux de notre travail, qui sont :

- Concevoir un système robotique intelligent contrôlé par un BCI qui, utilisant le paradigme des expressions faciales pour extraire des signaux EEG de l’individu, permette à l’utilisateur de piloter à distance une voiture robotique, en tenant compte des informations fournies par les capteurs.

- Implémenter les comportements du robot en approche des obstacles, en situation d’embourbement, ou en cas de perte de connexion.

- Évaluer la performance du système et la satisfaction des utilisateurs à partir des expériences menées au laboratoire.

## PLAN DU TRAVAIL

Nous avons subdivisé notre travail en quatre (4) chapitres qui sont :

1. Généralités sur le cerveau humain et le BCI
2. Conception du système
3. Outils et méthodologies
4. Expérimentations et résultats

C’est par là que nous mettons fin à l’introduction générale, nous vous invitons à passer du bon temps dans la lecture de la suite du travail.

# CHAPITRE 1 : GENERALITES SUR LE CERVEAU HUMAIN, L’ICM, LA MACHINE LEARNING ET LA CLASSIFICATION DES EEG

## INTRODUCTION

Ce chapitre présente les notions de base sur le cerveau humain, les interfaces cerveau-machine (ICM) et l’électroencéphalographie (EEG). Il s’agit de comprendre comment le cerveau humain produit et traite des signaux électriques, comment ces signaux peuvent être mesurés et interprétés par des dispositifs externes, et quelles sont les applications possibles des ICM basées sur l’EEG.

Il présente également les notions sur l’apprentissage machine et la classification, ainsi que les spécificités de la classification des données électroencéphalographiques (EEG). Il s’agit de comprendre comment les algorithmes d’apprentissage machine peuvent apprendre à partir de données, comment ils peuvent classifier des données en différentes catégories, et quels sont les défis et les critères de choix d’un classifieur adapté aux données EEG.

## LE CERVEAU HUMAIN

Cet organe miracle ne sait nullement cesser d’étonner l’homme moderne en dépit de son éventail de connaissance qui prend le large au fil des années. Notre humble tâche consiste à avoir une idée commune sur le concept du cerveau humain partant, bien évidemment, des définitions attribuées à cet organe dans d’autres domaines de la science. Nous en décrivons certains :

**Philosophiquement**, le cerveau est l’instrument qui nous permet de percevoir, de réfléchir et d’agir. Il est le moteur qui donne un sens à notre existence.

**Sociologiquement**, le cerveau est le maestro de notre organisme, le dirigeant tout en se dirigeant lui-même. Il est à l’origine de nos comportements et, par conséquent, de nos interactions avec les autres membres de la société.

**Scientifiquement**, le cerveau est un domaine clé car il recèle encore de nombreux mystères, que ce soit dans son développement, son fonctionnement normal et pathologique, ou ses capacités d’adaptation. Il est crucial aujourd’hui de déchiffrer les mécanismes qui sous-tendent nos capacités intellectuelles, nos émotions et les comportements moteurs qui en découlent.

En tant qu’ingénieur, ces perspectives sont essentielles pour comprendre comment nous pouvons concevoir des technologies, comme les interfaces cerveau-machine, qui peuvent interagir efficacement et de manière significative avec ces aspects complexes du cerveau humain.

## QUID LE CERVEAU HUMAIN ? CONSTITUTION

Au cœur de notre être, se trouve un orchestre complexe et harmonieux, le système nerveux central, composé du cerveau et de la moelle épinière. Il est le maître d’œuvre qui intègre les informations, commande nos mouvements et assure nos fonctions cognitives.

Le cerveau, pesant environ 1,3 kg et composé à 75% d’eau, est l’organe le plus protégé de notre corps. Il est enveloppé par trois couches protectrices, les méninges, et baigne dans le liquide céphalo-rachidien, qui amortit les chocs. Malgré sa petite taille, il consomme 15 à 20% de l’énergie produite par notre corps, principalement sous forme de glucose. Un réseau dense de vaisseaux sanguins le traverse, assurant un apport constant en oxygène.

Le cerveau est divisé en deux hémisphères, droit et gauche, reliés par le corps calleux. Chaque hémisphère est composé de plusieurs lobes : le lobe frontal, siège du raisonnement, du langage et de la coordination motrice volontaire ; le lobe pariétal, qui nous permet de prendre conscience de notre corps et de l’espace environnant ; le lobe occipital, qui intègre les messages visuels ; le lobe temporal, centre de l’audition, de la mémoire et des émotions ; le lobe limbique, qui traite les informations relatives aux émotions, aux affects et à la mémoire ; et le lobe de l’insula, qui nous permet de ressentir la douleur, les odeurs et le goût.

Enfin, le cervelet, qui contrôle notre équilibre et la coordination de nos mouvements, et le tronc cérébral, qui sert de point de passage entre les hémisphères cérébraux et la moelle épinière, complètent ce tableau.

Cette narration met en lumière la complexité et la beauté de notre cerveau, un organe qui, malgré les progrès de la science, conserve encore de nombreux mystères à dévoiler.

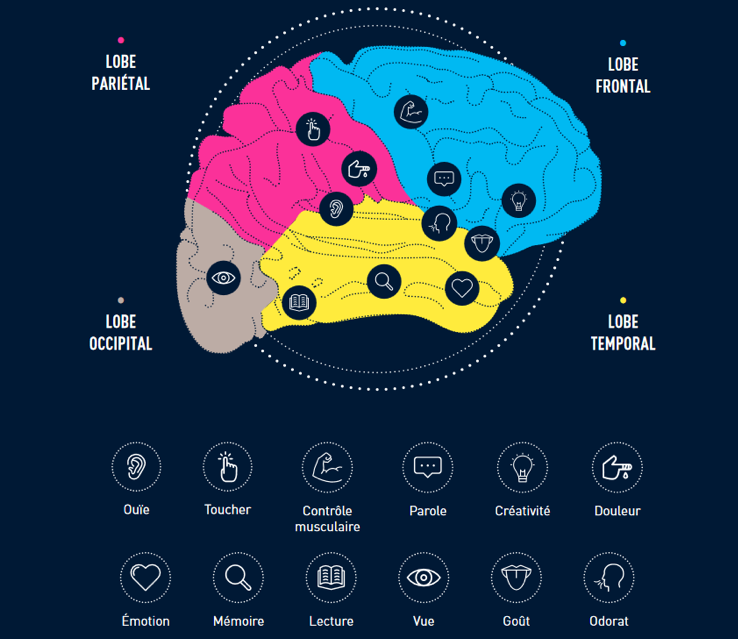


Figure 1. Le cerveau humain[[5]](#footnote-5)

## INTERET DU CERVEAU DANS LE PRESENT TRAVAIL

Le cerveau, une merveille de la nature, est un réseau complexe de 100 milliards de neurones, ces cellules nerveuses qui sont les unités de base de notre système nerveux. Ces neurones sont reliés par des axones, des prolongements qui sont protégés par une gaine appelée myéline. Cette myéline est produite par des cellules spéciales appelées oligodendrocytes, qui font partie d’un groupe de cellules connues sous le nom de cellules gliales, tout aussi nombreuses que les neurones.

Le cerveau est structuré en différentes régions, dont le cortex, ou substance grise, qui est la couche externe du cerveau et contient les corps cellulaires des neurones. En dessous du cortex se trouve la substance blanche, qui est composée des axones des neurones, enveloppés dans leur gaine de myéline. Le cerveau abrite également quatre cavités appelées ventricules cérébraux, où circule le liquide cérébrospinal. Au centre du cerveau, les noyaux gris centraux, également connus sous le nom de ganglions de la base, jouent un rôle crucial dans le contrôle du comportement et de l’apprentissage.

Le cerveau est un expert en communication. Les neurones communiquent entre eux par des signaux électriques, appelés influx nerveux. Ces signaux se propagent le long des axones jusqu’à la terminaison synaptique, où ils déclenchent la libération de substances chimiques appelées neurotransmetteurs. Ces neurotransmetteurs peuvent activer ou inhiber un autre neurone, permettant ainsi à l’influx nerveux de continuer son chemin.

Il existe plusieurs types de neurotransmetteurs, certains excitateurs comme le glutamate, d’autres inhibiteurs comme le GABA. Parmi les plus connus, on trouve la dopamine, la sérotonine, l’histamine et l’acétylcholine. Les neurones qui produisent la dopamine, situés dans une région profonde du cerveau appelée la substance noire, sont essentiels pour le contrôle du mouvement.

Ces connaissances sur le fonctionnement du cerveau sont fondamentales pour le développement des interfaces cerveau-machine (ICM). Les ICM sont des dispositifs qui permettent une communication directe entre le cerveau et un système externe, comme une voiture robotisée. En captant et en interprétant les signaux électriques du cerveau, les ICM peuvent traduire les intentions de l’utilisateur en commandes pour le système externe, permettant ainsi un contrôle direct par la pensée. C’est un domaine de recherche passionnant et prometteur, qui pourrait révolutionner notre interaction avec la technologie.

## LES INTERFACES CERVEAU-MACHINE

### PRINCIPE D’UNE INTERFACE CERVEAU

Une interface cerveau-machine (ICM) est un système de liaison directe entre un cerveau et un ordinateur, permettant à un individu d’effectuer des tâches sans passer par l’action des nerfs périphériques et des muscles. Ce type de dispositif permet de contrôler par la pensée un ordinateur, une prothèse ou tout autre système automatisé, sans solliciter ses bras, mains ou jambes. [Le concept remonte à 1973 et les premiers essais chez l’homme datent du milieu des années 90](https://www.inserm.fr/dossier/interface-cerveau-machine-icm/)[[6]](#footnote-6). Le principe d’une ICM est de capter les signaux cérébraux de l’utilisateur, de les traiter et de les traduire en commandes pour la machine, et éventuellement de fournir une rétroaction sensorielle à l’utilisateur.

### CLASSIFICATION DES INTERFACES CERVEAU-MACHINE

Les ICM peuvent être classées selon plusieurs critères, comme le type de signaux cérébraux utilisés, le mode d’acquisition des signaux, le type de tâches réalisées, ou le type de rétroaction sensorielle. Selon le type de signaux cérébraux utilisés, on distingue les ICM actives, les ICM réactives et les ICM passives. Les ICM actives utilisent des signaux cérébraux produits volontairement par l’utilisateur, comme l’imagination de mouvements ou l’attention visuelle. Les ICM réactives utilisent des signaux cérébraux provoqués par une stimulation externe, comme les potentiels évoqués visuels ou auditifs.

Les ICM passives utilisent des signaux cérébraux liés à l’état cognitif ou émotionnel de l’utilisateur, comme le niveau d’alerte ou le stress. Selon le mode d’acquisition des signaux, on distingue les ICM invasives, les ICM partiellement invasives et les ICM non invasives. Les ICM invasives nécessitent l’implantation d’électrodes à l’intérieur du cerveau, ce qui permet d’enregistrer des signaux de haute qualité, mais qui présente des risques d’infection, de rejet ou de lésion cérébrale. Les ICM partiellement invasives nécessitent l’implantation d’électrodes sous le crâne, mais au-dessus du cerveau, ce qui permet d’enregistrer des signaux de bonne qualité, mais qui présente également des risques d’infection ou de rejet. Les ICM non invasives nécessitent l’utilisation d’électrodes ou de capteurs à la surface du cuir chevelu ou du crâne, ce qui permet d’enregistrer des signaux de faible qualité, mais qui présente moins de risques pour la santé. Selon le type de tâches réalisées, on distingue les ICM de communication, les ICM de contrôle et les ICM de restauration.

Les ICM de communication permettent à l’utilisateur d’envoyer des messages ou des choix à la machine, comme des lettres, des chiffres, des mots ou des icônes. Les ICM de contrôle permettent à l’utilisateur de manipuler la machine, comme un curseur, un fauteuil roulant, un robot ou une prothèse. Les ICM de restauration permettent à l’utilisateur de récupérer une fonction perdue ou altérée, comme la vision, l’audition, la parole ou la motricité. Selon le type de rétroaction sensorielle, on distingue les ICM sans rétroaction, les ICM avec rétroaction visuelle, les ICM avec rétroaction auditive, les ICM avec rétroaction tactile ou les ICM avec rétroaction multisensorielle. Les ICM sans rétroaction ne fournissent aucune information à l’utilisateur sur le résultat de son action. Les ICM avec rétroaction visuelle utilisent des stimuli visuels, comme des images, des couleurs, des formes ou des textes, pour indiquer à l’utilisateur l’état de la machine ou la validité de son choix. Les ICM avec rétroaction auditive utilisent des stimuli auditifs, comme des sons, des voix, des musiques ou des bruits, pour informer l’utilisateur de la même manière que la rétroaction visuelle. Les ICM avec rétroaction tactile utilisent des stimuli tactiles, comme des vibrations, des pressions, des températures ou des textures, pour transmettre à l’utilisateur des sensations liées à la machine ou à son environnement. Les ICM avec rétroaction multisensorielle combinent plusieurs types de rétroaction sensorielle, pour augmenter l’immersion et l’efficacité de l’utilisateur.

### STRUCTURE GÉNÉRALE D’UN SYSTÈME D’INTERFACE CERVEAU-MACHINE

La structure générale d’un système d’interface cerveau-machine comprend quatre modules principaux : le module d’acquisition, le module de prétraitement, le module d’extraction et le module d’application. Le module d’acquisition est chargé de capter les signaux cérébraux de l’utilisateur, à l’aide d’électrodes ou de capteurs, et de les numériser pour les rendre exploitables par l’ordinateur. Le module de prétraitement est chargé de filtrer les signaux cérébraux, pour éliminer les bruits et les artéfacts, et de les segmenter en fenêtres temporelles, pour faciliter leur analyse. Le module d’extraction est chargé d’extraire des caractéristiques pertinentes des signaux cérébraux, en utilisant des méthodes de transformation ou de réduction de dimension, et de les classifier en catégories ou en commandes, en utilisant des algorithmes d’apprentissage automatique ou de reconnaissance de motifs. Le module d’application est chargé d’interpréter les commandes issues du module d’extraction, et de les transmettre à la machine, en utilisant des protocoles de communication ou de contrôle. Le module d’application est également chargé de fournir une rétroaction sensorielle à l’utilisateur, en utilisant des dispositifs de stimulation ou de restitution.

## LES INTERFACES CERVEAU-MACHINE À ÉLECTROENCÉPHALOGRAPHIE

### ACQUISITION DES SIGNAUX

#### MESURE DU POTENTIEL ÉLECTRIQUE CÉRÉBRAL

L’électroencéphalographie (EEG) est une technique non invasive qui permet de mesurer le potentiel électrique cérébral à la surface du cuir chevelu, à l’aide d’électrodes métalliques ou de capteurs secs. Le potentiel électrique cérébral est la somme des potentiels postsynaptiques des neurones corticaux, qui sont synchronisés et orientés de manière à générer un champ électrique détectable à l’extérieur du crâne. L’EEG est une technique qui présente plusieurs avantages, comme une haute résolution temporelle, un faible coût, une facilité d’utilisation et une portabilité. L’EEG présente également des inconvénients, comme une faible résolution spatiale, une sensibilité aux bruits et aux artéfacts, et une variabilité inter et intra-individuelle.

#### TYPES D’ÉLECTRODES

Il existe deux types principaux d’électrodes utilisées pour l’EEG : les électrodes métalliques et les capteurs secs. Les électrodes métalliques sont des disques métalliques, généralement en argent ou en or, qui sont fixés au cuir chevelu à l’aide d’un gel conducteur ou d’une pâte adhésive. Les électrodes métalliques permettent d’enregistrer des signaux EEG de bonne qualité, mais elles nécessitent une préparation du cuir chevelu, comme le nettoyage, le grattage ou le rasage, pour réduire l’impédance électrique. Les capteurs secs sont des dispositifs qui n’utilisent pas de gel ou de pâte, mais qui reposent sur des contacts mécaniques ou des matériaux piézoélectriques pour capter les signaux EEG. Les capteurs secs permettent d’enregistrer des signaux EEG de manière plus rapide et plus confortable, mais ils sont plus sensibles aux bruits et aux mouvements de la tête.

#### ÉLECTRODES ET ÉLECTROLYTES

Les électrodes et les électrolytes sont les éléments qui permettent de convertir les signaux électriques cérébraux en signaux électriques mesurables par l’EEG. Les électrodes sont les dispositifs qui sont en contact avec le cuir chevelu, et qui captent les variations du potentiel électrique cérébral. Les électrolytes sont les substances qui assurent la conductivité électrique entre les électrodes et le cuir chevelu, en créant une interface liquide-solide. Les électrolytes peuvent être des gels, des pâtes, des solutions salines ou des crèmes, qui contiennent des ions, comme le chlorure de sodium (NaCl) ou le chlorure d’argent (AgCl). Les électrodes et les électrolytes doivent être choisis en fonction de leurs propriétés, comme la résistance, la capacité, la polarisation ou la corrosion, pour garantir la qualité et la stabilité des signaux EEG.

#### NOMBRE ET POSITIONNEMENT DES ÉLECTRODES

Le nombre et le positionnement des électrodes utilisées pour l’EEG dépendent du type d’application et du niveau de précision souhaité. Il existe plusieurs systèmes de référence pour définir le positionnement des électrodes, comme le système 10-20, le système 10-10 ou le système 10-5. Le système 10-20 est le plus courant, et il consiste à placer les électrodes selon des lignes imaginaires qui divisent le cuir chevelu en 10% ou 20% de sa circonférence ou de sa longueur. Les électrodes sont nommées selon leur position et leur hémisphère, en utilisant des lettres et des chiffres. Les lettres indiquent le lobe cérébral correspondant, comme F pour frontal, P pour pariétal, O pour occipital ou T pour temporal. Les chiffres indiquent la distance par rapport à la ligne médiane, en utilisant des nombres pairs pour l’hémisphère droit et des nombres impairs pour l’hémisphère gauche. Par exemple, l’électrode Fp1 se situe sur le lobe frontal, à 10% de la ligne médiane, sur l’hémisphère gauche. Le système 10-20 comporte 21 électrodes, mais il peut être étendu à 64, 128 ou 256 électrodes, en utilisant le système 10-10 ou le système 10-5, qui divisent le cuir chevelu en 10% ou 5% de sa circonférence ou de sa longueur.

### TYPES DE SIGNAUX ÉLECTROENCÉPHALOGRAPHIQUES ET RYTHMES CÉRÉBRAUX

#### LES TYPES DE SIGNAUX ÉLECTROENCÉPHALOGRAPHIQUES

Les signaux électroencéphalographiques (EEG) sont les signaux électriques cérébraux mesurés à la surface du cuir chevelu, à l’aide d’électrodes ou de capteurs. Les signaux EEG sont caractérisés par leur amplitude, leur fréquence, leur phase et leur distribution spatiale. L’amplitude est la valeur maximale du signal EEG, exprimée en microvolts (µV). La fréquence est le nombre de cycles du signal EEG par seconde, exprimée en hertz (Hz). La phase est l’angle du signal EEG par rapport à un point de référence, exprimée en degrés ou en radians. La distribution spatiale est la répartition du signal EEG sur le cuir chevelu, exprimée en termes de localisation, de symétrie ou de cohérence. Les signaux EEG peuvent être classés en deux types principaux : les signaux EEG spontanés et les signaux EEG évoqués. Les signaux EEG spontanés sont les signaux EEG qui reflètent l’activité cérébrale spontanée de l’utilisateur, sans stimulation externe. Les signaux EEG spontanés sont souvent analysés en termes de rythmes cérébraux, qui sont des oscillations du signal EEG dans des bandes de fréquence spécifiques. Les signaux EEG évoqués sont les signaux EEG qui sont provoqués par une stimulation externe, comme visuelle, auditive, tactile ou électrique.

Les signaux EEG évoqués sont souvent analysés en termes de potentiels évoqués, qui sont des réponses du signal EEG à la stimulation, caractérisées par leur latence, leur amplitude et leur forme.

#### LES RYTHMES CÉRÉBRAUX

Les rythmes cérébraux sont des oscillations du signal EEG spontané dans des bandes de fréquence spécifiques, qui sont associées à des états physiologiques ou cognitifs de l’utilisateur. Il existe plusieurs types de rythmes cérébraux, comme le rythme delta, le rythme thêta, le rythme alpha, le rythme bêta, le rythme gamma ou le rythme mu.

Le rythme delta est le rythme le plus lent, avec une fréquence comprise entre 0,5 et 4 Hz. Il est observé principalement pendant le sommeil profond, et il est lié à la régénération des tissus cérébraux. Le rythme thêta est le rythme avec une fréquence comprise entre 4 et 8 Hz. Il est observé principalement pendant le sommeil léger, la relaxation, la méditation ou la créativité. Le rythme alpha est le rythme avec une fréquence comprise entre 8 et 13 Hz. Il est observé principalement pendant l’éveil calme, les yeux fermés, l’attention ou la mémorisation. Le rythme bêta est le rythme avec une fréquence comprise entre 13 et 30 Hz. Il est observé principalement pendant l’éveil actif, les yeux ouverts, le raisonnement ou la concentration. Le rythme gamma est le rythme le plus rapide, avec une fréquence supérieure à 30 Hz. Il est observé principalement pendant les processus cognitifs complexes, comme la perception, l’apprentissage ou la résolution de problèmes. Le rythme mu est un rythme particulier, avec une fréquence comprise entre 8 et 12 Hz. Il est observé principalement sur le cortex moteur, et il est modulé par l’imagination ou l’exécution de mouvements.

### PRÉTRAITEMENT DES SIGNAUX

#### ARTÉFACTS DANS LE SIGNAL ÉLECTROENCÉPHALOGRAPHIQUE

Les artéfacts sont des signaux indésirables qui se superposent au signal EEG, et qui peuvent altérer sa qualité et sa fiabilité. Les artéfacts peuvent avoir une origine physiologique ou non physiologique. Les artéfacts physiologiques sont des signaux produits par l’activité d’autres organes que le cerveau, comme les muscles, les yeux, le cœur ou la respiration. Les artéfacts non physiologiques sont des signaux produits par des sources externes, comme les appareils électriques, les câbles, les mouvements de la tête ou les interférences électromagnétiques. Les artéfacts peuvent être détectés par des méthodes visuelles, statistiques ou automatiques, et ils peuvent être supprimés par des méthodes de filtrage, de réjection ou de correction.

#### SUPPRESSION DES ARTÉFACTS

La suppression des artéfacts est le processus qui consiste à éliminer ou à réduire les signaux indésirables qui se superposent au signal EEG. Il existe plusieurs méthodes de suppression des artéfacts, comme le filtrage, la réjection ou la correction. Le filtrage est une méthode qui consiste à appliquer un filtre au signal EEG, pour atténuer ou supprimer les composantes fréquentielles qui correspondent aux artéfacts. Il existe différents types de filtres, comme les filtres passe-bas, les filtres passe-haut, les filtres passe-bande ou les filtres coupe-bande. La réjection est une méthode qui consiste à éliminer les segments du signal EEG qui contiennent des artéfacts, en utilisant des critères de seuil, de variance ou de corrélation. La correction est une méthode qui consiste à estimer et à soustraire les artéfacts du signal EEG, en utilisant des techniques de régression, de moyenne, de décomposition ou d’optimisation.

### EXTRACTION DES CARACTÉRISTIQUES DU SIGNAL

#### REPRÉSENTATION DU SIGNAL ÉLECTROENCÉPHALOGRAPHIQUE DANS LE DOMAINE TEMPOREL

La représentation du signal EEG dans le domaine temporel est la représentation la plus simple et la plus directe du signal EEG, qui consiste à tracer le potentiel électrique cérébral en fonction du temps. Cette représentation permet d’observer la forme, l’amplitude et la durée des signaux EEG, ainsi que les éventuels potentiels évoqués. Cette représentation ne permet pas de distinguer les différentes composantes fréquentielles des signaux EEG, ni de comparer les signaux EEG de différents canaux ou de différents sujets.

#### REPRÉSENTATION DU SIGNAL ÉLECTROENCÉPHALOGRAPHIQUE DANS LE DOMAINE FRÉQUENTIEL

La représentation du signal EEG dans le domaine fréquentiel est la représentation qui consiste à décomposer le signal EEG en une somme de signaux sinusoïdaux de différentes fréquences, amplitudes et phases. Cette représentation permet d’observer les différentes bandes de fréquence des signaux EEG, ainsi que les éventuels rythmes cérébraux. Cette représentation ne permet pas de distinguer les variations temporelles des signaux EEG, ni de localiser les sources cérébrales des signaux EEG. La représentation du signal EEG dans le domaine fréquentiel peut être obtenue par des méthodes de transformation, comme la transformée de Fourier, la transformée en ondelettes ou la transformée de Hilbert.

#### REPRÉSENTATION DU SIGNAL ÉLECTROENCÉPHALOGRAPHIQUE DANS LE DOMAINE TEMPS-FRÉQUENCE

La représentation du signal EEG dans le domaine temps-fréquence est la représentation qui consiste à analyser le signal EEG dans des fenêtres temporelles successives, et à appliquer une transformation fréquentielle à chaque fenêtre. Cette représentation permet d’observer les variations temporelles et fréquentielles des signaux EEG, ainsi que les éventuels événements transitoires ou non stationnaires. Cette représentation ne permet pas de localiser les sources cérébrales des signaux EEG, ni de réduire la dimensionnalité des signaux EEG. La représentation du signal EEG dans le domaine temps-fréquence peut être obtenue par des méthodes de spectrogramme, de scalogramme ou de bispectre.

#### REPRÉSENTATION DU SIGNAL ÉLECTROENCÉPHALOGRAPHIQUE DANS LE DOMAINE SPATIAL

La représentation du signal EEG dans le domaine spatial est la représentation qui consiste à analyser le signal EEG dans des régions spatiales définies, et à appliquer une transformation spatiale à chaque région. Cette représentation permet d’observer la distribution spatiale des signaux EEG, ainsi que les éventuels liens entre les régions cérébrales. Cette représentation permet également de réduire la dimensionnalité des signaux EEG, en éliminant les composantes redondantes ou non pertinentes. La représentation du signal EEG dans le domaine spatial peut être obtenue par des méthodes de cartographie, de localisation ou de séparation.

### APPLICATIONS DES INTERFACES CERVEAU-MACHINE

Les applications des interfaces cerveau-machine (ICM) sont les domaines d’utilisation des ICM, qui visent à améliorer la qualité de vie, la performance ou la santé des utilisateurs. Il existe plusieurs types d’applications des ICM, comme les applications de communication, les applications de contrôle, les applications de restauration, les applications de divertissement ou les applications de recherche. Les applications de communication sont les applications qui permettent à l’utilisateur d’envoyer des messages ou des choix à la machine, comme des lettres, des chiffres, des mots ou des icônes. Ces applications sont utiles pour les personnes souffrant de troubles de la communication, comme la paralysie, le lockie, le syndrome de l’enfermement ou l’aphasie. Les applications de contrôle sont les applications qui permettent à l’utilisateur de manipuler la machine, comme un curseur, un fauteuil roulant, un robot ou une prothèse. Ces applications sont utiles pour les personnes souffrant de troubles de la motricité, comme la tétraplégie, la paraplégie ou l’amputation. Les applications de restauration sont les applications qui permettent à l’utilisateur de récupérer une fonction perdue ou altérée, comme la vision, l’audition, la parole ou la motricité.

Ces applications sont utiles pour les personnes souffrant de troubles sensoriels, comme la cécité, la surdité, la dysarthrie ou la dyspraxie. Les applications de divertissement sont les applications qui permettent à l’utilisateur de se divertir, de se détendre ou de se former, comme des jeux, des musiques, des films ou des formations. Ces applications sont utiles pour les personnes souhaitant améliorer leur bien-être, leur plaisir ou leurs compétences. Les applications de recherche sont les applications qui permettent à l’utilisateur de participer à des études scientifiques, comme des expériences, des enquêtes, des tests ou des mesures. Ces applications sont utiles pour les chercheurs souhaitant collecter des données, des informations ou des connaissances.

## NOTION DE MACHINE LEARNING

La Machine Learning (ML) ou l’apprentissage machine est une branche de l’intelligence artificielle qui vise à créer des systèmes qui apprennent à partir de données, sans être programmés explicitement. L’apprentissage machine utilise des algorithmes qui se basent sur des modèles mathématiques ou statistiques pour représenter les données, et qui ajustent les paramètres de ces modèles en fonction des données observées. L’apprentissage machine permet aux ordinateurs d’apprendre par eux-mêmes et de réagir comme des humains dans la prise de décision ou la réalisation de prédictions, en améliorant leurs performances au fil du temps.

L’apprentissage machine nécessite un grand volume de données à fournir à l’ordinateur qui apprend. Ces données, appelées big data, doivent être analysées par des algorithmes d’apprentissage machine, qui cherchent à y trouver des motifs récurrents. En détectant ces motifs, les algorithmes apprennent et améliorent leurs performances dans l’exécution d’une tâche spécifique. Tout ce qui peut être stocké numériquement peut servir de données pour l’apprentissage machine. Par exemple, les signaux électriques collectés sur le cuir chevelu et convertis en données EEG numériques peuvent être traités par des algorithmes d’apprentissage machine.

Il existe quatre types principaux d’algorithmes d’apprentissage machine, selon le type d’apprentissage : supervisé, non supervisé, semi-supervisé ou par renforcement. L’apprentissage supervisé consiste à apprendre à partir de données étiquetées, c’est-à-dire associées à une classe ou à une valeur cible. L’objectif est de prédire la classe ou la valeur cible d’une nouvelle donnée, en utilisant un modèle appris à partir des données étiquetées. L’apprentissage non supervisé consiste à apprendre à partir de données non étiquetées, c’est-à-dire sans classe ou valeur cible. L’objectif est de découvrir la structure ou les caractéristiques des données, en utilisant un modèle qui regroupe ou qui réduit les données. L’apprentissage semi-supervisé consiste à combiner les deux types d’apprentissage précédents, en utilisant un ensemble limité de données étiquetées pour former les exigences de la classification, et en utilisant le modèle partiellement appris pour étiqueter le reste des données non étiquetées. L’apprentissage par renforcement consiste à apprendre par essai-erreur, en utilisant la technique d’exploration/exploitation, qui consiste à choisir une action, à observer les conséquences, et à choisir l’action suivante en tenant compte des récompenses ou des pénalités obtenues.

## GENERALITES SUR LA CLASSIFICATION

La classification est une tâche qui consiste à prédire la classe d'une donnée, parmi un ensemble fini de classes possibles. Il existe plusieurs types de classifieurs, qui peuvent être classés selon différents critères, comme le type de sélection de classe, la variation temporelle, la stabilité, la régularisation, etc. Par exemple, on peut distinguer les classifieurs génératifs, qui calculent la probabilité de chaque classe, des classifieurs discriminatifs, qui discriminent les classes directement. On peut aussi distinguer les classifieurs statiques, qui ne tiennent pas compte des informations temporelles, des classifieurs dynamiques, qui classifient une séquence de données. On peut encore distinguer les classifieurs stables, qui ont une faible complexité et une bonne robustesse, des classifieurs instables, qui ont une forte complexité et une mauvaise robustesse. On peut enfin distinguer les classifieurs régularisés, qui ont de bonnes performances de généralisation, des classifieurs non régularisés, qui ont de mauvaises performances face au bruit.

Il existe de nombreux algorithmes de classification, qui utilisent des principes et des méthodes différents pour apprendre un modèle à partir des données étiquetées, et pour prédire la classe d'une nouvelle donnée. Parmi les algorithmes de classification les plus utilisés dans le domaine des interfaces cerveau-machine (BCI), on peut citer :

- Les algorithmes linéaires, qui sont des algorithmes discriminatifs basés sur des fonctions linéaires pour la distinction des classes. Ils sont simples et populaires pour des applications de BCI. Par exemple, le Linear Discriminant Analysis (LDA) et la Machine à Vecteur de Support (SVM) sont des algorithmes linéaires courants pour des recherches sur les BCI.

- Les réseaux neuronaux, qui sont des ensembles de neurones artificiels qui permettent de réaliser un arbre de décision. Ils sont aussi très utilisés dans le domaine des BCI. Par exemple, le Multi Layer Perceptron (MLP) ou le classifieur Gaussien sont des réseaux neuronaux qui ont été testés avec succès pour des BCI asynchrones.

- Les algorithmes bayésiens non linéaires, qui sont des algorithmes génératifs qui produisent des frontières de décision non linéaires. Ils sont capables de rejeter les échantillons moins certains, mais ils sont moins utilisés que les algorithmes linéaires ou les réseaux neuronaux. Par exemple, le Modèle Caché de Markov ou le Réseau Bayésien Graphique sont des algorithmes bayésiens non linéaires.

- Les algorithmes des plus proches voisins, qui sont des algorithmes discriminatifs non linéaires et assez simples. Ils consistent à assigner une donnée à une classe en fonction de ses plus proches voisins, qui peuvent être d'autres données ou des prototypes de classe. Par exemple, le k Nearest Neighbors (kNN) ou la distance de Mahalanobis sont des algorithmes des plus proches voisins.

- Les algorithmes de combinaison de classifieurs, qui sont des algorithmes qui combinent plusieurs classifieurs pour réduire la variance et l'erreur de classification. Ils utilisent des techniques de renforcement ou de vote. Par exemple, le boosting ou le bagging sont des algorithmes de combinaison de classifieurs.  
  
- Le Linear Discriminant Analysis (LDA) est une technique d’apprentissage supervisé qui permet de classifier des données en fonction de leurs caractéristiques. Le LDA cherche à trouver une combinaison linéaire des caractéristiques qui maximise la distance entre les classes et minimise la variance à l’intérieur des classes. Le LDA suppose que les données suivent une distribution normale et que les covariances des classes sont égales. Le LDA utilise alors une règle simple pour attribuer une classe à une donnée, en fonction du produit scalaire entre un vecteur de poids et la donnée. Le vecteur de poids est calculé à partir des moyennes et des covariances des classes, et le seuil de décision est défini par la moyenne des moyennes des classes. Le LDA trace ainsi un hyperplan qui sépare les classes de manière optimale. Le LDA peut être étendu à des problèmes multi-classes en utilisant la stratégie du “One Versus the Rest”, qui consiste à séparer chaque classe du reste des classes.

Le LDA est un algorithme simple et efficace, qui peut être utilisé pour des applications d’interfaces cerveau-machine (BCI). Il a de faibles exigences en termes de ressources de traitement, et il donne généralement de bons résultats. Son principal inconvénient est qu’il est linéaire, et qu’il peut donc avoir de mauvaises performances sur des données complexes non linéaires.

L'objectif du LDA est de trouver, pour n'importe quel échantillon de la même distribution (pas nécessairement issu de l'ensemble d'entraînement), un bon prédicteur pour la classe y, étant donné une seule observation . Supposons que les fonctions de densité de probabilité conditionnelle et suivent toutes deux une distribution normale avec des paramètres de moyenne et de covariance (, Σ0) et (, Σ1), respectivement. Supposant en outre que les covariances de classe sont égales, c'est-à-dire, Σ0 = Σ1 = Σ, et qu'elles sont de rang complet, la constante de seuil c est donnée par la relation suivante :

*•* > *c*

Où = et *c* = •

Cela signifie que le critère pour qu'une entrée appartienne à une classe y est une fonction de cette combinaison linéaire d'observations connues. Géométriquement, cela signifie que l'observation appartient à y si le correspondant se trouve d'un côté d'un hyperplan perpendiculaire à , le lieu du plan étant défini par le seuil *c*.

## PERFORMANCE D’UN CLASSIFIEUR

La performance d’un classifieur est la capacité à prédire correctement la classe d’une donnée, parmi un ensemble fini de classes possibles, en fonction de ses caractéristiques. La performance d’un classifieur n’est jamais parfaite, et il faut donc définir un seuil de précision acceptable, qui dépend de la complexité des données à classifier. Pour les interfaces cerveau-machine (BCI) d’imagerie motrice, le seuil de précision minimal recommandé est de 65%.

Il existe plusieurs techniques pour évaluer la performance d’un classifieur, en utilisant un ensemble de données de test, qui sont différentes de celles utilisées pour apprendre le modèle de classification. Parmi ces techniques, on peut citer :

* L’exactitude, qui est le rapport entre le nombre de prédictions correctes et le nombre total de prédictions. Elle peut être trompeuse si les données sont déséquilibrées entre les classes, c’est-à-dire si une classe est plus fréquente que les autres.
* La matrice de confusion, qui est une matrice carrée qui compare les classes réelles et les classes prédites. Chaque ligne correspond à une classe réelle, et chaque colonne correspond à une classe prédite. La diagonale indique le pourcentage de prédictions correctes par classe. La Figure 40 ci-dessous montre la matrice de confusion de deux classes, appelées “Positive” et “Négative”.
* La précision, qui mesure la capacité du modèle à identifier correctement la classe positive. Elle permet de minimiser les faux positifs, c’est-à-dire les données qui sont prédites comme positives alors qu’elles sont négatives.
* La sensibilité, qui indique le taux d’observations positives qui ont été correctement prédites. Elle permet de minimiser les faux négatifs, c’est-à-dire les données qui sont prédites comme négatives alors qu’elles sont positives.
* La validation croisée, qui est une technique itérative qui divise l’ensemble de données en k segments de taille et de contenu aléatoires. À chaque itération, un segment est utilisé pour tester le modèle appris sur les k-1 segments restants, et l’erreur est calculée. À la fin des itérations, l’erreur finale est la moyenne des erreurs obtenues à chaque itération. La validation croisée permet d’avoir une idée de la performance du modèle sur des données inconnues. Il faut choisir avec soin le nombre k de segments, qui peut varier de 4 à 10.

## CLASSIFICATION DES DONNEES EEG

La classification des données électroencéphalographiques (EEG) est une tâche qui consiste à prédire la classe d’une donnée EEG, parmi un ensemble fini de classes possibles, en fonction de ses caractéristiques. Les caractéristiques sont des informations extraites du signal EEG traité, qui reflètent l’activité cérébrale du sujet. Pour classifier les données EEG, il faut utiliser un algorithme de classification, qui apprend un modèle à partir des données EEG étiquetées, et qui applique ce modèle aux nouvelles données EEG. La performance d’un algorithme de classification dépend de la qualité des données EEG utilisées pour l’entraînement, ainsi que de divers autres facteurs, comme le matériel d’acquisition, les capacités du sujet, les conditions climatiques, etc.

Les données EEG présentent des propriétés particulières, qui rendent la classification difficile. Parmi ces propriétés, on peut citer :

* Le bruit et les valeurs aberrantes, qui diminuent le rapport signal sur bruit (SNR) des signaux EEG, et qui peuvent fausser la classification.
* La dimensionnalité élevée, qui nécessite de réduire le nombre de caractéristiques à utiliser, en combinant intelligemment les caractéristiques de plusieurs canaux et segments de temps.
* Les informations temporelles, qui impliquent que les caractéristiques des BCI doivent tenir compte de la dynamique temporelle des motifs d’activité cérébrale.
* La non-stationnarité, qui signifie que les signaux EEG varient beaucoup dans le temps et entre les sessions, en fonction de divers facteurs, comme l’état psychologique et physiologique du sujet, le positionnement des capteurs, la qualité du contact, etc.
* Les petits ensembles de données pour l’entraînement, qui limitent la quantité de données disponibles pour apprendre le modèle de classification, et qui rendent le processus de collecte des données long et fastidieux pour les sujets.

La classification des données électroencéphalographiques (EEG) est une tâche qui consiste à prédire la classe d’une donnée EEG, en fonction de ses caractéristiques. Cette tâche présente deux problèmes majeurs : la malédiction de la dimensionnalité et le compromis biais-variance.

La malédiction de la dimensionnalité signifie que plus les caractéristiques sont nombreuses, plus il faut de données pour apprendre un modèle de classification efficace. Or, les données EEG ont souvent une dimensionnalité élevée et un ensemble d’apprentissage petit, ce qui rend la classification difficile.

Le compromis biais-variance signifie qu’il faut trouver un équilibre entre la complexité du modèle de classification et sa sensibilité aux données d’entraînement. Un modèle trop simple peut avoir un biais élevé, c’est-à-dire qu’il ne correspond pas à la meilleure correspondance possible. Un modèle trop complexe peut avoir une variance élevée, c’est-à-dire qu’il change beaucoup en fonction des données d’entraînement. Il faut donc utiliser des techniques de stabilisation, comme la régularisation ou la combinaison de classifieurs, pour réduire la variance.

## CHOIX DU CLASSIFIEUR POUR UNE ICM

Le choix d'un classifieur pour une Interface Cerveau-Machine (ICM) est crucial pour obtenir des résultats optimaux. L'exactitude de la classification, mesurée par le pourcentage de vecteurs correctement classifiés, est la métrique prédominante pour évaluer la performance d'un classifieur. Deux critères, proposés par Fabien Lotte et alii [57], guident le choix du classifieur optimal en fonction du timing d'implémentation de la BCI et des caractéristiques des données EEG.

Concernant le timing d'implémentation, les BCI synchrones, les plus courantes, trouvent leur efficacité avec des algorithmes tels que le SVM, les classifieurs dynamiques, et les combinaisons de classifieurs. Le SVM se distingue par sa régularisation, simplicité, et robustesse face à la dimensionnalité. Les classifieurs dynamiques captent les variations temporelles, tandis que la combinaison de classifieurs offre généralement de meilleurs résultats en réduisant la variance dans l'erreur de classification.

Pour les BCI asynchrones, moins fréquentes, les expériences sont limitées, mais les classifieurs dynamiques semblent moins performants que les statiques en raison de la difficulté à identifier le début de chaque tâche mentale.

En ce qui concerne les caractéristiques des données EEG, les classifieurs régularisés et discriminatifs s'avèrent plus performants en présence de bruit et de valeurs aberrantes. Le SVM est bien adapté aux données de haute dimensionnalité, les classifieurs dynamiques excellent dans l'exploitation de l'information temporelle, et la combinaison de classifieurs peut atténuer la non-stationnarité. Enfin, en cas de petits ensembles de données d'entraînement, des techniques simples comme le LDA avec peu de paramètres sont préférables.

## CONCLUSION

Ce chapitre a présenté les notions de base sur le cerveau humain, les interfaces cerveau-machine et l’électroencéphalographie. Il a permis de comprendre comment le cerveau humain produit et traite des signaux électriques, comment ces signaux peuvent être mesurés et interprétés par des dispositifs externes, et quelles sont les applications possibles des ICM basées sur l’EEG.

Nous avons également parlé des notions sur l’apprentissage machine et la classification, ainsi que les spécificités de la classification des données EEG. Il a permis de comprendre comment les algorithmes d’apprentissage machine peuvent apprendre à partir de données, comment ils peuvent classifier des données en différentes catégories, et quels sont les défis et les critères de choix d’un classifieur adapté aux données EEG.

# CHAPITRE 2 : CONCEPTION DU SYSTEME

## INTRODUCTION

Parler du système semi-autonome englobant à la fois le BCI, censé donner des commandes strictes à un robot, et la piCar-V, capable de se prendre seul en charge face à des situations étudiées dans ce travail, est une vision répondant à un cahier de charge qui nous a été confié après avoir vu les résultats de la totale direction menée seule par Cerebro.

Les objectifs de ce chapitre seront alors tirés directement du cahier de charge, tel est le socle qui nous mène à l’élaboration et à la réalisation de cette œuvre qualifiée de proof of concept pour l’ingénierie du futur.

## OBJECTIFS

Mettre en exergue d'une manière analytique et méthodique la capacité de la gestion BMI à faire une auto-analyse (debugging) des pannes/difficultés quand le robot est bloqué, à travers aussi un stress test.

Fiabilité (reliability) du contrôle BMI :  Quels sont les problèmes de connexion et action quand le robot se trouve à la limite de son rayon d'action.

Voulant répondre à l’implémentation de ces 2 paradigmes tout en se basant sur les moyens à notre disposition et qui pourraient matcher parfaitement avec le système BCI existant, nous nous sommes donnés des réponses concluant à la réalisation d’un système semi-autonome, qui selon nous, offrirait beaucoup d’élégance, de simplicité et d’ergonomie aux utilisateurs plutôt que de se baser à 1000% à l’utilisation unique du casque.

## CONCEPTION DU SYSTEME

Les règles de conception des systèmes automatiques se subdivisent en 2 grandes parties qui sont : la partie opérative et la partie commande.

Dans ce chapitre nous définissons la partie opérative comme étant l’ensemble des acteurs majeurs à la constitution de notre système. Ce sont des entités qui entre en contact les uns des autres afin de réaliser cette symbiose recherchée.

La partie commande désigne l’ensemble des éléments logiciels qui déterminent le comportement du système, les modèles, et les objets qui nous permettent de bien suivre la réalisation du système. Dans cette section, nous allons présenter la conception de notre système robot autonome avec BCI en détaillant ces deux parties.

### PARTIE OPERATIVE

La partie opérative de notre système se compose de deux sous-systèmes principaux : le BCI et le robot.

Le BCI est l’interface qui permet de traduire l’activité cérébrale de la personne qui conduit le robot en commandes pour le robot. Le robot est le système qui reçoit les commandes du BCI et qui les exécute en utilisant ses propres capacités. Les composantes de chaque sous-système sont présentées le prochain chapitre qui leur a été dédié. Le robot est l’élément central du système sur quoi porte la réflexion de ce travail orienté POC. Il est composé de plusieurs éléments comme la smart video car, la RPI 4 et les capteurs. De même que la partie BCI, ses constituants seront énumérés dans le chapitre sur les outils et méthodologies.

### PARTIE COMMANDE

La partie commande est celle qui apporte l’âme dans notre conception, elle décrit le fonctionnement intelligent du système et elle détermine en avance quels seraient les paramètres à prendre en compte quand on parlera des expérimentations et des résultats que nous souhaiterons présenter.

Nous voulons concevoir un système qui illumine les aspects suivants subdivisés en code :

**Premier code (Capteur ultrasonique et traitement d'image sur Raspberry Pi)**

- Entité : Raspberry Pi et capteur ultrasonique HC-SR04.

- Tâches : Mesure de distance, détection d'obstacles, et streaming vidéo. Utilise TensorFlow Lite pour le traitement d'image et la détection d'objets avec OpenCV.

- Apport en métriques :

a. Précision : Amélioration grâce à la détection d'obstacles et la classification d'objets en temps réel.

b. Réactivité : Évaluation basée sur le temps de réponse du système à un obstacle détecté.

c. Efficacité : Mesurée par la capacité à naviguer sans collisions en utilisant les données ultrasoniques et visuelles.

**Deuxième code (Serveur HTTP pour le contrôle du PiCar)**

- Entité : Serveur HTTP sur Raspberry Pi.

- Tâches : Réception des commandes de direction et de mouvement via HTTP, et contrôle du PiCar-V en conséquence.

- Apport en métriques :

a. Fiabilité : Basée sur la réussite des commandes envoyées au PiCar-V sans erreurs.

b. Latence : Temps de réponse du serveur à une commande HTTP.

c. Sécurité : Implique la robustesse du serveur à gérer les demandes sans crasher.

**Troisième code (Interface Cerveau-Ordinateur pour le contrôle du PiCar)**

- Entité : Interface Cerveau-Ordinateur (BCI) utilisant OpenViBE.

- Tâches : Traduction des signaux EEG en commandes de contrôle pour le PiCar-V via le serveur HTTP.

- Apport en métriques :

a. Précision : Exactitude de la classification des signaux EEG en commandes spécifiques.

b. Rapidité : Délai entre la pensée de l'utilisateur et la réaction du PiCar-V.

c. Expérience utilisateur : Mesurée par la facilité d'utilisation et la satisfaction de l'utilisateur en contrôlant le PiCar-V avec le mental.

Ces résumés montrent comment chaque code contribue à l'ensemble du projet en termes de métriques spécifiques, en indiquant clairement les responsabilités des différentes entités (capteurs, serveur, et interface BCI) et leur impact sur la performance globale du système.

### DIAGRAMMES

Pour illustrer la conception de notre système robot autonome avec BCI, nous allons présenter deux types de diagrammes : un diagramme de blocs fonctionnels, qui montre les relations entre les différents composants du système, les flux de données et de signaux, et les fonctions réalisées par chaque bloc ; et un diagramme de séquence, qui montre les interactions entre les différents composants du système, les événements et les messages, et le déroulement temporel du système. Ces diagrammes sont basés sur les éléments de conception que nous avons décrits précédemment.

#### Le diagramme de blocs fonctionnels

Le diagramme de blocs fonctionnels de notre système est le suivant :

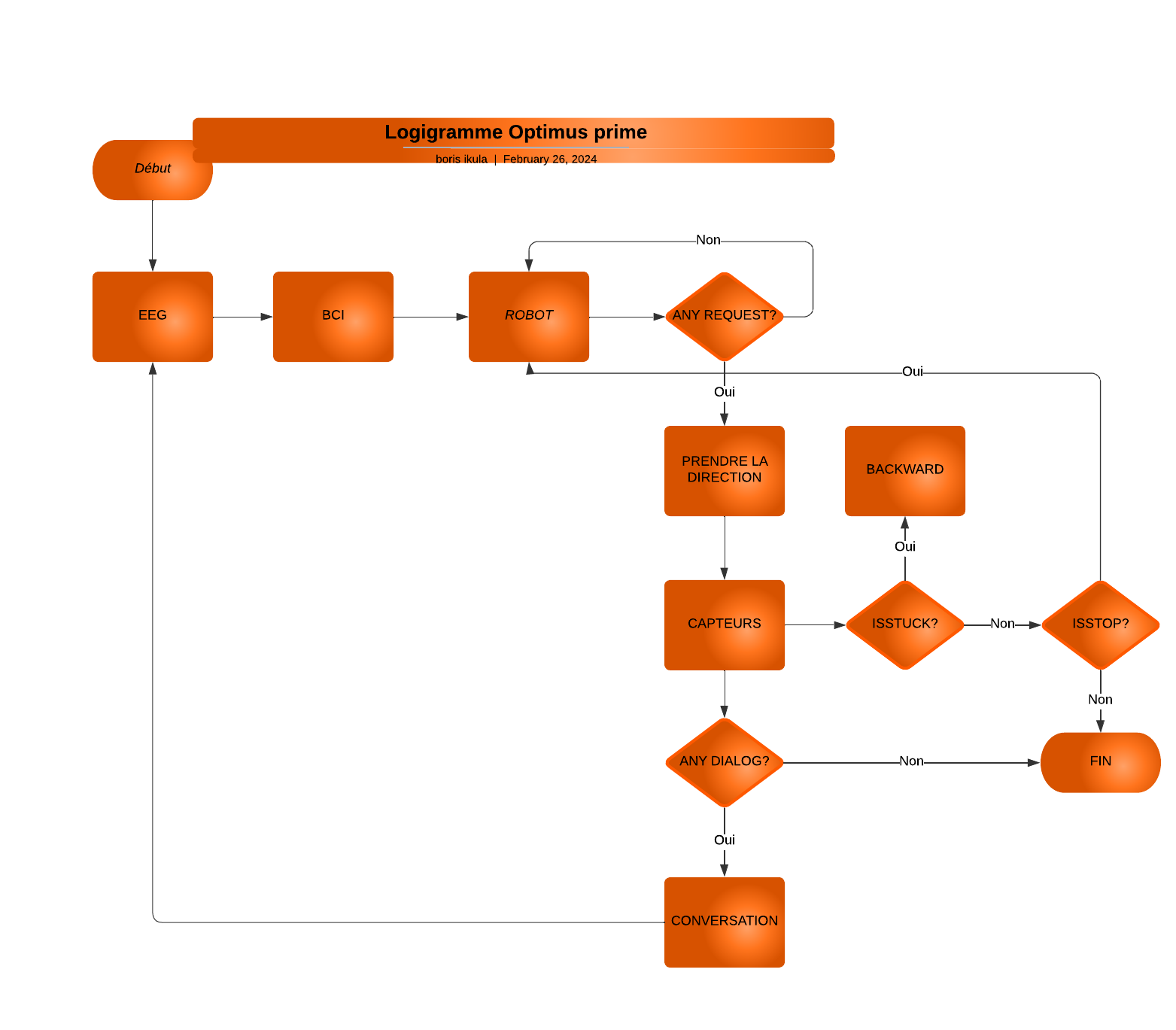


Figure 2. Logigramme Optimus prime

Ce diagramme montre que le BCI reçoit les signaux EEG de la personne qui conduit le robot, et les transforme en commandes pour le robot. Le robot reçoit les commandes du BCI, et les exécute en utilisant les technologies suivantes :

* Caméra : Permet de capturer les images de l’environnement et de les transmettre à l’écran de la personne qui conduit le robot, tout en estimant la taille de l’objet détecté.
* Capteur ultrason : Permet de mesurer la distance aux obstacles et de modifier la trajectoire du robot en cas de besoin.
* Microphone : Permet de capter les sons de l’environnement et de les transmettre à la personne pilotant le robot à travers le haut-parleur de l’ordinateur, et cette dernière utilisera le microphone de la machine pour envoyer le son au haut-parleur du robot.

#### Le diagramme de séquence

Ce diagramme illustre les interactions de chaque constituant de notre système dans le temps, et esquisse le déroulement des événements à partir de l’entrée, qui est confondue aux impulsions EEG reçues de l’utilisateur, jusqu’à la sortie, ici l’environnement ou le milieu.

Le BCI reçoit ainsi la capture des signaux, la transforme en commandes pour le serveur du robot qui va ensuite transmettre au robot à travers ses moteurs et ses capteurs (l’entrée de la caméra, l’ultrason et le micro). Les capteurs communiquent avec l’environnement pour détecter, dialoguer et prendre des décisions.

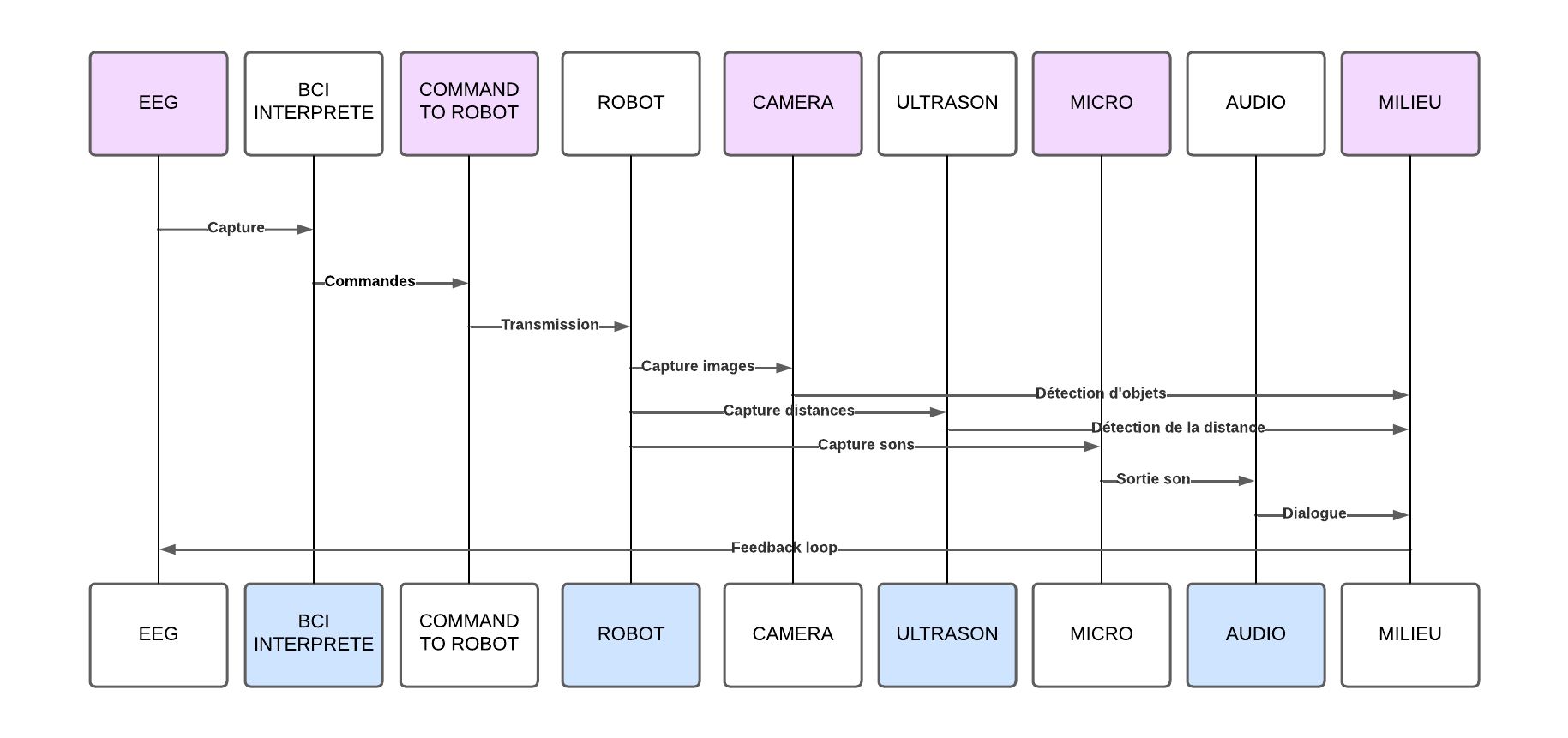


Figure 3. Diagramme de séquence du système

## CONCLUSION

Dans ce chapitre, nous avons présenté la conception de notre système robotique intelligent contrôlé par un BCI basé sur l’expression faciale. Nous avons défini les objectifs de notre travail, qui sont de mettre en évidence la capacité de la gestion BCI à faire une auto-analyse des pannes ou difficultés quand le robot est bloqué, et d’évaluer la fiabilité du contrôle BCI quand le robot se trouve à la limite de son rayon d’action.

Nous avons ensuite décrit les deux parties principales de notre système : la partie opérative, qui comprend le BCI et le robot, et la partie commande, qui comprend les codes qui déterminent le comportement du système et les métriques qui permettent de mesurer sa performance. Nous avons ainsi montré comment nous avons conçu un système semi-autonome, qui offre une solution élégante, simple et ergonomique aux utilisateurs, en combinant les commandes du BCI et les capacités du robot.

# CHAPITRE 3 : OUTILS ET METHODOLOGIES

## INTRODUCTION

Après avoir conçu notre système, nous nous basons sur les outils à prendre en compte pour nous permettre de nous conformer à la réalisation des objectifs arrêtés pour la réalisation de notre cahier de charge.

Ce chapitre parle de tous les constituants de notre système avec ses 2 parties majeures, la partie BCI et la partie voiture robot, et puis des méthodologies utilisées pour avoir un système robuste. Le chapitre contiendra donc deux sections qui, en intégrant les parties du système, détaillera tout ce qui est lié à l’aspect hardware et software du système conçu.

## OUTILS

Sachant que nous énumérons chaque élément du système, nous allons parler de prime à bord du hardware pour chuter au software.

### HARDWARE

#### EMOTIV EPOC+

EMOTIV EPOC+ L'EMOTIV EPOC+ est un dispositif de Brain-Computer Interface (BCI) avancé développé par la société EMOTIV. Il s'agit d'un appareil portable non invasif conçu pour mesurer l'activité électrique du cerveau (électroencéphalographie ou EEG) en temps réel. Il est équipé de 14 électrodes disposées selon le système international 10-20, qui couvrent les zones corticales impliquées dans la réalisation de tâches motrices imaginées. Ces électrodes captent les signaux électriques du cerveau et permettent de détecter les différentes activités cérébrales.

L'appareil est conçu pour être confortable à porter, avec un design ergonomique qui s'adapte à la forme de la tête de l'utilisateur. Cela permet une utilisation prolongée sans inconfort. Il utilise la technologie sans fil pour se connecter à un ordinateur ou à d'autres périphériques compatibles. Cela offre une liberté de mouvement à l'utilisateur et permet une utilisation pratique dans divers environnements.

EMOTIV fournit une gamme d'applications logicielles qui permettent de traiter et d'analyser les données EEG capturées par l'EMOTIV EPOC+.

Ces applications permettent notamment de visualiser les signaux cérébraux, de détecter des modèles d'activité et de contrôler des appareils externes en utilisant la pensée.



Figure 4. Casque Emoiv Epoc+

#### SUNFOUNDER SMART VIDEO CAR

C’est un kit de voiture robotique qui peut être contrôlé à distance par un Raspberry Pi. Il dispose d’une caméra, de quatre roues motrices, d’un capteur à ultrasons et d’un servomoteur.

[Il peut réaliser des fonctions telles que l’évitement d’obstacles, le streaming vidéo en temps réel et la reconnaissance d’objets](https://www.maxicours.com/se/cours/fonctionnement-des-composants-d-un-ensemble/). Raison pour laquelle nous l’avons reconduit dans notre travail comme robot principal où s’assoit toute la structure réalisée.

Sans trop détaillé les éléments constituant notre robot, comme l’a déjà fait notre ami Pierre dans son travail de mémoire, l’image suivante montre l’entière composition du piCar-V.



Figure 5. Equipement SunFounder Picar-V

#### RASPBERRY PI 4 MODEL B

Le RPI 4, comme nous aimons bien l’appeler, est un mini-ordinateur de la taille d’une carte de crédit qui peut exécuter divers systèmes d’exploitation. Il dispose d’un processeur quadricœur, de 2 Go de mémoire vive, de quatre ports USB (2 USB en noir : USB 2.0 qui permettent un transfert de données limité à 480 Mbps. 2 USB en bleu : USB 3.1 qui permettent de transférer les données à une vitesse de 10 Gbps, essentiels pour les données en temps réel), d’un port Ethernet, de 2 ports micro HDMI, d’un port audio, d’un lecteur de carte microSD et d’une connectivité sans fil. Il p ut être utilisé comme un ordinateur de bureau, un serveur, un centre multimédia ou une plateforme de développement.



Figure 6. Raspberry Pi 4

#### ACCESSOIRES

Ici nous avons voulu spécifier quelques éléments, quoique non cités dans les composants majeurs du système, qui jouent un rôle crucial dans le fonctionnement du système.

Ces éléments qui complètent le hardware, comprennent les batteries, les chargeurs, les câbles, l’adaptateur et les supports. Les batteries fournissent l’énergie nécessaire au fonctionnement du hardware. Les câbles permettent de connecter les différents éléments du hardware. Les adaptateurs permettent de convertir les signaux ou les tensions. Les supports permettent de fixer ou de positionner le hardware.

#### WEBCAM EXTERNE

Externe car ne faisant pas partie de l’arsenal prévu pour le picar-V, nous avons porté le choix sur un dispositif qui nous permettra d’avoir un micro intégré dans la caméra afin de bien jouer le rôle de webcam dans notre application sur l’IP call entre le robot et l’utilisateur. Le modèle utilisé est presqu’identique à celui présenté à la figure et à prix beaucoup plus bas.

C’est une caméra externe qui peut être connectée au Raspberry Pi via un port USB. Elle permet de capturer des images et des vidéos en haute définition. Elle dispose d’un microphone intégré, d’un autofocus, d’un zoom numérique et d’une correction automatique de l’éclairage, et l’annulaire s’allume au contact.



Figure 7. Webcam

#### BAFFLE POUR SORTIE AUDIO RPI 4

C’est un haut-parleur qui peut être connecté au Raspberry Pi via un port audio. Il permet de diffuser du son en stéréo. Il dispose d’un contrôle du volume, d’une prise casque et d’une entrée auxiliaire.



Figure 8. Baffle

#### HC-SR04

C’est un capteur à ultrasons qui permet de mesurer la distance entre le capteur et un objet. Il émet un signal sonore à haute fréquence et mesure le temps que met l’écho à revenir. Il peut détecter des objets situés entre 2 cm et 4 m.

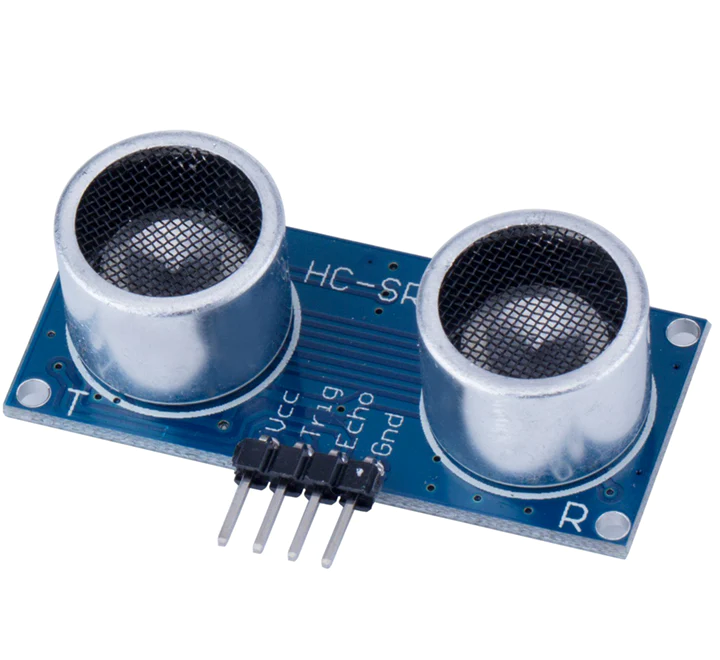


Figure 9. Ultrason

#### ESTIMATION COUT

Nous rappelons que la plupart des composants physiques utilisés dans notre système sont identiques à ceux utilisés dans le travail intitulé « Conception d’une interface cerveau-machine temps-réel asynchrone pour le pilotage à distance des systèmes robotiques à l’aide du Casque Emotiv EPOC+ ».

D’où le coût n’a pas beaucoup bougé dans ce tableau :

|  |  |  |  |
| --- | --- | --- | --- |
| Matériel | Quantité | Prix unitaire ($) | Prix total ($) |
| Emotiv EPOC+ | 1 | 699 | 699 |
| SunFounder Smart Video Car | 1 | 100 | 100 |
| Raspberry Pi 4B | 1 | 45 | 45 |
| Batterie 18650 8000 mAh 4.2V | 4 | 2 | 8 |
| Chargeur pour batterie 18650 | 1 | 4 | 4 |
| Carte microSD 32G + adaptateur | 1 | 4 | 4 |
| Ultrasonic HC-SR04 | 1 | 3 | 3 |
| Webcam externe | 1 | 10 | 10 |
| Baffle | 1 | 5 | 5 |
| Total | 12 |  | 878 |

Tableau 1 : Coût du projet

### 3.2.2. SOFTWARE

#### EMOTIV APPS

Ce sont des applications qui permettent de configurer, de visualiser et d’analyser les données du casque EMOTIV EPOC+. Elles comprennent EMOTIV App, EMOTIV Xavier, EMOTIV Insight et EMOTIV BCI. Elles sont écrites en C#, Java, Python et JavaScript.

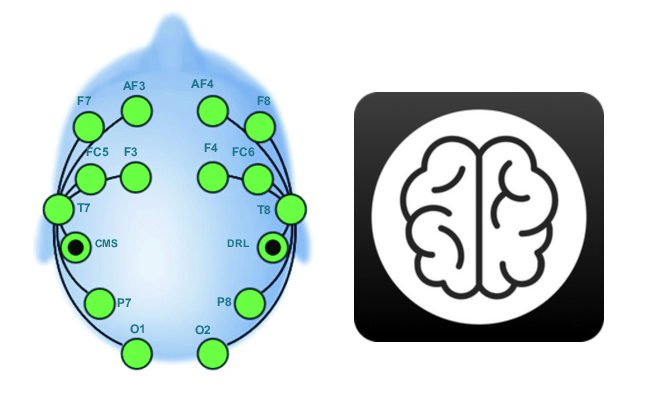


Figure 10. Vue des électrodes dans Emotiv Launcher

Emotiv Launcher est l’application que nous utilisons au switch-on du casque, elle nous permet de vérifier l’état du casque sur la tête de l’utilisateur, savoir si le contact est bon et si le pourcentage des signaux EEG lus est acceptable.

#### OPENVIBE

C’est une plateforme logicielle qui permet de concevoir, de tester et d’utiliser des interfaces cerveau-ordinateur (BCI). Elle permet de traiter les signaux EEG, de les classifier, suivant un épochage, de les filtrer et de les convertir en commandes. Elle est écrite en C++ et utilise des bibliothèques comme Qt, OpenGL et OpenAL.



Figure 11. OpenVibe

#### CYKIT

C’est un outil qui permet de connecter le casque EMOTIV EPOC+ au Raspberry Pi via un port USB. Il permet de recevoir les données brutes du casque et de les transmettre à OPENVIBE via un protocole TCP/IP. Il est écrit en Python et utilise des bibliothèques comme PyUSB, PyCrypto et Socket.

#### INTERPRÉTEUR PYTHON

C’est un programme qui permet d’exécuter des scripts écrits en Python. Il permet de réaliser des tâches de programmation de haut niveau, comme le calcul, la manipulation de données, le contrôle du hardware, etc. Il est écrit en C et utilise des bibliothèques comme NumPy, SciPy, Matplotlib, etc.

#### LIBRAIRIE SMART VIDEO CAR KIT

C’est une bibliothèque qui permet de contrôler le kit de voiture robotique SunFounder Smart Video Car. Elle permet de gérer les mouvements des roues, la rotation du servomoteur, la capture de la caméra, etc. Elle est écrite en Python et utilise des bibliothèques comme RPi.GPIO, Picamera, etc.

1. OPENCV

C’est une bibliothèque qui permet de réaliser des opérations de vision par ordinateur. Elle permet de traiter les images et les vidéos, de détecter des objets, des visages, des gestes, etc. Elle est écrite en C++ et utilise des bibliothèques comme Numpy, Scipy, Matplotlib, etc.

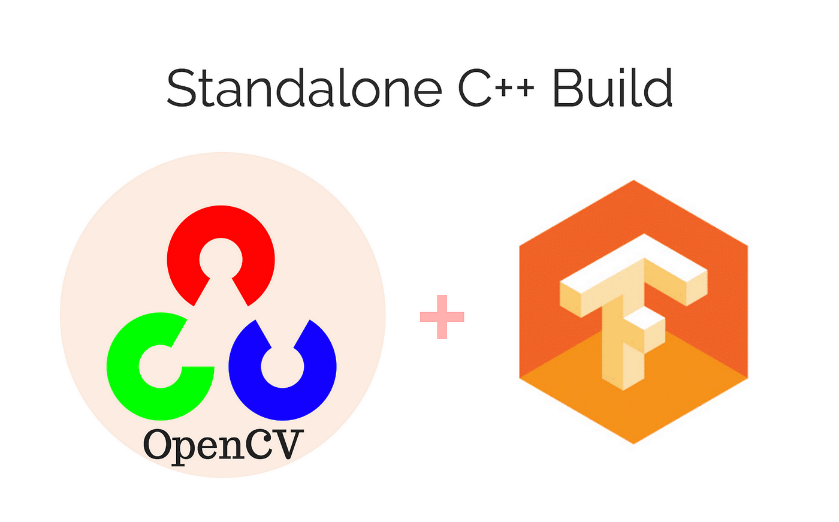


Figure 12. Trio C++, OpenCV et TensorflowLite

#### TENSORFLOW LITE

C’est un framework qui permet de déployer des modèles d’apprentissage automatique sur des appareils à faible puissance, comme le Raspberry Pi. Il permet de réaliser des tâches de reconnaissance d’images, de classification, de détection, etc. Il est écrit en C++ et utilise des bibliothèques comme Eigen, Gemmlowp, etc.

#### MOBILENETV1

C’est un modèle d’apprentissage automatique qui permet de réaliser de la reconnaissance d’objets sur des images. Il utilise une architecture de réseau de neurones convolutifs optimisée pour les appareils mobiles. Il peut reconnaître plus de 1000 classes d’objets différents. Il est écrit en Python et utilise des bibliothèques comme TensorFlow, Keras, etc.

#### C++

C’est un langage de programmation qui permet de réaliser des applications de bas niveau, comme le contrôle du hardware, le traitement des signaux, l’optimisation des performances, etc. Il utilise des bibliothèques comme STL, Boost, Qt, etc. Son utilisation dans ce projet nous permet d’augmenter les FPS du streaming de la vidéo OpenCV utilisant MobileNetV1 à l’aide de TFLite[[7]](#footnote-7)

1. SHELL

C’est un langage de script qui permet de réaliser des tâches de gestion du système, comme le lancement de programmes, la manipulation de fichiers, la communication entre processus, etc. Il utilise des commandes comme ls, cd, ps, etc.

1. WIRINGPI

C’est une bibliothèque écrite en C conçue pour faciliter l'accès aux GPIO (General Purpose Input/Output) du Raspberry Pi. Elle permet aux développeurs d'écrire des scripts et des programmes en C (et autres langages compatibles) qui interagissent directement avec les broches d'entrée et de sortie du Raspberry Pi, facilitant le contrôle de composants hardware tels que des LEDs, des moteurs, des capteurs, etc. WiringPi imite la fonctionnalité et la syntaxe de la bibliothèque Arduino Wiring, rendant la transition des projets Arduino vers Raspberry Pi plus intuitive pour les développeurs.

1. PUTTY

PuTTY est un émulateur de terminal open-source et gratuit, permettant une connexion sécurisée via SSH (Secure Shell) à des systèmes distants, tels que le Raspberry Pi. Il supporte également les connexions via Telnet, rlogin et les connexions série. Cet outil est crucial pour la configuration initiale, le débogage et la maintenance à distance du Raspberry Pi, offrant aux développeurs un moyen efficace de gérer leur système sans nécessiter un accès physique direct.

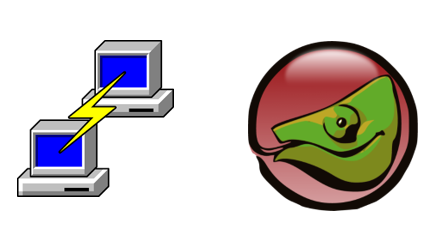


Figure 13. Putty et K-meleon Browser

1. K-MELEON

K-Meleon est un navigateur Web léger et personnalisable pour Windows, basé sur le moteur de rendu Gecko développé par Mozilla. Bien qu'il ne soit pas directement utilisé sur le Raspberry Pi, il peut être utilisé sur des systèmes de contrôle ou de monitoring fonctionnant sous Windows, pour accéder à des interfaces web du système, telles que des dashboards de contrôle ou des interfaces de programmation pour le système BCI et la voiture robot. Sa légèreté en fait un choix idéal pour des systèmes aux ressources limitées, où l'efficacité et la rapidité sont essentielles.

Le choix est penché sur K-meleon comme browser sur ce projet à cause de sa légèreté, sa faible utilisation de la RAM et sa faible consommation des ressources du processeur.

### INVENTAIRE ENERGETIQUE DES COMPOSANTS PHYSIQUES

L'évaluation de la consommation énergétique de chaque composant matériel est essentielle pour garantir l'autonomie et l'efficacité du système dans son ensemble. Cette section présente une analyse de la consommation énergétique des principaux composants matériels utilisés dans notre système.

A. EMOTIV EPOC+

* Consommation Énergétique: Environ 1,4 W
* Source d'Alimentation: Batterie rechargeable intégrée
* Autonomie: Jusqu'à 12 heures en utilisation continue

L'EMOTIV EPOC+ est optimisé pour une consommation énergétique faible, ce qui est essentiel pour les sessions prolongées d'utilisation. Sa batterie rechargeable garantit une utilisation continue sans nécessiter de changements fréquents ou de recharges, ce qui est crucial pour des applications mobiles ou à distance.

B. Raspberry Pi 4 Model B

* Consommation Énergétique: Environ 6 W (peut varier selon l'utilisation)
* Source d'Alimentation : Adaptateur secteur de 5 V et 3 A
* Autonomie : Non applicable (alimenté par une source d'énergie externe)

Bien que le Raspberry Pi 4 soit plus puissant que ses prédécesseurs, il maintient une consommation énergétique relativement basse. Cela permet son utilisation dans des projets nécessitant une certaine puissance de calcul sans un impact significatif sur la consommation énergétique totale du système.

C.SUNFOUNDER Smart Video Car

* Consommation Énergétique : Variable selon les opérations
* Source d'Alimentation: Batteries 18650
* Autonomie: 2-3 heures selon l'utilisation

La voiture robot SunFounder est conçue pour une utilisation modérée de l'énergie, avec une autonomie permettant des sessions d'utilisation suffisantes pour la plupart des applications éducatives et de loisirs. Les batteries 18650 offrent un compromis intéressant entre capacité, rechargeabilité et disponibilité.

D. Autres Composants

* Webcam Externe, HC-SR04, Baffle: Ces composants consomment relativement peu d'énergie (moins de 3 W chacun) et sont alimentés directement par le Raspberry Pi ou via une source externe de 5 V. Leur impact sur la consommation énergétique totale du système est minimal mais doit être pris en compte dans le cadre global.

## METHODOLOGIES

Dans ce présent travail, les méthodologies utilisées sont : la recherche scientifique et l’expérimentation au laboratoire.

Concernant la recherche scientifique, nous avons décidé d’effectuer des recherches approfondies sur l’état de l’art des BCIs et la semi-autonomie dans différents articles scientifiques sur la toile en mettant en exergue les points positifs et négatifs de chacun d’entre eux, dans le but de ressortir un travail aguerri basé sur les échecs et réussites de nos prédécesseurs dans l’art. Le mémoire de Pierre Sedi est la pièce maîtresse de cette œuvre de concept car nous nous sommes inspirés des paradigmes utilisés sous les expérimentations qui nous ont guidées.

Concernant la méthode expérimentale, nous y sommes allés par essai-erreur surtout dans l’obtention du comportement voulu tel que décrit dans le cahier de charge. Quoique pas certain, nous nous sommes dépassés avec plus de volonté, se mettant dans la peau des chercheurs scientifiques.

Nous avons ainsi conçu et réalisé un système robot autonome avec BCI, qui permet de piloter à distance un véhicule, le PiCar-V, en utilisant les signaux EEG captés par un casque, le EMOTIV EPOC+. Notre système intègre également des fonctionnalités de navigation intelligente, de réactivité aux conditions changeantes, et de communication interactive, grâce à l’utilisation de technologies telles que la caméra, le capteur ultrason, le microphone, le haut-parleur, le modèle MobileNetV1, et le contrôleur neuronal.

Pour évaluer la performance et la fiabilité de notre système, nous avons mené plusieurs tests et expériences, en utilisant des scénarios variés. Nous avons ainsi pu mesurer l’impact de différents facteurs, tels que la distance, la connexion, l’environnement, et les obstacles, sur la qualité du pilotage et la satisfaction de l’utilisateur.

## CONCLUSION

Le chapitre 3, dédié aux outils et aux méthodologies, constitue une étape cruciale dans la matérialisation de notre projet de système robot autonome semi-autonome. En explorant en détail les composants matériels, logiciels et les approches méthodologiques, nous avons tracé les fondations nécessaires à l'accomplissement des objectifs définis dans notre cahier des charges.

L'utilisation judicieuse d'outils matériels tels que l'Emotiv EPOC+, la voiture robotique SunFounder, le Raspberry Pi 4 Model B, et divers accessoires, a permis de construire une infrastructure robuste pour notre système. L'inventaire énergétique souligne l'efficacité énergétique de chaque composant, garantissant une autonomie optimale. Le choix de la webcam externe, du baffle pour la sortie audio, et du capteur HC-SR04 démontre une attention minutieuse aux détails, enrichissant les fonctionnalités du système tout en minimisant l'impact énergétique. L'écosystème logiciel, allant des applications Emotiv aux bibliothèques OpenVibe, Cykit, OpenCV, TensorFlow Lite, jusqu'aux langages de programmation C++, Python, et Shell, offre une palette complète pour la mise en œuvre cohérente du système.

# CHAPITRE 4. EXPERIMENTATIONS ET RESULTATS

## CRITERIUM

Ce chapitre fait une narration de tout ce qui constitue les résultats sur le pilotage de notre voiture robot que l’on souhaite rendre autonome dans des situations spécifiées au sein du chapitre de la conception du système.

Dans notre cas, s’agissant d’un *proof of concept*, nous avons pu implémenter des codes qui nous ont permis de tirer quelques métriques que nous avons jugées d’essentielles après avoir détaillé la partie commande du chapitre 2.

Dans notre travail, nous avions tenu à ajouter un quatrième code qui implique la télé présence de l’utilisateur du BCI en termes de communication avec les autres humains. Déclassé car celui-ci est qualifié de *hard ressource* pour notre Raspberry Pi qui se doit de rester fluide pour le bon fonctionnement de nos applications. En annexe nous trouverons les quatre codes et le protocole expérimental choisi pour présenter nos résultats.

Lors de la présentation des résultats, nous allons nous pencher sur les métriques enregistrées à l’intérieur du script principal, qui est un serveur d’écoute, comme du fichier log prélevant les événements dans le temps. En dehors du temps, y a aussi la distance qui est important pour affirmer si ce qu’on a fixé dans le code se passe comme prévu.

## EXPERIMENTATION

Ils étaient au nombre de trois (3), les sujets qui nous ont permis de mener ces expériences sur le comportement objectivé et souhaité de notre robot. Des humains en parfaite santé mentale et physique. Chaque sujet a eu droit au protocole expérimental complet prédéfini, le temps étant imparti pour tout un chacun.

### ENTRAINEMENT DU SUJET

D’entrer de jeu, l’expérience commence par le mode opératoire d’apprentissage élaboré dans le travail de P. Sedi sur la classification des expressions faciales, paradigme inspiré et maintenu pour le pilotage à distance de notre système semi-autonome, qui se résume en ceci :

* Utilisation de CyKit et OpenViBE : Configuration d'OpenViBE pour l'acquisition, le filtrage, et le traitement des signaux EEG capturés par le casque EPOC+.
* Filtrage Spatial et Temporel : Application de filtres pour améliorer la qualité des signaux EEG et extraction des caractéristiques pertinentes.
* Entraînement du Classifieur : Utilisation d'un classifieur pour interpréter les signaux EEG et déterminer les intentions de mouvement.

Nous signalons que nous avons ajouté une quatrième classe afin de contrôler l’avancement du robot au lieu de le laisser être automatique dès le début du programme.

Les expressions faciales retenues sont :

Visage détendu (garder son calme sans rien faire) pas de direction à prendre, sourcils levés pour la direction gauche, mâchoires serrées pour la direction droite et large sourire en faisant montrer les dents pour le devant.

Cette étape est illustrée ci-dessous montrant ainsi l’apparition de la classe ajoutée par rapport au mode opératoire dont nous nous basons. Après une quinzaine de minutes chaque sujet était capable de voir comment il pouvait être en mesure de piloter le robot par suite d’un moment d’exercice lui demandant de refaire les expressions faciales suivant les directions apprises en mode online.

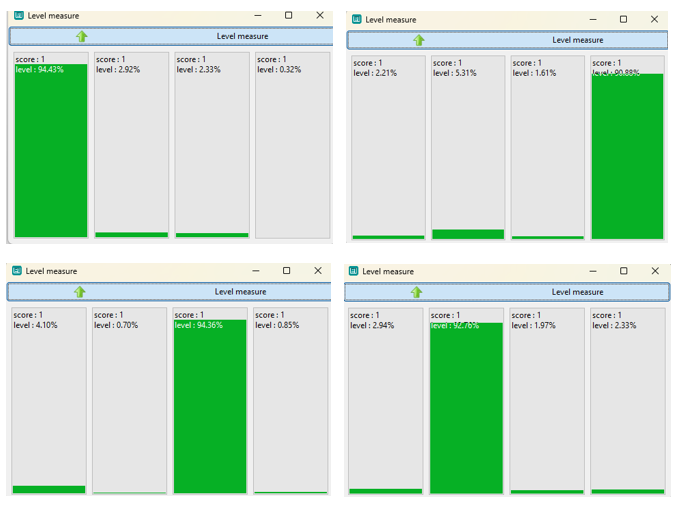


Figure 14. OpenVibe: Les 4 classes en mode online

Le script Python est appelé par OpenVibe pour permettre l’exécution, dans différents threads, de chaque classe lorsqu’elle est identifiée individuellement par l’EEG émanant du casque.

Une image contenant diagramme, texte, ligne

Description générée automatiquement

Figure 15. Design de l'arborescence des classes

### PILOTAGE DU ROBOT AVEC DES CAPACITES D’AUTONOMIE

Cette section est le cœur de notre travail qui a consisté à établir un système semi-autonome, en prenant ensemble un utilisateur du BCI et un robot en réel capacité de prendre des décisions sur des actions à opérer après que ce dernier ait reçu un ordre d’orientation.

Toujours avec nos sujets, en spécifiant que la dernière étape de l’apprentissage (en mode online comme précisé dans le mémoire de Sedi) sans se connecter au serveur qui se trouve d’ores et déjà placé sur le robot, était un aperçu du pilotage qui se fera directement sur celui-ci, nous allons connecter l’ICM à la partie restante du système, « Optimus Prime » notre robot.

Nous allons lancer le serveur principal sur le port 8000 du Raspberry Pi qui va également, comme écrit dans le code python du serveur *listen.py*, lancer le serveur SunFounder Picar-V sur le port 8001 à l’aide de l’outil Putty pour Windows.

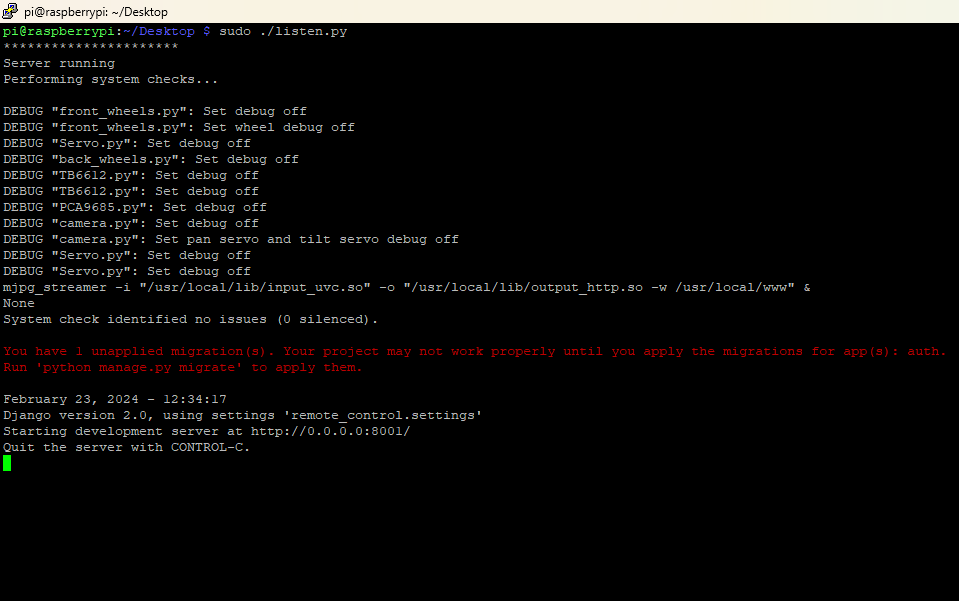


Figure 16. Lancement du serveur principal listen.py

Une fois lancé, le serveur est prêt à écouter les requêtes http en provenance de l’ICM via OpenVibe qui ouvrira son interface de colonne de niveau vu précédemment et qui permet une prise en main du pilotage à distance.

Le script de prise de décision du robot sera lancé ensuite pour enclencher les capteurs utilisés par Optimus Prime. Nous dupliquons la session Putty en cours pour exécuter le fichier compilé *CamUltrasonic*.

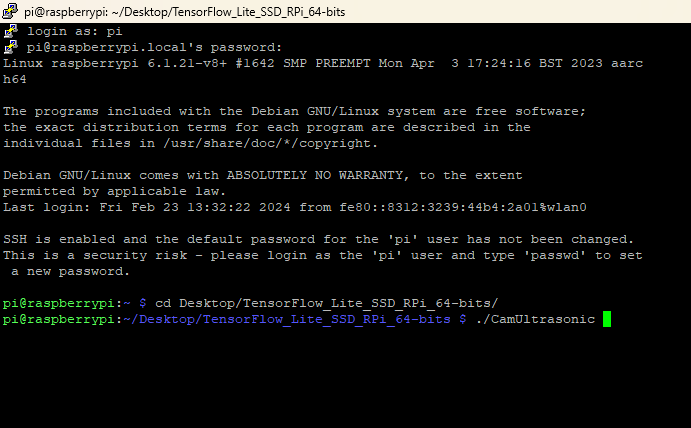


Figure 17. CamUltrasonic prêt à être exécuté

Ainsi, en temps réel nous obtenons les mesures de distances et de tailles des objets sur le robot directement. Afin de faciliter le pilotage du robot par l’utilisateur, nous avons créé une page html de visualisation en streaming des mesures prises par les capteurs sur le robot.

Ces données sont présentées dans un navigateur léger, K-Meleon que nous avons pris soin de choisir compte tenu de son workspace dans un processeur.

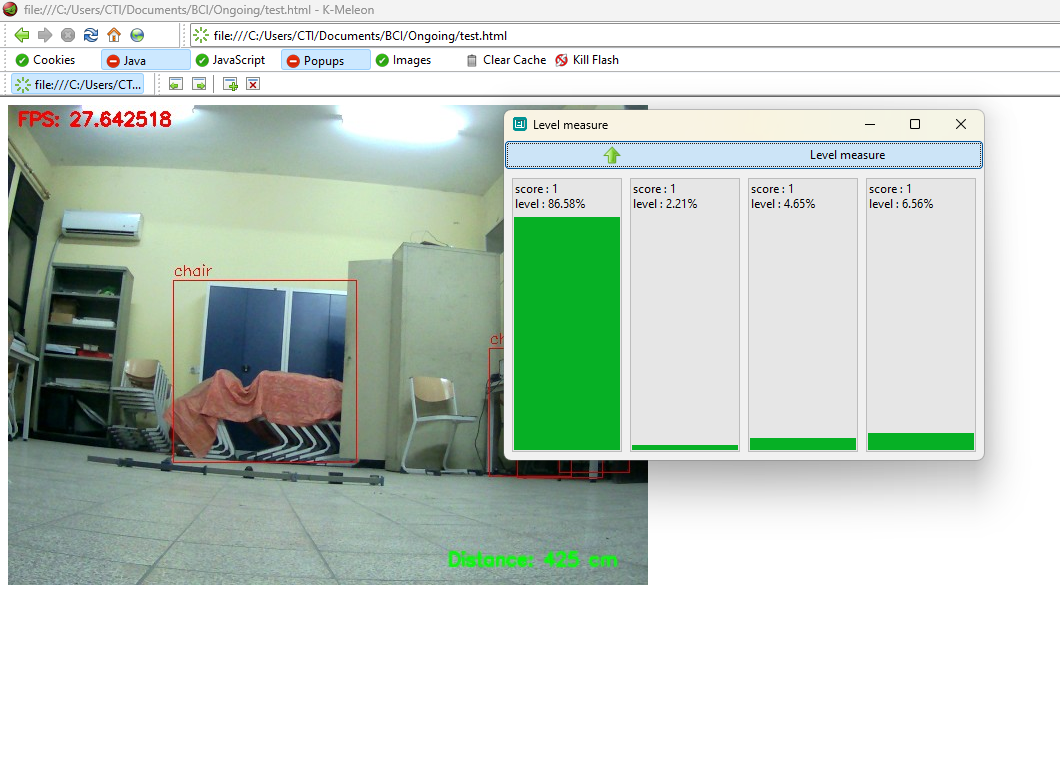


Figure 18. Streaming du webcam sur K-meleon browser

Le sujet n’a plus qu’à respecter le protocole du scénario tracé et observer le comportement du robot piloter dans les situations d’objet trop proche, d’embourbement accidentellement créé, et de perte de connexion.

## RESULTATS

Sur bases des expériences réalisées sur nos 3 sujets et sur les métriques que nous avons énumérées en début du chapitre, nous présentons les résultats suivants sur un tableau :

|  |  |  |  |
| --- | --- | --- | --- |
| Métrique | Sujet 1 | Sujet 2 | Sujet 3 |
| Entrée/sortie du mode débogage "unstuck" (en s) | 11:26:22 à 11:26:32 (10s) | 12:35:20 à 12:35:30 (10s) | 13:27:56 à 13:28:06 (10 s) |
| Entrée/sortie du mode débogage "conn" (en s) | 11:28:48 à 11:28:55 (7s) |  | 13:30:19 à 13:30:25 (6s) |
| Temps de réaction à l'erreur de connexion (en s) | 6 | 5 | 6 |
| Distance d'arrêt pour obstacle (< 39 cm) | Oui | Oui | Oui |
| Avertissement objet à 70%/Arrêt selon taille objet à 85% | Oui | Oui | Oui |

Tableau 2 : Résultats par métriques

Nous avons pris soin d’enregistrer les informations de chaque appel à une requête reçu par le serveur *raspberrypi.local :8000* et les différents comportements du robot dans le fichier *raspberry.log*, qui s’écrase automatiquement à chaque lancement du serveur.

Pour chaque sujet nous essayons d’interpréter les résultats que nous avons pu recueillir dans leur fichier .log renommé à chaque fois avant de pouvoir relancer le serveur d’écoute.

### Interprétation des résultats

Nous allons interpréter les enregistrements des fichiers de nos sujets.

Le Sujet 1 a commencé son pilotage à 11h23 avec le démarrage du serveur, puis a directement reçu des requêtes de vérification de l’embourbement du robot car dans l’autre session Putty la caméra et l’ultrasonique étaient déjà lancés. L’utilisateur a commencé par regarder le flux vidéo sur la page web créée et ensuite la distance ne bougeait pas pour le robot. Le capteur de distance, l’ultrasonique a envoyé ces requêtes pour prévenir l’utilisateur que le robot pourrait être bloqué.

Puis comme dans les réseaux locaux, le ping continu d’un terminal vers un autre, ici du serveur RPI 4 vers le PC (comme écrit dans le script en annexe), a subi un temps de non-réponse d’où le message reçu à la ligne 6 qui montre que l’état du Wi-Fi du RPI 4 est down.

Après qu’il soit de nouveau connecté au réseau et au PC, il y a eu un décalage dans l’arrivé des messages à cause de la connexion car le Sujet a demandé au robot d’aller de l’avant 5 fois de suite puis le robot s’est arrêté 4 fois puisque le code dit une fois forward, un second forward vient stopper l’avancement. La 5e fois étant la bonne, Optimus Prime est entré en mon débogage par suite d’un embourbement superficiel créé dans la salle (cfr. Protocole en annexe).

Ensuite le robot a reçu l’ordre de prendre la direction droite et sortir de la salle jusqu’à faire face à un objet qui se trouve à moins de 39cm de lui puis stopper.

Il prendra alors la direction gauche où il ira jusqu’à perdre la connexion au réseau comme indiqué à 11h28.

Le Sujet 2 commence le pilotage à 12h33 et lance quelques secondes après le forward que le robot exécute jusqu’à s’embourber devant l’objet de taille inférieure à l’objectif de l’ultrasonique créé pour simuler le stuck.

Sortie du stuck, le serveur reçoit la commande d’aller à droite en direction de la sortie de la salle. En moins de 39cm de l’objet il stoppe pour prendre la gauche comme prochaine commande. Il avancera à une distance où il constate que le PC n’est plus connecté. Sûrement un problème de connexion car le PC fixe dans le réseau et proche du modem.

Le scénario s’est arrêté car les batteries commençaient à lâcher et les leds sur le picar-V clignotaient.

Le Sujet 3, quant à lui, commence bien son pilotage tout comme pour le Sujet 2, un forward puis l’entrée en mode unstuck. Il perd la connexion au PC à cause de l’état du Wi-Fi du RPI 4 qui est sur down, puis se reconnecte.  
Le moment passé sans agir a envoyé un checkStuck qui ne sera pas considéré, ensuite le Sujet fait un forward right pour sortir le robot de la salle où il s’arrêtera à moins de 39cm de l’objet.

Puis il fait un forward left pour longer le couloir jusqu’à perdre la connexion lorsqu’il sort du réseau.

Le scénario avait prévu au départ un événement de conversation avec un étudiant au couloir, d’où le second foward envoyé à chaque fois par les utilisateurs pour stopper le robot et lancer le dialogue. Après avoir constaté que la conversation qui se faisait entre le RPI 4 et le PC resté dans la salle, via un autre port du RPI 4, consommait beaucoup de ressources, on s’est résolu de l’enlever du protocole.

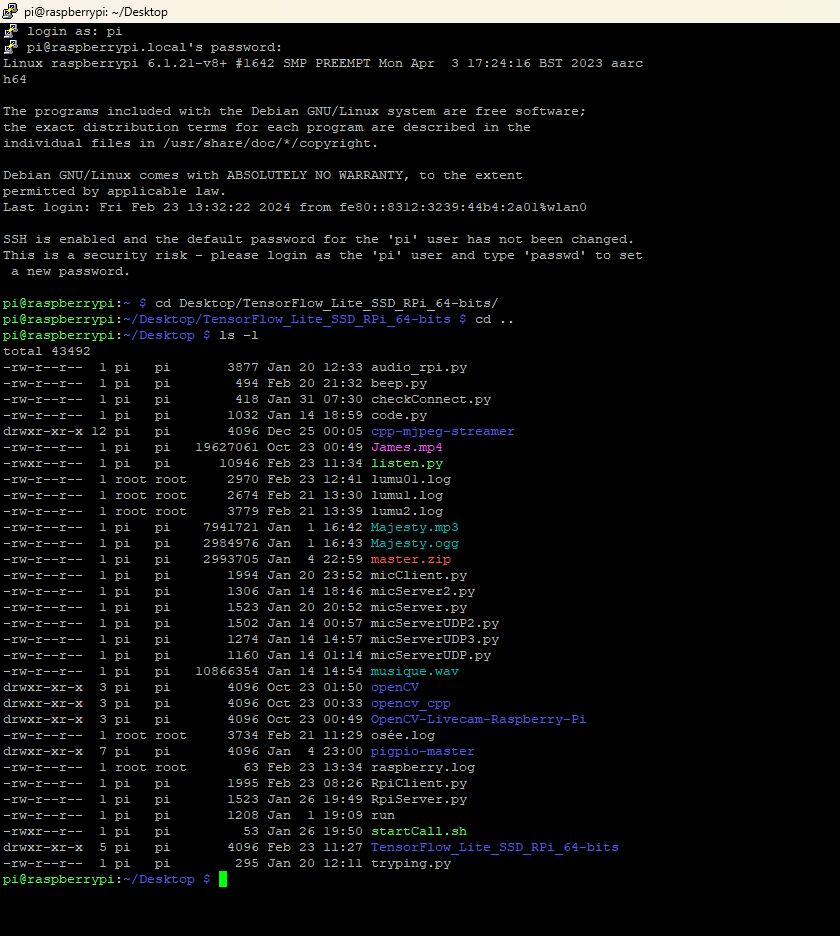




Figure 19. Les fichiers .log après leur sauvegarde

## FEEDBACK DES SUJETS

Dans ce travail, nous avons cherché à concevoir un système qui permettrait à l’utilisateur handicapé de bénéficier d’une assistance pour sa mobilité et son interaction avec son environnement, en utilisant une interface cerveau-ordinateur (BCI) et un véhicule robotique semi-autonome. Afin d’évaluer la réussite de notre quête, nous avons récolté les appréciations du système auprès des sujets qui ont participé aux expériences au laboratoire.

Nous avons recueilli leur feedback sur trois aspects principaux : le port du casque, l’apprentissage avec le casque et le pilotage du robot. Nous présentons dans cette section les résultats de cette collecte de feedback, ainsi que les implications pour l’amélioration de notre système.

### LE PORT DU CASQUE

Le casque Emotiv que nous avons utilisé pour capter les expressions faciales des sujets est un dispositif qui se porte sur la tête et qui comporte des électrodes, un gyroscope et une batterie. Il est relié à un ordinateur par un câble USB ou par une connexion sans fil. Le port du casque est une condition nécessaire pour utiliser le système, car il permet de contrôler le robot par la pensée. Nous avons donc demandé aux sujets de nous donner leur avis sur le confort, la praticité et l’esthétique du casque.

Les sujets ont globalement trouvé que le casque ne dérangeait pas du fait qu’ils en portaient pendant plusieurs heures. Ils ont apprécié le fait que le casque soit léger, souple et ajustable à la taille de leur tête. Ils ont également trouvé que le casque était plutôt confortable, car il ne serait pas trop leur crâne ni leurs oreilles, et qu’il ne provoquait pas de démangeaisons ni de transpiration. Deux sujets ont même déclaré qu’ils ne se rendaient pas compte qu’ils avaient encore le casque sur la tête, tellement il se portait comme un accessoire de beauté qu’on pouvait oublier sur soi.

### L’APPRENTISSAGE AVEC LE CASQUE

L’apprentissage avec le casque consiste à calibrer le casque en fonction des expressions faciales du sujet, afin de créer un profil personnalisé qui permettra au sujet de contrôler le robot. L’apprentissage avec le casque se fait à l’aide d’un logiciel fourni par Emotiv, qui demande au sujet de reproduire plusieurs fois les expressions faciales correspondant aux commandes du robot, comme le sourire, le clignement des yeux ou le froncement des sourcils. L’apprentissage avec le casque est une étape importante pour utiliser le système, car il permet d’optimiser la précision et la réactivité du casque. Nous avons donc demandé aux sujets de nous donner leur avis sur la durée, la facilité et l’intérêt de l’apprentissage avec le casque.

Les sujets ont généralement trouvé que la durée de l’apprentissage avec le casque était passable, car elle ne dépassait pas 15 minutes. Ils ont compris que cette durée était nécessaire pour assurer une bonne calibration du casque, et qu’elle n’était pas trop longue par rapport au temps d’utilisation du système. Un sujet a toutefois demandé si la possibilité de réduire le temps d’apprentissage pouvait être envisagée pour les horizons futurs, car il trouvait que c’était une contrainte pour les utilisateurs pressés ou impatients.

Les sujets ont également trouvé que la facilité de l’apprentissage avec le casque était satisfaisante, car ils n’ont pas rencontré de difficultés majeures pour reproduire les expressions faciales demandées.

Les sujets ont enfin trouvé que l’intérêt de l’apprentissage avec le casque était élevé, car ils ont ressenti une certaine curiosité et une certaine excitation à l’idée de contrôler le robot par la pensée. Ils ont trouvé que l’apprentissage avec le casque était une expérience originale et ludique, qui leur permettait de découvrir les capacités du casque et de se familiariser avec le système. Ils ont également trouvé que l’apprentissage avec le casque était une opportunité d’apprendre sur eux-mêmes, sur leurs expressions faciales et sur leur concentration.

### LE PILOTAGE DU ROBOT

Le pilotage du robot consiste à utiliser les expressions faciales captées par le casque pour envoyer des commandes au robot, comme avancer, tourner d’un côté ou s’arrêter. Le pilotage du robot se fait à l’aide d’une page web qui affiche les images captées par la caméra du robot, ainsi que les informations sur les objets détectés, les distances mesurées. Nous avons donc demandé aux sujets de nous donner leur avis sur l’agrément, l’effort et l’autonomie du pilotage du robot.

Les sujets ont unanimement trouvé que le pilotage du robot était agréable, car ils ont éprouvé du plaisir et de la satisfaction à conduire une voiture juste par les biais des expressions faciales. Ils ont apprécié le fait que le robot réponde à leurs commandes, et qu’il leur offre un retour visuel de qualité. Ils ont également apprécié le fait que le robot puisse prendre de lui-même des décisions de stopper, de rentrer en arrière et d’afficher les images sur l’écran, ce qui les émerveillait et les rassurait.

Les sujets ont aussi trouvé que le pilotage du robot ne demandait pas trop d’effort de leur côté, car ils n’avaient pas besoin de faire des gestes compliqués ni de prononcer des mots spécifiques pour contrôler le robot. Ils ont trouvé que les expressions faciales étaient des commandes naturelles et faciles à utiliser, et qu’elles ne les fatiguaient pas ni ne les stressaient.

### CONCLUSION

Pour tous les sujets, c’était la première fois qu’ils entendaient parler du BCI, du casque Emotiv et du système semi-autonome piloté de la sorte. Ils étaient enthousiastes à l’idée de refaire les expériences, tellement c’était une expérience positive pour eux.

Le feedback des sujets montre que le système que nous avons conçu répond aux besoins et aux préférences des utilisateurs, avec une ergonomie respectée. Le feedback des sujets nous permet également d’identifier des pistes d’amélioration pour le système, comme la réduction du temps d’apprentissage, l’augmentation de la fiabilité du casque, ou l’ajout de nouvelles fonctionnalités au robot. Nous tiendrons compte de ces remarques pour améliorer notre système dans le futur.

## DIFFICULTES RENCONTREES

Les objectifs fixés pour ce travail ont été d’une grande envergure, mais notre volonté de parvenir à l’atteinte de ceux-ci était beaucoup plus grande quoique la nature du chemin est restée parsemer d’embûches. Oui, nous avons rencontré des difficultés à concevoir et à réaliser un projet répondant au cahier de charge qui nous a été confié.

Parmi les difficultés rencontrées, liées à l’aspect technique et au développement, nous pouvons citer :

* Au tout début du projet, nous avons fait fasse à une limite d’implémentation du retour visuel du robot, car souhaitant avoir un rendu temps réel au niveau du webcam qui intègre l’openCV et le video streaming. Comme technologie, nous avons commencé avec un RPI 3 et du code écrit en python.   
  Le nombre de FPS était inférieur à 4 avec une latence considérable. La documentation qu’offre le site officiel *Q-engineering*[[8]](#footnote-8) sur le computer vision, qui combine la détection d’objet du modèle *MobileNetV1*[[9]](#footnote-9)avec TFLite et C++, nous a permis d’augmenter le FPS de 7 fois plus que ce qu’on avait auparavant, soit 29 FPS, compatible avec RPI 4.
* Les librairies stables pour les capteurs étaient difficiles voire pas sûres malgré le fait qu’on soit sur un environnement Linux, la plupart étant « depreciated » comme wiringPi qui se trouve actuellement que sur github. L’instabilité de la bibliothèque pyaudio avec ALSA qui rend très lourd l’utilisation de l’IP call qu’on a implémenté.
* Le capteur ultrason étant très instable et imprévisible dans le calcul des distances des objets lorsque le robot est en mouvement, il nous a fallu développer des algorithmes en tenant compte de ce facteur sans sortir de la logique de détection d’objet voulue.
* Du point de vue alimentation énergétique du robot, nous avons connu des difficultés qui nous ont retardées pendant l’étape des expérimentations et suscitant de la fatigue dans l’attente que les batteries soient de nouveau chargées. Les piles s’affaiblissaient très rapidement après l’ajout des capteurs sur le robot, et malgré la conception d’une alimentation en circuit parallèle des piles en série, nous étions appelés à les charger après 15 minutes d’utilisation.
* Nous avons eu aussi du mal à accéder à une infrastructure réseau stable pour une « safe connection », on a su se munir d’un modem de poche qui servait de réseau afin de pouvoir faire communiquer tous les éléments du système.

Ces problèmes majeurs ne nous ont pas empêchés de mener à bien cette croisade et nous ont rendus plus forts pour éclairer les zones d’ombre négligées et ignorées au départ. Nous avons appris de nos erreurs, de nos échecs et de nos réussites, et nous avons su nous adapter aux contraintes et aux imprévus.

## CONCLUSION

Dans ce chapitre, nous avons présenté les expérimentations et les résultats de notre travail sur le système robotique intelligent contrôlé par un BCI basé sur l’expression faciale. Nous avons décrit le protocole expérimental que nous avons suivi, qui comprenait deux grandes parties : l’entraînement du sujet au casque Epoc+ et le pilotage du robot avec des capacités d’autonomie.

Nous avons utilisé trois sujets pour tester notre système, et nous avons mesuré différentes métriques pour évaluer sa performance, telles que le temps, la distance, la précision, la fluidité, la réactivité et la satisfaction. Nous avons ainsi montré que notre système était capable de répondre aux commandes du sujet, de s’adapter aux situations imprévues, et de créer un lien affectif avec l’utilisateur.

# CONCLUSION ET PERSPECTIVES

En péroraison, ce projet avait pour objectif de concevoir un véhicule robotique semi-autonome piloté par une interface cerveau-ordinateur (BCI) et doté de capacités de vision par ordinateur, afin de fournir une assistance aux personnes handicapées dans leur déplacement et leur interaction avec leur environnement. Pour cela, nous avons exposé les principes théoriques et techniques qui sous-tendent les BCI, la vision par ordinateur et la robotique, ainsi que les enjeux éthiques, sociaux et juridiques liés à ce type de système.

Nous avons ensuite décrit le prototype que nous avons développé, en nous basant sur le travail de Pierre Sedi, qui a mis en œuvre une modalité de contrôle du robot basée sur l’expression faciale. Nous avons enrichi le prototype avec des fonctionnalités de détection d’objets, de mesure de distances, de capture de sons et de dialogue, en utilisant des capteurs et des algorithmes de vision par ordinateur. Nous avons effectué des tests du prototype dans différentes situations et évalué ses performances et son ergonomie.

Les résultats obtenus montrent que le prototype est capable de se déplacer et d’effectuer certaines actions de manière autonome, mais qu’il requiert aussi l’intervention humaine pour certaines situations ou décisions. Le prototype offre une alternative intéressante pour les personnes handicapées, qui peuvent contrôler le robot par la pensée et bénéficier d’un retour visuel et sonore.

Le prototype présente aussi des limites, notamment en termes de fiabilité, de robustesse, d’adaptabilité et de sécurité. Certaines difficultés rencontrées montrent à suffisance qu’une régulation énergétique est d’une importance capitale pour apporter de la stabilité et une saine utilisabilité du système. La télé présence évoquée dans le travail n’a pas pu être réaliste à cause des exigences en ressources des technologies disponibles sur le RPI 4. Le prototype nécessite également une calibration et un apprentissage préalables, ainsi qu’une connexion stable et sécurisée.

Ce projet apporte une contribution originale et innovante au domaine des BCI et de la robotique, en proposant un système qui combine plusieurs technologies et qui respecte les besoins et les préférences des utilisateurs. Ce projet ouvre aussi des perspectives d’approfondissement et d’amélioration, tant sur le plan technique que sur le plan éthique. Il serait intéressant de développer d’autres modalités de contrôle du robot, comme les ondes cérébrales ou les mouvements oculaires, ou d’intégrer d’autres capteurs, comme le toucher ou l’odorat.

Pourquoi pas avancer dans cet optique pour réaliser un robot balayeur de couloir au sein de la Faculté Polytechnique de l’Université de Kinshasa. Il serait également pertinent de réaliser des études plus poussées sur l’impact du système sur la qualité de vie, l’autonomie et la dignité des personnes handicapées, ainsi que sur les risques de manipulation, de discrimination ou de dépendance.

Pour conclure, nous sommes conscients que ce projet n’est pas parfait et qu’il peut être amélioré sur de nombreux aspects. Nous accueillons avec humilité les remarques, les suggestions et les critiques constructives qui nous permettront de corriger nos erreurs, de combler nos lacunes et de progresser dans notre apprentissage. Nous espérons également que ce projet suscitera l’intérêt et la curiosité d’autres chercheurs, étudiants ou professionnels, qui voudront collaborer avec nous pour développer, améliorer ou appliquer ce système.

Nous sommes ouverts à toute forme de coopération, d’échange ou de partage, dans le respect des principes éthiques et scientifiques qui guident notre travail. Nous pensons que ce projet a un potentiel important pour contribuer au bien-être et à l’inclusion des personnes handicapées, et nous souhaitons le faire évoluer dans ce sens.

# BIBLIOGRAPHIE

[L’enjeu d’une Interface Cerveau Ordinateur fonctionnant pour chacun - Hello Future Orange](https://hellofuture.orange.com/fr/lenjeu-dune-interface-cerveau-ordinateur-fonctionnant-pour-tout-le-monde/)

[Apprendre à contrôler une interface cerveau-ordinateur : le projet BrainConquest | Cairn.info](https://www.cairn.info/revue-realites-industrielles-2021-3-page-16.htm)

[Smart Robot Car Kit for Raspberry Pi - SunFounder](https://www.sunfounder.com/products/smart-video-car)

[Anatomie du cerveau : Comment fonctionne le cerveau humain ? (institutducerveau-icm.org)](https://institutducerveau-icm.org/fr/actualite/comprendre-le-cerveau-et-son-fonctionnement)

[Interface cerveau-machine (ICM) · Inserm, La science pour la santé](https://www.inserm.fr/dossier/interface-cerveau-machine-icm/)

[Install TensorFlow 2 Lite on Raspberry 64 OS - Q-engineering (qengineering.eu)](https://qengineering.eu/install-tensorflow-2-lite-on-raspberry-64-os.html)

[Computer vision - Q-engineering (qengineering.eu)](https://qengineering.eu/computer-vision.html)

[MobileNetV1 Explained | Papers With Code](https://paperswithcode.com/method/mobilenetv1)

[EMOTIV EPOC+ 14-Channel Wireless EEG Headset - EMOTIV](https://www.emotiv.com/epoc/)

P.Sedi, « Conception d’une interface cerveau-machine temps-réel asynchrone pour le pilotage à distance des systèmes robotiques à l’aide du casque Emotiv EPOC+ : Cas du robot Smart Video Car »

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. Clinical neurophysiology, 113(6), 767-791.

O. Akinari, T. Kouji, K. Toshihiro, O. Hiroki and K. Kenji, "Affective Stimuli for an Auditory P300 Brain-Computer Interface," *Frontiers in Neuroscience*, vol. 11, 2017, DOI: 10.3389/fnins.2017.00522

A. Nourmohammadi, M. Jafari and T. O. Zander, "A Survey on Unmanned Aerial Vehicle Remote Control Using Brain–Computer Interface," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 4, pp. 337–348, May 2018, DOI: 10.1109/THMS.2018.2830647

H. Gao, L. Luo, M. Pi, Z. Li, Q. Li, K. Zhao and J. Huang, "EEG-Based Volitional Control of Prosthetic Legs for Walking in Different Terrains," in IEEE Transactions on Automation Science and Engineering, vol. 18, no. 2, pp. 530–540, April 2021, DOI: 10.1109/TASE.2019.2956110

R. Li, X. Zhang, Z. Lu, C. Liu, H. Li, W. Sheng and R. Odekhe, "An Approach for Brain-Controlled Prostheses Based on a Facial Expression Paradigm," in Frontiers in Neuroscience, vol. 12, art. 943, Dec. 2018, DOI: 10.3389/fnins.2018.00943

V. F. Cardoso, D. Delisle-Rodriguez, M. A. Romero-Laiseca, F. A. Loterio, D. Gurve, A. Floriano, C. Valadão, L. Silva, S. Krishnan, A. Frizera-Neto and T. F. Bastos-Filho, “Effect of a Brain–Computer Interface Based on Pedaling Motor Imagery on Cortical Excitability and Connectivity,” *Sensors*, vol. 21, no. 6:2020, March 2021, DOI: 10.3390/s21062020

M. A. Romero-Laiseca, D. D. Rodriguez, V. Cardoso, D. Gurve, F. A. Loterio, J. H. P. Nascimento, S. Krishnan, A. Frizera and T. Bastos-Filho, “A Low-Cost Lower-Limb Brain-Machine Interface Triggered by Pedaling Motor Imagery for Post-Stroke Patients Rehabilitation,” in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 4, pp. 988–996, April 2020, DOI: 10.1109/TNSRE.2020.2974056

E. Petoku and G. Capi, "Mobile Humanoid Robot Control through Object Movement Imagery," 2022 4th International Conference on Control and Robotics (ICCR), Guangzhou, China, 2022, pp. 72–76, DOI: 10.1109/ICCR55715.2022.10053905

A. Mitocaru, M. S. Poboroniuc, D. C. Irimia, S. Placinta and A. Baciu, "Brain-Computer Interface Based Control for a Humanoid Robot," 2022 International Conference and Exposition on Electrical And Power Engineering (EPE), Iasi, Romania, 2022, pp. 714–718, DOI: 10.1109/EPE56121.2022.9959801

D. Göhring, D. Latotzky, M. Wang and R. Rojas, "Semi-autonomous Car Control Using Brain Computer Interfaces," in: *Intelligent Autonomous Systems 12. Advances in Intelligent Systems and Computing*, vol. 194, S. Lee, H. Cho, K.-J. Yoon and J. Lee, Berlin, Germany: Springer, 2013, pp. 393–408, DOI: 10.1007/978-3-642-33932-5\_37

J. Castillo, S. Müller, E. Caicedo, A. F. De Souza and T. Bastos, "Proposal of a Brain Computer Interface to command an autonomous car," 5th ISSNIP-IEEE Biosignals and Biorobotics Conference (2014): Biosignals and Robotics for Better and Safer Living (BRC), Salvador, Brazil, 2014, pp. 1–6, DOI: 10.1109/BRC.2014.6880969

H. Zeng, Y. Shen, X. Hu, A. Song, B. Xu, H. Li, Y. Wang and P. Wen, "Semi-Autonomous Robotic Arm Reaching With Hybrid Gaze–Brain Machine Interface," *Frontiers in Neurorobotics*, vol. 13, art. 111, 2019, DOI: 10.3389/fnbot.2019.00111

H. Zeng, Y. Wang, C. Wu, A. Song, J. Liu, P. Ji, B. Xu, L. Zhu, H. Li and P. Wen, "Closed-Loop Hybrid Gaze Brain-Machine Interface Based Robotic Arm Control with Augmented Reality Feedback," *Frontiers in Neurorobotics*, vol. 17, art. 60, 2017, DOI: 10.3389/fnbot.2017.00060

X. Chen, X. Huang, Y. Wang and X. Gao, "Combination of Augmented Reality Based Brain- Computer Interface and Computer Vision for High-Level Control of a Robotic Arm," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 12, pp. 3140–3147, Dec. 2020, DOI: 10.1109/TNSRE.2020.3038209

M. Ali, A. Anjum, O. Rana, A. R. Zamani, D. Balouek-Thomert and M. Parashar, "RES: Real-Time Video Stream Analytics Using Edge Enhanced Clouds," in IEEE Transactions on Cloud Computing, vol. 10, no. 2, pp. 792–804, 1 April-June 2022, DOI: 10.1109/TCC.2020.2991748

S. Siuly, Y. Li and Y. Zhang, "Electroencephalogram (EEG) and Its Background," in: *EEG Signal Analysis and Classification*, Health Information Science, S. Siuly, Y. Li and Y. Zhang, Cham, Switzerland: Springer, 2016, pp. 3–21, DOI: 10.1007/978-3-319-47653-7\_1

"EPOC+ Headset details," [Online]. Available: https://emotiv.gitbook.io/epoc-user-manual/using-headset/epoc+\_headset\_details. Accessed on: October 10, 2023.

# ANNEXES

## SCRIPTS

Le serveur principal listen.py :

|  |
| --- |
| #!/usr/bin/python  **import** **socketserver**  **from** **http.server** **import** BaseHTTPRequestHandler,HTTPServer  **import** **time**  **import** **http.client**  **import** **subprocess**  **from** **urllib.request** **import** urlopen  **import** **threading**  **from** **multiping** **import** multi\_ping  **import** **logging** # Importer le module logging  **from** **logging.handlers** **import** RotatingFileHandler # Importer la classe RotatingFileHandler  **import** **os**  **import** **pyaudio**  **from** **math** **import** pi  **import** **numpy** **as** **np**  actionsList = []  runningFW = **False**  runningLeft = **False**  runningRight = **False**  stuckRequests = **0**  silenceServer = **False**  setRunSpeed = **False**  disconnected = **False**  startDebugTime = **0**  backTime = **7**  isStop = **False**  PC = '192.168.0.100'  **if** os.path.exists("raspberry.log"): os.remove("raspberry.log")  logger = logging.getLogger(\_\_name\_\_)  logger.setLevel(logging.INFO) # Définir le niveau de sévérité à INFO  handler = RotatingFileHandler("raspberry.log", mode="w", maxBytes=**10**\***1024**\***1024**, backupCount=**0**) # Créer un objet handler pour écrire dans un fichier  formatter = logging.Formatter("%(asctime)s - %(levelname)s - %(message)s") # Définir le format du message  handler.setFormatter(formatter) # Associer le format au handler  logger.addHandler(handler) # Associer le handler au logger  **try**:  subprocess.Popen(["sudo /home/pi/SunFounder\_PiCar-V/remote\_control/start"], shell=**True**)  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  logger.info("Démarrage du serveur PiCar-V") # Utiliser logger.info pour enregistrer un message  **except**:  print("Error: Unable to run PiCar-V server.")  logger.exception("Erreur lors du démarrage du serveur PiCar-V") # Utiliser logger.exception pour enregistrer une erreur  **def** **make\_sinewave**(frequency, length, sample\_rate=**44100**):  length = int(length \* sample\_rate)  factor = float(frequency) \* (pi \* **2**) / sample\_rate  waveform = np.sin(np.arange(length) \* factor)  **return** waveform  **def** **piCar**(actionPiCar):  **global** setRunSpeed  **if** **not** setRunSpeed:  urlopen("http://0.0.0.0:8001/run/?speed=54")  setRunSpeed = **True**  urlopen("http://0.0.0.0:8001/run/?action="+actionPiCar)  wave = make\_sinewave(**1000**, **1**) # Créer un signal de 1000 Hz et 1 s  **def** **check\_wifi\_connection**():  **global** disconnected  **global** startDebugTime  **global** backTime  # Ajouter une variable pour compter le nombre de fois que le PC est injoignable  **global** unreachable\_count  **try**:  responses, no\_responses = multi\_ping([PC], timeout=**2**, retry=**0**)  # Augmenter le compteur de 1 si le PC ne répond pas  **if** len(responses)==**0**:  unreachable\_count += **1**  # Réinitialiser le compteur à zéro si le PC répond  **else**:  unreachable\_count = **0**  # Vérifier si le compteur est supérieur ou égal à 3  **if** unreachable\_count >= **3**:  print("Error: Pc is unreachable.")  logger.exception("Erreur de connexion au PC")  piCar("stop")  result = subprocess.check\_output("cat /sys/class/net/wlan0/operstate", shell=**True**)  operstate = result.decode().strip()  **if** operstate == "down":  logger.exception('down down down, Je viens de perdre la connexion au réseau')  **if** startDebugTime==**0**: startDebugTime = time.time()  debugMode('conn')  **else**:  print("Error: I'm in Wi-Fi but PC is out of Wi-Fi.")  logger.exception("Erreur: I'm in Wi-Fi but PC is out of Wi-Fi.")  # Afficher le code d'erreur  print("Error: PC is unreachable for 3 times or more.")  # Faire beeper le robot  p = pyaudio.PyAudio()  stream = p.open(format=pyaudio.paFloat32, channels=**1**, rate=**44100**, output=**1**,)  beeping = **True**  logger.info("Beeping est activé")  **if** beeping:  stream.write(wave.astype(np.float32).tostring())  time.sleep(**1**)  beeping = **False**  stream.stop\_stream()  stream.close()  p.terminate()  disconnected = **True**  # Sinon, si le PC répond, continuer normalement  **elif** len(responses) > **0**:  beeping = **False**  startDebugTime = **0**  backTime = **7**  **if** disconnected:  logger.info("De nouveau connecté")  disconnected = **False**  **except**:  print("Error: Unable to check Wi-Fi connection")  logger.exception("Erreur lors de la vérification de la connexion Wi-Fi")  timer = threading.Timer(**3**, check\_wifi\_connection) # Créer un objet timer qui appelle la fonction check\_wifi\_connection après 3 secondes  timer.start()  **def** **debugMode**(a):  **global** actionsList  **global** silenceServer  **global** backTime  **global** startDebugTime  silenceServer = **True**  print("Entering debug mode...")  logger.info("Entrée en mode debug") # Utiliser logger.info pour enregistrer un message  piCar('stop')  piCar('fwstraight')  #for i in range(0, 3):  **if** actionsList[-**1**][**0**] == 'forward':  piCar('backward')  #elif actionsList[-1][0] == 'backward':  # piCar('forward')  **elif** actionsList[-**1**][**0**] == 'fwleft':  piCar('fwleft')  piCar('backward')  time.sleep(**4**)  piCar('fwstraight')  backTime = **7**  startDebugTime -= **7**  **elif** actionsList[-**1**][**0**] == 'fwright':  piCar('fwright')  piCar('backward')  time.sleep(**4**)  piCar('fwstraight')  backTime = **7**  startDebugTime -= **7**  **if** a=='conn' **and** actionsList[-**1**][**0**] == 'forward':  logger.info("Fin du mode debug connexion")  time.sleep(backTime) #( int(actionsList[-1][1] - actionsList[-2][1]) )  startDebugTime -= backTime # for the next back, if it ever have to happen  **if** backTime < **7** : actionsList.pop() # si le robot est revenu en arriere pendant moins de 7 secondes, alors on enleve le dernier objet de la liste  **if** ( startDebugTime - actionsList[-**1**][**1**] ) < **7** :  backTime = startDebugTime - actionsList[-**1**][**1**]  **else**: backTime = **7**  **elif** a=='unstuck':  logger.info("Fin du mode debug unstuck")  time.sleep(**10**)  piCar('stop')  piCar('fwstraight')  logger.info("Sortie du mode debug")  silenceServer = **False**  **class** **MyHandler**(BaseHTTPRequestHandler):  **def** **do\_GET**(self):  self.send\_response(**200**)  self.send\_header('Content-type', 'text/plain')  self.end\_headers()  **global** silenceServer, runningFW, runningLeft, runningRight, stuckRequests, isStop  **if** silenceServer: **return**  **if** self.path == '/run/?action=fwleft':  **if** **not** stuckRequests: # Ajouter une condition pour vérifier si stuckRequests est False  actionsList.append(['fwleft',time.time()])  runningLeft = silenceServer = **True**  piCar('fwleft')  piCar('forward')  time.sleep(**7**)  piCar('fwstraight')  actionsList.append(['forward',time.time()])  runningLeft = silenceServer = **False**  runningFW = **True**  isStop = **False** # Mettre isStop à False  logger.info("Requête GET /run/?action=fwleft reçue et exécutée") # Utiliser logger.info pour enregistrer un message  **elif** self.path == '/run/?action=forward':  **if** **not** stuckRequests: # Ajouter une condition pour vérifier si stuckRequests est False  **if** runningFW:  piCar('stop')  runningFW = **False**  **else**:  actionsList.append(['forward',time.time()])  runningFW = silenceServer = **True**  piCar('forward')  #time.sleep(3)  silenceServer = **False**  isStop = **False** # Mettre isStop à False  logger.info("Requête GET /run/?action=forward reçue et exécutée") # Utiliser logger.info pour enregistrer un message  **elif** self.path == '/run/?action=fwright':  **if** **not** stuckRequests: # Ajouter une condition pour vérifier si stuckRequests est False  actionsList.append(['fwright',time.time()])  runningRight = silenceServer = **True**  piCar('fwright')  piCar('forward')  time.sleep(**7**)  piCar('fwstraight')  actionsList.append(['forward',time.time()])  runningRight = silenceServer = **False**  runningFW = **True**  isStop = **False** # Mettre isStop à False  logger.info("Requête GET /run/?action=fwright reçue et exécutée") # Utiliser logger.info pour enregistrer un message  **elif** self.path == '/run/?action=fwstraight':  **if** **not** stuckRequests: # Ajouter une condition pour vérifier si stuckRequests est False  piCar('fwstraight')  isStop = **False** # Mettre isStop à False  logger.info("Requête GET /run/?action=fwstraight reçue et exécutée") # Utiliser logger.info pour enregistrer un message  **elif** self.path == '/run/?action=backward': # Ne sera jamais appelE  #actionsList.append(['backward',time.time()])  silenceServer = **True**  piCar('backward')  time.sleep(**7**)  silenceServer = **False**  isStop = **False** # Mettre isStop à False  logger.info("Requête GET /run/?action=backward reçue et exécutée") # Utiliser logger.info pour enregistrer un message  **elif** self.path == '/run/?action=stop':  piCar('stop')  print('receive stop')  #time.sleep(7)  isStop = **True** # Mettre isStop à True  stuckRequests = **False** # Mettre stuckRequests à False  logger.info("Requête GET /run/?action=stop reçue et exécutée") # Utiliser logger.info pour enregistrer un message  **elif** self.path == '/run/?action=checkStuck':  **if** **not** isStop: # Ajouter une condition pour vérifier si isStop est False  **if** (runningFW **or** runningLeft **or** runningRight):  # Mettre stuckRequests à True  stuckRequests = **True**  print('The robot is stucked.')  debugMode('unstuck')  #time.sleep(20)  runningFW = runningLeft = runningRight = **False**  **else**:  # Mettre stuckRequests à False  stuckRequests = **False**  print('The robot is not stucked.')  logger.info("Requête GET /run/?action=checkStuck reçue et traitée")  httpd = socketserver.TCPServer(("", **8000**), MyHandler)  check\_wifi\_connection()  **try**:  httpd.serve\_forever()  **except** **KeyboardInterrupt**:  print("Shutting down the server...")  httpd.shutdown() |

Les configurations utilisées dans le travail :

|  |
| --- |
| g++ -o ultrasonic ultrasonic.cpp -lwiringPi  g++ -std=c++**17** -fexceptions -pthread -g -Wall -o avoidance avoidance.cpp -lwiringPi `pkg-config opencv4 --cflags --libs` `pkg-config gstreamer-**1.0** --cflags --libs` -lopenal -lsndfile -lpthread -I/home/pi/tensorflow -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/include -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/absl -I/home/pi/Desktop/cpp-mjpeg-streamer/include -L/home/pi/tensorflow/tensorflow/lite/tools/make/gen/linux\_aarch64/lib -L/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/build -ltensorflow-lite -lcurl -ldl  gcc -o ultra ultra.cpp -lpigpio -lpthread -lrt -ldl -lstdc++  g++ server2.cpp -lasound -lpthread -o server2  g++ -std=c++**17** -o server server.cpp -lpthread -lportaudio  g++ -std=c++**17** -fexceptions -pthread -g -Wall -o CamUltrasonic CamUltrasonic.cpp -lwiringPi `pkg-config opencv4 --cflags --libs` `pkg-config gstreamer-**1.0** --cflags --libs` -lopenal -lsndfile -lpthread -I/home/pi/tensorflow -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/include -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/absl -I/home/pi/Desktop/cpp-mjpeg-streamer/include -L/home/pi/tensorflow/tensorflow/lite/tools/make/gen/linux\_aarch64/lib -L/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/build -L/path/to/openal -L/path/to/sndfile -ltensorflow-lite -lcurl -ldl  g++ -std=c++**17** -fexceptions -pthread -g -Wall -o CamUltrasonic CamUltrasonic.cpp -lwiringPi `pkg-config opencv4 --cflags --libs` `pkg-config gstreamer-**1.0** --cflags --libs` -lopenal -lsndfile -lpthread -I/home/pi/tensorflow -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/include -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/absl -I/home/pi/Desktop/cpp-mjpeg-streamer/include -L/home/pi/tensorflow/tensorflow/lite/tools/make/gen/linux\_aarch64/lib -L/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/build -L/path/to/openal -L/path/to/sndfile -ltensorflow-lite -lcurl -ldl  g++ -std=c++**17** -fexceptions -pthread -g -Wall -o CamUltrasonic CamUltrasonic.cpp -lwiringPi `pkg-config opencv4 --cflags --libs` `pkg-config gstreamer-**1.0** --cflags --libs` -lopenal -lsndfile -lpthread -I/home/pi/tensorflow -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/include -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/absl -I/home/pi/Desktop/cpp-mjpeg-streamer/include -L/home/pi/tensorflow/tensorflow/lite/tools/make/gen/linux\_aarch64/lib -L/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/build -ltensorflow-lite -lcurl -ldl  g++ -std=c++**17** -fexceptions -pthread -g -Wall -o CamUltrasonic CamUltrasonic.cpp -lwiringPi `pkg-config opencv4 --cflags --libs` `pkg-config gstreamer-**1.0** --cflags --libs` -lopenal -lsndfile -lpthread -I. -I/home/pi/tensorflow -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/include -I/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/absl -I/home/pi/Desktop/cpp-mjpeg-streamer/include -L/home/pi/tensorflow/tensorflow/lite/tools/make/gen/linux\_aarch64/lib -L/home/pi/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers/build -ltensorflow-lite -ldl  sudo fuser -n tcp -k **8001**  sudo shutdown -h now ou sudo halt  python3 .\CyKIT.py **127.0.0.1** **5151** **6** openvibe+generic+nocounter+noheader+nobattery+ovdelay:**100**+**float**+ovsamples:**004** |

Les restes des scripts utilisés pour ce mémoire se retrouvent dans notre repository github. Vous pouvez utiliser ce lien : <https://github.com/newisdartagnan/TFE>

## PROTOCOLE EXPERIMENTAL

Dans cette partie, nous allons présenter le protocole expérimental que nous avons mis en place pour tester notre système robotique intelligent contrôlé par un BCI basé sur l’expression faciale. Notre protocole comprend deux grandes parties : l’entraînement du sujet au casque Epoc+ et le scénario du pilotage.

### L'entraînement du sujet au casque Epoc+ :

Cette étape se subdivise en 4 parties :

- Acquisition des signaux EEG du cerveau, perçus par le casque, via le dongle CyKit. Ici les directions choisies passent sur l'écran de l'ordinateur et le sujet essaie de suivre en reproduisant l'expression faciale retenue et correspondante à ladite direction.

- Entraînement des données récoltées par le serveur d'acquisition d'OpenVibe

- Classification des résultats de l'entraînement qui s'est produit suivant le paradigme d'expression faciale

- Mode online du pilotage

### Le scénario du pilotage

Dès que listen.py et CamUltrasonic sont lancés sur le RPI 4, on lance également le mode online de OpenVibe qui fait appel au script python des seuils de classes.

a. Démarrer le moteur du robot avec un forward (le temps pour une nouvelle détection étant de 5 secondes, le sujet maintient l'expression faciale pendant 4 à 5 secondes) : le robot avance jusqu’à rencontrer une situation de blocage simulée en mettant sur le chemin du robot un objet difficile à détecter car il est plus bas que son ultrason, à 4 cm du sol, dans le but de l’empêcher de continuer à avancer.

- Observation : Le robot tente seul de sortir de la situation d'embourbement après quelques secondes. Et le sujet tente de lancer un forward pendant que le robot recule mais cela ne s’exécute pas car le serveur est en pause pendant le débogage.

b. Faire un forward right pour que le robot sorte de la pièce. Une fois à l'extérieur, il avancera jusqu’à s’arrêter à moins de 39 cm de l’objet qu’il rencontre.

c. Faire un forward left pour que le robot suive le couloir jusqu’au bout, où il perdra la connexion et essaiera de revenir dans le réseau en répétant les derniers mouvements avant qu’il ne sorte de la connexion.

Rappel :

Correspondance des directions avec les expressions faciales :

1. Visage détendu (garder son calme sans rien faire) pas de direction à prendre,

2. sourcils levés pour la direction gauche (fwleft),

3. mâchoires serrées pour la direction droite (fwright),

4. large sourire en faisant montrer les dents pour le devant (forward).

## LOGS DES SUJETS

|  |
| --- |
| Sujet **1** :  **2024**-02-21 **11**:23:10,576 - INFO - Démarrage du serveur PiCar-V  **2024**-02-21 **11**:24:30,741 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:24:36,776 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:24:42,796 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:24:48,886 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:24:49,487 - ERROR - Erreur de connexion au PC  NoneType: None  **2024**-02-21 **11**:24:49,548 - ERROR - Erreur WiFi is down  NoneType: None  **2024**-02-21 **11**:24:49,697 - INFO - Beeping est activé  **2024**-02-21 **11**:24:54,899 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:24:54,934 - INFO - De nouveau connecté  **2024**-02-21 **11**:25:00,923 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:25:06,964 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:25:07,248 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **11**:25:13,065 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **11**:25:19,954 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **11**:25:43,101 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **11**:25:51,381 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **11**:25:59,721 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **11**:26:08,039 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **11**:26:09,029 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **11**:26:11,923 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **11**:26:16,021 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:26:22,184 - INFO - Entrée en mode debug  **2024**-02-21 **11**:26:22,398 - INFO - Fin du mode debug unstuck  **2024**-02-21 **11**:26:32,488 - INFO - Sortie du mode debug  **2024**-02-21 **11**:26:32,488 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:26:32,540 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **11**:26:39,476 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:26:45,571 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:26:51,625 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:27:03,556 - INFO - Requête GET /run/?action=fwright reçue et exécutée  **2024**-02-21 **11**:27:03,560 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:27:10,756 - INFO - Requête GET /run/?action=fwright reçue et exécutée  **2024**-02-21 **11**:27:27,096 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **11**:27:34,312 - INFO - Requête GET /run/?action=fwleft reçue et exécutée  **2024**-02-21 **11**:27:51,494 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **11**:27:58,591 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **11**:27:58,882 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **11**:28:48,233 - INFO - Entrée en mode debug  **2024**-02-21 **11**:28:48,233 - INFO - Fin du mode debug conn  **2024**-02-21 **11**:28:55,529 - INFO - Sortie du mode debug  **2024**-02-21 **11**:29:08,571 - INFO - Requête GET /run/?action=checkStuck reçue et traitée |

|  |
| --- |
| Sujet **2** :  **2024**-02-23 **12**:33:54,324 - INFO - Démarrage du serveur PiCar-V  **2024**-02-23 **12**:34:16,131 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-23 **12**:34:34,629 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:35:20,211 - INFO - Entrée en mode debug  **2024**-02-23 **12**:35:20,315 - INFO - Fin du mode debug unstuck  **2024**-02-23 **12**:35:30,374 - INFO - Sortie du mode debug  **2024**-02-23 **12**:35:30,375 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:36:02,476 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:36:36,686 - INFO - Requête GET /run/?action=fwright reçue et exécutée  **2024**-02-23 **12**:37:43,403 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:37:52,734 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-23 **12**:38:07,646 - INFO - Requête GET /run/?action=fwleft reçue et exécutée  **2024**-02-23 **12**:38:07,695 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-23 **12**:38:15,380 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:38:21,444 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:38:27,457 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:38:34,108 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-23 **12**:39:45,999 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:39:58,127 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:40:04,163 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:40:10,219 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:40:16,256 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:40:22,269 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:40:28,352 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:40:56,966 - ERROR - Erreur de connexion au PC  NoneType: None  **2024**-02-23 **12**:40:57,020 - ERROR - Erreur: I'm in Wi-Fi but PC is out of Wi-Fi.  NoneType: None  **2024**-02-23 **12**:40:57,178 - INFO - Beeping est activé  **2024**-02-23 **12**:40:58,450 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-23 **12**:41:02,378 - INFO - De nouveau connecté  **2024**-02-23 **12**:41:04,532 - INFO - Requête GET /run/?action=checkStuck reçue et traitée |

|  |
| --- |
| Sujet **3** :  **2024**-02-21 **13**:26:10,850 - INFO - Démarrage du serveur PiCar-V  **2024**-02-21 **13**:27:03,580 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **13**:27:38,282 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **13**:27:56,434 - INFO - Entrée en mode debug  **2024**-02-21 **13**:27:56,666 - INFO - Fin du mode debug unstuck  **2024**-02-21 **13**:28:06,798 - INFO - Sortie du mode debug  **2024**-02-21 **13**:28:06,799 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **13**:28:08,520 - ERROR - Erreur de connexion au PC  NoneType: None  **2024**-02-21 **13**:28:08,552 - ERROR - Erreur WiFi is down  NoneType: None  **2024**-02-21 **13**:28:08,702 - INFO - Beeping est activé  **2024**-02-21 **13**:28:14,177 - INFO - De nouveau connecté  **2024**-02-21 **13**:28:14,793 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **13**:28:31,819 - INFO - Requête GET /run/?action=fwright reçue et exécutée  **2024**-02-21 **13**:28:39,038 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **13**:28:54,574 - INFO - Requête GET /run/?action=fwleft reçue et exécutée  **2024**-02-21 **13**:28:54,634 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **13**:29:02,680 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **13**:29:10,989 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **13**:29:30,043 - ERROR - Erreur de connexion au PC  NoneType: None  **2024**-02-21 **13**:29:30,076 - ERROR - Erreur WiFi is down  NoneType: None  **2024**-02-21 **13**:29:30,202 - INFO - Beeping est activé  **2024**-02-21 **13**:29:31,146 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **13**:29:35,449 - INFO - De nouveau connecté  **2024**-02-21 **13**:29:37,182 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **13**:29:49,275 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **13**:29:55,396 - INFO - Requête GET /run/?action=forward reçue et exécutée  **2024**-02-21 **13**:30:11,483 - INFO - Requête GET /run/?action=stop reçue et exécutée  **2024**-02-21 **13**:30:19,570 - INFO - Entrée en mode debug  **2024**-02-21 **13**:30:19,724 - INFO - Fin du mode debug conn  **2024**-02-21 **13**:30:25,784 - INFO - Sortie du mode debug  **2024**-02-21 **13**:30:49,830 - INFO - Requête GET /run/?action=checkStuck reçue et traitée  **2024**-02-21 **13**:30:55,859 - INFO - Requête GET /run/?action=checkStuck reçue et traitée |

## PHOTO D’OPTIMUS PRIME

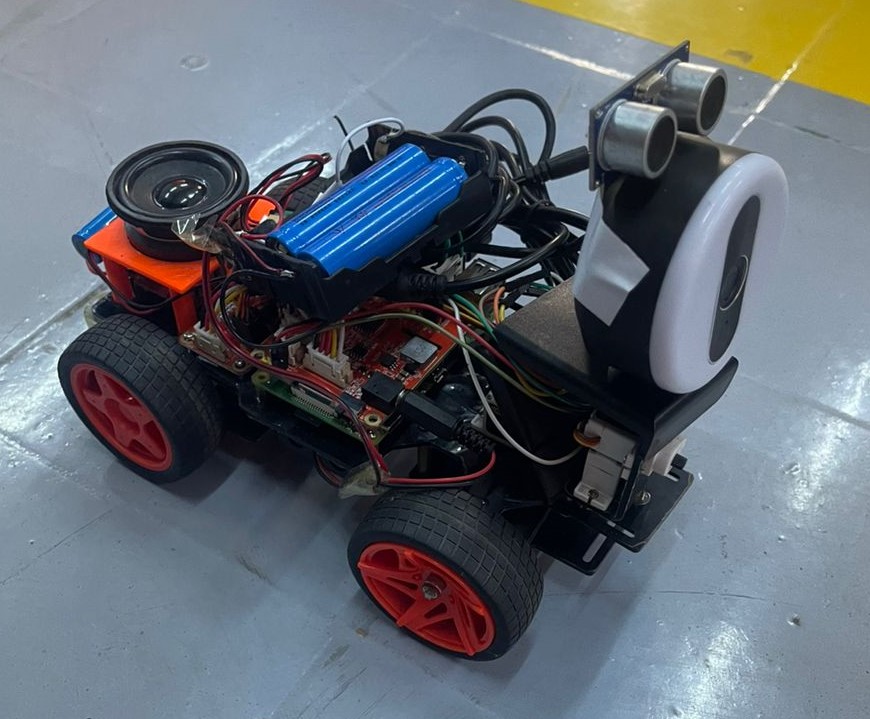


Figure 20. Optimus Prime

1. [L’enjeu d’une Interface Cerveau Ordinateur fonctionnant pour chacun - Hello Future Orange](https://hellofuture.orange.com/fr/lenjeu-dune-interface-cerveau-ordinateur-fonctionnant-pour-tout-le-monde/)

   [↑](#footnote-ref-1)
2. [Apprendre à contrôler une interface cerveau-ordinateur : le projet BrainConquest | Cairn.info](https://www.cairn.info/revue-realites-industrielles-2021-3-page-16.htm)

   [↑](#footnote-ref-2)
3. P.Sedi, « Conception d’une interface cerveau-machine temps-réel asynchrone pour le pilotage à distance des systèmes robotiques à l’aide du casque Emotiv EPOC+ : Cas du robot Smart Video Car »

   [↑](#footnote-ref-3)
4. [Smart Robot Car Kit for Raspberry Pi - SunFounder](https://www.sunfounder.com/products/smart-video-car) [↑](#footnote-ref-4)
5. [Anatomie du cerveau : Comment fonctionne le cerveau humain ? (institutducerveau-icm.org)](https://institutducerveau-icm.org/fr/actualite/comprendre-le-cerveau-et-son-fonctionnement) [↑](#footnote-ref-5)
6. [Interface cerveau-machine (ICM) · Inserm, La science pour la santé](https://www.inserm.fr/dossier/interface-cerveau-machine-icm/)

   [↑](#footnote-ref-6)
7. [Install TensorFlow 2 Lite on Raspberry 64 OS - Q-engineering (qengineering.eu)](https://qengineering.eu/install-tensorflow-2-lite-on-raspberry-64-os.html)

   [↑](#footnote-ref-7)
8. [Computer vision - Q-engineering (qengineering.eu)](https://qengineering.eu/computer-vision.html)

   [↑](#footnote-ref-8)
9. [MobileNetV1 Explained | Papers With Code](https://paperswithcode.com/method/mobilenetv1) [↑](#footnote-ref-9)