

Contrôle de drone par Électro-encéphalographie

Rapport de stage de deuxième année

Noé CARRAYROU
Étudiant en Robotique autonome

mai à septembre 2026

NTNU

Table des matières

1 introduction **3**
1.1 Résumé . . . 3
1.2 Abstract . . . 3
1.3 Présentation du laboratoire . . . 4
1.4 Objectifs . . . 4

2 Interfaces Cerveau-Machine (ICM) **5**
2.1 Notions sur le cerveau humain . . . 5
2.2 Méthodes d'enregistrement invasives et non-invasives . . . 5
2.3 Le Cortex Moteur . . . 6
2.4 Difficultés de l'EEG et de l'imagerie motrice . . . 6

3 Simulateur de drone **8**
3.1 Simulation physique . . . 8
3.1.1 Paramètres du modèle . . . 8
3.1.2 Evolution temporelle : . . . 9
3.2 Lois de commande . . . 9
3.2.1 Contrôle en position : . . . 10
3.2.2 Génération de l'attitude désirée : . . . 10
3.2.3 Contrôle en attitude . . . 11
3.2.4 Allocation des commandes . . . 11

4 Interface d'enregistrement EEG **12**
4.1 Travaux précédents . . . 12
4.2 Choix des commandes du drone . . . 12
4.3 Protocole d'enregistrement . . . 13
4.4 Connexion aux électrodes . . . 14

5 Réseaux de neurones **15**
5.1 Pré-traitement des données . . . 15
5.2 Réseau de Neurones Convolutifs (CNN) . . . 15
5.3 Optimisation des hyperparamètres . . . 16
5.4 Un réseau de neurones adapté à nos données : l'EEGNet . . . 16
5.5 Résultats . . . 17

6 Pilotage du simulateur par mesures EEG **18**

7 Conclusion **19**

8 Annexe **21**
8.1 Détail des meilleurs modèles obtenus . . . 21
8.2 Fiche d'évaluation du stage . . . 22

1 introduction

1.1 Résumé

Mon stage de deuxième année s'est déroulé à l'Université Norvégienne de Science et Technologie (NTNU), à Trondheim, en Norvège. Ce stage était supervisé par la professeur Marta Molinas, dans le laboratoire de cybernétique.

L'équipe de recherche dans laquelle j'étais a pour but d'étudier les ondes cérébrales de façon non invasive, et comment les employer dans différents domaines comme la médecine ou la robotique.

L'objectif de ce stage était de développer une interface homme-machine (IHM) permettant de piloter un drone à l'aide de mesures sur l'activité cérébrale, via électro-encéphalographie (EEG). Pour ce faire, il a fallut développer une interface afin d'enregistrer des mesures EEG, permettant d'entraîner un réseau de neurones. Ce réseau doit convertir ces signaux en commandes pour un drone. Une fois un réseau de neurones suffisamment précis obtenu, il fallait développer un simulateur de vol de drone afin de s'entraîner au pilotage, avant de passer aux vols réels.

Ce stage m'a permit d'acquérir des connaissances sur le machine learning, le fonctionnement du cerveau humain et l'acquisition de données dans le but d'entraîner un réseau de neurones.

1.2 Abstract

My second year internship was done at the Norvegian University of Science and Technology (NTNU), in Trondheim, in Norway. This internship was supervised by professor Marta Molinas, in the cybenetic department.

The research team I was in aimed at studying brain waves in a non-invasive way, and how to use it in different fields such as medicine or robotic.

The goal of this internship was to develop a human-machine interface (HMI) allowing us to pilot a drone thanks to measures on the brain activity, realised by electro-encephalography (EEG). To do that, I had to develop an interface to register the data in order to train a neural network. This network had to convert the data into commands for the drone. When the network is sufficiently trained, I had to develop a flight simulator to train to pilot the drone before flying the drone for real.

This internship helped me learn about machine learning, how the human brain works and how to collect data in order to train a neural network.

1.3 Présentation du laboratoire

L'Université Norvégienne de Science et Technologie (NTNU) est la plus grande université de Norvège. Elle regroupe plus de 43 000 étudiants chaque année, dont plus de la moitié y vient étudier les sciences et la technologie [4]. L'université regroupe différents centres et laboratoires de recherche, organisés par pôles thématiques. J'ai effectué mon stage dans le laboratoire de cybernétique, du département d'électronique.

Au sein du laboratoire, nous étions plusieurs stagiaires, doctorants et chercheurs travaillant sur différents sujets en lien avec les ondes cérébrales, dans le but de les appliquer à des domaines variés, comme la santé mentale, les troubles de la concentration ou le contrôle robotique.

Afin d'étudier et d'utiliser ces ondes, nous employons l'électro-encéphalographie, qui est une méthode non invasive permettant de mesurer l'activité cérébrale, à l'aide d'électrodes placées sur le cuir chevelu. Cela permet de suivre l'activité cérébrale et d'établir des programmes ou réseaux de neurones pouvant réagir à certains types d'activités, en fonction du but souhaité.

1.4 Objectifs

Afin de mettre en place un contrôleur de drone commandé par ondes cérébrales, il a fallu découper le projet en plusieurs parties :

1. Mise en place d'un simulateur de vol de drone simple, piloté en position
2. Définition de patterns cérébraux pertinents à classifier
3. Conception d'une interface d'enregistrement des signaux EEG correspondant aux patterns
4. Entraînement d'un réseau de neurones pour classifier les patterns
5. Entraînement au pilotage du drone simulé grâce aux ondes cérébrales
6. Pilotage du drone réel par mesures EEG

Dans le cadre de mon stage, afin de suivre l'activité cérébrale, nous utilisons un casque à 32 électrodes humides de MENTALAB, et le logiciel Explore Desktop permettant de communiquer en Bluetooth entre le casque et un ordinateur, le logiciel est disponible en open source [9].

Le drone utilisé durant ce stage est le drone Crazyflie 2.1 de Bitcraze, un drone quadricoptère miniature et open source.

2 Interfaces Cerveau-Machine (ICM)

2.1 Notions sur le cerveau humain

Le cerveau humain est composé de milliards de neurones qui communiquent entre eux grâce à de minuscules signaux électriques et chimiques. Ces signaux sont la conséquence de variations de tension électrique sur les membranes neuronales, principalement dues aux potentiels postsynaptiques (PPS) [2].

L'activité cérébrale peut être catégorisée en différentes bandes de fréquences, chacune correspondant à différents états d'activité cérébrale. Les principales bandes de fréquences sont :

1. Delta (0.5-4 Hz) : correspondant au sommeil profond
2. Theta (4-8 Hz) : correspondant au sommeil léger, à la relaxation et la créativité
3. Alpha (8-12 Hz) : correspondant à un état éveillé de relaxation, généralement associé à la méditation
4. Beta (13-30 Hz) : correspondant à une réflexion active, à la résolution de problème et au contrôle moteur
5. Gamma(30 Hz et plus) : correspondant à un haut niveau de traitement d'information et de fonctionnement cognitif

La bande Mu est généralement associée aux fréquences entre 8 et 12 Hz (comme l'activité Alpha), mais est enregistrée dans le cortex moteur ou au cortex somatosensoriel plutôt que dans le cortex visuel.

2.2 Méthodes d'enregistrement invasives et non-invasives

Placer une électrode très proche d'un neurone permet de détecter et enregistrer ces minuscules variations électriques. Cependant, cela nécessite une opération du cerveau très invasive pour la placer là.

Pour éviter cela, on emploie des systèmes non invasifs comme l'EEG, où les électrodes sont placées sur le crâne. Comme les électrodes sont plus éloignées des neurones, seuls les grands groupes de neurones produisant un signal simultané et suffisamment fort peut être détecté. Cette méthode est également plus sensibles à des variations d'impédance et de bruit.



FIGURE 1 – Électrodes placées sur le crâne

2.3 Le Cortex Moteur

Une des parties du cerveau produisant souvent des signaux suffisamment forts et coordonnés est le cortex moteur, contrôlant les mouvement volontaires des muscles. Quand on déplace un membre de notre corps, cela active le cortex moteur, qui génère des signaux détectables par EEG, principalement dans les bandes de fréquences Mu (8-12 Hz) et Beta (13-30 Hz).

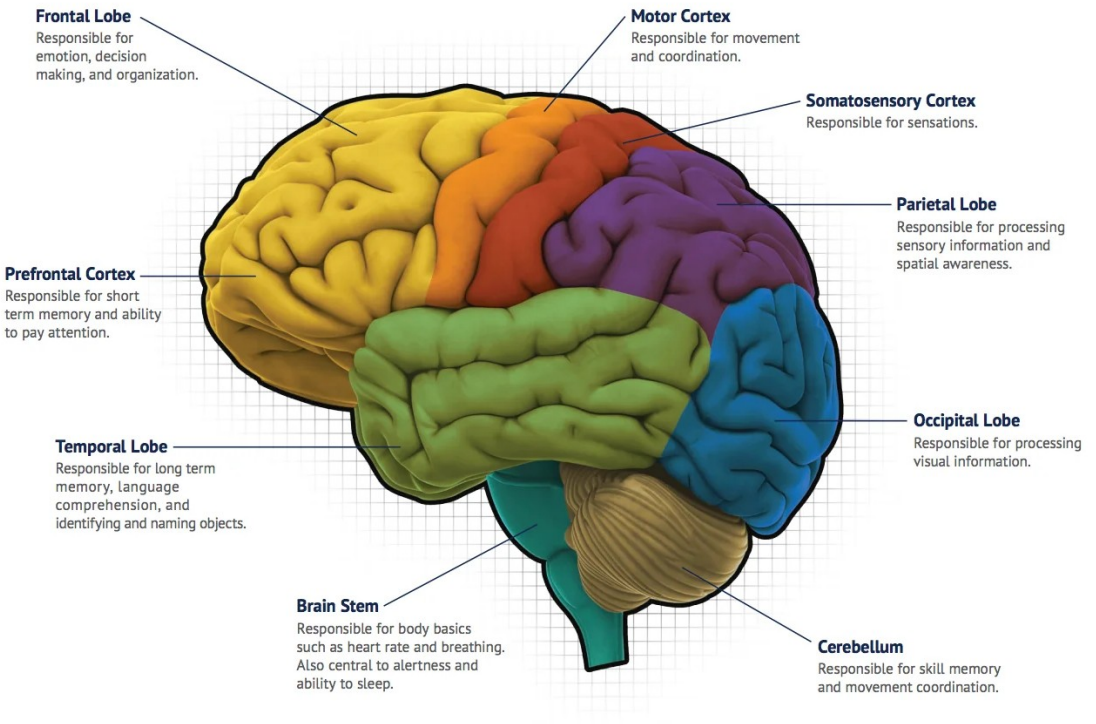


FIGURE 2 – Régions du cerveau humain

Afin d'avoir des données plus précise sur l'activité du cortex moteur, il est important d'augmenter la densité d'électrodes dans cette région là.

L'imagerie Motrice : Le fait d'imaginer faire un mouvement produit les mêmes signaux que de le faire réellement. Même si le signal est plus faible que lorsque le mouvement est réel, il reste distinguable à l'EEG. Cela permet d'obtenir des résultats sensiblement similaires chez les personnes avec des difficultés motrices par exemple.

2.4 Difficultés de l'EEG et de l'imagerie motrice

L'EEG permet de mesurer l'activité cérébrale sans envahir le cerveau, au prix d'une perte de précision. Par ailleurs, les électrodes que nous utilisons ici sont des électrodes humides, nécessitant l'application d'un gel spécial, afin de réduire la résistance électrique entre les électrodes et les neurones, pour obtenir une information plus complète. Cependant, tout ce processus prends du temps à mettre en place et nécessite l'aide d'une ou plusieurs personnes. Par ailleurs, la résistance n'étant pas toujours la même entre deux séances ni entre deux électrodes, cela ajoute de la variance entre les mesures.



FIGURE 3 – Mise en place du gel

Par ailleurs, l'imagerie motrice ne présente pas les mêmes résultats en fonction des individus, ni même sur un même individu au cours du temps. Ces différences sont appelées respectivement variabilités inter-sujets et variabilités intra-sujets [10]. d'autres facteurs comme la concentration, la fatigue ou l'état émotionnel peuvent également affecter la qualité du signal.

Pour résoudre ce problème, les chercheurs ont développés des algorithmes adaptatifs [11], ou du machine Learning [1]. L'emploi du Deep Learning est aussi envisageable, mais cela nécessite un grand jeu de données sur de nombreux individus, ce qui est très fastidieux à obtenir.

3 Simulateur de drone

3.1 Simulation physique

En premier lieu, il fallait établir un simulateur simple pour notre drone. Dans le cadre de ce stage, il s'agit d'un drone quadricoptère, dont les forces et couples générés par le robot ne dépendent que de la vitesse de rotation des hélices et de la géométrie du robot.

Dans notre cas, le drone réel a cette géométrie là, avec l'avant du drone dirigé vers le haut de la photo :



FIGURE 4 – photo du drone Crazyflie 2.1

Ce drone étant très similaire à celui étudié en cours [6], il nous suffit de reprendre ce qui a déjà été fait, en adaptant à notre situation.

3.1.1 Paramètres du modèle

Le modèle du quadrirotor est défini par les paramètres physiques suivants :

$m = 29$ g,	masse du drone,
$g = 9.81\ \mathrm{m\,s^{-2}}$,	accélération gravitationnelle,
$b = 1$,	coefficient de poussée,
$l = 46$ mm,	distance entre le centre du drone et un rotor,
$d = 0.1$,	coefficient de traînée.

On définit l'état de notre drone avec :

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \qquad \mathbf{v}_r = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \in \mathbb{R}^3 \qquad \boldsymbol{\omega}_r = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \in \mathbb{R}^3 \qquad \mathbf{R} \in SO(3)$$

La position des rotors dans le repère du robot est donnée par :

$$\mathbf{Q} = \frac{l}{2} \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

La force des rotors est toujours verticale dans le repère du robot, en orientant la verticale vers le bas, cela nous donne :

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

La vitesse de rotation des rotors est notée :

$$\mathbf{W} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}.$$

Ainsi, pour déterminer la force et le couple généré par le rotors, nous calculons :

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \mathbf{F} \begin{bmatrix} \omega_1|\omega_1| \\ \omega_2|\omega_2| \\ \omega_3|\omega_3| \\ \omega_4|\omega_4| \end{bmatrix} \qquad , avec : \mathbf{F} = \begin{bmatrix} b & b & b & b \\ -bl & 0 & bl & 0 \\ 0 & -bl & 0 & bl \\ -d & d & -d & d \end{bmatrix}$$

La force et le couple résultant deviennent donc :

$$\mathbf{F}_r = \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} \qquad \boldsymbol{\tau}_r = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}.$$

3.1.2 Evolution temporelle :

Le système inertiel évolue ainsi :

$$\dot{\mathbf{p}} = \mathbf{R}\,\mathbf{v}_r.$$

$$\dot{\mathbf{v}}_r = -\omega_r \wedge \mathbf{v}_r + \mathbf{R}^\top \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m}\mathbf{F}_r.$$

$$\dot{\mathbf{R}} = \mathbf{R}\omega_r\wedge$$

et, pour l'intégration discrète :

$$\mathbf{R}_{k+1} = \mathbf{R}_k \exp\left(\Delta t \omega_{r,k}\right).$$

$$\dot{\boldsymbol{\omega}}_r = \mathbf{I}^{-1}\left(-\omega_r \wedge \left(I\boldsymbol{\omega}_r\right) + \boldsymbol{\tau}_r\right).$$

3.2 Lois de commande

Le contrôleur est organisé en cascade. À partir de l'état désiré, il détermine une force et un couple permettant de rejoindre cet état [6] , ces deux informations permettent ensuite de remonter vers les vitesses de rotation des rotors produisant cette force et ce couple.

3.2.1 Contrôle en position :

L'erreur de position, exprimée dans le repère du drone, est définie par

$$\mathbf{e}_p = \mathbf{R}^\top(\mathbf{p}_d - \mathbf{p}).$$

L'erreur de vitesse est donnée par

$$\dot{\mathbf{e}}_p = \mathbf{R}^\top \dot{\mathbf{p}}_d - \mathbf{v}_r.$$

Le contrôleur calcule alors une accélération linéaire désirée dans le repère du drone :

$$\boxed{\dot{\mathbf{v}}_{r,d} = \mathbf{R}^\top \ddot{\mathbf{p}}_d - \boldsymbol{\omega}_r \wedge \mathbf{v}_r + 2\dot{\mathbf{e}}_p + \mathbf{e}_p}$$

À partir de cette accélération, la force désirée est obtenue par inversion de la dynamique du drone :

$$\boxed{\mathbf{F}_d = m\dot{\mathbf{v}}_{r,d} - m\mathbf{R}^\top \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + m\boldsymbol{\omega}_r \wedge \mathbf{v}_r}$$

3.2.2 Génération de l'attitude désirée :

l'axe vertical étant dirigé vers le bas, nous avons :

$$\mathbf{z}_d = -\frac{\mathbf{F}_d}{\|\mathbf{F}_d\|}.$$

L'axe $\mathbf{x}_d$ est ensuite obtenu en projetant l'axe $\mathbf{x}_0 = [1\ 0\ 0]^\top$ sur le plan orthogonal à $\mathbf{z}_d$:

$$\mathbf{x}'_d = \mathbf{x}_0 - (\mathbf{x}_0^\top \mathbf{z}_d)\mathbf{z}_d,$$

$$\mathbf{x}_d = \frac{\mathbf{x}'_d}{\|\mathbf{x}'_d\|}.$$

Le troisième axe est obtenu par

$$\mathbf{y}_d = \mathbf{z}_d \times \mathbf{x}_d.$$

L'attitude résultante est donc

$$\boxed{\mathbf{R}_F = \begin{bmatrix} \mathbf{x}_d & \mathbf{y}_d & \mathbf{z}_d \end{bmatrix}.}$$

La force de poussée des rotors étant verticale dans le repère du robot, il faut incliner le drone dans la direction où l'on souhaite qu'il se déplace afin que celui-ci converge correctement.

L'attitude désirée finale est obtenue en combinant cette orientation avec la consigne de lacet. Le roulis et le tangage sont pondérés par l'attitude issue de la force, tandis que le lacet est contrôlé par la consigne de lacet. Ce qui permet au drone de converger en postion et orientation simultanément.

3.2.3 Contrôle en attitude

L'erreur d'attitude est définie à partir de l'application logarithme sur $SO(3)$:

$$\mathbf{e}_R = \mathbf{R}^\top \log(\mathbf{R}_d \mathbf{R}^\top).$$

La vitesse angulaire désirée est obtenue à partir de

$$\boldsymbol{\omega}_d = (\dot{\mathbf{R}}_d \mathbf{R}_d^\top)^{\wedge -1}$$

L'accélération angulaire désirée est alors calculée par

$$\boxed{\dot{\boldsymbol{\omega}}_{r,d} = \left(\boldsymbol{\omega}_r \wedge \mathbf{R}_d^\top + \dot{\mathbf{R}}_d^\top\right) \boldsymbol{\omega}_d + \mathbf{R}_d^\top \dot{\boldsymbol{\omega}}_d + 2\dot{\mathbf{e}}_R + \mathbf{e}_R}$$

Le couple désiré est finalement donné par

$$\boxed{\boldsymbol{\tau}_d = \mathbf{I}\dot{\boldsymbol{\omega}}_{r,d} + \boldsymbol{\omega}_r \wedge \mathbf{I}\boldsymbol{\omega}_r.}$$

3.2.4 Allocation des commandes

La force et le couple désirés sont convertis en commandes pour les quatre rotors à l'aide de la matrice de commande $\mathbf{F}$:

$$\begin{bmatrix} -F_{d,z} \\ \tau_{d,x} \\ \tau_{d,y} \\ \tau_{d,z} \end{bmatrix} = \mathbf{F} \begin{bmatrix} \omega_1|\omega_1| \\ \omega_2|\omega_2| \\ \omega_3|\omega_3| \\ \omega_4|\omega_4| \end{bmatrix}.$$

La pseudo-inverse de $\mathbf{F}$ permet d'obtenir

$$\mathbf{W}_s = \mathbf{F}^+ \begin{bmatrix} -F_{d,z} \\ \tau_{d,x} \\ \tau_{d,y} \\ \tau_{d,z} \end{bmatrix}.$$

Enfin, les vitesses de rotation des rotors sont calculées par

$$\boxed{\omega_i = \mathrm{sgn}(W_{s,i})\sqrt{|W_{s,i}|}, \qquad i \in \{1,2,3,4\}.}$$

Pour s'approcher plus de la réalité, nous pourrions simuler les capteurs du drone, et utiliser un filtre de Kalman [7] pour qu'il puisse estimer son état à chaque instant. Cependant, comme notre drone réel intègre déjà toutes ces couches de calculs et ne se pilote qu'avec des commandes de direction (avant, arrière, haut, bas, etc...), il n'est pas nécessaire de complexifier plus notre simulation.

4 Interface d'enregistrement EEG

4.1 Travaux précédents

N'ayant initialement aucune connaissance sur le milieu l'EEG et très peu en ce qui concerne les réseaux de neurones, j'ai pu trouver beaucoup d'informations utiles dans les travaux d'autres chercheurs qui ont étés faits au laboratoire de cybernétique.

Dans le département de cybernétique de l'université, de nombreux travaux de recherches sont en cours sur différentes thématiques liées à l'EEG, ce qui permet d'avoir différents points de vues sur son utilisation.

Plusieurs travaux de recherches ont déjà été faits sur ce projet afin de contrôler un drone avec trois commandes distinctes [3] et appliquer cela à un essaim de drone [5] avec une très bonne fiabilité, cependant cela était limité à peu de commandes, afin d'avoir moins de variables sur un jeu de donnée relativement petit pour le Machine Learning.

Ces travaux nous donnent de très bonnes bases pour comprendre les méthodes d'enregistrement, de traitement des données et de distinctions entre différentes méthode de classification par Machine Learning [5].

Personnellement, je voulais explorer les méthode de classification à l'aide de réseaux de neurones, c'est donc ce que j'ai fait.

4.2 Choix des commandes du drone

Pour mon stage, j'étais relativement libre de choisir la direction à prendre, j'ai donc choisi de tenter de développer un contrôleur de drone par mesures EEG avec 7 commandes différentes, afin d'avoir plus de liberté de manœuvre. Cela ajoute également plus de difficultés par rapport à ce qui avait déjà été testé au laboratoire

Afin de commander un drone par imagerie motrice, il faut choisir les mouvements que l'on veut associer aux commandes du drone. Pour que le pilote reste libre d'un bon nombre de ses mouvements il ne faut pas que les mouvements sélectionnés soient aussi des mouvement de déplacement des jambes ou des yeux du pilote. Nous choisissons donc d'associer les mouvements des bras aux commandes du drone :

1. Repos : aucun mouvement des bras
2. Avancer : avancer les deux bras
3. Reculer : reculer les deux bras
4. Tourner à gauche : avancer uniquement le bras gauche
5. Tourner à droite : avancer uniquement le bras droit
6. Monter : monter les deux bras
7. Descendre : abaisser les deux bras

Ces différentes commandes sont relativement distinctes au niveau de l'activité cérébrale dans les bandes Alpha, Mu et Beta et par symétrie ou non dans le cortex, ce qui devrait théoriquement nous permettre de pouvoir les distinguer relativement facilement.

4.3 Protocole d'enregistrement

Afin d'enregistrer un jeu de donnée suffisant pour entraîner notre réseau de neurones, il nous faut un protocole d'enregistrement : Nous séparons nos données en dossiers correspondant aux différents sujets et sessions d'enregistrement. Chaque enregistrement dure 3 secondes et correspond à une des 7 classes décrites précédemment. Il est donc demandé au sujet de penser à faire une de ces 7 actions durant 3 secondes, puis ensuite d'associer l'enregistrement à une des classes, en appuyant sur une touche du clavier.

Le protocole commence par demander le nom du participant, afin de l'associer à un numéro unique pour l'anonymisation des données sur le réseau (un dictionnaire reste en local pour permettre de savoir quel nom est associé à quel nombre).

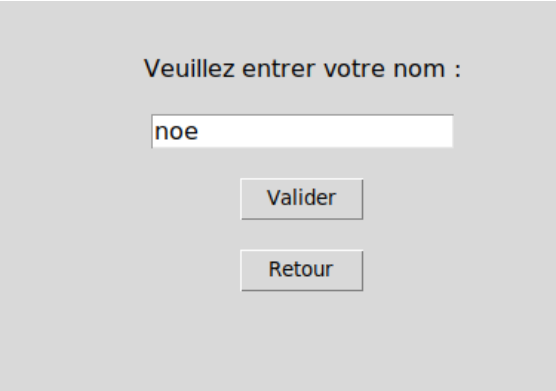


FIGURE 5 – Interface tkinter de demande de nom

Les consignes et attendus du protocole sont ensuite expliqués au participants, qui choisi quand il veut commencer un enregistrement.

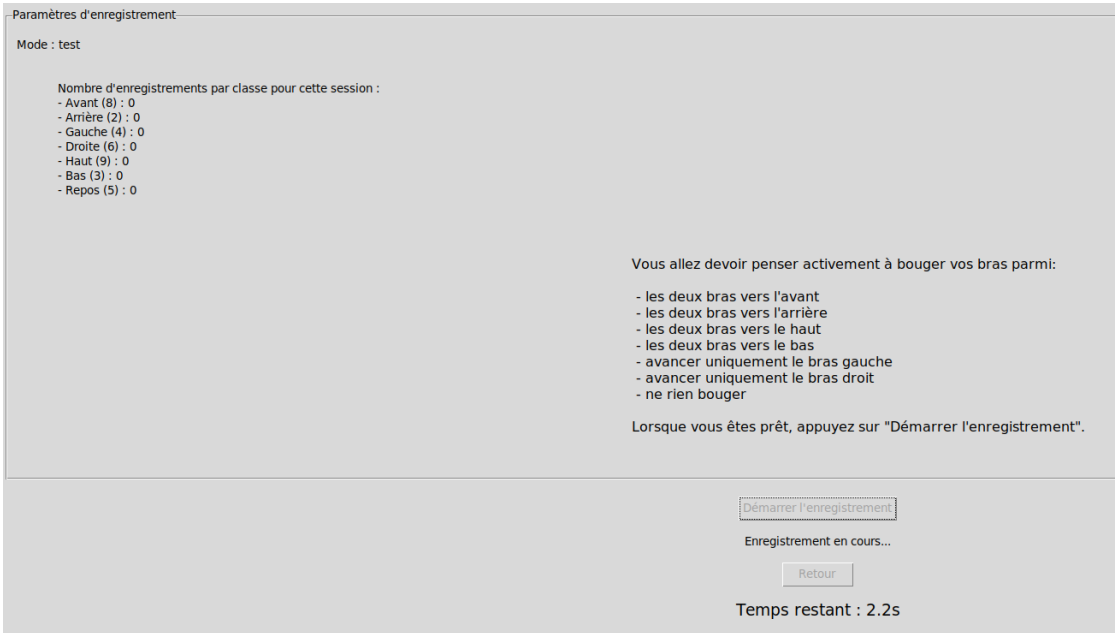


FIGURE 6 – Enregistrement des données

Ensuite, il est demandé au participant d'associer l'enregistrement à une des

classes. Afin de gagner du temps entre les enregistrements, il ne suffisait que de quelques appuis sur le clavier pour sélectionner la classe et passer à la suite.

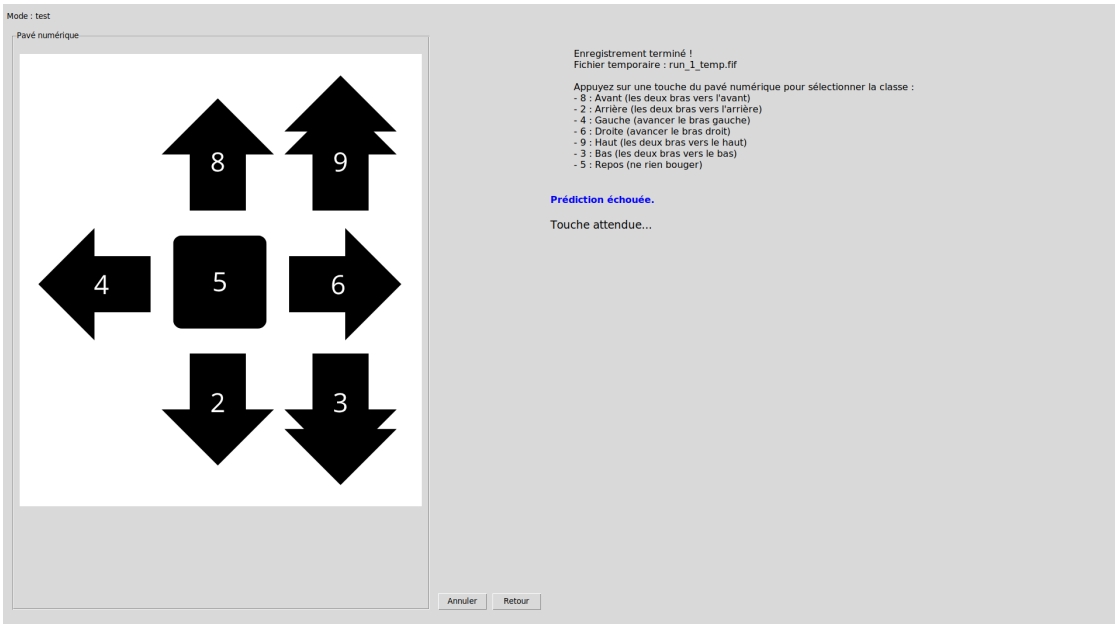


FIGURE 7 – Association des classes

Lors d'une session, il est demandé au participant d'enregistrer un nombre identique de classes, afin d'entraîner nos réseaux de neurones sur un jeu équilibré de données.

4.4 Connexion aux électrodes

Le casque à électrodes Mentalab se connecte en Bluetooth ou USB à un ordinateur équipé d'un logiciel dédié, disponible en open source. Elle permet de suivre l'impédance de chaque électrode pour ajuster le gel sur chaque électrode et ainsi avoir des données plus précises. L'application peut ensuite se connecter à notre code python pour l'enregistrement des données. La connexion est parfois instable, ce qui peut donner des enregistrements rognés ou plus longs par moments. Il arrive que la connexion soit perdue quelques instants, mais elle revient avec un peu de patience.

5 Réseaux de neurones

Une fois les données collectées, il nous faut un moyen de distinguer quels futurs enregistrement correspondent à quelle classe. Pour cela, on choisi d'utiliser un réseau de neurones entraîné sur les données collectés, pour classifier les données.

Nous pouvons lancer l'entraînement de réseaux de neurones directement avec les données brutes, mais cela risque d'être fastidieux et peu précis, car les réseaux auront plus de difficulté à identifier des critères de classification pertinents. Nous commençons donc à appliquer un pré-traitement à toutes nos données, afin de faciliter l'entraînement.

5.1 Pré-traitement des données

Nos données brutes collectées sont actuellement sous forme d'un tableau de données à 32 lignes (car 32 électrodes) et n colonnes (correspondant aux n prises de mesures temporelles de chaque enregistrement). Tout d'abord, nous devons réduire la longueur des fichiers trop grands, car ils contiennent des données qui pourraient être parasites. Ensuite, afin de faciliter l'identification, nous appliquons un filtre passe bande pour ne garder que les données entre 3 et 40 Hz. Ce qui permet de se focaliser sur les bandes d'activations qui nous intéressent (8-30 Hz), tout en restant ouvert à de potentielles information en dehors de ces fréquences.

Il ne faut pas oublier que chaque pré-traitement fait aux données d'entraînement devra être également applique aux données utilisées lors des essais par la suite. Comme nous souhaitons que le système reste réactif, nous devons garder notre pré-traitement des données relativement rapide, donc simple.

Pour avoir de la rapidité de commande, il nous faut ensuite décider à quelle fréquence nous souhaitons que le drone reçoive ses commandes. Une grande fréquence implique une petite fenêtre temporelle d'enregistrement EEG, donc moins d'informations pour la classification. Pour commencer, nous ferrons un compromis en prenant des fenêtres de 1 seconde, ce qui fait une fréquence de commande à 1 Hz. Si les résultats sont bons à cette fréquence, nous essayerons peut-être de la faire diminuer, sinon il faudra l'augmenter.

Par ailleurs, comme il était difficile d'obtenir des données en grande quantité, j'ai tenté d'augmenter artificiellement le jeu de données, en ajoutant y ajoutant des copies légèrement bruitées, mais cela n'a pas été concluant dans le temps qu'il me restait

5.2 Réseau de Neurones Convolutifs (CNN)

Le premier type de réseaux de neurones avec lequel je voulais expérimenter était le réseau de neurones convolutif (Convolutional Neural Network ou CNN en anglais) car il était relativement simple à implémenter et que c'était le seul réseau avec lequel j'étais un peu familier.

Les réseaux de neurones convolutifs utilisent un poids unique associé aux signaux entrant dans tous les neurones d'un même noyaux de convolution, ce qui permet d'avoir une empreinte mémoire plus faible que d'autres type de réseaux, tout en restant performants. Ces réseaux sont en général utilisés pour la classification d'images, de son ou de vidéos, ce qui n'est pas très différent de nos mesures

EEG quand on regarde le format de données (un tableau de valeurs variant avec le temps).

J'ai commencé par un réseau simple avec deux couches de convolution 1D et essayé de modifier petit à petit les paramètres pour obtenir de meilleurs résultats lors des sessions d'entraînement. Cependant, je me suis vite rendu compte que chercher les meilleurs paramètres par essai-erreur à la main serait une perte de temps et d'énergie.

5.3 Optimisation des hyperparamètres

Heureusement, le package tensorflow de python permet justement de rechercher les meilleurs hyperparamètres d'un modèle de réseau de neurones. Cela se fait en entraînant des modèles, avec des hyperparamètres variables, ce qui influe directement sur leurs performances. Il nous suffit alors de les comparer et de ne garder que le meilleur. Pour se faire, nous devons définir une zone de recherche pour les hyperparamètres : des bornes maximums et minimums ou une liste des valeurs à tester. A partir de là, il ne reste plus qu'à laisser le programme chercher les hyperparamètres donnant le réseau le plus précis dans la zone choisie.

Nous pouvons par la suite affiner le résultat en réduisant notre zone de recherche autour du meilleur résultat trouvé.

5.4 Un réseau de neurones adapté à nos données : l'EEGNet

Le CNN classique est plutôt bien adapté pour saisir des motifs spatiaux, mais il peine un peu plus pour ce qui est des motifs temporels. Or, les signaux EEG contiennent des informations essentielles sur le plan spatial (zones d'activations du cerveau) et sur le plan temporel (fréquence d'activation). C'est justement pour pouvoir extraire ces différentes composantes qu'a été développé le réseau de neurones EEGNet, qui est une variante du réseau de neurones convolutif. Plutôt que d'empiler des couches de convolution 1D, l'EEGNet utilise différents types de couches, afin de distinguer plus précisément les motifs particuliers des signaux EEG. [8]

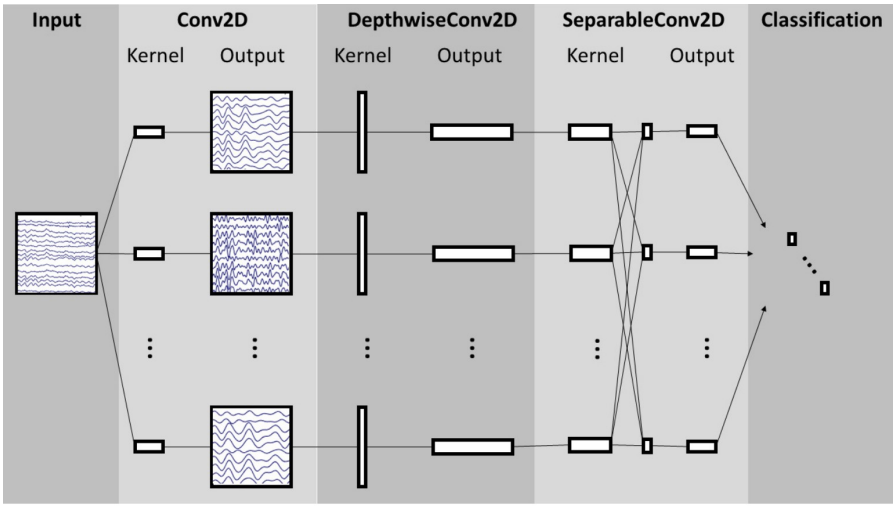


FIGURE 8 – Visualisation de l'architecture du réseau EEGNet

L'emploi de couches de convolution en deux dimensions, en longueur permet

de distinguer les motifs temporels et donc fréquentiels, tandis que ceux en largeur permettent une distinction spatiale des signaux. La combinaison de ces couches de convolution spécialisées permet de classifier plus précisément des signaux de type EEG, avec une architecture beaucoup plus compacte qu'un CNN classique, pour de meilleures performances.

Afin de voir si la taille du réseau de neurones (en termes de couches surtout) influe sur la précision, nous testons des versions étendues de nos réseaux.

5.5 Résultats

Comme le matériel d'enregistrement était partagé pour différents projets et que les enregistrements nécessitaient beaucoup de temps, je n'ai pas pu réaliser beaucoup de mesures pour entraîner les réseaux de neurones, ce qui a grandement limité les performances de ce projet.

modèle	CNN	EEGNet	EEGNet étendu
Précision	0.24716	0.21875	0.24715

Nos modèles sont légèrement plus précis que le hasard (0.1428), mais pas suffisamment pour êtres utilisés concrètement

Étonnamment, le modèle EEGNet a moins bien performé que le CNN, probablement à cause du manque de données et de l'adaptabilité plus souple du CNN. Quant à l'EEGNet étendu, celui-ci rattrape les performances du CNN, tout en étant un modèle bien plus réduit, ce qui démontre les performances d'un réseau de type EEGNet pour la classification de nos données EEG avec une architecture plus légère. Voir l'annexe pour plus de détails.

Les résultats sont cependant encourageant compte tenu de la taille de notre jeu de données. Avec plus de temps, nous aurions pu réaliser plus d'enregistrements, avec plus de sessions d'entraînement et de sujets divers, pour pouvoir obtenir un classificateur qui puisse être plus précis et plus souple.

6 Pilotage du simulateur par mesures EEG

Pipeline des données Afin de s'entraîner au pilotage du drone, il nous faut lier le simulateur de vol avec l'enregistrement de données EEG et le réseau de neurones. Pour cela, nous définissons deux threads : un pour le simulateur de vol et un pour réceptionner les données et les interpréter en commandes pour le simulateur. Les commandes sont stockées dans une file de priorité pouvant stocker un nombre limité de commandes au cas où il y a une désynchronisation des threads.

Système de Contrôle par EEG

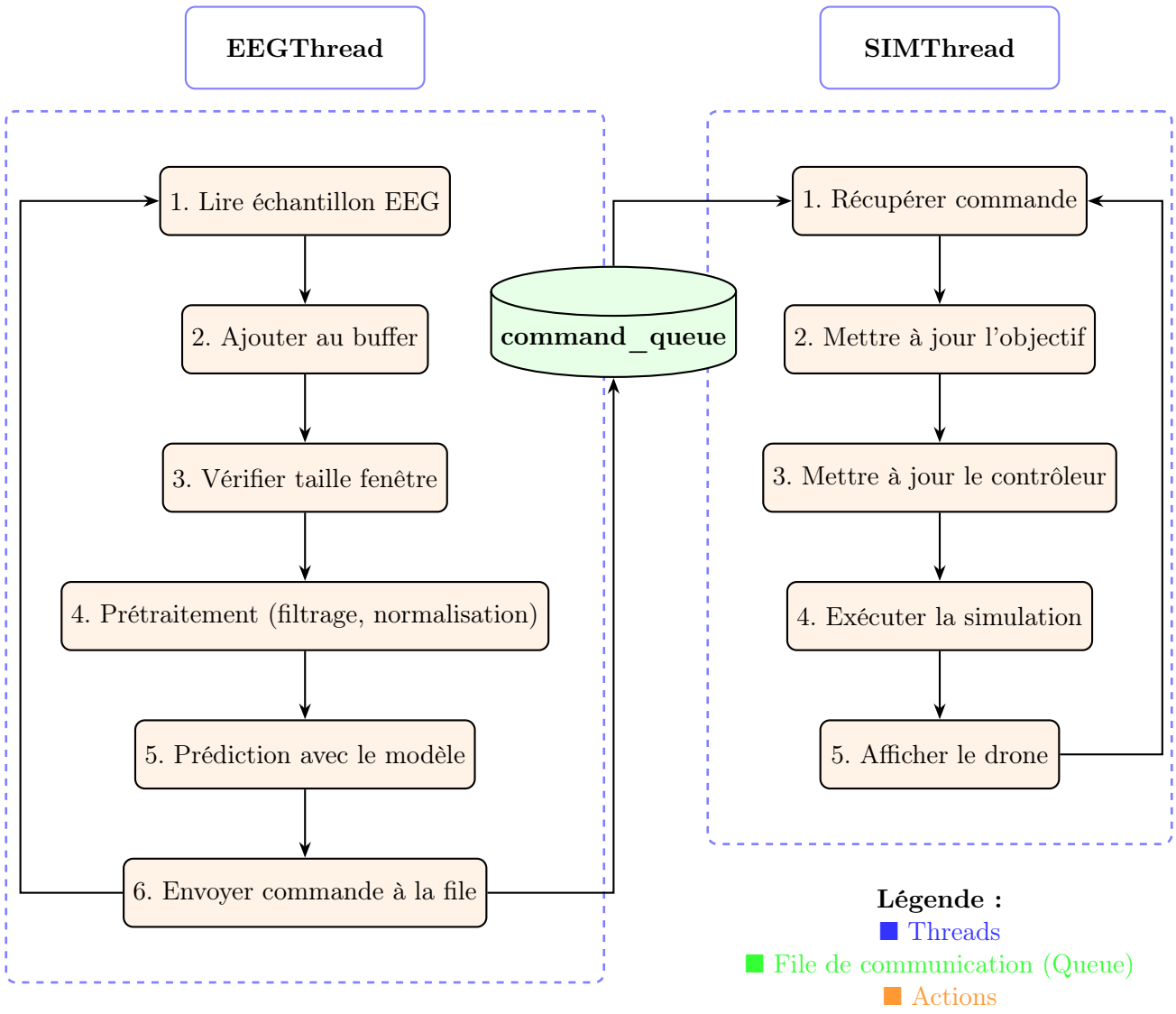


FIGURE 9 – Graphe des interactions entre les threads

Cette architecture n'a pas pu être testée avant la fin de mon stage car nos réseaux de neurones n'étaient pas assez précis.

7 Conclusion

Ce stage à l'Université Norvégienne de Science et Technologie avait pour but de développer une interface cerveau-machine permettant de piloter un drone à l'aide de mesures EEG. Le projet a été décomposé en plusieurs étapes : la mise en place d'un simulateur de vol de drone piloté en position, la définition des patterns d'imagerie motrice associés aux commandes du drone, la conception d'un protocole d'enregistrement des signaux EEG, l'entraînement de réseaux de neurones pour classifier ces signaux et l'association du contrôle inertiel du drone avec les commandes déduites des signaux EEG.

Le simulateur de vol, basé sur un modèle physique de quadrirotor avec un contrôleur inertiel, a été mis en place et fonctionne correctement. De même, le protocole d'enregistrement a permis de constituer un premier jeu de données, sur lequel différents types de réseaux de neurones (CNN, EEGNet et EEGNet étendu) ont pu être entraînés et comparés. Les précisions obtenues, bien que modestes, sont encourageantes compte tenu de la taille très limitée du jeu de données : le manque de mesures, dû au partage du matériel d'enregistrement entre plusieurs projets et au temps nécessaire à chaque session, reste la principale limite de ce travail. C'est également ce manque de données qui a empêché de tester en conditions réelles l'architecture complète reliant l'enregistrement EEG, le réseau de neurones et le simulateur.

De nombreuses perspectives s'ouvrent donc à la suite de ce stage. La priorité est de constituer un jeu de données plus important et plus varié, en multipliant les sujets et les sessions d'enregistrement, afin d'améliorer la précision de la classification. Une fois des réseaux suffisamment fiables obtenus, le pipeline complet pourra être testé sur le simulateur, puis sur le drone réel. À plus long terme, il serait intéressant de réduire la fenêtre temporelle de classification pour augmenter la réactivité du système, ou encore d'explorer des méthodes d'apprentissage adaptatives pour prendre en compte les variabilités inter- et intra-sujets des signaux EEG.

Au-delà des résultats techniques, ce stage m'a permis d'acquérir des compétences solides en traitement du signal, et en machine learning, mais aussi de découvrir le déroulement réel d'un projet de recherche, qui ne se passe jamais exactement comme prévu et exige des ajustements permanents. Travailler dans un environnement international, au sein d'une équipe de doctorants et de chercheurs, m'a également appris à communiquer et à collaborer efficacement sur un sujet complexe. Cette expérience conforte mon envie de poursuivre dans les domaines de la robotique autonome et de l'intelligence artificielle, et je remercie tous les membres du laboratoire de cybernétique pour leur accueil et leur accompagnement tout au long de ce stage.

Références

[1] Kjersti Brynestad et Erlend Vatsvåg. « An Asynchronous Motor Imagery-based Brain-Computer Interface for Two-dimensional Drone Control ». en. In : ().

[2] Kobe Christensen et Jim Hutchins. « Postsynaptic Potentials (PSPs) : The Source of EEG Signal ». en. In : (sept. 2024). Book Title : Advanced Neuroscience. URL : https://uen.pressbooks.pub/advneuro/chapter/postsynaptic-potentials-psps-the-source-of-eeg-signal/.

[3] Arvin Das. « Offline Evaluation of an EEG-Based Motor Imagery Brain-Computer Interface for Drone Control ». en. In : ().

[4] *Facts and figures - NTNU*. URL : https://www.ntnu.edu/facts.

[5] Daniel Nicolai Grønn. « A Modular Brain-Computer Interface Frame- work for Autonomous Swarm Drone Navigation ». en. In : ().

[6] Luc Jaulin. « Mobile robotics : Inertial https ://www.ensta-bretagne.fr/inmooc/ ». en. In : (2025).

[7] Luc Jaulin. « Mobile robotics : Kalman Filter https ://www.ensta-bretagne.fr/kalmooc/ ». en. In : (2023).

[8] Vernon J. Lawhern et al. « EEGNet : A Compact Convolutional Network for EEG-based Brain-Computer Interfaces ». en. In : (mai 2018). URL : https://www.alphaxiv.org/fr/abs/1611.08024 (visité le 28/08/2026).

[9] *Mobile EEG Systems | Mentalab | High Precision for Research*. en-US. URL : https://mentalab.com/.

[10] Simanto Saha et Mathias Baumert. « Intra- and Inter-subject Variability in EEG-Based Sensorimotor Brain Computer Interface : A Review ». English. In : *Frontiers in Computational Neuroscience* 13 (jan. 2020). ISSN : 1662-5188. DOI : 10.3389/fncom.2019.00087. URL : https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2019.00087/full.

[11] Aqsa Shakeel et al. « A Review of Techniques for Detection of Movement Intention Using Movement-Related Cortical Potentials ». en. In : *Computational and Mathematical Methods in Medicine* 2015 (2015), p. 1-13. ISSN : 1748-670X, 1748-6718. DOI : 10.1155/2015/346217. URL : http://www.hindawi.com/journals/cmmm/2015/346217/ (visité le 18/08/2026).

8 Annexe

8.1 Détail des meilleurs modèles obtenus

CNN

Paramètres	valeur
Dropout Rate	0.6
Conv1 filters	48
Conv1 kernel size	3
Conv2 filters	224
Conv2 kernel size	3
Conv3 filters	176
Conv3 kernel size	7
Conv4 filters	160
Conv4 kernel size	3
Conv5 filters	432
Conv5 kernel size	5
Conv6 filters	432
Conv6 kernel size	7
Dense units	416
learning rate	0.0013752
Précision	0.24716

EEGNet

Paramètres	valeur
Dropout Rate	0.68
Kernel Length	24
F1	80
F2	96
learning rate	0.00037201
Précision	0.21875

EEGNet étendu

Paramètres	valeur
Dropout Rate	0.65
Kernel Length	24
F1	16
F2	16
layer 1 filter	32
layer 1 kernel	16
layer 2 filter	16
layer 2 kernel	8
layer 3 filter	32
layer 3 kernel	32
layer 4 filter	64
layer 4 kernel	8
layer 5 filter	32
layer 5 kernel	16
learning rate	0.007775
Précision	0.24715

RAPPORT D'EVALUATION
ASSESSMENT REPORT

Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :
At the end of the internship, please return this report via mail or email to:

ENSTA – Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE
☎ 00.33 (0) 2.98.34.87.70 / stages.brest@ensta.fr

I - ORGANISME / *HOST ORGANISATION*

NOM / *Name* : Université Norvégienne de Science et Technologie (NTNU) _____

Adresse / *Address* : Høgskoleringen 6, 7491 Trondheim, Norvège _____
_____

Tél / *Phone (including country and area code)* +47 47 23 19 49 _____

Nom du superviseur / *Name of internship supervisor :* Mrs Marta Molinas_____

Fonction / *Function* : Professeur d'ingénierie cybernétique _____

Adresse e-mail / *E-mail address* : marta.molinas@ntnu.no _____

Nom du stagiaire accueilli / *Name of intern* Noé Carrayrou

II - EVALUATION / *ASSESSMENT*

Veuillez attribuer une note, en encerclant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**
*Please attribute a mark from **A (excellent)** to **F (very weak).***

MISSION / *TASK*

❖ La mission de départ a-t-elle été remplie ? A B C D E F
Was the initial contract carried out to your satisfaction?

❖ Manquait-il au stagiaire des connaissances ? ☒ oui/*yes* ☐ non/*no*
Was the intern lacking skills?

Si oui, lesquelles ? / *If so, which skills?* *Machine Learning* _____

ESPRIT D'EQUIPE / *TEAM SPIRIT*

❖ Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / *Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work)*

A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* *he adapted well*_____
_____

Version du 05/04/2019

COMPORTEMENT AU TRAVAIL / *BEHAVIOUR TOWARDS WORK*

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?
Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)?

A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here he was punctual*_____

_____

INITIATIVE – AUTONOMIE / *INITIATIVE – AUTONOMY*

Le stagiaire s'est –il rapidement adapté à de nouvelles situations ? A B C D E F
(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

Did the intern adapt well to new situations? A B C D E F
(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here some solutions were suggested*_____

_____

CULTUREL – COMMUNICATION / *CULTURAL – COMMUNICATION*

Le stagiaire était-il ouvert, d'une manière générale, à la communication ? A B C D E F
Was the intern open to listening and expressing himself /herself?

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here he was open*_____

_____

OPINION GLOBALE / *OVERALL ASSESSMENT*

❖ La valeur technique du stagiaire était : A B C D E F
Please evaluate the technical skills of the intern:

III - PARTENARIAT FUTUR / *FUTURE PARTNERSHIP*

❖ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?

Would you be willing to host another intern next year? ☒ oui/*yes* ☐ non/*no*

Fait à _____, le _____
In Trondheim, 21-09-2026 _____, on _____

Company stamp  _____*Intern's signature* 

Merci pour votre coopération
We thank you very much for your cooperation