

THALES



Projet de Fin d'Études

Mémoire

---

# Environnement Immersif Opérateur

---

*Auteur :*

M. Yoann SOLA

*Encadrant :*

M. Pierre LABORDE

*Jury :*

Pr. Laurent HARDOUIN

Pr. Luc JAULIN

Pr. Benoît CLÉMENT

Dr. Gilles LE CHENADEC

Version du  
24 août 2017



---

**Résumé** — Ce Projet de Fin d'Étude est né d'un besoin commun entre le service Interface Homme-Machine (IHM) et le service Soutien et Service Client (CSS) de Thales Systèmes Aéroportés : mettre au point un système immersif qui permette à la fois de tester l'ergonomie d'une IHM et de mesurer la performance de son utilisateur, dans un but de formation. Ce système, appelé Environnement Immersif Opérateur (EIO), doit répondre à de multiples exigences : être facilement déployable, être non-intrusif, être capable de suivre des scénarios de formation de natures variées, et offrir des moyens de contrôle, de supervision et de rejeu aux formateurs. Nous avons choisi d'utiliser les missions de patrouille maritime comme premier contexte de travail. Une multitude d'équipements ont été mis en œuvre afin de rendre nos formations plus réalistes : un siège dynamique, des moyens de communication au sein de l'équipage, un système sonore, un chauffage soufflant, ou encore un système de lumière connectée. Nous avons également utilisé le logiciel Sensor Hub afin de mesurer la performance de la personne formée. Celui-ci est capable de calculer l'OFS (Operator Functional State), un indicateur qui nous permet de connaître les moments où la personne a été dépassée par les événements et où il faudrait éventuellement retravailler l'IHM qui a été utilisée dans le scénario. Nous avons adopté une architecture logicielle basée sur des composants réutilisables et permettant une meilleure maintenabilité du système. Enfin nous avons pu éprouver l'EIO au moyen d'une expérience réalisée lors de notre rencontre avec une équipe du CESIM (CEntre de SIMulation) Santé de Brest. Cette expérience nous a permis de conclure que l'EIO remplissait une grande partie des exigences formulées initialement.

**Mots clés** : Formation, interface homme-machine, simulation, patrouille maritime, mesure de la performance, composant logiciel.

---

Thales Systèmes Aéroportés  
10 Avenue 1ère Dfl  
29200 Brest

ENSTA Bretagne  
2 Rue François Verny  
29200 Brest



# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Contexte et scénarios de formation</b>	<b>3</b>
1.1 La patrouille maritime . . . . .	3
1.2 Le système Amascos <sup>©</sup> . . . . .	4
1.3 Les scénarios de formation . . . . .	6
<b>2 Rendre une formation plus réaliste</b>	<b>9</b>
2.1 Reproduire les mouvements de l'avion . . . . .	9
2.1.1 Le logiciel de simulation . . . . .	9
2.1.2 Le poste de pilotage . . . . .	10
2.1.3 Le siège dynamique . . . . .	12
2.2 Permettre des communications internes et externes à l'équipage . . . . .	14
2.3 Gérer la température ressentie par la personne formée . . . . .	15
2.4 Créer une ambiance lumineuse convaincante . . . . .	16
2.5 Reproduire des odeurs réalistes . . . . .	17
<b>3 Mesure de la performance</b>	<b>19</b>
3.1 Les moyens usuels de mesure de la performance . . . . .	19
3.2 L'OFS : un moyen plus avancé de mesure de la performance . . . . .	21
3.3 Sensor Hub : un outil de supervision de l'OFS . . . . .	22
3.3.1 Présentation générale de Sensor Hub . . . . .	22
3.3.2 Les capteurs physiologiques présents dans l'EIO . . . . .	24
3.3.3 La charge cognitive : un exemple de modèle de Dimension . . . . .	32

---

3.3.4	Calcul de l'OFS . . . . .	38
<b>4</b>	<b>Intégration matérielle et logicielle</b>	<b>41</b>
4.1	Intégration matérielle . . . . .	43
4.1.1	Le poste de supervision . . . . .	43
4.1.2	Résumé de la configuration matérielle de l'EIO . . . . .	45
4.2	Intégration logicielle . . . . .	47
4.2.1	Le framework Naja : la programmation orientée composant . . . . .	47
4.2.2	Langages et protocoles utilisés . . . . .	50
<b>5</b>	<b>Expériences</b>	<b>53</b>
5.1	Contexte de l'expérience . . . . .	53
5.1.1	Le CESIM Santé de Brest . . . . .	53
5.1.2	La démonstration de l'EIO . . . . .	54
5.2	Résultats de l'expérience . . . . .	57
	<b>Conclusion</b>	<b>59</b>
	<b>Bibliographie</b>	<b>62</b>
	<b>Liste des sigles et acronymes</b>	<b>63</b>
	<b>Liste des figures</b>	<b>66</b>
	<b>Liste des tables</b>	<b>67</b>

# Introduction

Ce Projet de Fin d'Étude a été effectué au sein du service IHM<sup>1</sup> de Thales Systèmes Aéroportés, à Brest. Il est né d'un besoin commun entre ce service et le service CSS<sup>2</sup> : mettre au point un système immersif qui permette à la fois de tester l'ergonomie d'une IHM et de mesurer la performance de son utilisateur, dans un but de formation.

En effet, une des principales missions du service CSS est la formation des futurs utilisateurs de systèmes vendus par Thales. L'idée d'un simulateur permettant de reproduire des situations réalistes apporterait donc un plus à ces formations. Les activités de Thales étant fortement liées au domaine de la défense, les personnes formées par le service CSS sont souvent plongés dans un contexte militaire. Le système devra donc également reproduire la nature stressante de ces situations.

Le simulateur devra également servir d'outil de conception pour le service IHM. Ce service a pour vocation de développer des IHM intuitives, innovantes et qui nécessitent d'avoir régulièrement des retours d'utilisateurs. Le système devra donc être capable d'effectuer des mesures d'ergonomie au cours de son utilisation.

En résumé, le système immersif devra répondre à de multiples exigences :

- Être facilement déployable.
- Être capable de suivre des scénarios de formation de natures variées : patrouille maritime, guerre électronique, etc.
- Fournir une expérience réaliste de simulation aux personnes formées.
- Mettre progressivement les personnes formées dans une situation de stress, afin de les préparer au mieux à la réalité du terrain.
- Offrir des moyens de contrôle et de supervision aux formateurs.
- Effectuer des mesures de performances sur les personnes formées et être capable de fournir un compte-rendu à la fin de leur formation.
- Effectuer des mesures physiologiques sur les personnes formées de manière non-intrusive.
- Effectuer des mesures d'ergonomie sur les différentes IHM mises en œuvre.

---

1. Interface Homme-Machine

2. Soutien et Service Client

Un premier stage avait été effectué en 2016 par Florian VOGEL afin d'étudier le sujet. Florian a ainsi mis au point des scénarios de formation à la patrouille maritime, en se basant sur des enquêtes qu'il avait effectué auprès d'anciens opérationnels. Cela lui a permis de mettre en avant les facteurs d'immersion sur lesquels ses scénarios devront s'appuyer. Il a également effectué des recherches approfondies sur les sources de stress pouvant intervenir durant les missions de patrouille maritime, et sur la manière de les reproduire durant les formations. Il a enfin dressé une première liste du matériel nécessaire au futur simulateur, en se basant sur les enquêtes et études effectuées durant son stage.

Suite à cela, le service IHM se dota de nombreux équipements devant encore être testés et évalués afin de justifier leur présence dans le simulateur, appelé désormais l'« Environnement Immersif Opérateur (EIO) ».

L'objectif de mon stage est donc de construire une première version de cet EIO. Cela implique la réalisation de la partie matérielle, mais également logicielle du simulateur. Au niveau de la partie matérielle, il est nécessaire de tester chaque équipement séparément afin d'évaluer leur pertinence au sein du simulateur. Il faut ensuite réaliser l'intégration de tout ce matériel, en veillant à ce que l'ensemble des capteurs et actionneurs permette de placer l'EIO dans différentes configuration : il doit avoir une architecture modulaire. Pour la partie logicielle, un programme offrant les fonctionnalités suivantes doit être développé<sup>3</sup> :

- Contrôler tous les équipements et les faire communiquer ensemble.
- Permettre de charger et lancer des scénarios de formation.
- Superviser la formation et la personne formée, et permettre au formateur d'intervenir dans le déroulement du scénario.
- Acquérir et enregistrer toutes les informations, et pouvoir rejouer a posteriori des sessions de formation.
- Effectuer une synthèse de la session de formation et produire des statistiques à propos de la performance de la personne formée, mais également au sujet de l'ergonomie de l'IHM manipulée durant la formation.

Dans ce rapport de stage, je commencerai par présenter le contexte et les scénarios de formation utilisés avec l'EIO, puis j'expliquerai les moyens que nous avons utilisés pour rendre nos formations plus réalistes. J'aborderai ensuite les différentes manières de mesurer la performance d'un individu, avant de détailler l'intégration matérielle et logicielle. Enfin je terminerai par décrire les expériences que j'ai pu réaliser durant mon stage.

---

3. Ces exigences représentent des spécifications plus précises que celles énoncées précédemment (qui offraient un point de vue plus global de l'EIO).



# Chapitre 1

## Contexte et scénarios de formation

Nous allons dans ce chapitre présenter le contexte dans lequel se déroulent les scénarios de formation de l'EIO.

### 1.1 La patrouille maritime

Les avions de patrouille maritime ont pour missions de repérer et lutter depuis le ciel contre d'éventuels menaces sur zone.

Leurs missions peuvent être à caractère civil ou militaire et les menaces peuvent être de nature aérienne, marine (en surface) et sous-marine.

Les avions de patrouilles maritimes peuvent être des porteurs moyens (comme le Falcon 50) ou des porteurs lourds (comme l'Atlantique 2, représenté sur la Figure 1.1).



FIGURE 1.1 – Un avion Atlantique 2 (ATL 2) de la marine nationale effectuant une mission de patrouille maritime en haute altitude.

Leurs missions consistent à patrouiller à proximité des côtes et des zones d'intérêts économiques afin d'analyser le trafic maritime et aérien.

Ils doivent donc identifier la provenance et la nature des bateaux, mais également veiller à la sécurité des biens et des personnes. Ils sont ainsi amenés à rechercher des personnes ou des navires naufragés, ou encore à organiser l'assistance ou le secours de navires en détresse. Ils doivent aussi surveiller et anticiper les actes de piraterie, lutter contre le déchargement en mer de produits pouvant nuire à l'écologie maritime et contrôler les activités de pêche.

Pour remplir cet ensemble varié d'activités, les avions de patrouille maritime embarquent à leur bord de nombreux équipements : RADAR<sup>1</sup>, caméra EOIR<sup>2</sup>, détecteur de RADAR ou RESM<sup>3</sup>, AIS<sup>4</sup>, etc.

## 1.2 Le système Amascos<sup>©</sup>



FIGURE 1.2 – Une console Amascos Touch.

Amascos<sup>©</sup> est une solution IRS développé par Thales : Intelligence, Reconnaissance et Surveillance.

1. RAdio Detecting And Ranging : Détection et calcul de la distance par ondes radioélectriques.
2. Electro-Optic InfraRed : Optique électronique infrarouge.
3. Radar Electronic Support Measures : Mesures de soutien électronique et radar.
4. Automatic Identification System : Système d'identification automatique.

Il s'agit d'un système multimissions intégré, modulaire, évolutif et adaptable. Il est connecté à l'ensemble des capteurs de l'avion et permet de nombreux traitements : radar, optronique, acoustique, etc. Il dispose d'un système d'exploitation lui permettant d'enregistrer toutes les données et gérer les télécommunications.

La console Amascos Touch (voir Figure 1.2) est le système de contrôle et de commande (système C2) de Amascos, et permet donc de manipuler les fonctionnalités de l'ensemble des capteurs de l'avion.

Sur la Figure 1.3, nous pouvons observer des captures d'écran de l'IHM de Amascos Touch. Elle est composée de deux écrans tactiles disposés verticalement et possède des menus circulaires permettant de naviguer facilement parmi les fonctionnalités du système.

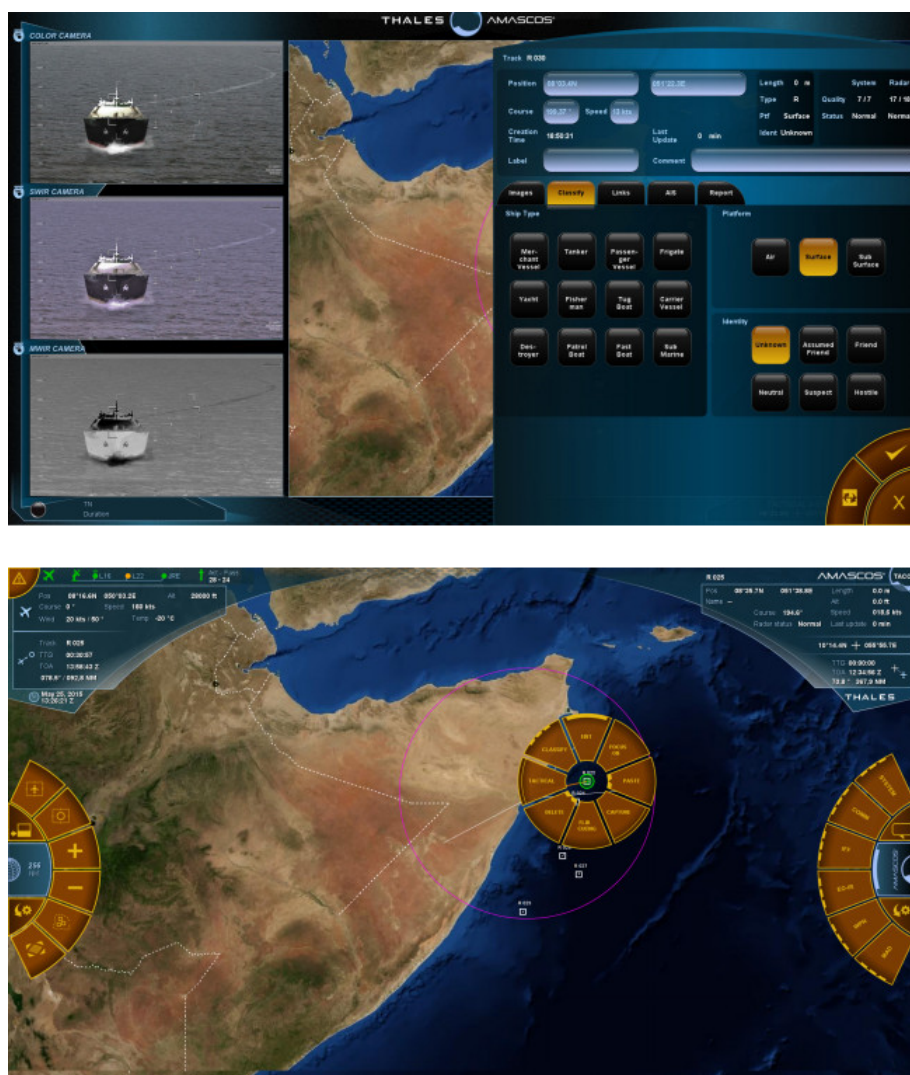


FIGURE 1.3 – Captures d'écran de Amascos Touch : écran du haut et écran du bas.

La personne chargée de mettre en œuvre Amascos Touch est le « Sensor System Operator (SSO) », aussi appelé parfois le « radariste ». Il a pour rôle d'identifier l'ensemble des véhicules rencontrés.

Le SSO est dirigé par le « Tactical Commander (TACCO) ». Ce dernier est le chef de la mission et représente l'opérateur indispensable à toute mission de patrouille maritime, qui a pour tâche de garder à jour la situation tactique de la mission et de suggérer des itinéraires au pilote de l'avion. Ils se basent pour cela sur le travail effectué par le SSO.

### 1.3 Les scénarios de formation

Actuellement, l'EIO utilise des scénarios de patrouille maritime.

En effet la personne formée, aussi appelé le sujet de la simulation, prend le rôle d'un SSO durant une mission de patrouille maritime.

Nous supposons dans ces scénarios que le SSO dispose des compétences nécessaires pour également assurer le rôle de TACCO.

Le sujet va ainsi devoir effectuer plusieurs tâches au cours du scénario : identifier des véhicules de tout type, détecter un sous-marin suspect, organiser le sauvetage d'un navire, suggérer des itinéraires, etc.

La formation fait intervenir d'autres personnes :

- Le formateur : Il observe et mesure les performances et les actions du sujet de la simulation. Il a également le pouvoir de modifier à la volée le scénario de la formation.
- Le pilote de l'avion : Il s'agit d'un acteur dont le rôle est faire suivre un certain itinéraire à l'avion de la simulation. Il doit également communiquer en permanence avec le SSO, afin de s'approcher de certains véhicules en particulier.
- Des acteurs supplémentaires : Ce sont des personnes qui se situeront dans une pièce à part et qui auront pour rôle de répondre aux sollicitations du SSO. En effet, la personne formée sera amenée à interroger certains navires par radio afin d'obtenir des informations que ne peuvent pas fournir les capteurs de l'avion. Les acteurs devront alors interpréter les équipages de ces navires.

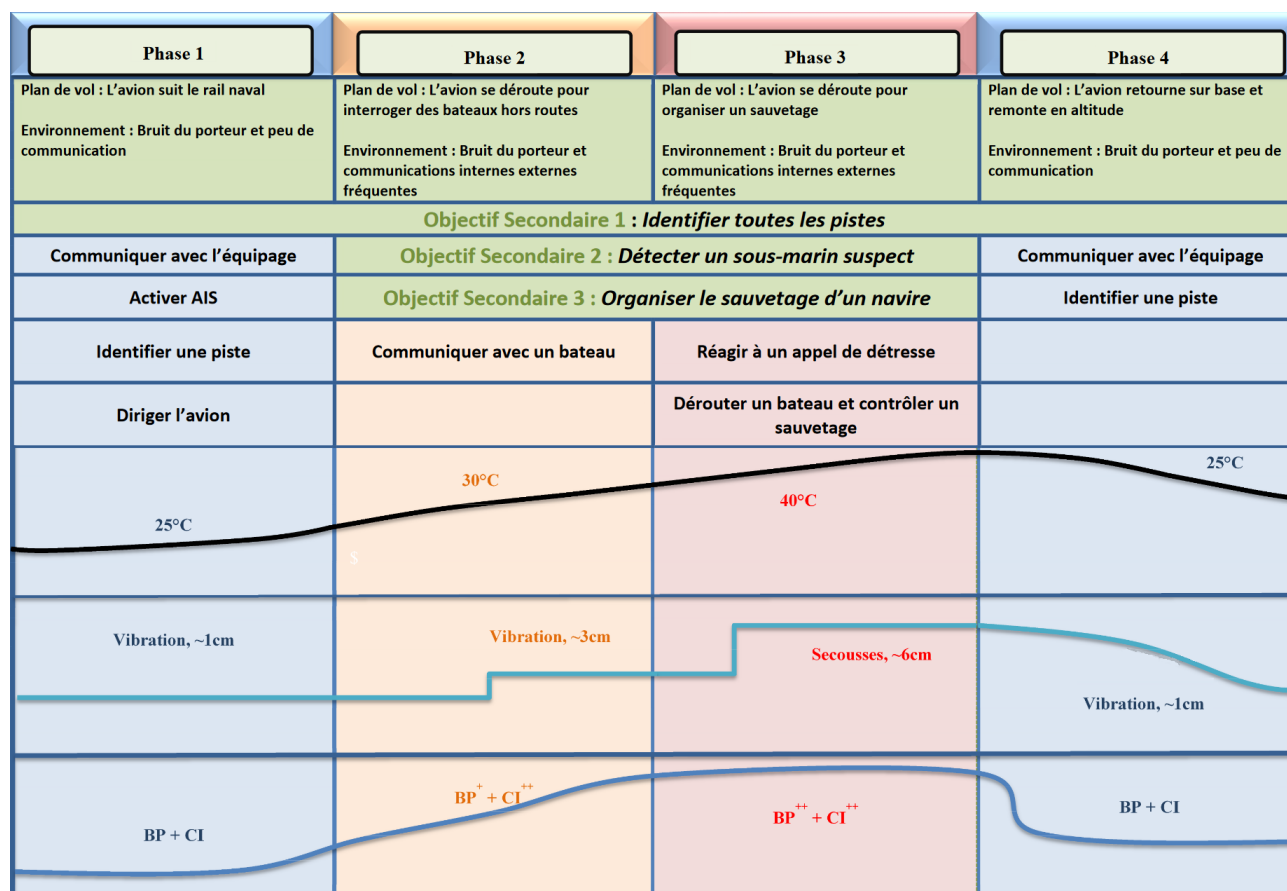


FIGURE 1.4 – Exemple de scénario de patrouille maritime. BP signifie « Bruit du porteur », et CI signifie « Communications internes ».

Les scénarios de formation comportent plusieurs phases (illustrées sur la Figure 1.4) :

- Tout d'abord, le scénario demande peu de tâches à effectuer au SSO, permettant à ce dernier de s'habituer aux fonctionnalités de Amascos Touch.
- Progressivement, de plus en plus de tâches sont demandées simultanément au SSO, augmentant ainsi sa charge cognitive. Les conditions expérimentales se durcissent également : la chaleur augmente, les vibrations et le bruit ambiant s'intensifient, et les communications internes sont plus fréquentes.
- Puis le scénario atteint son paroxysme, confrontant ainsi le SSO à un environnement hostile. C'est à ce moment-là que nous pouvons observer si la personne formée conserve de bonnes performances même en situation de stress et de fatigue mentale.
- Enfin, au bout d'un certain temps, le scénario devient de plus en plus « calme » et les conditions expérimentales reviennent à la normale. Cela permet de ramener progressivement la personne formée à la réalité et d'achever la séance de formation.

Bien que l'EIO ait été principalement déployé pour des scénarios de patrouille maritime, ce dernier a été conçu pour fonctionner avec des scénarios de tout type et faisant intervenir des véhicules aériens, terrestres, marins ou sous-marins.

En effet, il suffirait pour cela de changer simplement la configuration des équipements décrits dans les chapitres suivants.

# Chapitre 2

## Rendre une formation plus réaliste

Nous allons nous intéresser dans ce chapitre à la manière dont nous avons rendu les formations plus réalistes grâce à l'EIO.

### 2.1 Reproduire les mouvements de l'avion

#### 2.1.1 Le logiciel de simulation

Lors des premiers développements de l'EIO, lorsque nous testions l'ensemble du matériel dont nous disposions, nous avons commencé par nous servir du logiciel de simulation de vol X-Plane (voir Figure 2.1 ci-après).

Cette simulation de vol est notamment réputée pour disposer d'un moteur physique très réaliste et performant, et pour proposer un grand nombre de modèles d'avion simulés. Elle nous a donc permis de rapidement mettre en œuvre notre matériel, en ayant très peu d'éléments logiciels à paramétrer.

X-Plane étant un logiciel propriétaire, nous ne pouvions pas avoir une emprise totale sur la simulation.

Nous nous sommes donc ensuite tournés vers une simulation de vol appelée « Chronos », qui a été intégralement développée par Thales Systèmes Aéroportés.

Nous pouvons ainsi choisir de simuler des véhicules terrestres, marins, sous-marins et aériens, mais nous pouvons aussi changer la zone géographique où se déroule la simulation, choisir la nature et le nombre de véhicules peuplant cette zone, changer des paramètres environnementaux, etc.



FIGURE 2.1 – Jaquette du logiciel de simulation de vol X-Plane (à gauche) et capture d'écran d'une session de simulation (à droite).

Nous pouvons également extraire de la simulation toutes sortes d'informations qui pourraient encore améliorer le réalisme de la formation : récupérer des données pouvant servir à la commande de divers équipements (comme par exemple le siège dynamique de la section 2.1.3), extraire une vue du cockpit de l'avion simulé et l'afficher sur l'écran du pilote du scénario, et même extraire différentes vues de l'environnement extérieur et les afficher sur plusieurs écrans répartis dans la salle de formation et imitant des hublots de l'avion simulé.

La simulation envoie périodiquement des données de navigation à la console Amascos Touch : l'avion que nous voyons bouger sur la console correspond au même avion qui est simulé par Chronos. Cela permet de former un ensemble cohérent entre ce que voit le pilote depuis le cockpit et ce que voit le SSO sur son IHM.

## 2.1.2 Le poste de pilotage

Comme expliqué dans la section 2.1.1, la personne incarnant le pilote dans le scénario de formation possède son propre poste de travail, constitué d'un écran montrant une vue du cockpit de l'avion simulé et de contrôleurs lui permettant de piloter l'avion.



Concernant les contrôleurs, nous avons dans un premier temps choisi le système de contrôle de vol X52 de la marque Saitek (voir Figure 2.2), composé d'un joystick et d'une manette des gaz.



FIGURE 2.2 – Le système de contrôle de vol X52 de Saitek.

Nous avons par la suite utilisé l'ensemble de contrôleurs HOTAS Warthog de la marque Thrustmaster (voir Figure 2.3), déjà utilisé par le passé par Thales Système Aéroportés dans d'autres projets. Nous avons porté notre choix sur ces contrôleurs car il s'agit d'une réplique exacte des contrôleurs utilisés dans l'avion d'attaque A-10C de l'U.S. Air Force, ce qui procure donc plus de réalisme au poste de pilotage de l'EIO.



FIGURE 2.3 – Le système de contrôle de vol HOTAS Warthog de la marque Thrustmaster.

Les contrôleurs du pilote sont connectés à l'ordinateur exécutant la simulation Chronos et permettent donc de commander l'avion simulé dans la zone géographique du scénario.

### 2.1.3 Le siège dynamique

Durant toute la durée du scénario de formation, la personne formée se trouve assise sur un siège servant à reproduire le mouvement du véhicule alors simulé par l'EIO.

En effet nous utilisons le siège dynamique GS-105 développé par l'entreprise Geko Systems (voir Figure 2.4).



FIGURE 2.4 – Le siège dynamique GS-105 de Geko Systems.

Le dossier du siège est composé de deux parties : la partie basse est fixe, tandis que la partie haute est mobile. Ainsi, la partie mobile du dossier permet de reproduire des secousses dans le plan horizontal : vers l'avant, vers l'arrière et sur les côtés. Les secousses sont périodiquement « rafraîchies » par le siège afin d'imiter les forces d'inertie exercées par les mouvements de l'avion.

La partie fixe du siège peut quant à elle reproduire des vibrations pouvant par exemple imiter la mise en marche du moteur de l'avion.

Ce siège dynamique est donc parfaitement adapté à l'EIO. Il permet de procurer des sensations réalistes au SSO qui siègera dessus. De plus, sa taille (similaire à la taille d'un fauteuil de bureau) lui permet d'être facilement transportable.

Il est relié à l'ordinateur exécutant la simulation et reçoit donc des commandes lui permettant d'imiter les mouvements de l'avion simulé par Chronos.

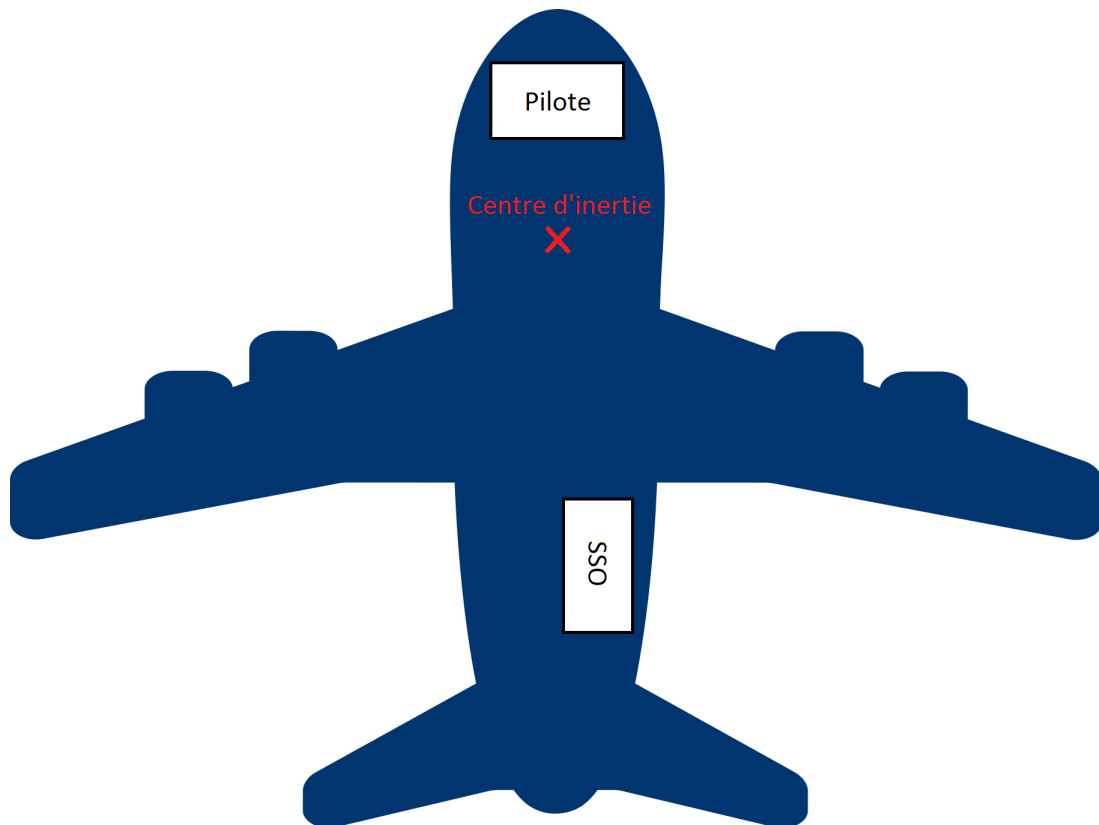


FIGURE 2.5 – Positions du pilote et du SSO dans un porteur lourd de patrouille maritime.

Une autre chose importante à prendre en compte est la position relative du SSO dans l'avion (voir Figure 2.5) : le SSO ne se trouve pas dans la direction longitudinale de l'avion, mais dans une direction transversale.

Alors que le pilote regarde dans la même direction que la direction de vol de l'avion, le SSO se trouve lui dans une direction perpendiculaire.

Le SSO est également plus éloigné du centre de gravité de l'avion que le pilote, et ressent donc plus les secousses dû au vol de l'avion : les bras de levier des forces d'inertie sont plus forts.

Il faut donc modifier les forces d'inertie extraites de Chronos (qui sont calculées pour l'endroit exact où le pilote de l'avion est assis) en les amplifiant et en les appliquant dans une direction perpendiculaire à la direction de vol de l'avion.

## 2.2 Permettre des communications internes et externes à l'équipage

Comme expliqué dans le chapitre 1, la personne formée sera amenée à effectuer des communications internes à l'équipage (avec le pilote de l'avion), mais aussi externes (avec des acteurs).



FIGURE 2.6 – Le micro-casque Gamecom 777 de Plantronics.

Ainsi le pilote et le SSO seront tous les deux équipés d'un micro-casque Gamecom 777 de la marque Plantronics (voir Figure 2.6) leur permettant de communiquer entre eux. Ce micro-casque permettra également au SSO de recevoir des communications radio provenant de l'extérieur : appel de détresse, interrogation de l'équipage (interprété par des acteurs) de certains navires, etc.



FIGURE 2.7 – Le kit audio GigaWorks T3 de Creative.

Nous avons également mis en place un système audio 2.1 au moyen du kit GigaWorks T3 de la marque Creative (voir Figure 2.7), constitué de deux enceintes et d'un caisson de basse. Ce système permet de reproduire fidèlement le bruit ambiant régnant dans l'avion.

Les micro-casques que nous avons choisis conviennent parfaitement car ils n'isolent pas complètement le SSO et le pilote. Ils leur permettent de continuer à entendre le bruit généré par les enceintes, placées de part et d'autre du SSO.

Le caisson de basse est quant à lui placé sur la même table que la console Amascos Touch, ce qui permet de faire vibrer l'ensemble et d'apporter une touche supplémentaire d'immersion.

## 2.3 Gérer la température ressentie par la personne formée

Afin de respecter les scénarios de formation décrits dans la section 1.3, l'EIO doit être capable de réguler la température ressentie par la personne formée, suivant les différentes phases du scénario choisi.



FIGURE 2.8 – Le chauffage soufflant AM09 de Dyson.

Nous avons pour cela choisi d'utiliser le chauffage soufflant AM09 de la marque Dyson (voir Figure 2.8), qui est capable au choix de produire un courant d'air chaud ou ambiant. Il permet donc, dans une zone relativement restreinte, de contrôler la température via ses deux modes de fonctionnement.

Ainsi le formateur disposera d'une télécommande lui permettant de régler la température ressentie par le SSO en fonction des différentes phases du scénario de formation.

Cet équipement convient parfaitement à l'EIO car il permet de faire ressentir une plage de température plutôt importante (de 15 à 37°C selon le constructeur) à un individu, ce qui nous permet de durcir les conditions de la formation.

Une prochaine étape serait de trouver un moyen de contrôler automatiquement cet appareil, ou d'acquérir un autre chauffage qui pourrait cette fois être programmable.

## 2.4 Créer une ambiance lumineuse convaincante

Le pilote de l'avion, le SSO et éventuellement le formateur se trouvent dans une salle spécialement dédiée à la formation. Afin d'apporter un degré supplémentaire de réalisme, il est donc nécessaire de reproduire une ambiance lumineuse immersive.



FIGURE 2.9 – L'ensemble de lumière connectée Hue de Philips : une lampe (à gauche) et son bridge (à droite).

Ainsi, nous avons choisi d'éclairer la pièce de formation avec l'ensemble de lumière connectée Hue de la marque Philips (voir Figure 2.9).

Cet ensemble est constitué de plusieurs lampes réparties dans la salle et connectées en Wi-Fi à un « bridge ». Ce dernier est capable de contrôler toutes les lampes et offre une interface permettant de le programmer facilement. Les lampes Hue permettent de générer des ambiances lumineuses de différentes couleurs.

Ce système de lampes connectées et programmables satisfait pleinement les besoins de l'EIO. En effet en fonction des différentes phases du scénario, plusieurs ambiances sont programmables via l'interface du bridge. Nous pouvons par exemple prévoir un éclairage tamisé dans les phases de calme en début et en fin de scénario, puis une lumière rouge lors de phases plus stressantes, pouvant ainsi suggérer un certain état d'alerte.

## 2.5 Reproduire des odeurs réalistes

Nous sommes en train d'étudier un moyen pour reproduire des odeurs de manière réaliste dans la salle de formation.

Nous songeons notamment à entrer en contact avec la société Exhalia (basée à Dinard en Ille-et-Vilaine) qui a réussi à développer un système très efficace permettant de reproduire de très nombreuses odeurs.

Le système s'apparente à un ventilateur dans lequel des billes encapsulant des odeurs sont insérées. Le ventilateur est ensuite capable de diffuser l'odeur sélectionnée dans une salle entière.

Ce système a notamment été utilisé avec succès par l'équipe du CESIM<sup>1</sup> Santé de Brest, que nous avons été amenés à rencontrer (voir le chapitre 5).

Un tel dispositif serait un réel plus pour l'EIO, puisque l'odorat est l'un des sens jouant le plus sur l'immersion d'un individu : l'odeur de nourriture peut par exemple nous donner immédiatement faim.

---

1. Centre de SIMulation





# Chapitre 3

## Mesure de la performance

Mesurer la performance d'une personne tout au long de sa formation est primordial afin de pouvoir effectuer un suivi et ainsi adapter la formation en conséquence.

### 3.1 Les moyens usuels de mesure de la performance

La mesure de performance durant une formation peut en premier lieu passer par des indicateurs basiques comme le nombre d'objectifs principaux et secondaires remplis par la personne formée, le temps qu'elle a mis à les remplir, ou encore sa capacité à détecter des informations et à les communiquer au reste de l'équipe.

Dans le cadre de la patrouille maritime, elle sera évaluée principalement sur le nombre de pistes radar qu'elle aura correctement détectées, ainsi que sur le temps écoulé entre l'apparition du véhicule dans le scénario de la formation et son identification par le SSO. Le formateur prêtera notamment attention à sa maîtrise du matériel : interaction avec l'IHM, gestion des différents modes radar, etc. Un autre critère important sera sa capacité de réaction face à des événements imprévus. Nous pouvons par exemple observer sa prise d'initiatives lors de la réception d'un message de détresse provenant d'un navire en proie à un incident. La personne formée devra alors dérouter d'elle-même un second bateau, afin qu'il puisse venir en aide au premier.

Il est également intéressant d'observer l'évolution de la personne formée en lui faisant suivre dans un premier temps un scénario simple<sup>1</sup>, avant de la confronter à un scénario plus difficile et éprouvant.

---

1. Moins de pistes radar sont à identifier, il n'y a pas d'évènements imprévus, la personne est moins secouée, le bruit ambiant est plus faible et la chaleur de la pièce est moins élevée.

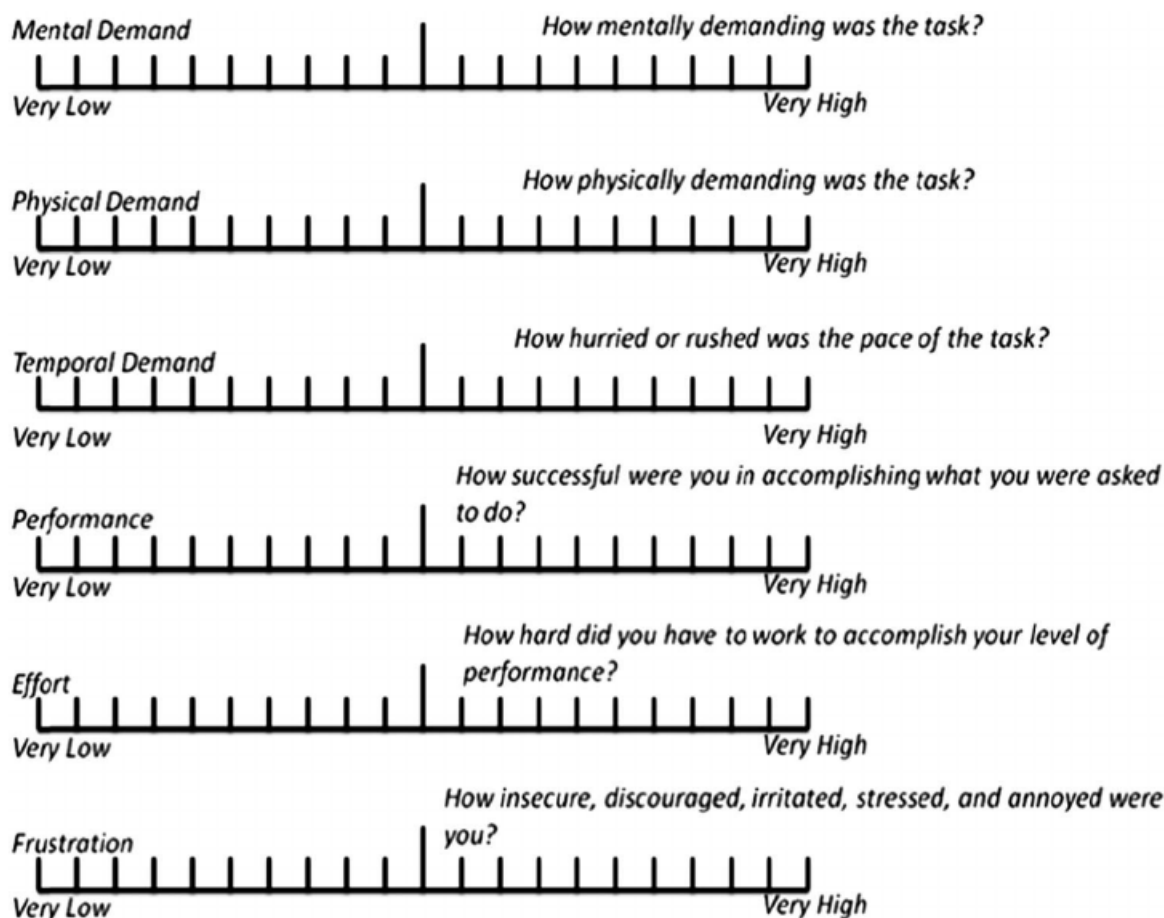


FIGURE 3.1 – Un exemple de questionnaire de type NASA-TLX.

Enfin, il existe un outil mis au point par la NASA appelé NASA-TLX<sup>2</sup> et permettant de recueillir le ressenti d'une personne sur les tâches qui lui ont été demandées, comme expliqué dans [11].

Il s'agit d'un questionnaire demandant de qualifier différentes composantes par une note allant de 0 à 20, ce qui permettra d'évaluer par la suite si la charge de travail demandée à la personne était adaptée ou non à ses capacités de travail. Les composantes devant être notées sont la demande mentale, la demande physique, la demande temporelle, la performance, l'effort et la frustration (voir Figure 3.1).

Les domaines de la fusion de données et de la classification ont cependant rendu possible la mise au point de nouveaux moyens de mesure.

---

2. NASA-Task Load Index

## 3.2 L'OFS : un moyen plus avancé de mesure de la performance

D'après [4], l'Operator Functional State<sup>3</sup> (OFS) est un indicateur permettant de savoir de manière globale si un opérateur, à un instant donné, est apte ou non à réaliser la tâche qui lui est confiée, en prenant en compte le contexte de la tâche, les capacités de l'opérateur, ainsi que son état actuel.

Différents éléments sont pris en compte lors de l'évaluation de l'OFS d'un opérateur :

- Les caractéristiques de l'environnement dans lequel la tâche est effectuée : la pression, la température, le bruit, les vibrations et secousses, la quantité d'oxygène présente dans l'air, etc.
- L'état actuel (à un instant donné) de l'opérateur : sa fréquence cardiaque, sa fréquence respiratoire, sa pression artérielle, l'amplitude de ses ondes cérébrales, son activité oculaire, sa fatigue, son stress, etc.
- Les caractéristiques de la tâche qui doit être réalisée par l'opérateur : la charge cognitive et la charge physique qu'elle va requérir, sa durée, le fait que l'opérateur puisse être au courant de tout ou partie de l'état de la tâche, etc.

Dans un vrai contexte de travail (et non dans le cadre d'une formation), la connaissance de l'OFS peut permettre de prévenir des éventuelles baisses de performance d'un opérateur et aide à maintenir des conditions optimales de travail.

Dans le cadre de l'EIO, l'évaluation de l'OFS nous permet de savoir si la personne formée est dépassée ou non par les événements, et donc d'ajuster en conséquence les scénarios de la formation afin de fournir une expérience plus personnalisée. Cette donnée nous est fortement utile car elle nous permet de détecter des failles dans l'ergonomie de l'IHM mise en œuvre dans l'EIO. Nous pouvons alors identifier les points de l'IHM qui sont à revoir : les emplacements des menus, les icônes, les couleurs, etc.

Nous devons désormais nous interroger sur comment est calculé cet OFS à partir de toutes les caractéristiques précédemment citées.

---

3. État fonctionnel de l'opérateur

## 3.3 Sensor Hub : un outil de supervision de l'OFS

### 3.3.1 Présentation générale de Sensor Hub

Sensor Hub (dont le logo est représenté sur la Figure 3.2) est un logiciel développé au Québec par l'entité Thales Research and Technology, en collaboration avec l'Université Laval et l'Institut National de la Recherche Scientifique.

Il s'agit d'une solution mobile et en temps réel de supervision et d'interprétation automatique de l'état d'un être humain, dans le but d'améliorer sa performance, son apprentissage et sa sécurité.



FIGURE 3.2 – Le logo de l'outil Sensor Hub.

Comme nous pouvons le voir sur la partie gauche de la Figure 3.3 (ci-après), Sensor Hub supporte à la fois des capteurs environnementaux (GPS, température, etc) et des capteurs physiologiques (Électrocardiogramme (ECG), activité oculaire, etc).

Nous pouvons aussi voir sur la partie de droite de cette même figure que Sensor Hub est capable de fournir de nombreuses informations sur l'état d'une personne : OFS, stress, charge mentale, etc.

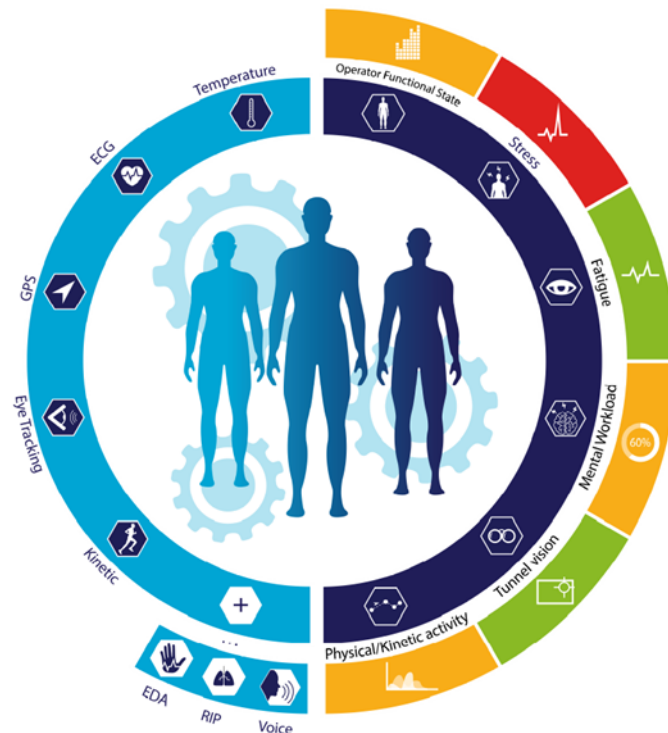


FIGURE 3.3 – Une vue globale de l'outil Sensor Hub.

Comme expliqué dans [20], Sensor Hub va collecter les données issues de tous les capteurs, et va les interpréter afin de pouvoir en extraire diverses caractéristiques.

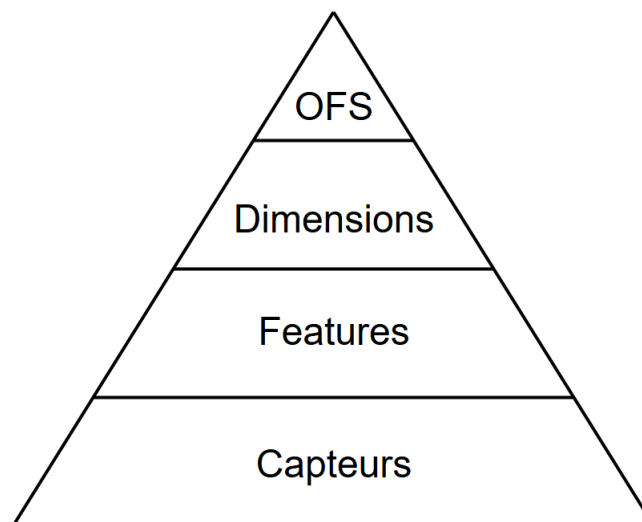


FIGURE 3.4 – Une représentation pyramidale des informations fournies par Sensor Hub.

La pyramide présente sur la Figure 3.4 illustre parfaitement ce processus :

- Les capteurs figurent tout en bas de la pyramide. Il s'agit là de l'acquisition des données brutes issues des capteurs physiologiques et environnementaux, obtenues grâce aux différents pilotes logiciels présents au sein de Sensor Hub.
- Les « Features » correspondent aux caractéristiques des différents signaux représentant les données brutes. Elles sont obtenues grâce à des techniques de traitement du signal appliquées aux signaux des différents capteurs : calcul des amplitudes et des fréquences, manière dont évoluent les courbes, etc.
- Les « Dimensions » figurent à un niveau d'abstraction plus élevé. Il s'agit des différents concepts psychologiques et neurophysiologiques mentionnés précédemment : le stress, la charge cognitive, la fatigue, etc. Elles sont obtenues grâce à des modèles paramétrés dans le logiciel. En effet, Sensor Hub dispose d'un modèle pour chaque Dimension, lui permettant d'interpréter les Features et de déterminer en conséquence l'état de l'individu. Ces modèles peuvent être de différentes natures : équations linéaires, ensemble d'arbres de décision, machines à vecteurs de support, méthode des  $k$  plus proches voisins, réseaux de neurones, etc.
- L'Operator Functional State représente le plus haut degré d'abstraction et est l'information la plus importante fournie par Sensor Hub.

Dans la suite de cette section, nous allons tout d'abord lister les différents capteurs que nous avons utilisés, puis nous allons détailler un exemple de modèle de Dimension, et nous finirons par expliquer comment Sensor Hub calcule le fameux OFS.

### 3.3.2 Les capteurs physiologiques présents dans l'EIO

Comme expliqué précédemment, Sensor Hub supporte un grand nombre de capteurs, de natures très variées. Pour l'EIO nous avons décidé de porter notre choix sur trois capteurs : une montre connectée, une ceinture pectorale, et un bandeau frontal.

#### La montre connectée de Motorola

La montre connectée que nous avons choisie est une Motorola MOTO 360 Sport (voir Figure 3.5).



FIGURE 3.5 – La montre connectée MOTO 360 Sport de Motorola.

Nous avons choisi cette montre car elle nous permet de mesurer la fréquence cardiaque<sup>4</sup> de l'individu utilisant l'EIO, de manière non-intrusive. Elle utilise le principe de la photopléthysmographie.

Comme expliqué dans [14], cette méthode de mesure requiert seulement des LED vertes et des photodiodes. Le principe est le suivant :

1. Dans un premier temps, les LED vertes présentes dans la montre vont émettre de la lumière verte en direction de la peau du poignet.
2. Cette lumière va alors traverser la peau et atteindre le sang coulant dans les veines situées au niveau du poignet.
3. Ensuite le sang, qui est de couleur rouge, va absorber une partie de cette lumière verte, avant de réfléchir le reste en direction de la montre. Ainsi plus le débit de sang est important, plus la lumière verte sera absorbée et moins elle sera réfléchie vers la montre.
4. Enfin, la montre va mesurer l'intensité de la lumière verte qui a été réfléchie par le sang, au moyen de photodiodes dont elle est équipée. Grâce aux variations d'intensité lumineuse qu'elle mesurera, elle pourra connaître les variations du débit sanguin, et ainsi déterminer la fréquence cardiaque.

En effet, ce sont les battements du cœur qui provoquent des variations régulières dans le débit sanguin.

---

4. Nombre de battements cardiaques (ou pulsations) par unité de temps.

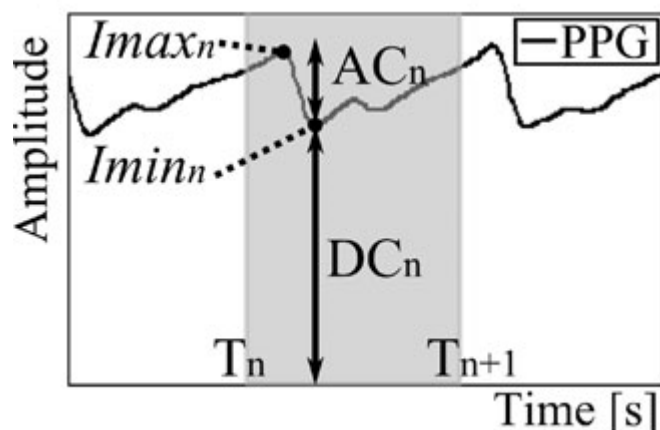


FIGURE 3.6 – Un exemple de photopléthysmogramme.

Sur la Figure 3.6, issue de [14], nous pouvons voir un exemple de photopléthysmogramme issu d'un capteur fonctionnant sur le même principe que notre montre connectée. Ce signal, représentant donc l'intensité lumineuse de la lumière verte réfléchiée par le sang, est périodique et comporte une composante continue (sa valeur est notée  $DC_n$  sur la figure). Cette composante continue provient de la lumière réfléchiée et dispersée par la peau (et non le sang). Nous voyons que le motif périodique de ce signal (de  $T_n$  à  $T_{n+1}$  sur le graphe ci-dessous) adopte déjà la forme d'un battement cardiaque, ce qui confirme bien la méthode de mesure expliquée précédemment.

### La ceinture pectorale BioHarness

La ceinture pectorale que nous avons utilisé est le modèle BioHarness de la marque Zephyr (voir Figure 3.7).



FIGURE 3.7 – La ceinture pectorale BioHarness de Zephyr.



Elle nous permet de mesurer à nouveau la fréquence cardiaque de l'individu, mais également sa fréquence respiratoire<sup>5</sup>. Le fait d'effectuer deux mesures différentes de la fréquence cardiaque (via la montre connectée et la ceinture pectorale) apporte plus de précision aux calculs, car Sensor Hub effectue ensuite une fusion des données provenant des différents capteurs.

Nous avons choisi d'utiliser ce capteur avec l'EIO parcequ'il est fiable (voir ci-après) et très simple à mettre en œuvre, ce qui permet de conserver la facilité de déploiement de l'EIO : la ceinture doit seulement être enfilée par la personne que nous voulons évaluer.

Comme expliqué dans [12], la ceinture pectorale contient des électrodes permettant de détecter les impulsions électriques générées par le cœur. Ces électrodes doivent être en contact avec la peau et sont composés de tissus électriquement conducteurs et d'élasthanne<sup>6</sup> antimicrobien argenté. Les impulsions électriques détectées sont générées par le cœur et transmises à travers la peau jusqu'aux électrodes, permettant ainsi de calculer la fréquence cardiaque d'une personne.

La fréquence respiratoire est mesurée au moyen d'un capteur capacitif propriétaire présent dans la ceinture pectorale et composé de couches de tissus électriquement conducteurs, de mousse et de polytéréphtalate d'éthylène<sup>7</sup> élastique.

Toujours d'après [12], la fréquence respiratoire sera quantifiée grâce à la mesure de la capacité du capteur capacitif. En effet, la capacité va être modifiée à cause des variations de l'impédance<sup>8</sup> du capteur, qui sera elle-même modifiée à cause des mouvements de la cage thoracique. Étant donné que le capteur est élastique, son impédance va par exemple augmenter lorsque sa propre taille va s'accroître, entraînant une variation de sa capacité. Ainsi lorsqu'une personne va inspirer, sa cage thoracique va s'ouvrir, et l'impédance du capteur capacitif va augmenter. À l'inverse lorsque celle-ci va expirer, sa cage thoracique va se fermer, et l'impédance du capteur capacitif va diminuer. La ceinture pectorale pourra calculer la fréquence respiratoire en mesurant la fréquence des inspirations et expirations.

La ceinture Zephyr BioHarness est très fiable quant à la mesure de la fréquence respiratoire. Cela a été démontré dans [10], qui présente une étude effectuée sur 12 personnes et comparant les mesures de fréquence respiratoire effectuées par la ceinture Zephyr BioHarness et par le capteur Metamax 3B (un système analysant le gaz inspiré et rejeté à chaque respiration et étant extrêmement fiable). Les résultats des deux capteurs furent alors très proches, avec une erreur type<sup>9</sup> de  $5.8\% \pm 1.5\%$ .

---

5. Nombre de cycles respiratoires (inspiration et expiration) par unité de temps.

6. Aussi connu sous la marque « Lycra ».

7. Aussi connu sous la marque « Mylar ».

8. Correspond à l'opposition d'un circuit électrique au passage d'un courant alternatif sinusoïdal.

9. L'erreur type d'une méthode d'estimation est l'écart type de l'estimateur utilisé.

### Le bandeau frontal Muse

Le bandeau Muse (voir Figure 3.8), commercialisé par l'entreprise InteraXon, est un dispositif permettant de mesurer l'activité électrique du cerveau et de produire des électroencéphalogrammes (EEG).



FIGURE 3.8 – Le bandeau frontal Muse.

Cette activité électrique provient de la différence de voltage entre les différents neurones du cerveau. D'après [19], elle reflète l'activité synchrone de neurones possédant la même orientation spatiale.

Les EEG montrent les oscillations de cette activité électrique au cours du temps, pour différentes fréquences. Pour mesurer ce voltage, plusieurs électrodes doivent être placées sur la tête de la personne subissant l'électroencéphalographie. Les emplacements des électrodes et leur dénomination sont définis par le « système international 10-20 », standardisé par l'American Clinical Neurophysiology Society (ACNS).

Nous pouvons observer une illustration du système international 10-20 sur la Figure 3.9. Une partie de la nomenclature utilisée dans ce standard est également représentée sur la Table 3.1.

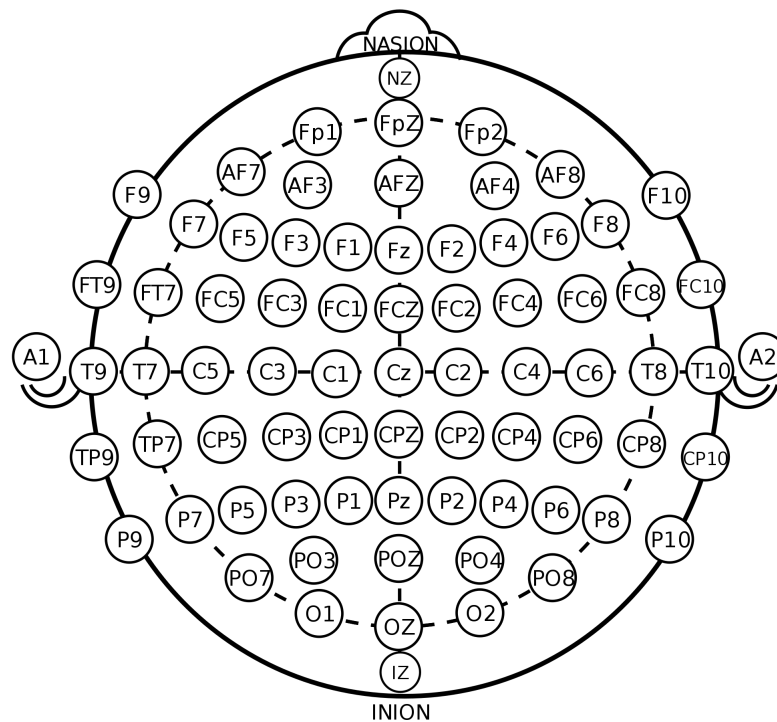


FIGURE 3.9 – Représentation du système international 10-20.

Symbole	Signification	Symbole	Signification
A	Lobe d'oreille	AF	Antérieur Frontal
C	Central	CP	Centropariétal
F	Frontal	FC	Frontocentral
FT	Frontotemporal	N	Nasion
O	Occipital	P	Pariétal
PO	Pariéto-occipital	T	Temporal

TABLE 3.1 – Une partie de la nomenclature du système international 10-20.

Les électrodes utilisées en électroencéphalographie sont usuellement placées sur le crâne avec un gel électriquement conducteur. Le bandeau Muse utilise quant à lui des « électrodes sèches », plus faciles à utiliser.

De plus, il transmet ses données via la technologie Bluetooth, éliminant ainsi les très nombreux câbles connectés normalement à chaque électrodes.

Dans [19], il est expliqué que le bandeau Muse est capable de mesurer 4 emplacements du système international 10-20 : FP1, FP2, TP9 et TP10.

Enfin, toujours dans [19], des chercheurs thaïlandais ont essayé de prouver que ce bandeau était fiable en mesurant l'activité cérébrale de moines bouddhistes lorsque ceux-ci méditaient, mais aussi durant d'autres activités.

Ils ont notamment pu exploiter les mesures du bandeau Muse en utilisant l'analyse temps-fréquence. En effet, ces techniques issues du traitement du signal permettent d'obtenir une représentation temps-fréquence d'un signal, montrant ainsi l'amplitude des différentes fréquences du signal qui apparaissent au cours du temps.

Les chercheurs ont décidé de regrouper certaines fréquences entre elles afin de pouvoir former des bandes de fréquences caractéristiques des ondes cérébrale (voir Table 3.2).

Nom	Fréquences	Activité
$\delta$ - delta	Moins de 4 Hz	Les plus lentes et fortes ondes cérébrales, très basse fréquence avec haute amplitude. Habituellement associé avec le sommeil profond. Aussi trouvé durant la méditation profonde.
$\theta$ - thêta	4 - 8 Hz	Somnolence, sommeil léger, relaxation profonde ou méditation.
$\alpha$ - alpha	8 - 12 Hz	Relaxation, état calme ou tranquillité.
$\beta$ - bêta	12 - 30 Hz	Conscience éveillée ordinaire, état d'alerte, raisonnement actif, anxiété ou concentration. La plupart des gens agissent dans cette bande de fréquence durant la journée.
$\gamma$ - gamma	Plus de 30 Hz	Les ondes cérébrales les plus rapides. Hyperactivité ou traitement d'informations provenant de différentes régions du cerveau.

TABLE 3.2 – Les bandes de fréquences caractéristiques des ondes cérébrales.

La représentation temps-fréquence d'un signal est obtenu en utilisant la transformée de Fourier à court terme (TFCT), puis en calculant le spectrogramme à partir de celle-ci. En effet, soit  $x$  le signal dont nous voulons la représentation temps-fréquence, la TFCT de  $x$  est donnée par l'équation 3.1 :

$$TFCT_x(t, f) = \int_{-\infty}^{+\infty} [x(\tau) \cdot w(\tau - t)] \exp^{-j2\pi f\tau} d\tau \quad (3.1)$$

avec  $w$  une fonction de pondération (dite aussi de « fenêtrage »),  $t$  le temps et  $f$  la fréquence en Hz.

Le spectrogramme de  $x$  est alors calculé à partir de la norme au carré de sa TFCT :

$$\text{spectrogramme}_x(t, f) = |TFCT_x(t, f)|^2 \quad (3.2)$$

Les chercheurs de [19] sont arrivés à obtenir les spectrogrammes représentés sur la Figure 3.10, à partir de mesures effectuées avec le bandeau Muse sur un moine bouddhiste.

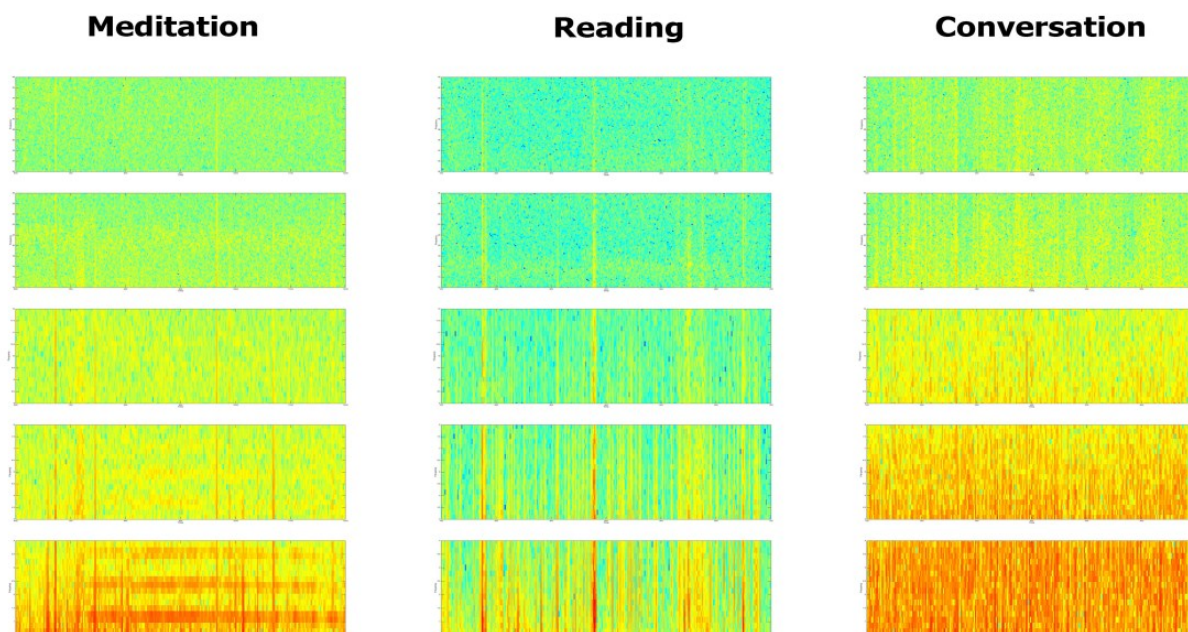


FIGURE 3.10 – Exemple de spectrogrammes obtenus grâce au bandeau frontal Muse placé sur un moine bouddhiste effectuant différentes activités.

Il y a donc 5 spectrogrammes pour chaque activité (méditation, lecture et conversation), correspondant de bas en haut aux ondes delta, thêta, alpha, bêta et gamma. Pour chaque spectrogramme, l'axe horizontal correspond au temps, l'axe vertical correspond aux fréquences et la couleur correspond à l'amplitude<sup>10</sup>.

Nous pouvons par exemple observer que lors de la méditation, les ondes cérébrales sont principalement des ondes delta. Cela correspond bien à ce qui était annoncé par la Table 3.2. En effet, étant donné que les moines bouddhistes s'entraînent à la méditation depuis de nombreuses années, ils sont donc capables de se plonger dans un état de méditation profonde, ce qui génère surtout des ondes delta.

10. Plus la couleur est proche du bleu, plus l'amplitude est faible; plus elle est proche du rouge, plus l'amplitude est élevée

[19] conclut donc que le bandeau Muse est un capteur très fiable et qu'il est adapté au domaine de la recherche.

Il convient également parfaitement à l'EIO, puisqu'il est très simple à déployer et non-intrusif, comme expliqué précédemment.

### 3.3.3 La charge cognitive : un exemple de modèle de Dimension

Les modèles utilisés afin de représenter les Dimensions dans Sensor Hub doivent être prédictifs. Ils ont pour but de prédire des caractéristiques abstraites en fonction des données issues des capteurs. Ainsi bien souvent les modèles de Sensor Hub se basent sur les techniques issues du domaine de l'apprentissage automatique<sup>11</sup>.

Par exemple, d'après [8], le modèle utilisé pour représenter la charge cognitive dans Sensor Hub est une « stochastic gradient boosting machine ». Il s'agit d'un algorithme utilisant des ensembles d'arbres de décision.

Dans cette section, nous rappellerons donc dans un premier temps le principe des arbres de décision ainsi que de la méthode de descente du gradient, puis nous expliquerons le fonctionnement des algorithmes de « gradient boosting », et nous finirons par la variante stochastique de ces algorithmes utilisé dans le modèle de la charge cognitive.

#### Les arbres de décision

Tout d'abord, les arbres de décision sont des modèles d'apprentissage automatique supervisés<sup>12</sup> et non paramétriques<sup>13</sup>. Ils sont utilisables aussi bien pour la classification<sup>14</sup> que pour la régression<sup>15</sup>.

Afin d'expliquer le principe des arbres de décision, nous allons utiliser un exemple simple de classification pris dans [13].

Cet exemple consiste à prédire si une personne va pratiquer ou non une activité en extérieur, en fonction de la valeur de 4 variables environnementales. Pour cela, nous disposons des 14 observations présentes sur la Table 3.3. Chaque observation est constituée des valeurs des 4 variables environnementales (Prévision, Température, Humidité et Vent) et de la valeur correspondante de la variable « Activité en extérieur » qui doit être prédite.

---

11. Aussi appelé par la traduction anglaise « Machine Learning ».

12. Qui utilisent des données d'entraînement pour lesquels nous connaissons d'avance la classe à laquelle ils appartiennent.

13. Qui ne présupposent pas de forme particulière pour la fonction de prédiction qui doit être déterminée.

14. Lorsque l'algorithme doit déterminer la classe d'un élément parmi un ensemble discret de possibilités.

15. Lorsque l'algorithme doit déterminer une valeur appartenant à un ensemble continu. L'arbre est alors appelé un arbre de régression.

	Prévision	Température	Humidité	Vent	Activité en extérieur
1	Couvert	Froid	Normal	Oui	Oui
2	Couvert	Chaud	Élevé	Non	Oui
3	Couvert	Chaud	Normal	Non	Oui
4	Couvert	Moyen	Élevé	Oui	Oui
5	Pluvieux	Froid	Normal	Oui	Non
6	Pluvieux	Moyen	Élevé	Oui	Non
7	Pluvieux	Froid	Normal	Non	Oui
8	Pluvieux	Moyen	Élevé	Non	Oui
9	Pluvieux	Moyen	Normal	Non	Oui
10	Ensoleillé	Chaud	Élevé	Non	Non
11	Ensoleillé	Chaud	Élevé	Oui	Non
12	Ensoleillé	Moyen	Élevé	Non	Non
13	Ensoleillé	Froid	Normal	Non	Oui
14	Ensoleillé	Moyen	Normal	Oui	Oui

TABLE 3.3 – Un ensemble d'observations devant servir à l'élaboration d'un arbre de décision devant décider si une personne va pratiquer ou non une activité en extérieur.

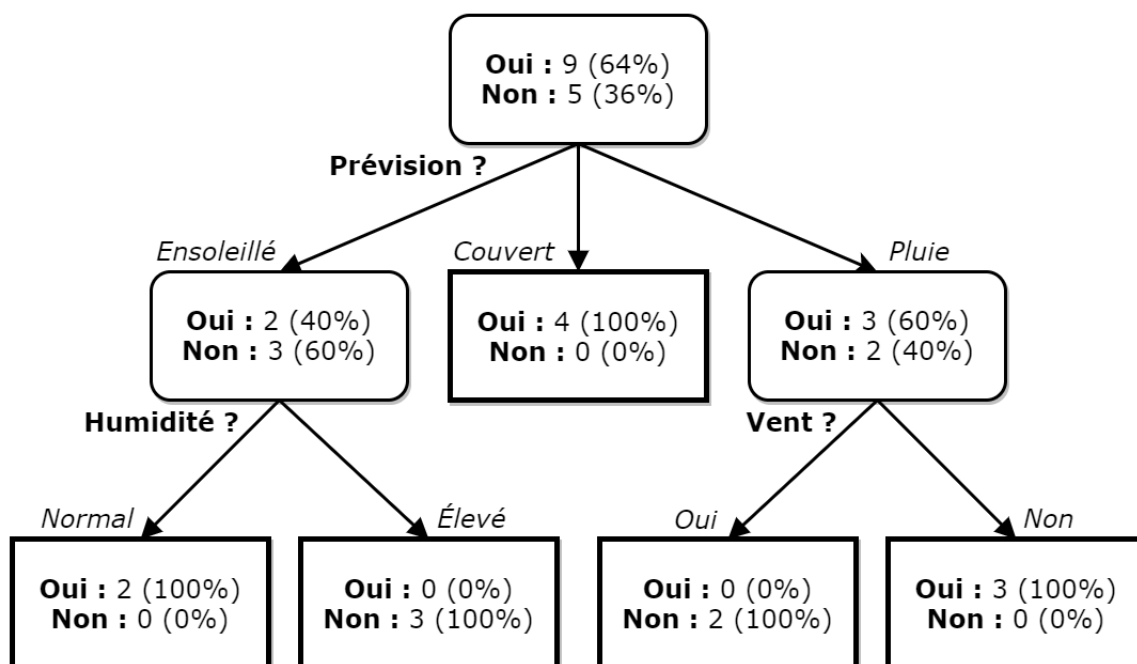


FIGURE 3.11 – Un exemple d'arbre de décision : il doit ici décider si une personne va pratiquer ou non une activité en extérieur.

L'arbre de décision de la Figure 3.11 est ensuite créé à partir de ces données. Chaque bloc correspond à un « nœud » de l'arbre, auquel est associé un « critère de segmentation » permettant de diviser le nœud en plusieurs branches. Ici les critères de segmentation correspondent aux différentes valeurs des variables environnementales. Dans le cas de variables continues, le critère de segmentation associé consiste à regarder si la variable est supérieure ou inférieure à une certaine valeur, créant ainsi deux branches distinctes.

Nous pouvons observer l'évolution de la distribution des observations dans les différents nœuds, au fur et à mesure que nous descendons dans l'arbre de décision. Nous voyons ainsi que le premier nœud (aussi appelé la « racine » de l'arbre) contient l'ensemble des 14 observations de la Table 3.3. Les nœuds sont ensuite répartis dans les différentes branches en fonction des critères de segmentation, jusqu'à ce que les nœuds terminaux (aussi appelés « feuilles » de l'arbre) soient composés d'un ensemble homogène d'observations : soit que des observations correspondant à la pratique d'une activité en extérieur (que des « Oui »), soit l'inverse (que des « Non »).

Enfin, nous pouvons noter que la température n'est pas une variable discriminante<sup>16</sup> car elle ne figure pas ici comme un critère de segmentation.

La méthode de création d'arbres n'a pas été explorée en détail ici car nous avons effectué un simple rappel sur le fonctionnement général des arbres de décision, nécessaire pour la suite de cette section. Des explications supplémentaires pourront être trouvés dans [18].

### La méthode de descente du gradient

Les algorithmes dit de « gradient boosting » se basent sur le même principe que la méthode de descente du gradient. Pour rappel, cette méthode d'optimisation itérative est utilisée afin de déterminer un minimum local d'une fonction.

Soit  $x$  un scalaire et  $f(x)$  la fonction dont nous voulons trouver un minimum local. Nous commençons par choisir une valeur particulière  $f(x^{(0)})$  qui sera le point de départ de notre descente du gradient. Soit  $\eta$  un scalaire, la  $i^{\text{ème}}$  itération de l'algorithme s'écrit :

$$x^{(i)} = x^{(i-1)} - \eta \frac{df(x^{(i-1)})}{dx} \quad (3.3)$$

$x^{(i)}$  est donc déterminé en suivant la direction opposée de la pente au point  $f(x^{(i-1)})$ , le scalaire  $\eta$  servant à régler le « pas » de l'algorithme. Ainsi  $x^{(i)}$  est plus proche du minimum local que  $x^{(i-1)}$ .

Nous continuons ainsi jusqu'à ce que nous ayons :  $x^{(i)} = x^{(i-1)}$ . Nous avons alors trouver un minimum local.

---

16. Elle n'apporte aucune information permettant la classification correcte des observations.



En prenant un point de départ  $x^{(0)}$  différent, nous pourrions éventuellement trouver un autre minimum local.

Plus nous nous rapprochons du minimum, plus la pente est faible en valeur absolue, et plus les déplacements de  $x$  sont faibles. La Figure 3.12 illustre parfaitement cela.

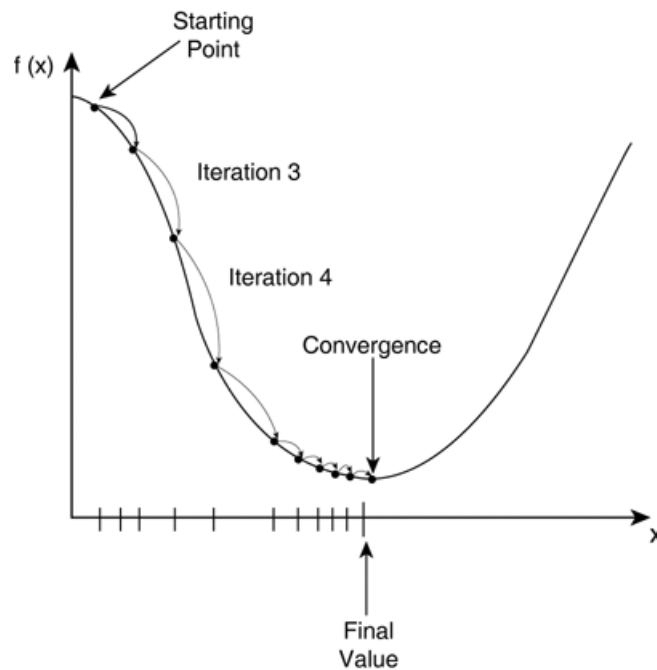


FIGURE 3.12 – Illustration de la méthode de descente du gradient, avec des mouvements de plus en plus faibles à l'approche du minimum.

Lorsque  $x$  est un vecteur, l'algorithme est similaire et seule l'équation 3.3 devient :

$$x_j^{(i)} = x_j^{(i-1)} - \eta \frac{\partial f(x_j^{(i-1)})}{\partial x_j} \quad (3.4)$$

avec  $x_j$  la  $j^{\text{ème}}$  composante du vecteur  $x$ .

Ainsi ce sont toutes les composantes du vecteur  $x$  qui se déplacent indépendamment et simultanément vers le minimum local.

### Les algorithmes de gradient boosting

Le « boosting » est une technique d'apprentissage automatique qui consiste à combiner plusieurs classifieurs faibles <sup>17</sup> afin de former une sorte de super-classifieur.

Les algorithmes de « gradient boosting » reposent eux sur les arbres de décision et utilisent un principe similaire à la méthode de descente du gradient.

Soit  $x$  un vecteur et  $h(x)$  un classifieur ou un régresseur (nous l'appellerons prédicteur dans la suite).

Soit  $L$  la « fonction de perte » de  $h$  : plus le classifieur  $h$  effectue de mauvaises prédictions, plus les valeurs de  $L$  s'accroissent.

Dans le cas du gradient boosting, la fonction de perte  $L$  doit être différentiable.

Nous allons ici utiliser l'erreur quadratique comme fonction de perte :

$$L(y, x) = (y - h(x))^2 \quad (3.5)$$

avec  $h(x)$  la valeur prédite par le classifieur  $h$  au point  $x$ , et  $y$  la vraie valeur qui devait être prédite.

Nous allons ici chercher à minimiser la fonction :

$$f(x) = \sum_{i=1}^N L(y_i, h(x_i)) \quad (3.6)$$

c'est-à-dire la perte sur l'ensemble des points  $(x_i, y_i)$ .  $x$  est donc ici le vecteur dont chaque composante est un point  $x_i$  pour lequel le classifieur effectue une prédiction  $h(x_i)$ , alors que la vraie valeur est  $y_i$ .  $N$  est donc le nombre total de points.

Dans la méthode de descente du gradient,  $x$  devenait un minimum de  $f$  grâce à l'équation 3.3.

Ici nous voulons que  $h(x)$  devienne un minimum de  $f$ , puisque nous voulons trouver un prédicteur qui minimise la perte totale exprimée dans l'équation 3.6, devenant ainsi un prédicteur efficace.

Les algorithmes de gradient boosting fonctionnent de manière itérative :

1. Nous choisissons comme point départ  $h^0(x) = c$ , avec  $c$  une constante minimisant  $f(x)$ . Nous choisissons donc  $c$  tel qu'il minimise  $\sum_{i=1}^N L(y_i, c)$ .

---

17. Un classifieur est dit « faible » lorsque qu'il est capable de reconnaître deux classes au moins aussi bien que le hasard ne le ferait : in ne setrompe pas plus d'une fois sur deux en moyenne, si la distribution des classes est équilibrée.

2. À la  $i^{\text{ème}}$  itération, pour  $j = 1, 2, \dots, N$ , nous calculons  $r_{ji} = -\frac{\partial L(y_j, h^{(i-1)}(x_j))}{\partial h^{(i-1)}(x_j)}$ . Cela est faisable puisque la fonction de perte  $L$  est différentiable. Dans le cas de l'erreur quadratique, nous avons  $\frac{\partial L(x)}{\partial h} = -2(y - h(x))$ . Cette étape se rapproche de l'équation 3.4 vu dans la méthode de descente du gradient.
3. Nous avons donc un ensemble  $(x_j, r_{ji})$  pour  $j = 1, 2, \dots, N$ . Nous utilisons désormais cet ensemble afin de construire un arbre de régression<sup>18</sup> qui prédise  $r_{ji}$  à partir de  $x_j$ . Cette arbre estime donc le gradient de tous les points  $x_j$ . Appelons cet arbre  $T_g^{(i)}$ , puisque nous sommes toujours à la  $i^{\text{ème}}$  itération.
4. Soit  $K$  le nombre de feuilles de l'arbre  $T_g^{(i)}$ . Ces feuilles segmentent donc l'espace des variables prédictives en région  $R_k$  pour  $k = 1, 2, \dots, K$ . Si nous mettons un point  $x_j$  en entrée de l'arbre  $T_g^{(i)}$ , il finira donc par atterir dans une des régions  $R_k$ . Désormais nous voulons associer une constante  $\eta_k$  à chaque région  $R_k$ , tel qu'elle minimise la perte dans une région définie par

$$\sum_{x_j \in R_k} L(y_j, h^{(i-1)}(x_j) + \eta_k) \quad (3.7)$$

Ceci est résolu comme  $k$  problèmes simples de minimisation indépendants.

Nous voyons alors émerger un second arbre : sa structure est donnée par  $T_g^{(i)}$ , mais les prédictions situées aux feuilles concernent désormais  $\eta_k$ .

5. Comme dans la descente du gradient, nous effectuons une « mise à jour » de la valeur de  $h(x)$  :

$$h^{(i)}(x) = h^{(i-1)}(x) + \sum_{k=1}^K \eta_k I(x \in R_k) \quad (3.8)$$

Ici  $I(x \in R_k)$  est une fonction indicatrice qui a pour valeur 1 lorsque  $x$  se situe dans la région  $R_k$ , et 0 sinon.

Comme expliqué précédemment, le second terme de l'équation 3.8 est un arbre dérivé de  $T_g^{(i)}$ .

De plus, nous observons que les équations 3.7 et 3.8 se ressemblent. En effet,  $\eta_k$  a bien été déterminé de manière à ce que le classifieur  $h$  ici mis à jour produise le moins possible de perte.

6. Nous répétons les étapes 2 à 5 un nombre  $M$  de fois.
7. Nous obtenons finalement notre nouveau prédicteur  $h^{(M)}(x)$ , qui minimise la perte totale  $f$  définie à l'équation 3.6. Étant donné qu'un arbre a été ajouté à chaque passage par l'étape 5,  $h^{(M)}(x)$  consiste simplement en une somme d'arbres de régression, plus la constante de départ  $c$  déterminée à l'étape 1.

Il existe plusieurs algorithmes de gradient boosting, et des explications supplémentaires peuvent être trouvées dans [5].

18. Il s'agit d'un arbre de décision utilisé dans le cadre d'une régression et non d'une classification.

## L'algorithme stochastique de gradient boosting

Enfin, la charge mentale de Sensor Hub est modélisée par une « stochastic gradient boosting machine », comme mentionné dans [8].

Cette variante de l'algorithme de gradient boosting présenté précédemment a été développé dans [6].

Sa nouveauté réside dans le fait qu'à chaque itération, l'arbre de régression est construit à partir d'un sous-ensemble de l'ensemble  $(x_j, r_{ji})$  (vu à la page 37). L'ensemble n'est donc pas utilisé dans son entier, mais est sous-échantillonné aléatoirement (d'où le terme « stochastique »). Il est observé dans [6] que cette modification a pour effet d'améliorer la précision des prédictions.

Ainsi, la charge mentale est la variable prédite par cette stochastic gradient boosting machine, à partir des données issues des capteurs physiologiques.

### 3.3.4 Calcul de l'OFS

Comme expliqué dans [9], il existe de nombreuses méthodes permettant de calculer l'OFS.

Ainsi, à l'instar de la charge mentale étudiée dans la section 3.3.3, Sensor Hub se base sur des algorithmes issus de l'apprentissage automatique pour calculer l'OFS.

Dans [9], plusieurs algorithmes sont comparés : la méthode des  $k$  plus proches voisins , les machines à vecteurs de support, les réseaux de neurones de type « feed-forward », ou encore les arbres de décision et de régression.

Une des principales problématiques à prendre en compte est la manière dont les données d'entraînement vont être recueillies.

En effet, tous les algorithmes précédemment cités ont besoin d'être « entraînés » afin de pouvoir effectuer des prédictions précises de l'OFS. Il faut donc lui fournir des exemples variés et représentatifs de calcul d'OFS, pour les besoins de cette phase d'entraînement. Il faut donc enregistrer des exemples de données de capteurs et de Dimensions (cf la section 3.3.1) pour lesquelles nous connaissons l'OFS (qui sera « étiqueté » à ces données).

Il existe différentes façons de recueillir ces données :

- Nous pouvons demander à des personnes effectuant certaines tâches de noter leur état : elles devront dire si la tâche qu'elles sont en train d'effectuer leur paraît facile, difficile, plus facile ou difficile qu'une autre tâche, etc. Nous construirions ainsi une échelle relative pour chaque personne nous permettant de quantifier son OFS au cours du temps, tout en enregistrant l'ensemble de ses données physiologiques et psychologiques.

- Nous pouvons également manipuler nous même l'OFS de certaines personnes en changeant les conditions expérimentales d'une tâche donnée. Par exemple, nous savons qu'en induisant de la fatigue ou du stress et en augmentant la charge cognitive requise pour effectuer la tâche, l'OFS de la personne diminuera, puisqu'elle sera moins apte qu'avant à accomplir la tâche en question. Nous pouvons ainsi arriver à nouveau quantifier l'OFS au cours du temps, pour une plusieurs tâches.
- Nous pouvons aussi baser l'évaluation de l'OFS d'une personne sur des mesures de performances simples, comme celles mentionnées dans la section 3.1. Dans le cas de tâches de recherche visuelle par exemple, nous pourrions comparer les temps de réaction d'une personne d'un exercice sur l'autre, afin de quantifier son OFS au cours du temps.

[9] compare également différents niveaux de granularité concernant la mise en œuvre des modèles d'apprentissage automatique :

- Un ensemble de modèles entraînés spécifiquement pour chaque tâche, tout individu confondu.
- Un ensemble de modèles entraînés spécifiquement pour chaque individu, toute tâche confondue.
- Utiliser un seul modèle global entraîné à partir des données provenant de toutes les tâches et de toutes les personnes effectuant ces tâches. Ils introduisent alors une nouvelle mesure : la « performance dynamique décontextualisée ». Il s'agit d'un concept utilisant une mesure de performance simple (comme le temps de réaction précédemment évoqué) effectué sur un individu pour une tâche donnée, à un moment donné. La performance de l'individu est alors considérée soit comme « fonctionnelle », soit comme « sous-fonctionnelle », en comparaison de la performance globale de l'individu sur cette même tâche.

Les résultats de cette comparaison sont qu'entraîner un seul modèle global est tout aussi performant en terme de prédictions correctes qu'entraîner un ensemble de modèles spécifiques à chaque tâche ou à chaque individu. Entraîner un seul modèle global permet donc un gain considérable de temps et de moyens.

Ainsi Sensor Hub possède un seul modèle prédisant l'OFS, entraîné à partir de données provenant d'un grand nombre de personnes et d'un grand nombre de tâches.



# Chapitre 4

## Intégration matérielle et logicielle

Dans ce chapitre, nous présenterons des détails plus techniques concernant l'intégration matérielle et le paradigme de programmation orientée composant.

Avant de commencer le chapitre à proprement parler, le schéma représenté sur la Figure 4.1 (voir la page suivante) peut aider à mieux visualiser comment s'organise l'EIO.

La configuration de l'EIO présentée sur ce schéma est un exemple d'utilisation différent de ce que nous avons pu voir jusqu'à présent.

Le scénario de formation fait en effet intervenir plus d'acteurs : nous avons ici un expert en Guerre électronique et un accousticien. Ils viennent aider le SSO en analysant respectivement le spectre électromagnétique et les ondes accoustiques reçues par l'avion.

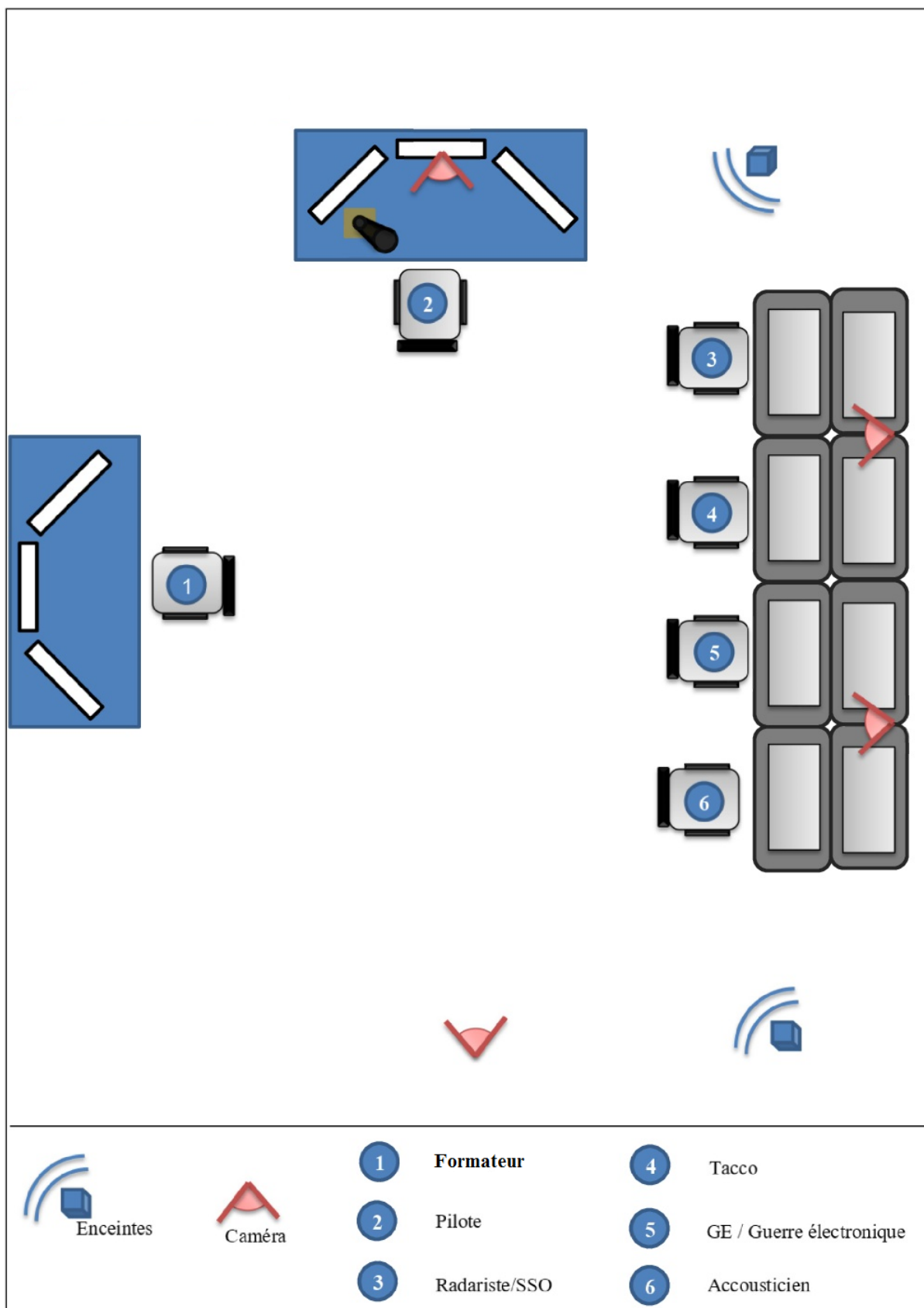


FIGURE 4.1 – Exemple de configuration de l'EIO.



## 4.1 Intégration matérielle

### 4.1.1 Le poste de supervision

Après le poste de pilotage et le poste de la console Amascos Touch présentés dans les chapitres 1 et 2, un troisième poste entre en jeu dans l'EIO : le poste de supervision du formateur.

#### L'affichage des données de Sensor Hub

Pour commencer, le formateur dispose d'un écran lui permettant d'afficher les données calculées par Sensor Hub.

Nous désirons seulement afficher les informations que nous jugerons utiles pour le formateur.

Sensor Hub est en effet capable de délivrer beaucoup d'informations. Certaines de ces informations ne sont pas très parlantes et n'existent que pour les besoins des calculs : il s'agit des Features (voir la section 3.3.1). Celles-ci peuvent être vues comme des variables intermédiaires, qui ne sont nécessaires que pour le calcul de variables plus abstraites (comme les Dimensions ou l'OFS).

Nous avons ainsi choisi de n'afficher que six informations sur l'écran du formateur : la fréquence cardiaque, la fréquence respiratoire, le stress, la fatigue, la charge mentale et l'OFS. Il s'agit de variables plus parlantes pour un être humain que les signaux des capteurs ou les Features.

Sensor Hub fonctionne sur une architecture client-serveur. Dans le cas de l'EIO, le serveur est exécuté sur une tablette tactile Galaxy Tab S2 de la marque Samsung (voir Figure 4.2) et le client sur un PC Windows standard.



FIGURE 4.2 – La tablette tactile Galaxy Tab S2 de Samsung.

La montre connectée de Motorola, la ceinture pectorale BioHarness et le bandeau frontal Muse (présentés dans la section 3.3.2) sont tous connectés en Bluetooth à la tablette tactile sur laquelle est exécuté le serveur. Celui-ci est chargé de calculer toutes les données, comme expliqué dans le chapitre 3.

Le client Sensor Hub récupère les informations calculées par le serveur afin de les afficher dans des graphes.

### La régie vidéo

Le formateur possède également un second écran lui servant de régie vidéo. Plusieurs caméras IP<sup>1</sup> lui permettent en effet de superviser la salle de formation et la personne formée.

La régie vidéo nous permet de placer le formateur dans une autre salle que celle où se déroule la formation, pour ne pas briser l’immersion, tout en continuant à pouvoir surveiller visuellement la personne formée.

Il pourra par exemple être dans la salle de formation pendant les premières sessions, puis aller dans une autre salle lorsque la personne formée se sera familiarisée avec la console Amascos Touch.

Nous avons choisi d’utiliser des caméras DCS-2230L de la marque D-Link (voir Figure 4.3), celles-ci sont très facilement déployables : elles sont légères et disposent d’un socle permettant de le placer facilement sur des surfaces planes.

De plus elle dispose d’un mode de capture vidéo en « vision nocturne » nous permettant de continuer à superviser la formation même lorsque l’éclairage devient faible pour les besoins du scénario.



FIGURE 4.3 – La caméra IP DCS-2230L de D-Link.

---

1. Aussi appelé « caméras en réseau »

Ces caméras permettent de fournir des flux vidéo possédant une définition « Full HD », soit  $1920 \times 1080$  pixels.

Les vidéos capturées par ces caméras peuvent être encodées et envoyées sur le réseau au moyen de plusieurs protocoles. Nous avons choisi d'encoder les vidéos au format H.264<sup>2</sup> et de les envoyer sur le réseau grâce au protocole RTSP<sup>3</sup>. Ces deux choix sont classiques dans le domaine de la vidéo sur réseau, car ils permettent de diffuser efficacement les flux vidéos.

### 4.1.2 Résumé de la configuration matérielle de l'EIO

La connexion entre le serveur Sensor Hub et son client s'effectue grâce à l'utilisation du routeur Wi-Fi RT-AC88U de la marque Asus (voir Figure 4.4).



FIGURE 4.4 – Le routeur Wi-Fi RT-AC88U de Asus.

La tablette tactile et l'ordinateur qui affiche les données de Sensor Hub sont tous les deux connectés en Wi-Fi à ce routeur et peuvent ainsi communiquer ensemble.

De plus, les caméras IP, l'Amascos Touch, le PC de simulation et de pilotage et le bridge Hue sont tous également connectés en Wi-Fi à ce routeur.

---

2. Aussi appelé « MPEG-4 AVC (Advanced Video Coding) » ou encore « MPEG-4 Part 10 ».

3. Real Time Streaming Protocol (Protocole de Streaming en Temps-Réel), un protocole de communication situé sur la couche applicative du modèle OSI (niveau 7).

La Figure 4.5 résume l'ensemble du matériel utilisé dans l'EIO, ainsi que les connexions entre chaque équipement.

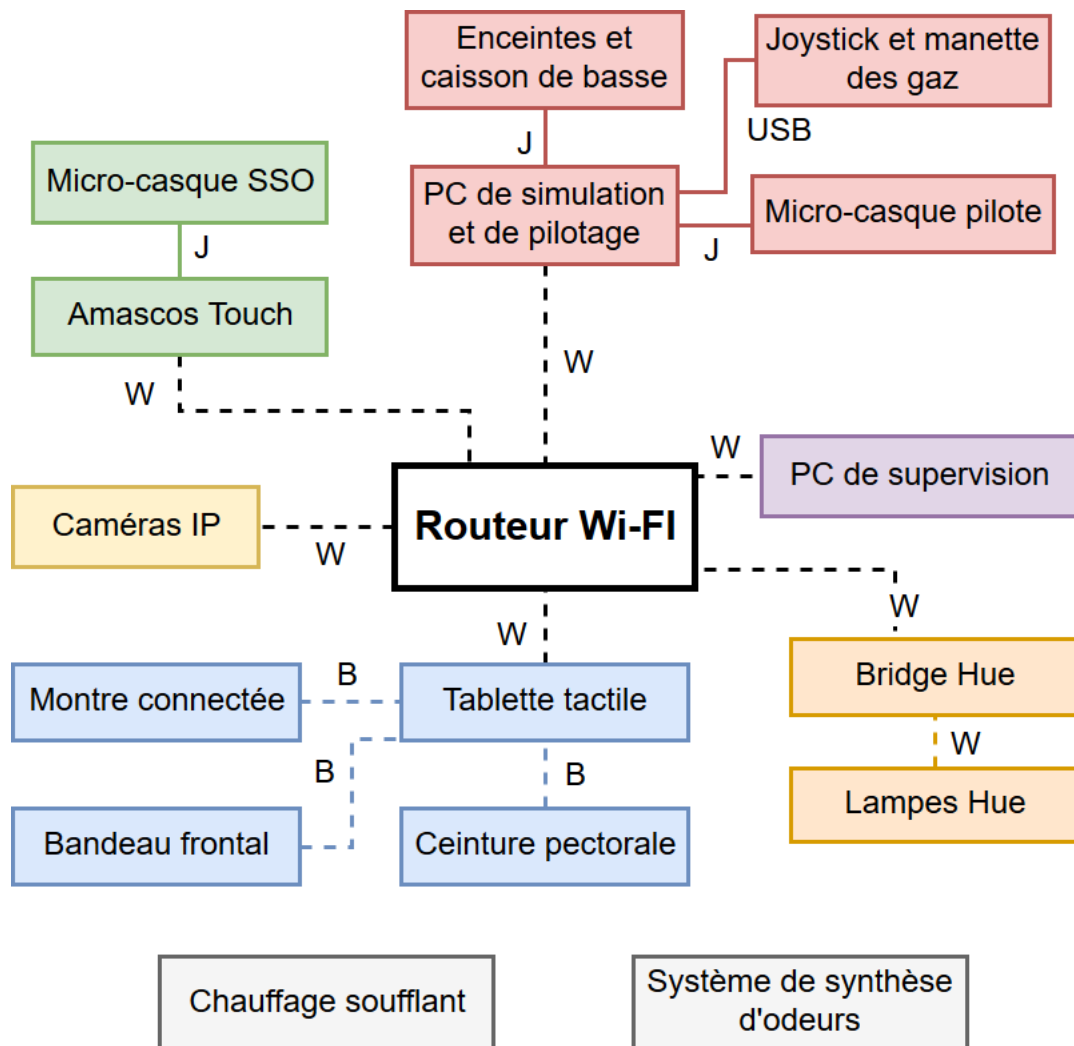


FIGURE 4.5 – Schéma global du matériel de l'EIO. W = Wi-Fi, J = Jack, B = Bluetooth.

## 4.2 Intégration logicielle

### 4.2.1 Le framework Naja : la programmation orientée composant

Naja est un framework Java développé et utilisé par Thales Systèmes Aéroportés qui permet de mettre en place des pratiques de développement logiciel à base de composant pour réaliser des IHM.

Il met en œuvre des composants appelés « composants Jacomo ». Les composants sont connectés les uns avec les autres afin de former une architecture logicielle. Un composant Jacomo est une « brique logicielle » réutilisable.

La figure 4.6 montre la représentation d'un composant Jacomo. Le composant possède un type unique et des paramètres qui peuvent être modifiés par d'autres composants durant l'exécution d'une application.

Un composant Jacomo est capable de produire des événements et de fournir des services à d'autres composants, tout en consommant des événements et en utilisant les services d'un autre.

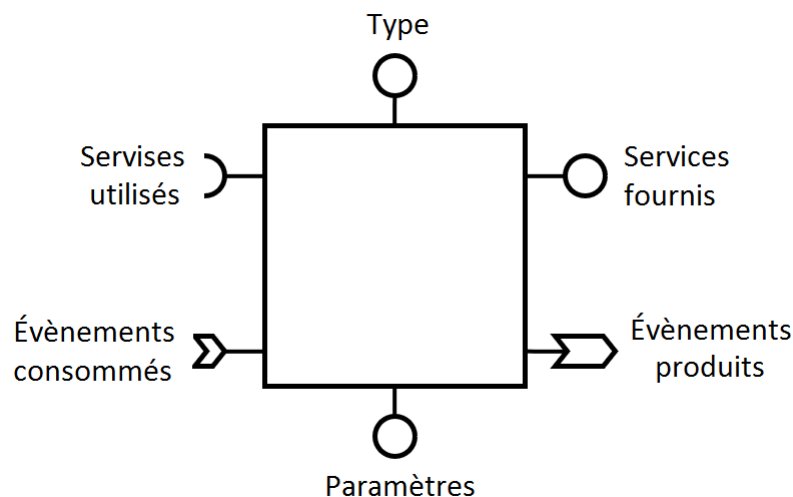


FIGURE 4.6 – La représentation d'un composant Jacomo.

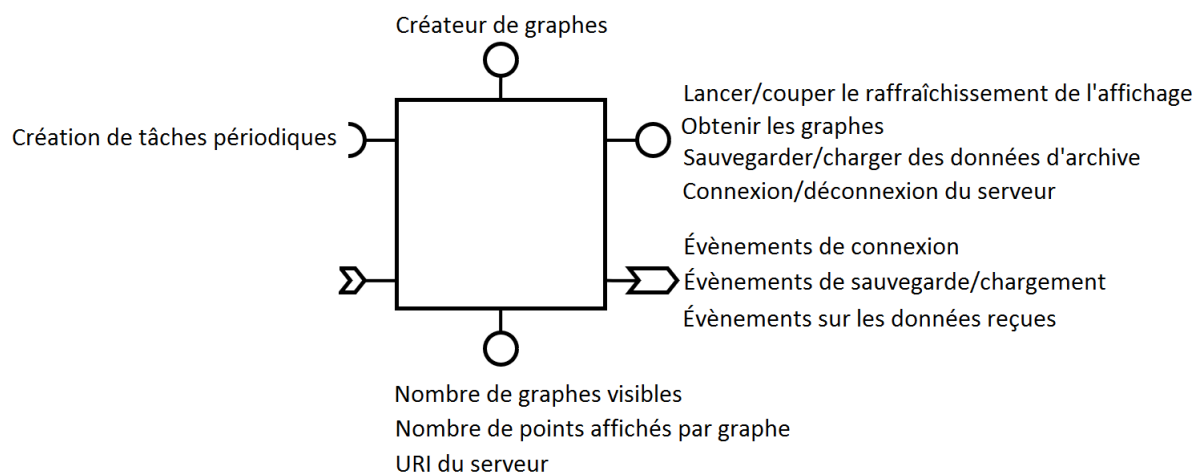


FIGURE 4.7 – Le composant Jacomo d’exploitation du client Sensor Hub permettant d’afficher des graphes.

Nous pouvons observer un exemple de composant Jacomo sur la Figure 4.7 : il s’agit du composant permettant d’exploiter un client Sensor Hub (présenté à la section 4.1.1) afin de pouvoir tracer des graphes.

Les avantages de ce paradigme de programmation sont multiples :

- La réutilisabilité : il est possible de réutiliser les composants Jacomo afin de gagner du temps de développement dans des projets ultérieurs., car chaque composant est conçu indépendamment des autres.
- La maintenabilité : il est facile de corriger seulement un composant à la fois, sans avoir à modifier le projet dans sa globalité et risquer ainsi de perdre sa stabilité. Il est nécessaire pour cela de respecter les interfaces fournies par les composants.
- La spécialisation : une équipe de développement peut se diviser en plusieurs sous-équipes, chacune s’occupant de ne travailler que sur un seul composant.

Un composant Jacomo a donc été développé pour chaque fonctionnalité de l’EIO : le créateur de graphes, la régie vidéo, le contrôle du siège, l’IHM de supervision, etc.

La Figure 4.8 (en page suivante) représente un schéma très simplifié de l’architecture utilisée dans l’EIO.

Les composants Jacomo représentés ici ont été développés pour l’EIO. Seul les principaux services utilisés sont affichés, afin de ne pas surcharger le schéma.

Par exemple, le contrôleur du siège récupère les données de la simulation auprès du client Chronos, et fournit un résumé des forces qu’il applique au siège à l’IHM de supervision, afin que cette dernière puisse les afficher en temps réel.

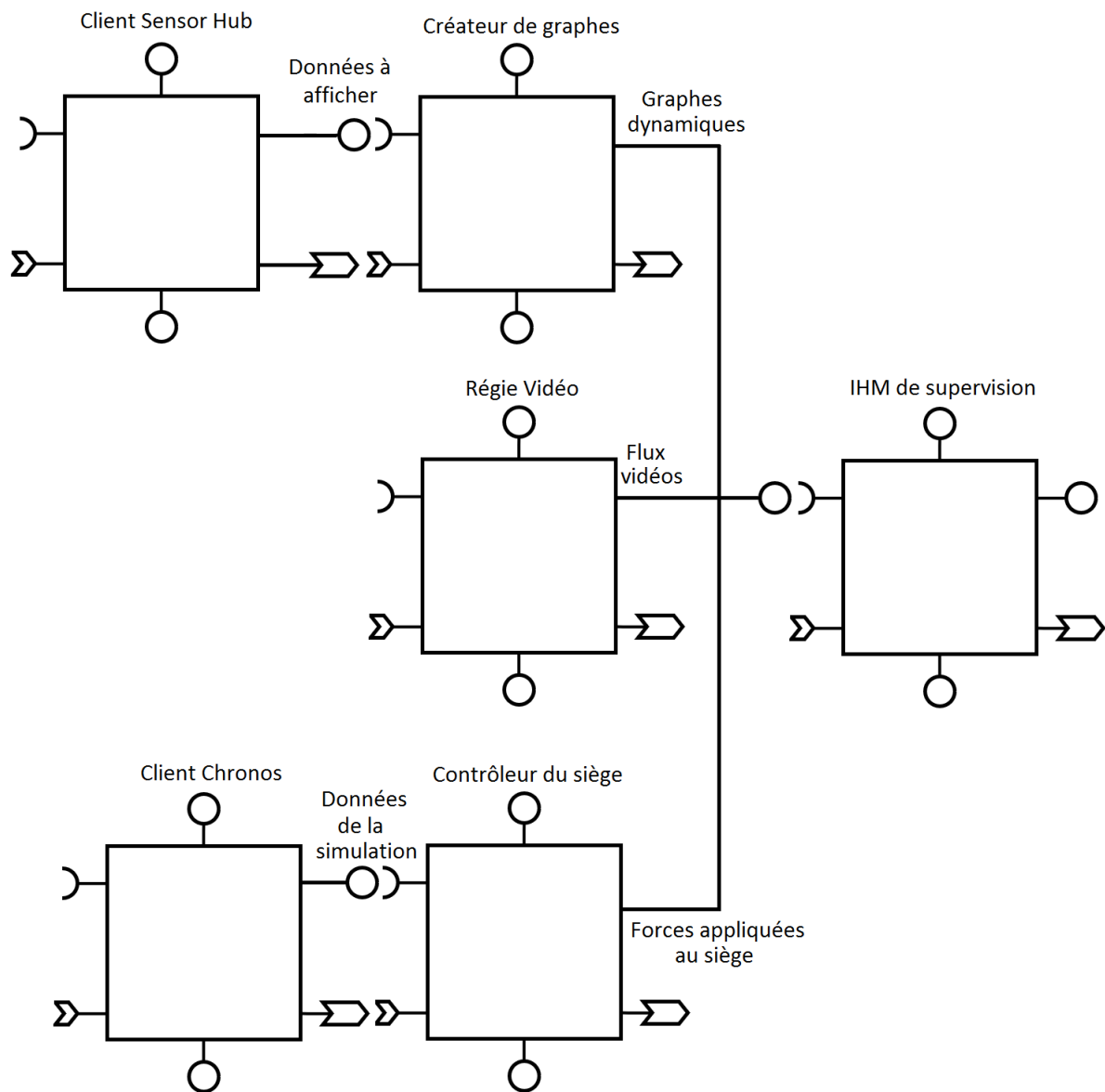


FIGURE 4.8 – Les composants Jacomo utilisés dans l'EIO.

## 4.2.2 Langages et protocoles utilisés

Tous les composants Jacomo de l'EIO sont développés dans le langage Java.

Ces composants utilisent plusieurs protocoles réseaux, implémentés par des bibliothèques logicielles. Nous allons dans cette section expliquer le fonctionnement des deux protocoles les plus importants de l'EIO : les websockets et l'architecture CORBA.

En effet, la communication entre le client et le serveur Sensor Hub se fait grâce aux websockets. Les informations calculées par le serveur Sensor Hub sont mis au format JSON<sup>4</sup>, puis sont envoyés périodiquement au client Sensor Hub via une websocket.

De plus, les données de la simulation Chronos permettant de piloter le siège dynamique sont récupérées au moyen d'une architecture CORBA (voir la section 2.1). La régie vidéo présentée à la section 4.1.1 utilise également une architecture CORBA : elle est basée sur un serveur CORBA offrant des fonctionnalités utiles comme l'affichage ou l'enregistrement des vidéos.

### Les websockets

Le protocole WebSocket est un protocole de la septième couche du modèle OSI (Open Systems Interconnection) : la couche applicative.

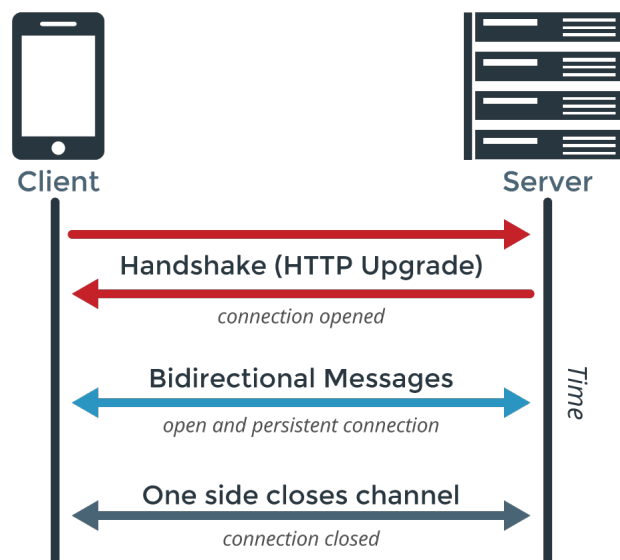


FIGURE 4.9 – Le fonctionnement du protocole WebSocket.

---

4. JavaScript Object Notation



Il a pour vocation de contourner les limites du protocole HTTP<sup>5</sup>, notamment le fait que la communication en HTTP soit « half duplex »<sup>6</sup>.

Le protocole HTTP reposant sur une architecture client/serveur, le client doit envoyer une requête, puis attendre la réponse du serveur. Le serveur ne peut donc pas envoyer de lui-même des informations au client.

Le fonctionnement du protocole (illustré par la Figure 4.9) est le suivant :

- La communication doit être initialisée par le client. Ce dernier envoie au serveur une requête HTTP indiquant via le champ « upgrade » que la communication doit désormais être faite en WebSocket, et non en HTTP. Cette étape est appelée le « handshake ».
- La communication peut ensuite se faire en « full duplex » : le serveur et le client peuvent simultanément s'envoyer des messages. Le serveur peut désormais envoyer des informations au client quand il le souhaite et n'a plus à attendre les sollicitations du client.
- La communication peut être stoppée à n'importe quel moment par le client ou le serveur.

## L'architecture CORBA

CORBA (Common Object Request Broker Architecture) est une architecture logicielle permettant de créer des composants complètement interopérables : ils peuvent travailler ensemble tout en étant développés dans des langages différents et en étant exécutés sur des ordinateurs différents, et même sur des systèmes d'exploitation différents.

Cette architecture est également de type client/serveur et met en œuvre des composants appelés « ORB<sup>7</sup> » et le protocole GIOP<sup>8</sup> associé.

En CORBA, le client est capable d'invoquer des méthodes (des fonctionnalités offertes par un objet) proposées par un serveur.

---

5. Hypertext Transfer Protocol

6. La transmission des données ne peut pas se faire simultanément dans les deux directions

7. Object Request Broker

8. General Inter ORB Protocol

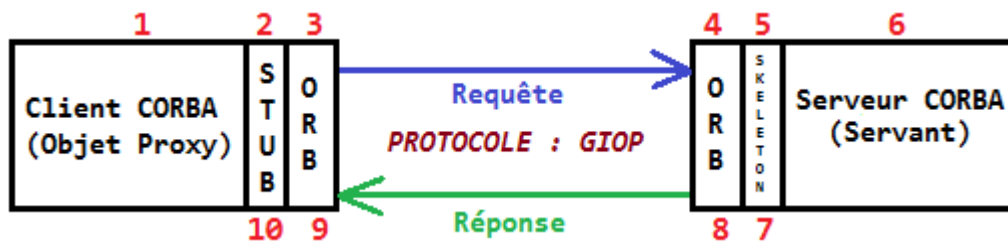


FIGURE 4.10 – Le fonctionnement de l'architecture CORBA.

Le fonctionnement général de l'architecture CORBA est le suivant (voir Figure 4.10) :

1. Le client CORBA invoque la méthode du serveur qu'il veut utiliser, qui est représentée par un objet appelé « Proxy ».
2. Le Stub est un objet côté client qui est chargé de préparer la requête CORBA adéquate qui doit être envoyée au serveur. Il s'agit de l'étape de « marshalling » ou encore de « sérialisation » de la requête CORBA.
3. L'ORB du client envoie la requête CORBA à travers le réseau, via le protocole GIOP.
4. L'ORB du serveur réceptionne la requête CORBA.
5. Le Skeleton est l'objet côté serveur qui a pour rôle de désérialiser (étape d'« unmarshalling ») la requête CORBA du client. Il s'agit du processus inverse de ce qui est effectué par le Stub à l'étape 2 : le Skeleton doit rendre la requête compréhensible par le serveur.
6. Le serveur CORBA exécute la méthode appelée par le client, qui est représentée par un objet appelé « Servant ». Le Servant est en fait une instance de la classe qui implémente la méthode en question.
7. Le Skeleton sérialise le résultat de la méthode du serveur en une réponse CORBA (marshalling).
8. La réponse CORBA est envoyée à travers le réseau par l'ORB du serveur.
9. L'ORB du client réceptionne la réponse CORBA.
10. Le Stub désérialise la réponse CORBA pour la rendre compréhensible par le client (unmarshalling).

Le client CORBA a ainsi pu faire appel à une méthode du serveur CORBA, ce dernier étant exécuté dans un processus différent. De plus, le client et le serveur peuvent très bien utiliser deux implémentations différentes de l'architecture CORBA, sans que cela ne pose problème.

# Chapitre 5

## Expériences

Au cours de ce Projet de Fin d'Étude, nous avons eu la possibilité de réaliser une démonstration de l'EIO. Cela a été l'occasion de valider le bon fonctionnement de l'EIO.

### 5.1 Contexte de l'expérience

#### 5.1.1 Le CESIM Santé de Brest

Le CESIM (CEntre de SIMulation) Santé est un Groupement d'Intérêt Scientifique (GIS) partagé entre l'Université de Bretagne Occidentale (UBO) et le Centre Hospitalier Régional Universitaire (CHRU) de Brest. Les locaux du CESIM se situent dans la faculté de médecine de Brest.

Il s'agit d'un centre de formation par la simulation. Leurs activités sont principalement liées au milieu hospitalier, bien que de nouvelles collaborations surviennent de temps à autre (avec les pompiers par exemple). Leurs simulations se basent sur de nombreux équipements de pointe : des mannequins simulateurs de patients pouvant reproduire des situations convulsions ou de saignements, des simulateurs de poumons, d'intubation, d'accouchements, etc.

Une de leurs équipes est ainsi venue voir l'avancée du développement et les résultats de ce stage.

### 5.1.2 La démonstration de l'EIO

L'EIO possédant une architecture modulaire, cette démonstration a été l'occasion pour nous d'essayer une nouvelle configuration (voir Figure 5.2).

Nous nous sommes ainsi tournés vers une configuration semblable à celle utilisée dans des porteurs légers de patrouille maritime (comme le Diamond DA42 par exemple, présenté en Figure). Le pilote et le SSO sont désormais côte à côte et regardent dans la même direction.



FIGURE 5.1 – Le porteur léger Diamond DA42.

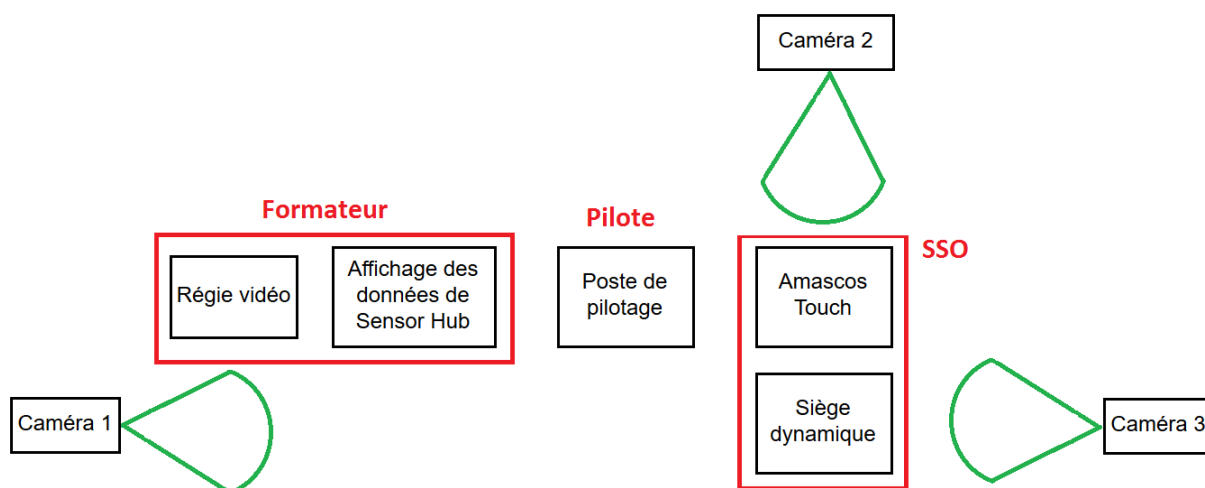


FIGURE 5.2 – La configuration spatiale de l'expérience avec le CESIM (en vue de dessus).

Le pilote et le SSO sont tous les deux incarnés par deux personnes du CESIM. Deux formateurs sont cette fois présents dans la salle : l'un pour le SSO et l'autre pour le pilote. En effet, les personnes du CESIM ignorent comment fonctionne Amascos Touch, mais également comment piloter l'avion de la simulation. Il est donc nécessaire de passer par une phase d'apprentissage de ces deux rôles.

C'est le SSO qui portera les trois capteurs physiologiques : la montre connectée, la ceinture pectorale et le bandeau frontal.

Nous pouvons observer ci-après deux photos d'ensemble de la démonstration. Celles-ci sont légèrement sombre, étant donné que nous avons créé une ambiance lumineuse tamisée dans la salle de formation.

La Figure 5.3 montre une vue d'ensemble de l'EIO, qui reprend la configuration spatiale présentée sur la Figure 5.2.

La Figure 5.4 montre (de gauche à droite) le formateur du pilote, le pilote, le SSO et le formateur du SSO.

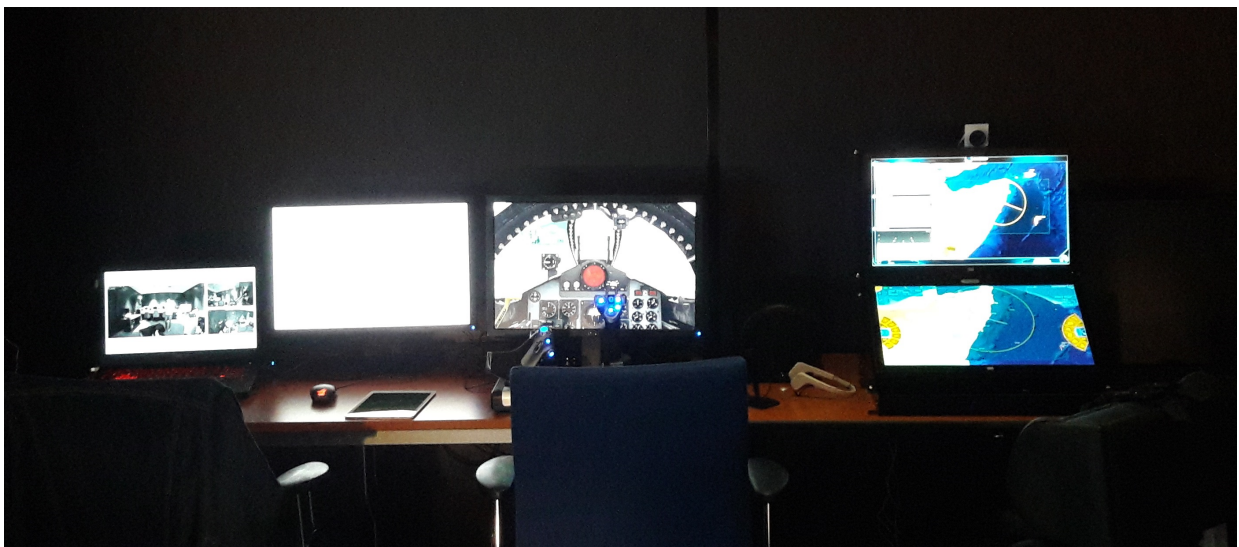


FIGURE 5.3 – Photo d'ensemble de l'EIO.



FIGURE 5.4 – Photo des formateurs, du pilote et du SSO.

La Figure 5.5 montre la régie vidéo de l'EIO.

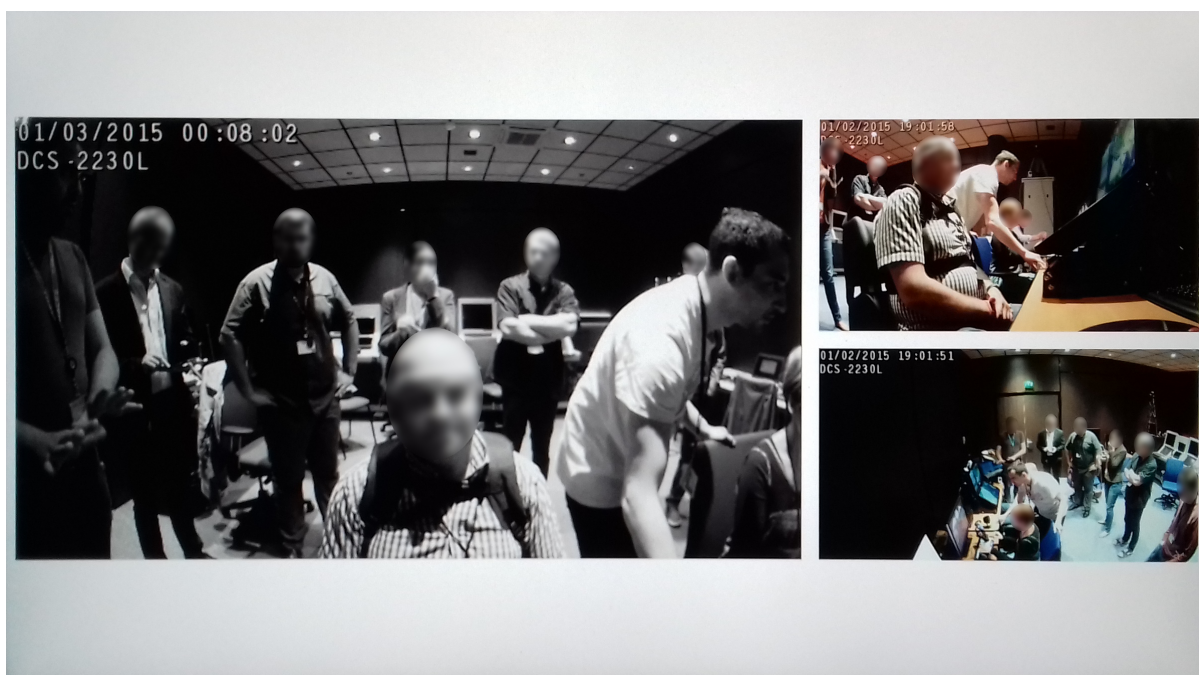


FIGURE 5.5 – Photo de la régie vidéo.

Le formateur dispose de deux vues d'ensemble situées sur le côté droit, ainsi que d'une plus grande vue sur la gauche se concentrant surtout sur le SSO (le SSO étant la personne formée).

Nous voyons que la caméra de gauche passe correctement en affichage nocturne (nous avons mis une ambiance lumineuse tamisée) et valide encore une fois le choix du modèle de caméra.

## 5.2 Résultats de l'expérience

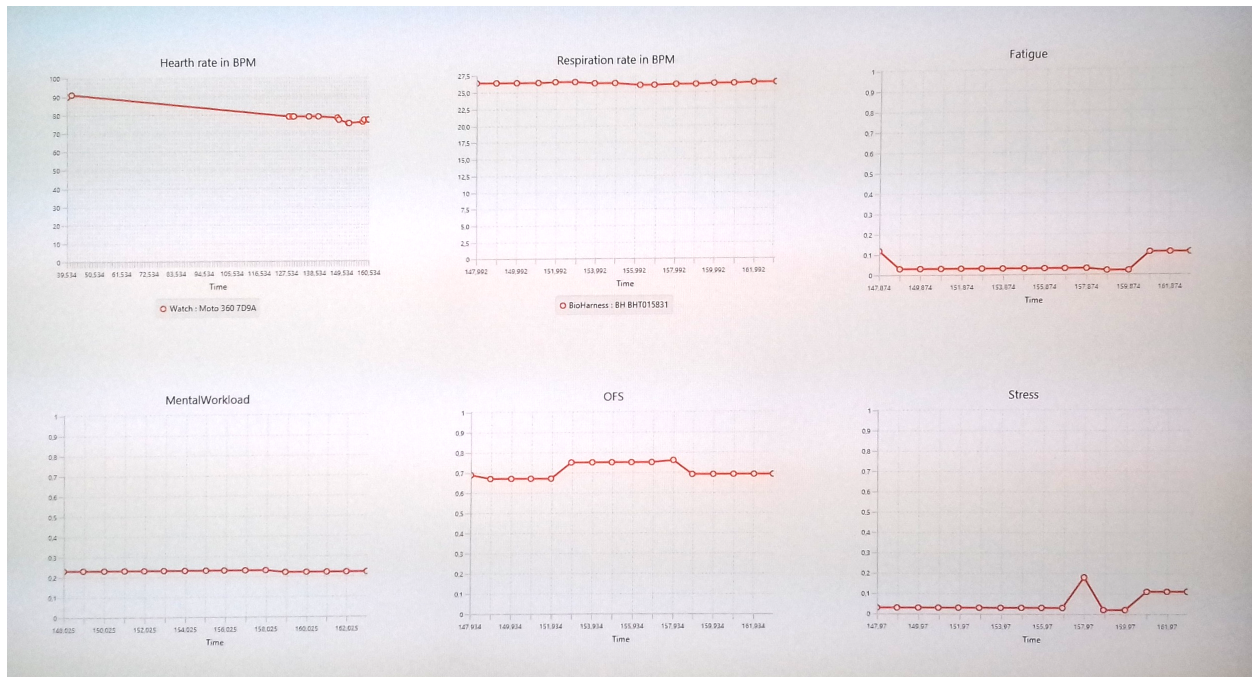


FIGURE 5.6 – Photo de l'affichage des données de Sensor Hub.

La Figure 5.6 correspond à l'affichage des données issues de Sensor Hub. Six caractéristiques du SSO sont représentées : (sur la première ligne, de gauche à droite) la fréquence cardiaque en battement par minute, la fréquence respiratoire, la fatigue, (sur la seconde ligne, de gauche à droite) la charge mentale, l'OFS et le stress.

L'axe horizontal de ces six graphes représente le temps écoulé depuis le début de la démonstration, en seconde. Les axes verticaux de la fréquence cardiaque et de la fréquence respiratoire sont exprimés en battement par minute (BPM). La fatigue, la charge mentale, l'OFS et le stress sont normalisés : leur axe vertical va de 0 à 1.

Par exemple, plus l'OFS est proche de 1, plus le SSO est efficace dans la réalisation de sa tâche. Nous pouvons ainsi voir qu'une augmentation de la fatigue (par exemple à la fin de la courbe de fatigue), provoque une diminution de l'OFS. Cela est bien en accord avec la définition de l'OFS donnée dans le chapitre 3.

Il est à noter que lorsque la photo a été prise, le graphe de la fréquence cardiaque a mis un certain temps avant d'afficher des valeurs : au début de la courbe de fréquence cardiaque (« Heart rate in BPM »), peu de valeurs sont présentes. Cela est dû au fait que Sensor Hub met un certain temps à effectuer la calibration de la montre connectée.



# Conclusion

Ma contribution à l'Environnement Immersif Opérateur s'est faite en plusieurs temps. Tout d'abord, j'ai testé et validé l'ensemble du matériel qui était prévu pour l'EIO. J'ai également sélectionné de nouveaux équipements qui pourraient apporter plus de fonctionnalités et de réalisme à la formation.

J'ai réussi à faire fonctionner et communiquer tout ces équipements ensemble, afin d'être capable de suivre des scénarios de formation.

J'ai finalement créé l'architecture de l'EIO, ainsi que tous les outils de base qui lui sont associés.

J'ai donc réussi à remplir une partie des exigences initiales :

- Que ce soit pour la partie matérielle ou logicielle, l'EIO possède une architecture modulaire lui permettant d'exécuter des scénarios de formation variés. De plus, cette architecture lui permet d'être très facilement déployable dans le cadre de démonstrations ou de phase de création d'IHM.
- L'ensemble des équipements que nous avons choisis est parfaitement adéquat et confère une dimension immersive aux formations.
- L'EIO est capable de mesurer la performance d'un individu de manière non-intrusive et offre des moyens de supervision au formateur.
- Nous sommes capables d'enregistrer les données provenant de Sensor Hub et les flux vidéo des caméras de surveillance, afin de pouvoir rejouer en partie des sessions de formation.

Nous pouvons ajouter que l'ensemble des composants logiciels que j'ai développés pour l'EIO pourront être réutilisés pour de futurs projets de Thales.

Mes encadrants sont ainsi très satisfaits de l'ensemble de mon travail.

Les prochaines améliorations que j'estime nécessaires pour l'EIO concernent le rejeu de mission et le contrôle à la volée.

Le rejeu de mission doit à terme être capable de proposer une « timeline » permettant de restituer l'état de l'ensemble des composants de l'EIO à un instant donné du scénario.

Le contrôle à la volée devra permettre au formateur de modifier en temps réel le scénario de formation et la configuration de certains équipements.



# Bibliographie

- [1] B.K. Bracken, N. Palmon, V. Romero, J. Pfautz, and N. J. Cooke. A prototype toolkit for sensing and modeling individual and team state. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 58, 2014.
- [2] G. Durantin, J.-F. Gagnon, S. Tremblay, and F. Dehais. Using near infrared spectroscopy and heart rate variability to detect mental overload. *Behavioural Brain Research*, 259, 2014.
- [3] K. Durkee, A. Geyer, S. Pappada, A. Ortiz, and S. Galster. Real-time workload assessment as a foundation for human performance augmentation. *Lecture Notes in Computer Science - Foundations of Augmented Cognition*, 8027, 2013.
- [4] RTO Human Factors and Medicine Panel Task Group. Operator functional state assessment. Technical report, North Atlantic Treaty Organization (NATO), 2004.
- [5] J.H. Friedman. Greedy function approximation : A gradient boosting machine. *The Annals of Statistics*, 29, 2001.
- [6] J.H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38, 2002.
- [7] J.-F. Gagnon, O. Gagnon, D. Lafond, and S. Tremblay. A systematic assessment of operational metrics for modeling operator functional state. In *3rd International Conference on Physiological Computing Systems - Volume 1 : PhyCS, Lisbon, Portugal*, 2016.
- [8] J.-F. Gagnon, D. Lafond, M. Parent, and S. Tremblay. Modélisation psychophysiologique de la charge mentale : Entraînement d'un modèle générique. In *Semaine de l'Intelligence Numérique 2016, Québec, Canada*, 2016.
- [9] O. Gagnon, M. Parizeau, D. Lafond, and J.-F. Gagnon. Comparing methods for assessing operator functional state. In *IEEE 2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA) - San Diego, CA, USA (2016.3.21-2016.3.25)*, 2016.
- [10] J. Hailstone and Andrew E. Kilding. Reliability and validity of the zephyr<sup>TM</sup> bioharness<sup>TM</sup> to measure respiratory responses to exercise. *Measurement in Physical Education and Exercise Science*, 15, 2011.

- 
- [11] S.G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50, 2006.
- [12] J.-H. Kim, R. Roberge, J. B. Powell, A. B. Shafer, and W. Jon Williams. Measurement accuracy of heart rate and respiratory rate during graded exercise and sustained exercise in the heat using the zephyr bioharness<sup>TM</sup>. *International Journal of Sports Medicine*, 34, 2012.
- [13] P. Lemberger, M. Batty, M. Morel, and J.-L. Raffaëlli. *Big Data et Machine Learning : Les concepts et outils de la data science*. Dunod, 2016.
- [14] Y. Maeda, M. Sekine, and Tamura T. The advantages of wearable green reflected photoplethysmography. *Journal of Medical Systems*, 35, 2011.
- [15] S.C. Mukhopadhyay. Wearable sensors for human activity monitoring : A review. *IEEE Sensors Journal*, 15, 2015.
- [16] M. Parent, J.-F. Gagnon, T.H. Falk, and S. Tremblay. Modeling the operator functional state for emergency response management. In *ISCRAM 2016 Conference Proceedings - 13th International Conference on Information Systems for Crisis Response and Management, Rio de Janeiro, Brazil, 2016*.
- [17] N. Regis, F. Dehais, C. Tessier, and J.-F. Gagnon. Ocular metrics for detecting attentional tunnelling. In *Human Factors and Ergonomics Society Europe Chapter Annual Meeting, October 2012, Toulouse, France, 2012*.
- [18] S.R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems Man and Cybernetics*, 21, 1991.
- [19] D. Surangsriat and A. Intarapanich. Analysis of the meditation brainwave from consumer eeg device. In *IEEE SoutheastCon 2015 - Fort Lauderdale, FL, USA (2015.4.9-2015.4.12)*, 2015.
- [20] S. Tremblay, J.-F. Gagnon, D. Lafond, M. Rivest, and F. Couderc. Sensor-hub : A real-time data integration and processing nexus for adaptive c2 systems. In *6th International Conference on Adaptive and Self-Adaptive Systems and Applications, Venice, Italy, 2014*.
- [21] G.F. Wilson and C.A. Russell. Operator functional state classification using multiple psychophysiological features in an air traffic control task. *Human Factors The Journal of the Human Factors and Ergonomics Society*, 45, 2003.

# Liste des sigles et acronymes

<b>ACNS</b>	<i>American Clinical Neurophysiology Society</i>
<b>CESIM</b>	<i>CEntre de SIMulation</i>
<b>CSS</b>	<i>Customer Support Services : Soutien et service client</i>
<b>ECG</b>	<i>Électrocardiogramme</i>
<b>EEG</b>	<i>Électroencéphalogramme</i>
<b>EIO</b>	<i>Environnement Immersif Opérateur</i>
<b>IHM</b>	<i>Interface Homme-Machine</i>
<b>INRS</b>	<i>Institut National de la Recherche Scientifique</i>
<b>NASA-TLX</b>	<i>NASA-Task Load Index</i>
<b>OFS</b>	<i>Operator Functional State : État fonctionnel de l'opérateur</i>
<b>PPG</b>	<i>Photopléthysmogramme</i>
<b>SSO</b>	<i>Sensor System Operator</i>
<b>TACCO</b>	<i>Tactical Commander</i>
<b>TRT</b>	<i>Thales Research and Technology</i>
<b>TSA</b>	<i>Thales Systèmes Aéroportés</i>



# Liste des figures

1.1	Un avion ATL 2 de la marine nationale en mission de patrouille maritime.	3
1.2	Une console Amascos Touch. . . . .	4
1.3	Captures d'écran de Amascos Touch. . . . .	5
1.4	Exemple de scénario de patrouille maritime. . . . .	7
2.1	Jaquette et capture d'écran de la simulation de vol X-Plane. . . . .	10
2.2	Le système de contrôle de vol X52 de Saitek. . . . .	11
2.3	Le système de contrôle de vol HOTAS Warthog de la marque Thrustmaster.	11
2.4	Le siège dynamique GS-105 de Geko Systems. . . . .	12
2.5	Positions du pilote et du SSO dans un porteur lourd de patrouille maritime.	13
2.6	Le micro-casque Gamecom 777 de Plantronics. . . . .	14
2.7	Le kit audio GigaWorks T3 de Creative. . . . .	14
2.8	Le chauffage soufflant AM09 de Dyson. . . . .	15
2.9	L'ensemble de lumière connectée Hue de Philips. . . . .	16
3.1	Un exemple de questionnaire de type NASA-TLX. . . . .	20
3.2	Le logo de l'outil Sensor Hub. . . . .	22
3.3	Une vue globale de l'outil Sensor Hub. . . . .	23
3.4	Une représentation pyramidale des informations fournies par Sensor Hub. .	23
3.5	La montre connectée MOTO 360 Sport de Motorola. . . . .	25
3.6	Un exemple de photopléthysmogramme. . . . .	26
3.7	La ceinture pectorale BioHarness de Zephyr. . . . .	26
3.8	Le bandeau frontal Muse. . . . .	28
3.9	Représentation du système international 10-20. . . . .	29

---

3.10	Exemple de spectrogrammes obtenus grâce au bandeau frontal Muse . . . .	31
3.11	Un exemple d'arbre de décision. . . . .	33
3.12	Illustration de la méthode de descente du gradient . . . . .	35
4.1	Exemple de configuration de l'EIO. . . . .	42
4.2	La tablette tactile Galaxy Tab S2 de Samsung. . . . .	43
4.3	La caméra IP DCS-2230L de D-Link. . . . .	44
4.4	Le routeur Wi-Fi RT-AC88U de Asus. . . . .	45
4.5	Schéma global du matériel de l'EIO. W = Wi-Fi, J = Jack, B = Bluetooth.	46
4.6	La représentation d'un composant Jacomo. . . . .	47
4.7	Le composant Jacomo d'exploitation du client Sensor Hub permettant d'afficher des graphes. . . . .	48
4.8	Les composants Jacomo utilisés dans l'EIO. . . . .	49
4.9	Le fonctionnement du protocole WebSocket. . . . .	50
4.10	Le fonctionnement de l'architecture CORBA. . . . .	52
5.1	Le porteur léger Diamond DA42. . . . .	54
5.2	La configuration spatiale de l'expérience avec le CESIM (en vue de dessus).	54
5.3	Photo d'ensemble de l'EIO. . . . .	55
5.4	Photo des formateurs, du pilote et du SSO. . . . .	56
5.5	Photo de la régie vidéo. . . . .	56
5.6	Photo de l'affichage des données de Sensor Hub. . . . .	57



# Liste des tables

3.1	Une partie de la nomenclature du système international 10-20. . . . .	29
3.2	Les bandes de fréquences caractéristiques des ondes cérébrales. . . . .	30
3.3	Un ensemble d'observations devant servir à l'élaboration d'un arbre de décision.	33