

# Wearable and Self-Aware Machine Learning System for Online Cognitive Workload Monitoring and Drone Control

Présentée le 2 décembre 2020

à la Faculté des sciences et techniques de l'ingénieur  
Laboratoire des systèmes embarqués  
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

**Fabio Isidoro Tiberio DELL'AGNOLA**

Acceptée sur proposition du jury

Prof. D. Floreano, président du jury  
Prof. D. Atienza Alonso, Prof. D. R. Barrettino, directeurs de thèse  
Prof. L. Marchal-Crespo, rapporteuse  
Prof. T. Theocharides, rapporteur  
Prof. W. Karlen, rapporteur





There comes a point when a dream becomes reality  
and reality becomes a dream.  
— Frances Farmer

To my parents...



# Acknowledgments

I would like to start the acknowledgments with an anecdote. When I was little, I used to spend my holidays with my family at the campsite in Morges. I remember one day, during a bike trip to Lausanne and next to the Université de Lausanne (UNIL), I thought: *"Wow, people there should be incredibly smart! That's not for me!"* But eventually, I ended up doing a Ph.D. at the École Polytechnique Fédérale de Lausanne (EPFL). With these few lines, I want to say that we have to believe in ourselves and do not discard any options, even if some of them seem unreachable. Definitely, to successfully reach this unexpected point in my life, I received precious help and support of many people. Therefore, it is my pleasure to thank those who made this journey possible and allowed me to live this intense and enriching experience.

First of all, I would like to express my deepest gratitude to my thesis director **Prof. David Atienza** for his constant and excellent supervision. Since the beginning, he has established a friendly and social working environment. He gave me the freedom to investigate this attractive topic, which allowed me to learn a lot. Although we entered a relatively new field of research, he was constantly some steps further than me and always ready to help. Even if he was always some steps further, I always felt like having him on my side, more than me working for him. Having such a great example and reference helped me not to lose the right way and allowed me to grow my research-thinking capabilities to be able to get my Ph.D. successfully.

Then, I would like to offer my special thanks to my thesis co-director **Prof. Diego Barrettino** for trusting in my capabilities, for helping me to start this journey, and in particular, for correcting my thesis and for providing valuable and constructive suggestions.

Moreover, I wish to acknowledge the jury members, **Prof. Dario Floreano**, **Prof. Walter Karlen**, **Prof. Laura Marchal-Crespo**, and **Prof. Theocharis Theocharides**, who honored me by agreeing to evaluate this work and provided highly valuable feedback.

For their silent and selfless daily assistance, I would like to thank our administra-

## Acknowledgments

---

tive assistants **Homeira Salimi** and **Francine Eglese**, and our IT masters, **Rodolphe Buret** and **Mikaël Doche**.

Then, I would like to thank all my officemates: **Rubén Braojos** for being such an example, **Elisabetta De Giovanni** for taking the time to listen to my problems, **Farnaz Forooghifar** for her amazing homemade cake, and **Leila Cammoun** for her kindness and help. They all created such a pleasant atmosphere and were one of the reasons why I enjoyed going to work every day.

Next, I would like to thank **Hossein Mamaghanian**, the gentlemann who handed me his place in ELG 133. It was for me a privilege to get his place and to take care of the elephant on the whiteboard.

I am also particularly grateful for the support given by all the present and past scientists from Embedded System Laboratory (ESL). In particular, I would like to thank **Francisco Rincon**, **Srinivasan Murali**, **Pablo Garcia**, **Adriana Arza**, **Miguel Peon**, **Dennis Majoe**, **Marina Zapater**, **Alexandre Levisse**, **Amir Aminifar**, and **Tomas Teixeira**, for taking the time to discuss my work and to answer my questions.

I would also like to extend my thanks to all the past and present Ph.D. students from the lab. First, **Grégoire Surrel**, **Una Pale**, **Renato Zanetti**, and **Alessandro Fontanella**, who actively helped me obtain the results presented in this work.

Second, for all the awesome moments we spent together, thanks to **Artem Andreev**, **Damián Pascual**, **Ali Pahlevan**, **Loris Duch**, **Soumya Basu**, **Dionisije Sopic**, **Arman Iranfar**, **Yasir Qureshi**, **Halima Najibi**, **Benoît Denkinger**, **Andrew Simon**, **Anthony Thomas**, **Flavio Ponzina**, **Fabio Montagna**, and many others.

Finally, I would like to thank all the people who joined ESL for a project or internship and contributed to prove that this work is not only theoretical. Thanks to **Victoriano Montesinos**, **Arthur Gay**, **Quentin Herzig**, **Kay Lächler**, **Rodrigo Mariño**, **Karen Flores**, **Giulio Masinelli**, **Tanguy Rossel**, **Niloofar Momeni**, and **Carlos Megías**. It was a privilege to supervise your projects.

I would also like to thank those people from other departments with whom I collaborated during my Ph.D.. In particular, my thanks go to **Stefano Mintchev**, **Alexandre Cherpillod**, **Carine Rognon**, and **Alice Concordel**, from the Laboratory of Intelligent Systems (LIS); To **Prof. José del R. Millán**, **Ricardo Chavarriaga**, and **Ping-Keng Jao**, from the Defitech Chair on Brain-Machine Interface of the Center for Neuroprosthetics and Bioengineering Institute (CNBI); And to **Prof. Silvestro Micera**, **Martina Coscia**, and **Jenifer Miehlebradt**, from the Translational Neural Engineering (TNE) Laboratory. I had the chance to work with them in the Symbiotic Drone project, where I gained a lot of experience that made me grow as a scientist and as a person. I

would also like to thank the workshop for the printed circuit board design at EPFL. In particular, my thanks go to **Peter Bruehlmeier**, who handled the routing of my PCB layout and was always available to help me.

All this would not have been possible without the help of my friends. In particular, thanks to **Sandra** and **Andrea Bernaschina** for bringing me to the right place and at the right moment, and for the precious advice given by **Gioele Balestra** to enroll at EPFL, so that I was able to find a Ph.D. position that best suited to my interests. My special thankfulness goes to **Annick Baur** for bringing some sun to Lausanne almost every Wednesday. I appreciated all our *"parenthèses suspendues un peu hors du temps"*, during which we had enjoyable running sessions and pleasant conversations. I am glad I have met such a unique and great friend. I am also thankful to **Mauro Rossi**, a mountain guide with whom I had the chance to share a lot of great adventures in the mountains. Now, we can eventually go for our serious summit projects. I shall also take this opportunity to thank my friends from the legendary Demo Team Airolo, namely, **Amos Gianella**, **Robert Brogioli**, **Simone Leonardi**, **Mattia Scanzio**, **Nahyeli Mottini**, **Francesca Leoni**, **Paola Forni**, **Monica Frisoni**, **Claudia D'Andrea**, and **Elisa Scossa-Romano**, as well as all other members of the Swiss Ski and Snowboard School of Airolo. I have been absent for a while, but I will probably soon request a tower flyby.

While in Lausanne, I had the chance to serve as a firefighter at the fire station of Chamberonne. For that, I want to thank **Benedict Fasel**, who introduced me to the officers allowing me to join the team. It was for me a privilege to work there, where I learned a lot. Therefore, I would like to extend my thanks to all members of the SDIS Chamberonne, and especially to **Boris Ouvry**, and **Elisa Mühleman** for their kind friendship. My gratitude also goes to all my friends and colleagues firefighters from Faido. In particular, I am thankful for being so understanding concerning my long absence and for saving my place. Let me also express my thankfulness to all the people I forget to mention, but that I met during my life, gladdened my journey, and contributed to making me grow as a person and becoming a scientist. Please forgive me if I forget to mention someone in particular, here the list is too long, so thank you all.

The next lines are to express my highest gratitude to my parents **Dina** and **Silvano**. Their infinite love, continuous guidance, and support have allowed me to achieve many goals in my life, especially this one. There are no words to describe how thankful I am, neither to express my love for them. Special thanks also go to my

## Acknowledgments

---

brother **Sandro**, to his wife **Pamela**, and especially to my little nephew **Aron**, who always spreads positive energy. Of course, my thanks go to all my aunts, uncles, and cousins as well.

Last but not least, my very great appreciation goes to **Christa Schwendimann**. She was able to understand my dreams, my needs, and my problems. For many years, she selflessly devoted herself to support me in a million ways. Thanks Christa, for all that you have done for me. I will never forget.

*Lausanne, November 5, 2020*

Fabio Dell'Agnola

# Abstract

The use of robots in search and rescue is gaining particular interest, but singular skills are required to ensure efficient deployments in real missions. To face this problem, there is a need to develop more intuitive control interfaces. Moreover, to ensure high performance during high cognitive demanding tasks, such as in search and rescue missions, there is a need to combine both human's and robot's skills. Respectively, humans and robots can adapt to new situations and optimize the execution of repetitive tasks. In this regard, novel share-control techniques have been developed to adapt the human-robot interaction, but to dynamically adapt this interaction, the information about the human state is missing.

To address these problems, I first developed a novel wearable system that enhances the control of drones providing a more intuitive flying experience. As shown in chapter 2, this wearable system tracks the upper body movements and translates them into commands for a drone. The system has been tested with a simulator and demonstrated for the teleoperation of a real drone. Moreover, to ensure enduring operations, I proposed a method that drastically reduces communication, and consequently, improves energy efficiency by 11.9%.

Second, in chapter 3, I presented a machine-learning approach for monitoring the cognitive workload level of a drone operator involved in search and rescue missions. My model combines the information of different features extracted from physiological signals, such as respiratory activity, electrocardiogram, photoplethysmogram, and skin temperature, acquired in a non-invasive way. To reduce both subject and day inter-variability of the signals, I explored different feature normalization techniques. Moreover, I adjusted the learning method for support vector machines to allow subject-specific optimizations. On a test set acquired from 34 volunteers, the proposed model distinguished between low and high cognitive workloads with an average accuracy of 87.3% and 91.2%, while controlling a drone simulator using both a traditional controller and the proposed FlyJacket design, respectively.

## Abstract

---

Third, in chapter 4, I presented the integration of the method developed for cognitive workload monitoring, on a new single wearable embedded system, that also integrates the proposed drone controller design. On the hardware side, it includes a multi-channel physiological signals acquisition and a low-power processing platform that is suited for cognitive workload monitoring. On the software side, the proposed system includes novel energy-aware bio-signal processing and embedded machine learning methods. Moreover, to exploit the trade-offs between the required accuracy of the available energy of the system, I presented a new application of the concept of a scalable machine-learning method with different power-saving levels. Results showed that the proposed self-aware approach yields an increase of 78% of the battery lifetime without really affecting the classification accuracy.

The proposed system, comprising a drone controller integrating a unit for cognitive workload monitoring, lays the foundations for the development of new-generation human-robot interfaces. With the information about the human state, we can close the loop of traditional share-control techniques, which will be able to dynamically adapt the level of interaction with semi-autonomous machines based on the need for the operator.

**Keywords:** Cognitive Workload, Online Monitoring, Physiological Signals, Sensor Fusion, Machine Learning, Human-Robot Interaction, Wearable, Embedded Systems, Drones, Search and Rescue Missions.



# Zusammenfassung

Bei der Suche und der Rettung gewinnt der Einsatz von Robotern besonderes Interesse. Um effiziente Einsätze in realen Missionen zu gewährleisten, sind aber einzigartige Fähigkeiten unabdingbar. Um dieses Problem zu lösen, müssen intuitivere Steuerungsschnittstellen entwickelt werden. Ausserdem muss eine hohe Leistung bei kognitiv anspruchsvollen Aufgaben gewährleistet werden, z.Bsp. ist es notwendig, bei Such- und Rettungseinsätzen, die Fähigkeiten von Mensch und Roboter zu kombinieren. Entsprechend können Mensch und Roboter sich an neuere Situationen anpassen und die Ausführung von sich wiederholenden Aufgaben optimieren. In dieser Hinsicht wurden neuartige Techniken der gemeinsamen Steuerung entwickelt, um die Mensch-Roboter-Interaktion anzupassen. Um diese Interaktion dynamisch anzupassen, fehlen aber die Informationen über den menschlichen Zustand.

Um diese Probleme anzugehen, habe ich zunächst ein neuartiges tragbares System entwickelt, das die Steuerung von Drohnen verbessert und ein intuitiveres Flug-erlebnis ermöglicht. Wie im Kapitel 2 gezeigt, verfolgt dieses tragbare System die Bewegungen des Oberkörpers und setzt sie in Befehle für die Drohne um. Das System wurde mit einem Simulator getestet und wurde für die Teleoperation einer echten Drohne demonstriert. Um einen dauerhaften Betrieb zu gewährleisten, habe ich darüber hinaus eine Methode vorgeschlagen, die die Kommunikation zwischen Mensch und Drohne drastisch reduziert. Infolgedessen wird die Energieeffizienz um 11.9% verbessert.

Des weiteren habe ich im Kapitel 3 einen maschinellen Lernansatz für die Überwachung der kognitiven Arbeitsbelastung eines Drohnenoperators vorgestellt, der an Such- und Rettungseinsätzen beteiligt ist. Mein Modell kombiniert die Informationen verschiedener Merkmale, die aus physiologischen Signalen extrahiert werden wie z.B. Atmungsfrequenz, Elektrokardiogramm, Photoplethysmogramm und Hauttemperatur, die auf nicht-invasive Weise erfasst werden. Um die Variabilität der Signale sowohl zwischen den Probanden als auch zwischen den Tagen zu reduzieren, untersuchte ich verschiedene Techniken zur Merkmalsnormalisierung. Darüber

## Abstract

---

hinaus passte ich die Lernmethode für Support-Vektor-Maschinen an, um fachspezifische Optimierungen zu ermöglichen. Auf einem Testsatz, der von 34 Freiwilligen erworben wurde, unterschied das vorgeschlagene Modell zwischen niedrigen und hohen kognitiven Arbeitsbelastungen mit einer durchschnittlichen Genauigkeit von 87,3% und 91.2%, während ein Drohnensimulator sowohl mit einem traditionellen Controller als auch mit dem vorgeschlagenen FlyJacket-Design gesteuert wird.

Drittens habe ich in Kapitel 4 die Integration, der für die Überwachung der kognitiven Arbeitsbelastung entwickelten Methode, auf einem neuen einzelnen, tragbaren eingebetteten System vorgestellt. Das System integriert auch das vorgeschlagene Design des Drohnen Controllers. Auf der Hardwareseite umfasst es eine mehrkanalige physiologische Signalerfassung und eine leistungsarme Verarbeitungsplattform, die für die Überwachung der kognitiven Arbeitsbelastung geeignet ist. Auf der Softwareseite umfasst das vorgeschlagene System eine neuartige energiebewusste Biosignalverarbeitung und eine eingebettete Methode des maschinellen Lernens. Um darüber hinaus die Kompromisse zwischen der erforderlichen Genauigkeit der verfügbaren Energie des Systems auszunutzen, stellte ich eine neue Anwendung des Konzepts einer skalierbaren maschinellen Lernmethode mit verschiedenen Energiesparstufen vor. Die Ergebnisse zeigten, dass der vorgeschlagene selbstbewusste Ansatz zu einer Erhöhung der Batterielebensdauer um 78% führt, ohne die Klassifizierungsgenauigkeit signifikant zu beeinträchtigen.

Das vorgeschlagene System, bestehend aus einem Drohnen-Controller, der eine Einheit zur Überwachung der kognitiven Arbeitsbelastung integriert, legt den Grundstein für die Entwicklung von Mensch-Roboter-Schnittstellen der neuen Generation. Mit den Informationen über den menschlichen Zustand können wir die Schleife traditioneller Techniken zur gemeinsamen Steuerung schliessen, die in der Lage sein werden, den Grad der Interaktion mit halbautonomen Maschinen auf der Grundlage der Bedürfnisse des Bedieners dynamisch anzupassen.

Stichwörter: Kognitive Arbeitsbelastung, Online-Überwachung, Physiologische Signale, Sensor-Fusion, Maschinelles Lernen, Mensch-Roboter Interaktion, Tragbar, Eingebettete Systeme, Drohnen, Such- und Rettungseinsätze.

# Résumé

L'utilisation de robots dans le domaine de la recherche et du sauvetage a récemment suscité un intérêt particulier. Cependant, des compétences singulières sont nécessaires pour assurer une bonne efficacité dans des missions réelles. Pour faire face à ce problème et faciliter le travail des sauveteurs, il est nécessaire de développer des interfaces de contrôle plus intuitives. De plus, il est connu que les humains, contrairement aux robots jouissent d'une grande capacité d'adaptation aux situations inconnues alors que les robots ont une forte capacité à optimiser l'exécution de tâches répétitives. Donc, afin de garantir des performances élevées lors de l'exécution de tâches cognitivement exigeantes, il est nécessaire de combiner les compétences humaines et robotiques. À ce sujet, des études récentes ont développé de nouvelles techniques de contrôle, qui grâce à l'intelligence artificielle, arrivent à adapter le niveau d'interaction entre l'homme et les robots. Cependant, afin d'adapter dynamiquement cette interaction entre l'homme et le robot, il manque l'information sur l'état du pilote.

Pour résoudre ces problèmes, j'ai d'abord développé un nouveau système embarqué portable qui améliore le contrôle des drones en offrant une expérience de vol plus intuitive et immersive. Comme présenté dans le chapitre 2, ce système portable traduit les mouvements du haut du corps en commandes pour le drone. Le système a été testé avec un simulateur et utilisé pour la téléopération d'un vrai drone. De plus, pour assurer une longue durée de fonctionnement, j'ai proposé une méthode qui réduit considérablement la communication et qui par conséquent améliore aussi l'efficacité énergétique de 11.9%.

Dans un second temps, dans le chapitre 3, j'ai présenté une méthode pour surveiller l'effort cognitif d'un pilote de drones. Mon modèle combine des caractéristiques extraites depuis des signaux physiologiques, tels que l'activité respiratoire, l'électrocardiogramme, le photopléthysmogramme et la température cutanée. Ces mesures sont acquises de manière non invasive. Pour réduire l'inter-variabilité des signaux

## Abstract

---

entre les sujets et entre les jours, j'ai aussi exploré différentes techniques pour normaliser ces caractéristiques physiologiques. De plus, j'ai introduit une nouvelle méthode d'apprentissage pour des machines à vecteurs de support, qui convient aux optimisations requises par chaque sujet. Sur un ensemble de tests acquis parmi 34 volontaires, le modèle proposé fait la distinction entre des efforts cognitives faibles et élevées, avec une précision moyenne de 87.3% et 91.2%, en contrôlant un simulateur de drone avec à la fois une télécommande traditionnelle ou le nouveau design, appelé FlyJacket.

Finalement, dans le chapitre 4, j'ai présenté l'intégration de la méthode développée pour la surveillance de l'effort cognitif sur un nouveau et unique système embarqué, qui intègre également le nouveau contrôleur de drone proposé. Côté hardware, il comprend des capteurs pour l'acquisition des signaux physiologiques et une plateforme de traitement adapté à la surveillance cognitive de la charge de travail. Du côté logiciel, le système portable proposé comprend un nouveau traitement du signal sensible à la consommation énergétique et des méthodes d'apprentissage automatique intégrées. De plus, pour exploiter les compromis entre la précision requise et de l'énergie disponible du système, j'ai présenté une nouvelle application du concept d'une méthode d'apprentissage automatique évolutive avec différents niveaux énergétiques. Les résultats ont montré que l'approche proposée permet une augmentation de 78% de la durée de vie de la batterie sans dégradation significative de la précision de la classification.

Le système proposé, comprenant un contrôleur de drone et intégrant une unité de surveillance de l'effort cognitif, pose les bases pour le développement d'interfaces homme-machine de nouvelle génération. Avec les informations sur l'état du pilote, nous pouvons compléter les techniques traditionnelles de contrôle partagée, qui pourront adapter dynamiquement le niveau d'interaction avec des machines semi-autonomes en fonction du besoin de l'opérateur.

Mots clefs : Charge cognitive, Monitoring, Signaux physiologiques, Fusion de capteurs, Apprentissage automatique, Interaction homme-machine, Technologie portable, Système embarqué, Drones, Missions de recharge et sauvetage.

# Riassunto

L'utilizzo dei robot sta riscuotendo un particolare interesse anche nel campo della ricerca e del salvataggio. Tuttavia, per garantirne un buon utilizzo in missioni reali, sono necessarie competenze elevate. Per affrontare questo problema e facilitare il lavoro dei soccorritori, è necessario sviluppare interfacce di controllo più intuitive. Inoltre, è risaputo che gli esseri umani possono adattarsi a nuove situazioni, mentre i robot possono ottimizzare l'esecuzione di compiti ripetitivi. Di conseguenza, per garantire l'esecuzione di compiti cognitivamente molto pesanti, è necessario combinare le abilità di entrambi, vale a dire quelle dell'uomo e quelle dei robot. A questo proposito, sono state recentemente sviluppate delle tecniche di controllo condiviso, le quali cercano di adattare il livello di interazione tra uomo e robot, ma per poterlo fare dinamicamente, manca l'informazione sullo stato del pilota.

Per affrontare questi problemi, ho dapprima sviluppato un nuovo sistema integrato e indossabile che rende più intuitivo il controllo dei droni. Come mostrato nel capitolo 2, questo sistema traccia i movimenti della parte superiore del corpo e li traduce in comandi per il drone. Il sistema proposto è poi stato testato con un simulatore e utilizzato per il controllo di un vero drone. Inoltre, per garantire un funzionamento duraturo, ho proposto un metodo che riduce la comunicazione e di conseguenza, migliora l'efficienza energetica del 11.9%.

In secondo luogo, nel capitolo 3, ho presentato un metodo per monitorare il livello dello sforzo cognitivo di un pilota di droni. Il mio modello combina diverse caratteristiche estratte da segnali vitali, come l'attività respiratoria, l'elettrocardiogramma, il fotopletismogramma e la temperatura cutanea, i quali sono acquisiti in modo non invasivo. Per ridurre l'intervariabilità dei segnali sia del soggetto che del giorno, ho esplorato diverse tecniche di normalizzazione. Inoltre, ho introdotto un nuovo metodo di apprendimento per macchine a vettori di supporto, il quale è adatto alle ottimizzazioni richieste dai vari soggetti. Sulla base di un campione di test acquisito da 34 volontari, il modello proposto ha riconosciuto deboli e alti sforzi cognitivi con

## Abstract

---

una precisione media del 87.3% e del 91.2%, mentre i partecipanti controllavano un simulatore di drone utilizzando rispettivamente un telecomando tradizionale o il nuovo controllore, chiamato FlyJacket.

Infine, nel capitolo 4, ho presentato l'integrazione del metodo per il monitoraggio dello sforzo cognitivo, su un singolo sistema imbarcato e indossabile che a sua volta integra anche il controllore di drone proposto nel capitolo 2. Dal lato hardware, il sistema include una serie di sensori per l'acquisizione di segnali vitali e una piattaforma di elaborazione a basso consumo energetico, adatta al monitoraggio dello sforzo cognitivo. Sul lato software, il sistema include nuovi algoritmi a basso consumo per l'elaborazione dei segnali e l'integrazione di nuovi metodi di apprendimento automatico. Inoltre, per sfruttare i compromessi tra la precisione richiesta e l'energia disponibile del sistema, ho presentato una nuova applicazione del concetto di apprendimento automatico con diversi livelli di risparmio energetico. I risultati hanno mostrato che l'approccio proposto produce un aumento del 78% della durata della batteria senza realmente influire sulla precisione della classificazione.

Il sistema proposto, comprendente un dispositivo di controllo per droni che integra a sua volta un'unità per il monitoraggio dello sforzo cognitivo, il quale getta le basi per lo sviluppo di interfacce uomo-macchina di nuova generazione. Con le informazioni sullo stato del pilota, possiamo completare le tradizionali tecniche di controllo condiviso, che saranno in grado di adattare dinamicamente il livello di interazione con macchine semi-autonome e in base alle necessità dell'operatore.

Parole chiave: Sforzo cognitivo, Monitoraggio, Segnali vitali, Sensor Fusion, Apprendimento automatico, Interazione uomo-macchina, Dispositivo indossabile, Sistemi imbarcati, Droni, Missioni di ricerca e salvataggio.

# Contents

<b>Acknowledgments</b>	<b>i</b>
<b>Abstract (English/Deutsch/Français/Italiano)</b>	<b>v</b>
<b>List of figures</b>	<b>xvii</b>
<b>List of tables</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Limitations of Unmanned Aerial Vehicles in Search and Rescue . . . . .	4
1.2 Teleoperation of Unmanned Aerial Vehicles . . . . .	6
1.2.1 A little bit of history . . . . .	6
1.2.2 Traditional Human Drone Interface (UDI) . . . . .	7
1.2.3 Graphical User Interface (GUI) . . . . .	8
1.2.4 Natural User Interface (NUI) . . . . .	9
1.2.5 Drone Autonomy . . . . .	12
1.2.6 Share-Control . . . . .	13
1.3 Thesis Contributions . . . . .	15
1.3.1 Wearable Drone Controller (WDC) . . . . .	17
1.3.2 Cognitive Workload Detection Method . . . . .	17
1.3.3 Wearable Cognitive Workload Monitoring System (CWMS) . . . . .	18
1.4 Thesis Outline . . . . .	19
<b>2 Wearable Drone Controller</b>	<b>21</b>
2.1 Introduction . . . . .	21
2.2 Natural User Interface: State of the Art . . . . .	24
2.3 Design of a Natural-User Interface for Drone Control . . . . .	28
2.3.1 Proposed Wearable Drone Controller . . . . .	29
2.3.2 Control Strategy: Design and Evaluation . . . . .	30
2.3.3 Wearable Sensor Node for Drone Control: Hardware Design . . . . .	32

## Contents

---

2.3.4	Wearable Drone Controller: Software Design . . . . .	36
2.4	Experimental Validation . . . . .	39
2.4.1	Open-loop Experiment with Virtual Reality . . . . .	40
2.4.2	Closed-loop Experiment with Virtual Reality . . . . .	44
2.4.3	Experiment with a real drone . . . . .	47
2.5	Experimental Results . . . . .	50
2.5.1	Body Posture Evaluation . . . . .	51
2.5.2	FlyJacket performance vs. Vicon . . . . .	53
2.5.3	FlyJacket used to fly a real drone . . . . .	54
2.5.4	Power Saving by Reducing the Communication Rate . . . . .	55
2.6	Conclusion . . . . .	57
<b>3</b>	<b>Cognitive Workload Detection Method</b>	<b>61</b>
3.1	Introduction . . . . .	61
3.2	Cognitive Workload Monitoring: State of the Art . . . . .	64
3.3	Cognitive Workload Detection Method . . . . .	68
3.4	Signals Acquisition and Preprocessing . . . . .	69
3.4.1	Physiological Process behind Cognitive Workload . . . . .	70
3.4.2	Signals Preprocessing . . . . .	71
3.5	Features Extraction and Selection . . . . .	71
3.5.1	Feature Extraction . . . . .	71
3.5.2	Features Normalization . . . . .	77
3.5.3	Features Selection . . . . .	78
3.6	Cognitive Workload Monitoring . . . . .	78
3.6.1	Model for Cognitive Workload Monitoring . . . . .	79
3.6.2	Training of the Universal Background Model . . . . .	80
3.6.3	Training of the Subject-Specific Model . . . . .	80
3.7	Experimental Setup . . . . .	81
3.7.1	Search and Rescue Drone Simulator . . . . .	83
3.7.2	Study Protocol 1: Use of a Gamepad . . . . .	84
3.7.3	Study Protocol 2: Use of FlyJacket . . . . .	86
3.7.4	Acquired Signals . . . . .	87
3.7.5	Research Participants . . . . .	87
3.8	Experimental Results . . . . .	87
3.8.1	Self-perception of induced cognitive workload . . . . .	87
3.8.2	Features discriminant power emphasized by normalization . . . . .	89
3.8.3	Physiological featuring of cognitive workload . . . . .	90
3.8.4	Classifiers for cognitive workload monitoring . . . . .	92



3.8.5	Classification improved with the SSM . . . . .	95
3.8.6	Emulated online cognitive workload monitoring . . . . .	98
3.9	Conclusion . . . . .	99
<b>4</b>	<b>Wearable and Multimodal Cognitive Workload Monitoring System</b>	<b>101</b>
4.1	Introduction . . . . .	101
4.2	Cognitive Workload Monitoring Systems: State of the Art . . . . .	104
4.2.1	Bio-signal acquisition with fitness devices . . . . .	106
4.2.2	Bio-signal acquisition with healthcare devices . . . . .	106
4.2.3	Embedded systems and the concept of self-awareness . . . . .	107
4.3	Self-Aware and Energy-Scalable System for Cognitive Workload Monitoring . . . . .	108
4.3.1	Self-Aware Energy Management . . . . .	110
4.3.2	Embedded Machine-Learning Algorithm . . . . .	114
4.4	Multi-Sensor Hardware Design . . . . .	115
4.4.1	Micro-Controller Unit . . . . .	116
4.4.2	Bio-Sensing Modules . . . . .	118
4.4.3	Communications . . . . .	119
4.4.4	Power Management . . . . .	120
4.5	Real-Time Software Design and Implementation . . . . .	121
4.5.1	Signal Acquisition . . . . .	123
4.5.2	Filtering and Delineation . . . . .	124
4.5.3	Feature Extraction and Classification . . . . .	126
4.5.4	Synchronization . . . . .	127
4.6	Setup and Validation Process . . . . .	128
4.6.1	Hardware Validation Process . . . . .	128
4.6.2	Software Validation Process . . . . .	130
4.6.3	Validation of the Self-Aware and Energy-Scalable Method . . . . .	130
4.7	Results . . . . .	131
4.7.1	Multi-Channel Acquisition Platform: Hardware Validation . . . . .	132
4.7.2	Online Cognitive Workload Monitoring: Software Validation . . . . .	133
4.7.3	Evaluation of Computation Complexity . . . . .	135
4.7.4	Evaluation of System Sensitivity and Energy-Scalable Levels . . . . .	137
4.7.5	Evaluation of the Energy-Scalable Machine-Learning Method . . . . .	139
4.8	Conclusions . . . . .	143

**Contents**

---

<b>5</b>	<b>Conclusions and Perspectives</b>	<b>145</b>
5.1	Summary and Conclusions . . . . .	145
5.1.1	Wearable Drone Controller . . . . .	146
5.1.2	Cognitive Workload Detection Method . . . . .	146
5.1.3	Wearable Cognitive Workload Monitoring System . . . . .	147
5.2	Future Work . . . . .	147
5.2.1	Suggestions for Drone Control Improvements . . . . .	147
5.2.2	Perspectives in Cognitive Workload Characterization . . . . .	148
5.2.3	Future of Online Cognitive Workload Monitoring . . . . .	149
5.2.4	Human-State: An Input for New Share-Control Techniques . . .	151
5.3	Retrospective . . . . .	152
	<b>Bibliography</b>	<b>172</b>
	<b>Acronyms</b>	<b>173</b>
	<b>Curriculum Vitae</b>	<b>i</b>

# List of Figures

1.1	When everyone else runs away, a firefighter runs into the fire. . . . .	1
1.2	Search and rescue missions: Applications. . . . .	3
1.3	Woodcut print of a kite from John Bate's 1635 book, The Mysteries of Nature and Art. . . . .	6
1.4	Remote piloting of a Queen Bee. . . . .	7
1.5	Human-Machine Interface: Traditional controller. . . . .	8
1.6	Graphical User Interface [132]: State of the Art. . . . .	9
1.7	Example of Natural User Interface (NUI) shown in Minority Report, a Spielberg's movie. . . . .	10
1.8	Autonomous Drones: State of the Art. . . . .	12
1.9	Share-control [135]: State of the Art. . . . .	13
1.10	An advanced share-control loop for drone operations in search and rescue missions. . . . .	15
1.11	Simplified control loop for drone operations in search and rescue missions. . . . .	16
2.1	Problems of current drone controllers. . . . .	22
2.2	Drone Controller: State of the Art . . . . .	25
2.3	Natural User Interface (NUI) based on moving platforms. . . . .	25
2.4	Natural User Interface (NUI) based on video capture systems [157]. . .	26
2.5	Natural User Interface (NUI) based on wearable devices [70]. . . . .	27
2.6	Natural User Interface (NUI) based on a camera motion capture system. .	29
2.7	Context diagram of the proposed wearable drone controller. . . . .	30
2.8	Drone-control identification loop. . . . .	31
2.9	Block diagram of the proposed wearable drone controller. . . . .	32
2.10	Prototype of the proposed wearable drone controller. . . . .	33
2.11	Graphical representation of the open-loop approach applied for the design of the control strategy. . . . .	41

## List of Figures

---

2.12 Setup of the open-loop experiment. . . . .	42
2.13 Sensor placement for the open-loop experiment. . . . .	43
2.14 Protocol of the open-loop experiment. . . . .	44
2.15 Graphical representation of the control loop applied to compare the proposed wearable drone controller with a controller based on a cam- era motion capture system. . . . .	45
2.16 Setup for drone control systems comparison in a virtual environment. . . . .	46
2.17 Sensor placement for the closed-loop experiment. . . . .	46
2.18 Protocol of the closed-loop experiment. . . . .	47
2.19 Graphical representation of the control loop applied to fly a real drone. . . . .	48
2.20 Setup used for flying a real drone with FlyJacket. . . . .	49
2.21 Map showing the covered flight trajectory of the real drone and the identified points of interest (i.e., red and green circles). . . . .	49
2.22 View from the camera of the drone. . . . .	50
2.23 Subjective evaluation of the control strategy. . . . .	51
2.24 Physiological response of the research participants. The error bars show the standard deviation, the three asterisks indicate a significant difference with $p < 0.001$ , and n.s. means not significant. . . . .	52
2.25 FlyJacket vs. Vicon. . . . .	53
2.26 Teleoperation of a real drone. Difference between the FlyJacket com- mands and the corresponding response of the drone orientation. . . . .	54
2.27 Evaluation of both the transmission (Tx) rate and the introduced Root Mean Square Error (RMSE) as a function of different thresholds. . . . .	56
3.1 Overview of the process for the design of a cognitive workload moni- toring method. . . . .	68
3.2 Schematic representation of the signal processing and feature extrac- tion processes. . . . .	72
3.3 Setup used to control the simulator with a gamepad and the FlyJacket design. . . . .	82
3.4 Protocol of the experiment with the gamepad. . . . .	85
3.5 Protocol of the experiment with FlyJacket. . . . .	86
3.6 Cognitive workload level perceived by the participants. . . . .	88
3.7 Comparison of different normalization methods. . . . .	89
3.8 Effect of different normalization methods. . . . .	90
3.9 Comparison between the best classifiers on cross-validation. . . . .	94
3.10 Comparison of a Universal Background Model (UBM) vs. a Subject- Specific Model (SSM). . . . .	99

4.1	Overview of the proposed self-aware and energy-scalable wearable system for online cognitive workload monitoring. . . . .	109
4.2	Self-aware concept of energy-scalable embedded machine-learning algorithms and methods for online cognitive workload monitoring. . .	111
4.3	Switching principle of the Self-Aware Manager . . . . .	113
4.4	Block diagram of the proposed wearable system. . . . .	116
4.5	Prototype of the proposed wearable system. . . . .	117
4.6	Delineated points and parameters extracted from each physiological signal. . . . .	126
4.7	Validation Method. . . . .	128
4.8	Feature extraction evaluation. The metrics are the relative error related to the reference implementation and the Spearman's correlation coefficient. . . . .	135
4.9	Current consumption of the proposed levels while always acquiring all signals and acquiring only the required ones. . . . .	141
4.10	Current consumption and classification accuracy for the different approaches, namely, a single-level model, a multi-level model acquiring all signals, and the proposed multilevel model acquiring only the required signals. . . . .	142



# List of Tables

2.1	Comparison of 9-axis Inertial Measurement Units (IMUs) with on-chip 16-bit Analog-to-Digital Converters (ADCs), combining a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. . . . .	35
2.2	Average current consumption and run time of the proposed sensor node.	57
3.1	Summary of the state-of-the-art studies addressing cognitive workload monitoring. . . . .	65
3.2	Measurable physiological manifestations related to induced cognitive workloads and affected by the Sympathetic Nervous System (SNS), the Parasympathetic Nervous System (PSNS), and Hypotalai-Adrena (HPA) axis. . . . .	70
3.3	Performance of combining both Recursive Features Elimination (RFE) and classification methods for feature selection. . . . .	91
3.4	List of the most important features for cognitive workload detection. .	93
3.5	Performance of a Universal Background Model (UBM) vs. a Subject-Specific Model (SSM) on a test set collected from participants using a gamepad (Study 1, subsection 3.7.2). . . . .	95
3.6	Performance of a Universal Background Model (UBM) vs. a Subject-Specific Model (SSM) on a test set collected from participants while FlyJacket (Study 2, subsection 3.7.3). . . . .	96
4.1	State-of-the-art studies on cognitive workload monitoring using wearables . . . . .	105
4.2	Bio-sensors characteristics . . . . .	118
4.3	Power modes of the Microcontroller Unit (MCU) (from Run/active down to Standby mode), with the corresponding current consumption and wakeup time to Run mode. . . . .	120
4.4	Power modes of the different peripherals, with the corresponding current consumption and wakeup time. . . . .	122

## List of Tables

---

4.5	Distributed processing using interrupts and timers . . . . .	122
4.6	Selected sampling frequency, processing window length, and buffer size used at the acquisition layer for each signal. . . . .	124
4.7	Delineated points and bio-parameters for each physiological signal . .	125
4.8	Maximum points per minute, number of parameters per period, and buffer size used at the acquisition layer for each signal. . . . .	125
4.9	Evaluation of the fiducial points delineated from the physiological signal acquired with the proposed system. The metrics are sensitivity, positive predictive value, mean error, and standard deviation of the mean error. . . . .	132
4.10	Evaluation of the fiducial points delineated from each physiological signal. The metrics are sensitivity, positive predictive value, mean error, and standard deviation of the mean error. . . . .	134
4.11	Computation time (ms) of a 60-second signal window processing. . . .	136
4.12	Accuracy and computational costs for different scaled models. . . . .	138
4.13	Average current consumption of the different components operating under the proposed energy-scalable levels. In bracket the values while always acquiring all signals. . . . .	140
4.14	Average current consumption of the different energy-scalable models of each level. In bracket the values while always acquiring all signals. .	142
4.15	Comparison of the different approaches in terms of accuracy, time spent in the various levels, average current consumption, and estimated battery life. . . . .	143



# 1 Introduction

An unusual event, even exceptional, a fact that breaks or temporally changes the natural course of things, these are factors that suddenly make the alarm of Search and Rescue (SAR) teams sound. This alarm denotes a stringent request for assistance. From that moment on, it is time for the SAR teams to go there, where everyone else runs away (Fig. 1.1). Often, several complex and dangerous tasks are waiting for them. Although rescuers are aware of that, it is not always clear how difficult it will be, especially at the beginning, when chaos dominates. In SAR fast information and timing are of primary importance. Having the right information from the very beginning is crucial to evaluate what happens, to efficiently adopt the proper series of measures aimed at the search for people who are in distress or imminent danger, and to provide first aid assistance. These measures are typically determined by the characteristics of the event, that is, the number of victims and both vastness and type of terrain the mission has to be conducted over.



Figure 1.1 – When everyone else runs away, a firefighter runs into the fire.

In general, to prevail against the event, the attempt is to adopt an offensive tactic.

## Chapter 1. Introduction

---

This tactic consists of the deployment of enough means to efficiently counteract the event, namely, an adequate supply of rescuers, specialists, materials, vehicles, etc. However, uncontrollable factors, such as weather conditions, may prevent SAR teams from deploying some equipment and force them to apply a defensive tactic. For instance, to quickly reach remote areas or to quickly cover extended surfaces, rescuers typically use helicopters. However, depending on both weather conditions and time of the day, aerial support may only be provided by a restricted group of professionals, or maybe not at all. In this case, rescuers have no other solution than to apply any other measure to at least mitigate as much as possible the wrong evolution of the event. To this aim, teams of specialists are regularly training and ready to operate. Training is probably the most important activity in SAR, as it allows us to face an event without being caught unprepared. It is where rescuers learn how to use the equipment, operate under time pressure and in difficult conditions. Training is also a way to find new strategies, to evaluate the risks, and consequently set the limits to reduce such risks.

Nowadays, to limit rescuers' exposure to risky situations, SAR teams make use of unmanned robotic vehicles. The use of Unmanned Aerial Vehicles (UAVs), commonly known as drones, was originally limited to the military sphere, but recently, it gained a particular interest in SAR. Thanks to their limited costs of deployment and versatility, drones find different interesting applications in SAR. First of all, a drone can be used to have a better understanding of the situation upon arrival, but also during the intervention. Gathering information from a chaotic place is hard and time consuming, because accesses are often difficult and resources relatively limited. A drone can facilitate and accelerate this task, especially providing information that is not available from a ground perspective [91, 144, 177]. Drones can also be used for both outdoor and indoor inspections, to search for sources of fire, source of gas escape, or any other hazardous materials. For instance, as a firefighter, we used drones for forest fire monitoring (Fig. 1.2a), to determinate the perimeter, to measure the rate of spread, and to find remaining hot spots. Moreover, drones can be used to search for missing people, victims caught by an avalanche (Fig. 1.2b), victims of earthquakes, etc. For instance, drones have been used in Nepal to identify people trapped in remote areas and evaluate the damage caused by the earthquake of 2015. Due to the overwhelming needs, Nepal suffered from a shortage of available manned helicopters. Therefore, the use of drones allowed to leave precious helicopters for effective rescue missions. Furthermore, drones can be used to establish communication with victims, or to provide them with first assistance; namely, with a life jacket (Fig. 1.2c), an automated external defibrillator (Fig. 1.2d),

---

water, oxygen, or simply moral support. In this regard, Swiss Air Rescue (REGA) will begin to use drones in 2021 to access areas that cannot be reached by helicopters ([www.rega.ch](http://www.rega.ch)). This Rega drone will be able to autonomously scan large search areas and identify people in real time, using infrared, thermal, and daylight cameras. An additional example from SwissDrones is the SDO 50 V2, which is also capable of providing real-time data and visual information in case of natural disasters to help first responders and civil protection organizations with better situational awareness ([www.swissdrones.com](http://www.swissdrones.com)). This drone has been able to successfully perform several critical tests including aerial surveillance missions for emergency rescue and disaster management in China. However, even though these drones find many applications in the field of SAR, limitations exist in their effective and efficient utilization in real missions [89, 136].



(a) Forest Fire



(b) Mountain Rescue



(c) Lake Rescue



(d) Basic Life Support

Figure 1.2 – Search and rescue missions: Applications.

### 1.1 Limitations of Unmanned Aerial Vehicles in Search and Rescue

From the technological point of view, the search and rescue context is extremely unfriendly. Although there is a vast literature on research efforts towards the development of unmanned SAR tools, this research effort stands in contrast to the practical reality in the field, where unmanned SAR tools have great difficulty finding their way to the end-users [57]. In general, rather than a replacement for human rescuers, robotic technology should be intended for performing physical tasks that are outside of human capabilities, facilitate the execution of a particular task or augment rescuers' performance. If the technology performs worse than a human, it will rarely find employments in real SAR. To be adopted in the SAR field, rescue robots have to show high capabilities as well as reliability and robustness. Moreover, the technology must be easy to use and should be deployed extremely quickly. In principle, rescuers try to use the simplest technologies that are good enough to solve specific problems. In fact, most of the robotic platforms in use by SAR teams can be used by everyone. In contrast, platforms that are sensitive or complicated to use are often discarded due to the challenges related to real-world deployments and the consequent elevated risk of failure.

In my humble opinion, drones belong to this second category; namely, interesting, but critical platforms. More than the drone itself, critical is the Human-Machine Interface (HMI), a mean used by a human to interact with the drone. In other words, the weakness is the remote controller, a device used to fly a drone, or control any other remote UAV. Most of these controllers are simple electronic devices, with buttons, switches, and levers, configured to remotely control the various Degrees of Freedom (DoF) of the drone. However, although the technology of such devices is relatively simple and applicable to control different UAVs, an important limitation exists; that is, the intuitiveness of the control strategy in use. For instance, to decrease or increase the altitude of a drone, you have to respectively move forward or backward a lever of the remote controller. This configuration, or mapping, of the different commands is not really intuitive, and under stress situations the problematic is even more emphasized. In fact, operating in extreme conditions, dealing with the scarcity of human resources, and having the feeling of urgency in finding victims, demands an important cognitive effort. Therefore, singular skills are required to ensure an effective and efficient deployment of drones in real missions. Thus, the use of drones in the SAR field is limited to highly trained professionals [28]. However, the risk of not being able to perform a mission or to crash the drone is still extremely high.

### **1.1. Limitations of Unmanned Aerial Vehicles in Search and Rescue**

---

To address this problem and facilitate rescuers operating with drones under stressful conditions, there is a need to work on the improvement of current HMIs. For instance, an improvement of the control strategy would decrease the training requirements for the teams and, in general, would improve the reliability of the tool in harsh conditions, like those in SAR operations. Low operator training requirements are in fact other important criteria for the adoption of robots in SAR, but this aspect is often not addressed by the research community [40]. In this regard, researchers have focused on the improvement of current HMIs. For instance, nowadays, many novel HMIs often include a screen for visual feedback, which helps the pilot by providing more information, but it does not really simplify the control strategy. On the other hand, more recent studies used a completely different approach, which relies on the use of upper-body gestures recorded with external motion tracking systems. Although these solutions seem to be more intuitive to use, these tracking systems mainly relies on different cameras [192] or bulky platforms [81, 162], which are difficult to carry into a disaster area.

Another solution that can facilitate rescue operations is the use of semi-autonomous robots. Respectively, humans and robots have the ability to adapt to new situations and to optimize the execution of repetitive tasks. Therefore, to ensure high performance during unique and high cognitive demanding tasks, such as during search and rescue missions, there is a need to combine both human's and robot's skills. Thanks to recent enhancements in share-control techniques [25] and the achieved level of drone autonomy [57], we can definitely combine both human's and robot's abilities, and bring robot teleoperation to the next level. However, to dynamically adapt this human-robot interaction, important information is missing in the loop, that is, the human state, and in particular, the cognitive workload of the operator.

The cognitive workload is defined as the mental effort required to perform a single or a combination of tasks [23], and it has been proven to affect operators' performance, especially in the case of high-risk and high-demanding situations [103, 115, 175]. Thus, monitoring cognitive workload could provide an important feedback about the human state, information that could adaptively support the operators according to their specific needs, improve their performance, and potentially decrease hazards. In this regard, different studies showed how to assess cognitive workload monitoring from physiological signals and provided numerous solutions that are suitable for real-time applications [23, 77, 146]. However, in the field of SAR missions with drones, there are no clear indications about the method and physiological signals to use. Moreover, most of the current studies rely on offline processing performed on a

## Chapter 1. Introduction

---

computer, a methodology that is not suitable for SAR missions.

### 1.2 Teleoperation of Unmanned Aerial Vehicles

Since antiquity, there have been stories of men attempting to fly, typically by jumping off a tower with birdlike wings, stiffened cloaks or other devices, but most attempts ended in serious injury or death. The legend of Icarus is one of the best known.

#### 1.2.1 A little bit of history

The first form of man-made aircraft may have been the kite (Figure 1.3), which was invented in Asia before the 5th century BC [36]. During the 3rd century AD, the Chinese invented the sky lantern [41], an airborne lantern used for military signaling. The sky lantern is a precursor of the hot air balloon demonstrated in 1783 by the French brothers Joseph-Michel and Jacques-Étienne Montgolfier. The Montgolfier brothers developed a hot air balloon and demonstrated the flight of a first unmanned balloon [35]. Similar unmanned balloons were used in 1848 by Austrian soldiers, who filled the balloons with explosives and attacked the city of Venice [35]. An interesting aspect is that to control such unmanned flying objects the inventors used a rope, which makes it as the first user interface adopted for the control of unmanned flying objects.



Figure 1.3 – Woodcut print of a kite from John Bate’s 1635 book, *The Mysteries of Nature and Art*.

## 1.2. Teleoperation of Unmanned Aerial Vehicles

---

The evolution of unmanned flying vehicles continued over the years, even if I think the main objective was to develop manned vehicles. In fact, inventors developed unmanned platforms mainly to test the feasibility of putting a human on it. An indisputable approach that is mandatory for the development of manned flying vehicles (FAA Regulations). A revolution for the teleoperation of unmanned vehicles is the first radio control patented by the inventor Nicola Tesla in 1898 [178]. This radio-control was initially intended for the teleoperation of a boat.

The first recorded use of radio control techniques for the teleoperation of UAV dates back to First World War, only 14 years after the first powered aircraft from the Wright brothers at Kitty Hawk in North Carolina. In 1917, Prof. Archibald Montgomery Low used in fact the radio-control technology for the first pilotless winged aircraft in history, the Ruston Proctor Aerial Target [54]. This project led the way for further research of UAVs, such as the Hewitt-Sperry Automatic Airplane, also known as "Flying Bomb", or "Aerial Torpedo", which went from Britain to USA, yielding the Kettering Bug, an upgraded version of aerial torpedo developed by the US Army. Although promising demonstrations, the war ended before they could have been used in a combat scenario. During the Second World War and the Cold War, UAV

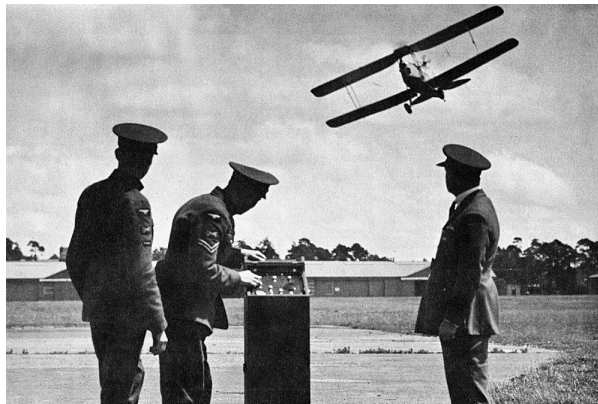


Figure 1.4 – Remote piloting of a Queen Bee.

technology have been vastly enhanced. However, due to the warfare orientation and the elevated costs of development, the use of UAVs was limited to the military sphere.

### 1.2.2 Traditional Human Drone Interface (UDI)

Only in the 1960's, thanks to advances in telerobotics matured in different fields, such as nuclear power plants, medicine, undersea and space missions [121], but also

## Chapter 1. Introduction

---

the drastic reduction of the development costs, the market of UAVs opened to the private sphere. Moreover, breakthroughs in semiconductor technology allowed the miniaturization of radio-controlled components, which yielded in the development of traditional controllers, such as the one shown in Figure 1.5. However, later studies



Figure 1.5 – Human-Machine Interface: Traditional controller.

demonstrated that this type of controller is neither natural nor intuitive, as the user requires training and concentration during operation [68, 118]. Being able to interact and command robots in natural and efficient ways is crucial, especially in demanding operations, such as in search-and-rescue or indoor navigation [53].

### 1.2.3 Graphical User Interface (GUI)

With the advent of computers, researchers developed new technologies, which allowed the creation of new forms of human-machine interaction. Sketchpad, the first graphical computer-aided design program was developed by Ivan Sutherland in 1963 [171]. In 1968, Douglas Engelbart presented a first live public demonstration of the mouse, used to manipulate text-based hyperlinks [47]. In 1973, Alan Kay went beyond text-based hyperlinks and used for the first time a Graphical User Interface (GUI) as the main interface [88]. Such a GUI allowed the user to provide commands, each typically characterized by several instructions, and to automatically receive direct feedback of what has been done. Initially intended for the human-computer interaction, this technology has been applied for drones teleoperation, few novel examples are shown in Figure 1.6.

In general, a GUI is characterized by both a control and a feedback panel, typically displayed on the same screen, as shown in 1.6b. However, hybrid versions exist, which embed both a traditional-like controller and a touch screen for guidance and visual feedback, as shown in 1.6a. Nowadays, Virtual Reality (VR) glasses (1.6c) are replacing screens to provide a more immersive interaction. Although GUIs improved



## 1.2. Teleoperation of Unmanned Aerial Vehicles



Figure 1.6 – Graphical User Interface [132]: State of the Art.

both intuitiveness and level of immersion, these interfaces are not yet the most efficient option, that is, a barrier still subsists between the communication from human to machine. One step towards overcoming such a barrier is the development of Natural User Interfaces (NUIs) [53].

### 1.2.4 Natural User Interface (NUI)

To allow users to engage with machines as they would like, researchers started applying a user-centered design approach [53]. The idea behind this approach is to design an interface based on how users will likely use it, by studying their behavior in real-world tests, and focusing on both requirements and limitations of the end-users. Following this user-centered approach, researchers were able to develop new NUIs [160], which allow to harmoniously handle both virtual and real objects. For this purpose, the most common used mediums are voice [51, 66], tags [43], hand gestures [70, 167], and body gestures [108, 157]. Figure 1.7 shows an example of NUI presented in *Minority Report*, a Spielberg's movie, which has been a canonical reference point for designers.

Voice control, originally developed for human-computer interaction, is probably one of the most natural interfaces that could be used to interact with a machine. In fact, this type of interface involves the speech, which is the most used way of interaction between humans. Although the use of the voice offers a solution that is both intuitive and appealing to the user [53], only few studies in the literature use it to interact with drones. These studies show the weaknesses of voice control, but demonstrate as well some important limitations. An example is the work presented in [145], where a voice controller was developed to recognize and send commands to a fixed wing



Figure 1.7 – Example of Natural User Interface (NUI) shown in *Minority Report*, a Spielberg's movie.

semi-autonomous UAV. Another exploratory study, considering voice commands for the interaction with robots in a simulated environment, concluded that to control a drone lower-level commands (e.g., left and right) are generally preferred by users [85]. From one side, the use of voice allows the development of intuitive and hands-free interfaces, but on the other hand, real-flight tests showed that ambient wind noise and conversation can lower the reliability of the voice recognition system [145]. Thus, in SAR missions, where both high noise and frequent communications are ordinary, voice control does not find a future.

In a few studies, researchers also used visual tags, in other words, specific pixel patterns that uniquely encode a predefined set of commands, which robustly and accurately define what a drone has to do. The use of visual tags is a common practice in robotics, since no additional device apart from a simple monochromatic camera is needed. For instance, this approach has been used to control underwater robots [43] and drones as well [53]. For this purpose, the on-board front facing camera of the drone can be used, but to avoid the need for being in the visual field of the drone, an additional cameras placed in front of the pilot. From one side, such an interaction is simple, since a non-expert user can pick up a predefined set of tags and interact with the drone in a reliable and safe way. However, the use of a predefined set of tags can limit the level of interaction to a limited set of actions, such as take off, hover, land, etc. To face this problem, authors in [53] used a binary coding tag system developed for producing configurable dictionaries with arbitrary size and number of tags [61]. Although this approach allows to expand the reliable and unique command list that can be sent to the drone, having too many options may be confusing and in the

## 1.2. Teleoperation of Unmanned Aerial Vehicles

---

rush the pilot may not be able to find the right tag that has to be shown to the drone. Therefore, the use of visual tags is not really suitable for SAR missions, where there is no time to search between the many tags which one is the right one to use.

In some sense, an evolution of the visual tag interaction is given by the use of hand gestures. In fact, hand-gesture interaction involves both static signs and dynamic gestures [164, 167], which make it one of the most used NUIs in literature. Static signs are expressed with a motionless and predetermined finger configuration, while dynamic gestures are characterized by hand motions [87]. The recognition of both static and dynamic gestures is addressed in different ways. Initial hand gesture based NUIs depended on sensors embedded in glove-based devices that directly measured spatial position and joint angles [13, 52]. Such devices were massive and limited the interaction between the user and the machine [87]. A development of new sensors equipped with depth cameras, such as the Kinect™ [107] and the Leap Motion™ [190], busted the development of new hand gesture based NUIs [104, 138]. Initially, a gesture recognition system based on depth imagery and a depth-based hand gesture database for drone control was developed in [101]. Subsequently, a robot that recognized hand gesture commands was successfully tested in real world scenarios [4]. Further developments of both cameras and depth-based hand gestures algorithms improved both recognition accuracy and precision [73]. These deficiencies were in fact limiting the performance of the hand/finger poses extraction, which was low due to limited camera resolution, high noise and missing data [101]. The hand-gesture interaction is an interesting approach and users find it intuitive, if a direct transformation between hand and drone movement is applied [53]. However, this method takes possession of the hands, which cannot be used to execute any possible parallel task. Therefore, hand-free solutions are more interesting in that sense.

The first form of visual body interaction reproduces the instinctive behavior of baby animals, who follow their parents everywhere. Similarly, the idea is applied to let the drone follow its owner, or its user. To this aim, drones are equipped with an on-board camera, and computer vision algorithms are used to detect and track the user's position. Finally, additional algorithms are applied to let the drone keep both distance and point of view, even if the user moves or runs [134]. This solution could be interesting for SAR missions, as the rescuers could freely execute their activity without having to care about the drone. However, it implies that someone, either a human or a ground robot, needs to move around to cover the area of interest, and this can be an evident problem in collapsed or dangerous environments.

### 1.2.5 Drone Autonomy

The improvement of user interfaces is also due to recent enhancements in drone autonomy, which drastically simplify the Human-Machine teleoperations. The well-documented problems of the Human-Machine teleoperation have been addressed by three lines of research, which aim to release the work of the pilot by increasing the level of robots' autonomy [57].

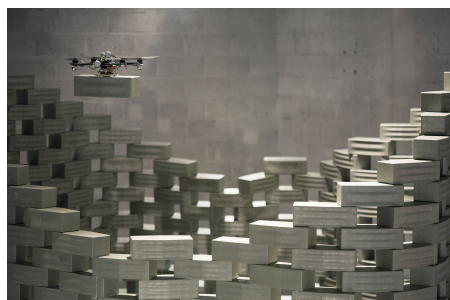
The first line of research focuses on the sensory-motor autonomy, which aims to translate high-level human commands into combinations of platform-dependent control signals (such as pitch, roll, yaw angles or speed). For instance, current methods of sensory-motor autonomy allow us to reach a given altitude, perform circular trajectory, move to global positioning system (GPS) coordinates, follow pre-programmed trajectory using GPS waypoints, and maintain a specific position [112].

The second line of research focuses on the reactive autonomy, which integrates sensory-motor autonomy to maintain current position or trajectory in the presence of external perturbations (such as wind or electro-mechanical failure), maintain a safe or predefined distance from ground, avoid obstacles, coordinate with moving objects, take off and landing [102, 116].

The third line of research focuses on the cognitive autonomy, which includes reactive autonomy to perform simultaneous localization and mapping to recognize objects or persons, to plan (i.e., for battery recharge), to resolve conflicting information, and to learn [11, 49, 117, 176]. Two examples are shown in Figure 1.8, where a drone autonomously flights in a forest (Fig. 1.8a) and the cooperation of autonomous drones assembles an architecture installation (Fig. 1.8b).



(a) Autonomous forest flight [102].



(b) Flight assembled installation [11].

Figure 1.8 – Autonomous Drones: State of the Art.

## 1.2. Teleoperation of Unmanned Aerial Vehicles

Although all levels of drone autonomy can contribute to relieve the workload of the pilot, at present in SAR missions, a drone cannot be fully autonomous. In fact, none of today's commercial drones have sufficient control autonomy to complete a SAR mission without skilled human supervision, which makes those operations slow, dangerous and not scalable [57]. The current level of autonomy allows an optimized execution of some repetitive tasks, but it does not replace the human's ability to adapt to new situations. Thus, to combine both human's and robot's abilities, recent studies focus on share-control techniques [25].

### 1.2.6 Share-Control

The approaches applied to increase robot autonomy are not yet reliable enough to leverage in the SAR field. The problem is that SAR missions are unique and it is extremely difficult to develop a technology that works in all possible scenarios a SAR mission may hide. Therefore, in order to efficiently perform SAR missions, a coexistence/collaboration between humans and robots is required. In fact, robots can be optimized to repeat tasks and enter places that are inaccessible for rescuers whereas humans have the ability to adapt to new situations. In this regard, share-control systems have been investigated to modulate the human-robot interaction and dynamically adapt the level of assistance [25, 135].

Shared control has been shown to help operators of robotic equipment in a wide range of applications, including transport [1] and assistive technology [26, 189]. As shown in Figure 1.9, in current share-control techniques, both human and robot

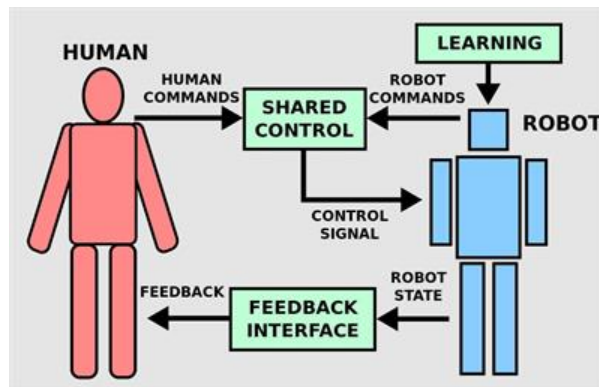


Figure 1.9 – Share-control [135]: State of the Art.

commands are fused to obtain a control signal, that is, a sequence of commands to

## Chapter 1. Introduction

---

control the robot. Moreover, artificial-intelligence methods are applied to provide the robot with a learning ability and consequently optimize the robot commands transmitted to the shared control. Finally, a feedback interface is used to inform the human about the robot state. In particular, shared control can increase safety and reliability as well as decrease user workload. However, if the level of assistance is not well matched to the user's instantaneous needs and abilities, it may not only reduce user acceptance, but could also be detrimental. Thus, to ensure an efficient interaction between rescuers and drones, it seems that something is missing.

The level of assistance does not just depend on the environmental context (as measured by the robot's sensors), but also on performance measures [25] and the user's control signals [154], which have to be both computed at runtime. In my humble opinion, the complete control loop for drone teleoperation in SAR missions should be as the one depicted in Figure 1.10. The electric motors of the drone are controlled by a Low-Level Control (LL CTRL) inner loop. The low-level control, also called flight controller, is basically the system of the drone that controls drone's rotors to produce the desired outcome. The flight controller might be of different types, and it should also include both manual or automatic means for starting and stopping the motors, selecting and regulating the speed, regulating or limiting the torque, and potentially protecting against overloads and electrical faults. Then, a Remote Controller (R CTRL) is used to interact with the drone and provide the desired combinations of platform-dependent commands (such as pitch, roll, yaw angles or speed). The remote controller allows the pilot to remotely control the drone using radio signals. Then, a High-Level Control (HL CTRL) loop should be added to provide a certain level of autonomy to the drone. This high-level control is used to target the three levels of increasing autonomy described in section 1.2.5. Moreover, a Share Control (SH CTRL) should be included to increase the performance of the human-machine interaction.

Finally, the last element in the loop is the Human State Monitoring System (HSMS). Such a monitoring system is the key to provide the necessary feedback, which allows the share-control to dynamically adjust the level of autonomy based on the real needs of the pilot. For instance, we could use such information to adjust the speed of the drone automatically and relieve the pilot from this task. Moreover, we could also adapt the reactivity of the drone by adjusting the gains of the controller. In this way, we can enhance drones' teleoperation by addressing the fact that pilots have different abilities and varying performances.

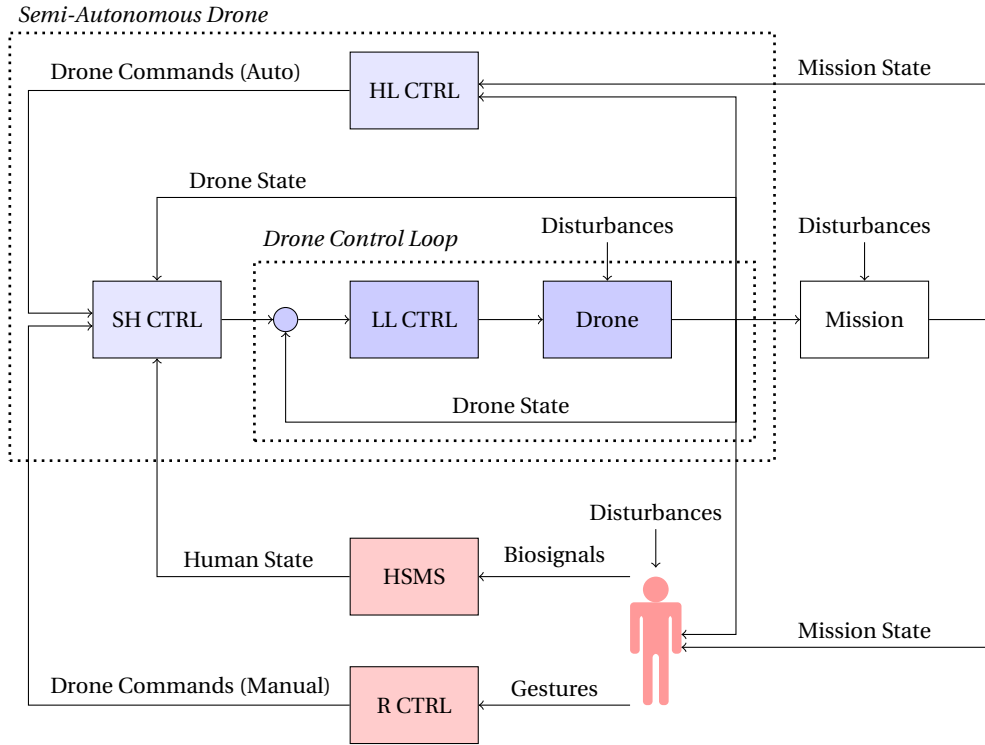


Figure 1.10 – An advanced share-control loop for drone operations in search and rescue missions. Respectively, a Low-Level Control (LL CTRL) and a High-Level Control (HL CTRL) are responsible for controlling the electric motors and the autonomy of the drone. The pilot typically uses a Remote Controller (R CTRL) to interact with the drone. The Share-Control (SH CTRL) unit dynamically adapts the level of autonomy based on information provided by the drone and by a Human State Monitoring System (HSMS).

Without the knowledge of the human state, a share-control cannot be aware of the difficulties encountered by both humans and robots. As shown in Figure 1.9, this is the case of current share-control techniques, which do not include such feedback. Therefore, there is a need to investigate the feasibility of providing information about the human state.

### 1.3 Thesis Contributions

Being the information of the human state not yet available in the loop, and the fact that current drone controllers are not at all intuitive, both the human state

## Chapter 1. Introduction

monitoring and the improvement of the existing manual controller are the focus of this work. As human state information, I decided to target cognitive workload, as most of the SAR missions require a particular mental effort at the beginning to understand what to do and to decide how to handle the situation. Tasks that are often handled in a limited amount of time, but are really demanding. To this aim, I explored different methods to characterize the cognitive workload in a continuous and noninvasive way. For the improvement of the manual controller, I investigated the integration in a wearable system of a recent method that translates upper-body movements tracked by a 3D camera system into commands for a drone [108]. In particular, the contributions of this work are shown in Fig. 1.11, and summarized next.

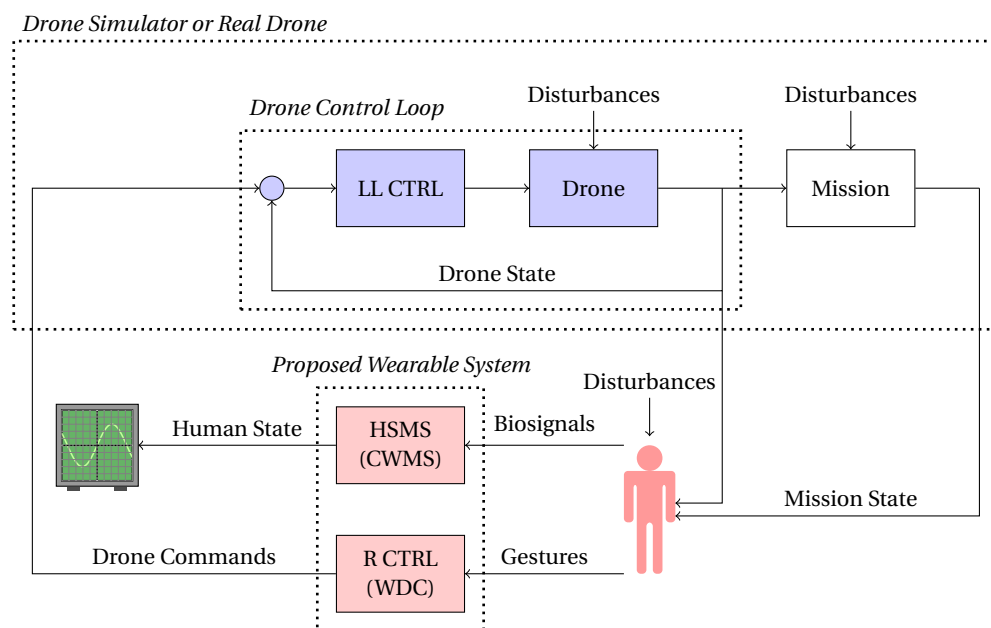


Figure 1.11 – Simplified control loop for drone operations in search and rescue missions. Red boxes highlight the focus of this work. Both the implemented drone simulator and the selected real drone are simple and only include a Low-Level Control (LL CTRL). As a Human State Monitoring System (HSMS), I developed a Cognitive Workload Monitoring System (CWMS). As a Remote Controller (R CTRL), I developed a Wearable Drone Controller (WDC).



### 1.3.1 Wearable Drone Controller (WDC)

To provide a more portable NUI for drone teleoperation, I developed a new Wearable Drone Controller (WDC), which is based on upper body movements. The proposed system has been integrated into a soft exoskeleton, called FlyJacket [153], and validated for the teleoperation of both a simulator and a real drone. To this aim, I ran different experiments and participated in demonstrations during public events, such as Cybathlon and EPFL Open Days.

Moreover, to find the best position that a user should assume while using the proposed wearable system, I analyzed the physiological response of different users. The study showed that sitting is the best position compared with standing and lying.

Finally, to optimize the battery lifetime of the proposed system, I presented a method that drastically reduces the communication traffic. The method was tested while driving a real drone with the proposed wearable system and showed an 11.9% reduction of the total energy consumption.

### 1.3.2 Cognitive Workload Detection Method

To induce different levels of cognitive workload related to search and rescue missions, I modified a virtual-reality based drone simulator to include parallel tasks, such as objects recognition. The simulator was used for the acquisition of a database of different physiological signals acquired from 34 participants involved in a cognitively demanding simulated, but immersive SAR mission, 24 using a traditional controller and 10 using the proposed wearable controller, called FlyJacket.

To characterize the physiological response of cognitive workload, I performed an exhaustive investigation of relevant features extracted from physiological signals and I selected the most representative ones.

To reduce both inter-subject and inter-day variability, I explored different feature normalization techniques showing that a normalization that considers both subject and day improves the classification results.

To further reduce the inter-subject variability, I provided a new learning method based on Support Vector Machines (SVMs) suitable for a subject-specific optimization. This SVM based method uses two regularization terms, one for learning the general behaviour, and another one for tuning the model to fit the characteristics of

a particular data subset.

Finally, I proved the ability of the proposed method to detect low and high levels of cognitive workload with both traditional controllers and new advanced controllers, such as the new FlyJacket design, achieving an accuracy of 87.3% and 91.2%, respectively. These results were obtained on an unseen dataset acquired from 34 subjects while flying a drone simulator and mapping a graphic representation of a damaged area. These results are better than the latest state-of-the-art studies.

### 1.3.3 Wearable Cognitive Workload Monitoring System (CWMS)

With this work, I demonstrated the feasibility of using a wearable device to monitor cognitive workload from physiological signals acquired in a non-invasive way. To achieve this result, I developed a new wearable multi-channel signal-acquisition and processing platform, which implements the algorithms presented in Chapter 3. To succeed, we had to address the stringent processing and memory constraints, which limit the performance and the execution of the algorithms in embedded devices. Dealing with such constraints was indeed a challenge, as the proposed multimodal cognitive workload monitoring method requires the acquisition and the processing of a large amount of data. Therefore, some basic optimizations were applied to avoid delays and reduce large memory requirements, which can affect both functionality and performance. Then, the embedded algorithms and methods have been validated for online monitoring of low and high levels of cognitive workload, achieving an accuracy of 75%.

Finally, to optimize energy consumption, I explored the trade-offs between the required accuracy and the available energy of the system. In this regard, I relied on the fact that complex classifiers typically reach high detection performance, but they frequently lack in energy efficiency. On the other hand, simple classifiers cannot always guarantee high detection performance, but they are energetically more efficient. Therefore, while dynamically adapting the level of complexity, I showed that it is possible to gain in energy efficiency without seriously compromising the detection accuracy. To this aim, I proposed a self-aware approach that exploits a scalable machine-learning method with different power-saving levels. This approach is extended to the acquisition level, which yields an increase of 78% of the battery life and an acceptable accuracy lost with respect to the best universal background model presented in chapter 3 (i.e., from a theoretical 80.32% to 77.65%).

### 1.4 Thesis Outline

The rest of this thesis includes three main chapters, which are self-containing and structured to include sections covering related works, applied methods, experimental results, and a conclusion. A final chapter concludes this thesis with a summary and possible future work. More precisely, each chapter presents the following contents.

Chapter 2 presents the development of a new Wearable Drone Controller (WDC), which captures upper-body gestures and translates them into commands for the teleoperation of a drone. As a first result, a comparison with a commercial optical camera system for motion capture shows that the proposed WDC is accurate while tracking the gestures required to drive a drone. Then, after being integrated into a soft exoskeleton, called FlyJacket, the proposed WDC has been experimentally validated for the teleoperation of both a simulator and a real drone. Moreover, this chapter includes an investigation to find the best position that a user should assume while using the proposed WDC. To this aim, I presented an analysis of the physiological response of different users showing that sitting is the best position compared with standing and lying. Finally, as an optimization for the design of such a novel WDC, I presented a method that drastically reduces the communication traffic, and consequently improves the battery lifetime of the proposed system by 11.9%.

Chapter 3 details the methods used to access cognitive workload detection from physiological signals acquired in a non-invasive way. In particular, I proposed a machine learning algorithm for a continuous cognitive-workload monitoring. The proposed multi-modal cognitive workload monitoring model combines the information of 25 features extracted from physiological signals, such as respiratory activity, electrocardiogram, photoplethysmogram, and skin temperature, acquired in a non-invasive way. To reduce both subject and day inter-variability of the signals, I explored different feature normalization techniques and I introduced a modified learning method for support vector machines, which is suitable for subject-specific optimizations. On a new test set acquired from 34 volunteers, the proposed subject-specific model is able to distinguish between low and high cognitive workloads with an average accuracy of 87.3% and 91.2% while controlling a drone simulator using both a traditional controller and a new-generation controller called FlyJacket, respectively. Finally, the results reported in this chapter showed that the proposed model is suitable for both traditional controllers and new advanced controllers, such as FlyJacket.

Chapter 4 presents the hardware/software co-design of a new wearable embedded

## Chapter 1. Introduction

---

system for online cognitive workload monitoring. This new wearable includes, on the hardware side, a multi-channel physiological signals acquisition (respiration cycles, heart rate, skin temperature, and pulse waveform) and a low-power processing platform. On the software side, this wearable embedded system includes the bio-signal processing algorithms presented in chapter 3 and the application of the self-aware concept for scalable energy embedded machine learning algorithms and methods for online cognitive workload monitoring. The results showed that the proposed wearable system can continuously monitor multiple bio-signals, compute their key features, and provide accurate detection of high cognitive workload levels with an accuracy of 75% and a time resolution of 1 minute. Such an online cognitive workload monitoring system could provide valuable inputs to decision making instances, such as the operator's state and performance, thus, allowing potential adaptive support to the operator.

Finally, Chapter 5 concludes this work and provides pointers for future work in this domain.

## 2 Wearable Drone Controller

### 2.1 Introduction

The use of Unmanned Aerial Vehicles (UAVs), commonly known as drones, was originally limited to the military sphere, but in the last years, it has become pervasive in both professional and domestic environments. In fact, thanks to their commercial accessibility and their large degree of versatility, the utilization of drones has exploded in different fields (e.g., aerial mapping, search and rescue, transportation and delivery, and even for private leisure [68]). Nowadays, drones are extremely interesting devices for search and rescue applications. For instance, once a disaster occurs, rescuers have to gather information to immediately evaluate each emergency situation upon arrival. This first task is crucial to take the proper decisions and effectively master the emergency. However, gathering information from a chaotic place is hard and time consuming, because accesses are often difficult and resources relatively limited. A drone, or a network of drones, can facilitate and accelerate this task, especially providing information, which is not available from a ground perspective. Moreover, drones can be used to establish communication with victims, or to provide them first assistance (e.g., with water, oxygen, or moral support).

Even though drones find many applications in the field of search and rescue, an effective and efficient utilization in real missions is still in an embryonic stage. One of the main problems is that current Human-Machine Interfaces (HMIs), such as joysticks, keyboards, or touch screens, are neither natural nor intuitive, as the user requires training and concentration during operation [68, 118]. Drone teleoperation with traditional remote controllers is in fact a challenging task, which becomes even more demanding during long-term operations [133] or under stressful conditions

## Chapter 2. Wearable Drone Controller

---

[194]. In many cases, this limits the use of robots to highly trained professionals [28, 29]. The complexity of current Human-Machine Interfaces (HMIs) may result in a loss of control bringing the user to unexpected and catastrophic situations (Fig. 2.1), with the risk of crashing the drone, and consequently, severely compromising the outcome of a mission. Therefore, there is a growing demand for new HMIs to enhance the control and better interact with drones [141, 155]. The development of more intuitive control interfaces could improve flight efficiency, reduce errors, and allow users to shift their attention from the task of control to the evaluation of the information provided by the drone.



Figure 2.1 – Current controllers are neither natural nor intuitive, and can lead the user to unexpected situations with catastrophic outcomes. For instance, in 2015, a drone crashed onto a slalom ski course and nearly hit skier Marcel Hirscher (CNN).

In the last decade, researchers have focused on the improvement of traditional HMIs by looking for more natural and intuitive solutions. For instance, recent studies make use of external tracking systems to translate upper body movements into command for flying robots, or drones [120, 137, 157]. Following the same objective, other researchers have used moving platforms [81, 162] combined with virtual reality to control real drones [32]. However, the common weakness of all these studies is portability. In fact, most of the HMIs presented in these studies rely on bulky setups, which are not easily transportable. Therefore, further investigation is needed to find new solutions that are both intuitive to use and portable.

By comparing the approaches found in the literature, we can identify different modes of operations. For instance, while using traditional controllers, operators typically

hold the controller in a standing position, but in principle, they could assume almost any pose (e.g., sitting or lying). On the other hand, platforms such as Birdly [162] and Hypersuit [81] impose a lying position, which may be more closely associated with the idea of flying [108]. To the best of my knowledge, there is no study indicating which one is the most suitable position and which one requires less physical strength. Therefore, while designing a new portable HMI, this aspect has to be considered. While operating at the most comfortable position, the user will not only be able to perform with the least physical effort, but it will probably help to reduce the barrier of current HMI.

In general, while designing portable devices, one of the main issues is energy consumption [151]. In fact, elevated energy consumption affects the battery lifetime and consequently also the operational time. To increase the operational time, we could increase the battery capacity, but this approach will affect the size of the device and consequently also the portability. Instead, a smarter approach is to reduce the energy consumption. Thus, for the application targeted in this work, a particular attention was paid to memory storage and communications [152].

### Contributions of this Chapter

The development of a wearable system for drone teleoperation in search and rescue missions is the focus of the work in this chapter, which proposes the following contributions:

- To provide a more portable solution, I developed a wearable embedded system for drone control, which is based on upper body movements. The proposed system has been integrated into a soft exoskeleton, called FlyJacket [153], and validated for the teleoperation of both a simulator and a real drone. To this aim, I ran different experiments and participated in demonstrations during public events, such as Cybathlon and EPFL Open Days.
- Moreover, to find the best position that a user should assume while using the proposed wearable system, I analyzed the physiological response of different users. The study showed that sitting is the best position compared with standing and lying.
- Finally, to optimize the battery lifetime of the proposed system, I presented a method that drastically reduces the communication traffic. The method

## Chapter 2. Wearable Drone Controller

---

was tested while driving a real drone with the proposed wearable system and showed a 11.9% reduction of the total energy consumption.

### Publications

This work yielded the following publication and patents:

- C. Rognon, S.Mintchev, F. Dell'Agnola, A. Cherpillod, D. Atienza and D. Floreano. "FlyJacket: an Upper-Body Soft Exoskeleton for Immersive Drone Control." in IEEE Robotics and Automation Letters, 3(3):2362-2369, 2018. Video: [youtu.be/L0FTPYkLKHl](https://youtu.be/L0FTPYkLKHl)
- C. Rognon, D. Floreano, S.Mintchev, A. Concordel, F. Dell'Agnola, D. Atienza, "Jacket for embodied interaction with virtual or distal robotic device", International Patent Application No. PCT/IB2017/055410, filed on September 8, 2017.
- J.Miehlbradt, F. Dell'Agnola, A. Cherpillod, M. Coscia, F. Artoni, S.Mintchev, D. Floreano, D. Atienza, S.Micera. "Teleoperation with a wearable sensor system", UK Patent Application No. 1810285.5, filed on June 22, 2018.

### Chapter Outline

The rest of the chapter is organized as follows. Section 2.2 gives an overview of the state of the art about Natural User Interfaces (NUIs), focusing on drone controllers based on body gestures. Section 2.3 describes the proposed wearable embedded system for drone control based on upper-body movements. Section 2.4 describes the steps applied to experimentally validate the system and Section 2.5 reports the consequent results. Finally, Section 2.6 provides the main conclusions of this chapter.

## 2.2 Natural User Interface: State of the Art

In general, but especially in Search and Rescue (SAR) missions, robot teleoperation with traditional controllers (Fig. 2.2a) is a non-intuitive and challenging task. For this reason, the use of robots in the SAR field is still restricted to simple missions and highly trained professionals [28, 29]. Over the last decade, researchers have focused on the improvement of traditional HMIs by looking for more immersive



## 2.2. Natural User Interface: State of the Art

(Fig. 2.2b) or accessible (Fig. 2.2c) solutions, by including a screen for visual feedback or using smartphones, respectively. However, from the usage point of view, these new interfaces are not different than a traditional controller.



Figure 2.2 – Drone Controller: State of the Art

More recently, the research community has focused on the development of gesture-based NUIs [68] to control the flight by following three different approaches, thus providing more natural and intuitive solutions. The first approach makes use of moving platforms on which a person can lie horizontally with the arms spread out to control the flight. Two examples are shown in Figure 2.3, where a user can control the flight of a simulated bird or wing-suit by moving both hands and arms using a platform such as Birdly (Somniacs SA, Zurich, Switzerland) [162] or Hypersuit (Theory, Paris, France) [81]. Although these platforms were designed to let people fly like a bird in virtual reality, it has been shown that they can also be used for immersive control of a real drone [32]. Despite the impressive rendering of flight experiences, these platforms are bulky and heavy, which prevents their usability in real-world drone operation [153].

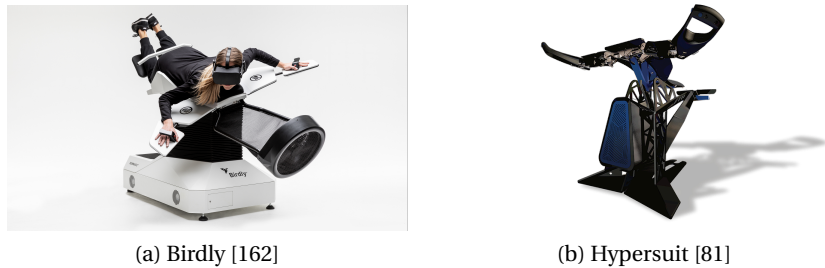


Figure 2.3 – Natural User Interface (NUI) based on moving platforms.

The second approach relies on the use of upper-body gestures recorded with exter-

## Chapter 2. Wearable Drone Controller

nal motion tracking systems, such as Microsoft Kinect™ [107] and Vicon™ [192]. This approach is used in different studies and allow a person to control a drone by using a larger variety of hand or body gestures [53, 108, 120, 137, 157]. However, as shown in Figure 2.4, the required equipment is still quite substantial. In fact, both a motion tracking system and a computer are needed to acquire, process and translate the gestures into commands that can be finally sent to the drone. Moreover, such motion tracking systems, Vicon for instance, often require an accurate calibration process previous utilization. This process requires both time and clean environments, elements that are not available in SAR.



Figure 2.4 – Natural User Interface (NUI) based on video capture systems [157].

The third approach relies on wearable sensors. The possibility of monitoring human postures and gesture with wearable sensors has been demonstrated in different studies [5, 92] and opened the doors to outdoor measurement, such as alpine ski monitoring [50]. Wearable devices have been used as well in robotics, where it has been shown that wearable sensors embedded in exoskeletons can enhance both control intuitiveness and immersion [24, 119]. Indeed, human-robot interfaces can be improved by focusing on natural human gestures captured by wearable sensors. Following this approach, a different study used Electromyogram (EMG) sensors, placed on the forearm, to track hand gesture and control a drone [167]. Although this approach is quite intuitive, the use of the hands prevents the execution of parallel tasks. In another study [70], the authors developed a control strategy based on pointing gestures. This method is intuitive as well and particularly indicated for the teleoperation of multiple robots. In fact, as shown in Figure 2.5, the user only needs to indicate which robot has to move and where it has to go. Although this interaction makes use of the hand, its use is limited in time. Therefore, with the solution proposed in [70], the user has most of the time its hands free. However, few

## 2.2. Natural User Interface: State of the Art

limitations still exist. First of all, this approach is limited to short range operations, as small errors in the detection of the angles would result in an important drift in a distant pointed position. Moreover, the user needs to be in visual contact with the drone permanently, but this problem could be solved with the use of extended reality. Another problem is the level of immersion. The use of virtual reality could partially solve this issue, but the immersion would be limited to the environment. In other words, the user could feel of being located at the place where the robot is seen (self-location), but he would not be able to experience the sense of owning the distant robot (self-identification). In fact, to experience a self-identification with a robot, a link between user and robot should be established. To this aim, the robot should mirror the human's motions, posture, or bodily signals [20]. Therefore, a further investigation considering the body self-consciousness paradigms is needed to help relocate the user's perception into the distal drone to improve immersion and piloting performances.

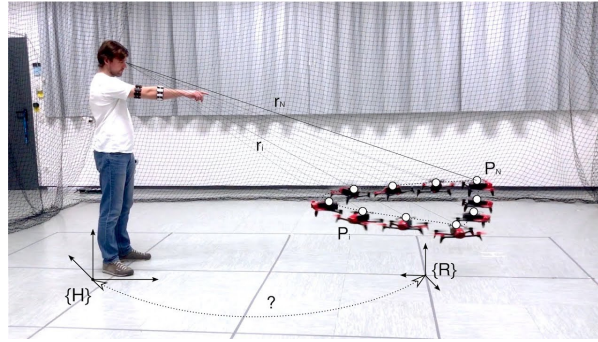


Figure 2.5 – Natural User Interface (NUI) based on wearable devices [70].

Although a lot of effort has gone into HMI improvements [95], an efficient deployment of drones in SAR missions is still a challenge. One of the main reasons is the lack of an appropriate NUI. Traditional solutions are in fact neither natural nor intuitive [68, 118]. Recent studies based on moving platforms [32] and visual motion tracking systems [108] seem to address this problem. However, most of these solutions lack portability, which makes them unsuitable for SAR missions. Moreover, the control strategies are typically based on gestures that are predefined by the developers, who do not really consider the needs of the final users and force them to learn and adapt to the proposed solution. This training process is often not addressed by the research community [40], and in SAR is quite important as rescuers need to train on many topics. The use of wearable sensors is covered by only

few studies [70, 167], where the control is carried out from a third-person view or through a monitor, which limits the immersion. Therefore, there is a need for further investigation to find easy-to-use and portable solutions.

### 2.3 Design of a Natural-User Interface for Drone Control

The design of a novel natural-user interface for drone control was handled by a collaboration between three laboratories of École Polytechnique Fédérale de Lausanne (EPFL), namely, Translational Neural Engineering (TNE), Laboratory of Intelligent Systems (LIS), and Embedded System Laboratory (ESL). In a first phase, we performed an open-loop experiment (described in subsection 2.4.1) to identify the spontaneous gesture-based interaction strategies of naive individuals with a distant device, such as a drone. The idea was to exploit this information of the upper-body motion to develop a natural data-driven body-machine interface to efficiently control this drone. Moreover, we also wanted to investigate different postures (i.e., lying, sitting, and standing) to identify which one is the most suitable for the teleoperation of a drone. In this first phase of the project, my role was limited to the analysis of the comfort/discomfort of these different postures, to identify the position that requires the least physical strength. Therefore, as I did not take part in the investigation aimed to identify the gesture-based interaction with a drone, I will only provide the important information relative to my work. However, all the details can be found in [108, 109].

In a second phase, we performed a closed-loop experiment (described in subsection 2.4.2) to validate the gesture-based interaction strategy. This analysis was done at TNE and all details can be found in [108, 109]. In this phase, my role was to propose and validate an equivalent, but portable solution, that is, a wearable system (subsection 2.3.1) implementing the aforementioned gesture-based interaction strategy (subsection 2.3.2). As shown in Figure 2.6, the system proposed in [108, 109] was in fact based on an eight-camera motion capture system from Vicon [192], which is not easily portable and requires an important calibration process previous utilization.

Finally, in the third and last phase of this collaboration, my wearable system was embedded in a soft upper-body exoskeleton called FlyJacket. The jacket was combined with a glove to include the control of speed, take off, and landing. Both the FlyJacket fabric and the glove were designed and developed at LIS, all details about the fabric design can be found in [153]. In this phase, we organized an experiment (described in subsection 2.4.3) to validate the teleoperation of a real drone with the FlyJacket

## 2.3. Design of a Natural-User Interface for Drone Control

---



Figure 2.6 – Natural User Interface (NUI) based on a camera motion capture system.

design.

### 2.3.1 Proposed Wearable Drone Controller

With the goal of combining portability and intuitiveness, I developed a novel wearable embedded system for human-machine interactions, which has been designed and tested for the teleoperation of a real drone. The proposed system reflects the intuitive behaviour of the solution proposed in [108], but it is fully wearable and consequently also portable. As shown in Figure 2.7, the controller consists of a wearable body sensor network, which is composed of a central sensor node (red square on the human's chest) connected with additional sensors (red circles). The central node acquires the signals for the upper-body movement tracking, combines them, and translates them into commands that are finally transmitted to the drone.

It has been demonstrated that measurements of the torso orientation are enough to fly a drone at constant speed with relatively high performance [108]. Therefore, I designed a wearable body sensor node that tracks the motion of the torso and decodes the movements into commands for the drone as proposed in [108]. However, in order to allow an implementation of more advanced flight styles, I proposed a modular design that could be extended to implement more advanced control strategies. For instance, the proposed design can be used with the following configurations:

- one single central node, thus, only one Inertial Measurement Unit (IMU),
- a central node with two additional IMUs,
- a central node with four additional IMUs.

These three configurations are suggested by previous studies [5, 12, 50, 149], which

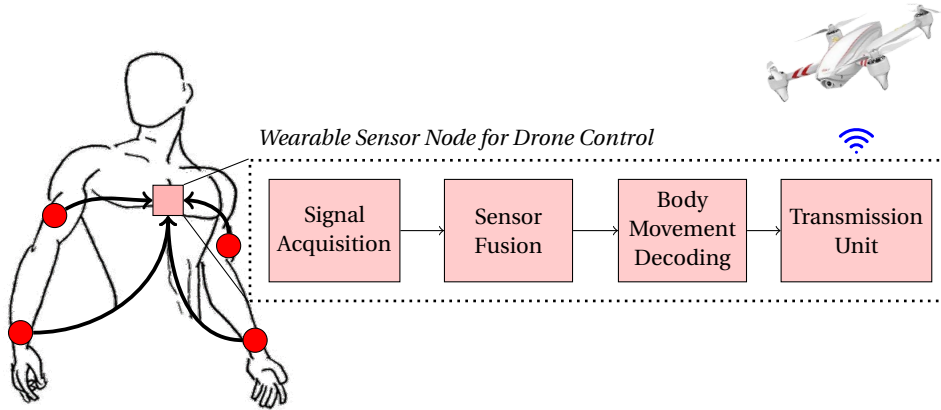


Figure 2.7 – Context diagram of the proposed wearable drone controller. A central wearable sensor node (red square on the human’s chest) embeds the methods grouped by the dashed-line box. Additional nodes (represented by red circles), or simply Inertial Measurement Units (IMUs), are used to track the movements of arms.

demonstrate the possibility of tracking the movement of the body segments with IMUs.

The proposed central node is intended to be placed on the chest or the back of the torso. In principle, both options are valid [12]. However, I prefer the back rather than the chest, as it facilitates to embed the sensors in a jacket, which may have a zipper or buttons on the chest side. The additional IMUs can be placed on the upper arms and forearms to record the motion of these body segments [12], and can be used, for instance, to control the Degrees of Freedom (DoF) that were not considered in [108].

### 2.3.2 Control Strategy: Design and Evaluation

The general idea behind the method applied to find the control strategy implemented in the proposed drone controller is shown in Figure 2.8. The drone controller, in particular the control strategy, is the part that has to be identified. The control loop starts with an action from the user, who applies a command (1) based on its desired trajectory and the real trajectory of the drone (3). Then, a drone controller translates the action of the user into commands (2) for the drone. Finally, to identify a proper control strategy, we measured the actions of the user and the commands applied to

### 2.3. Design of a Natural-User Interface for Drone Control

the drone.

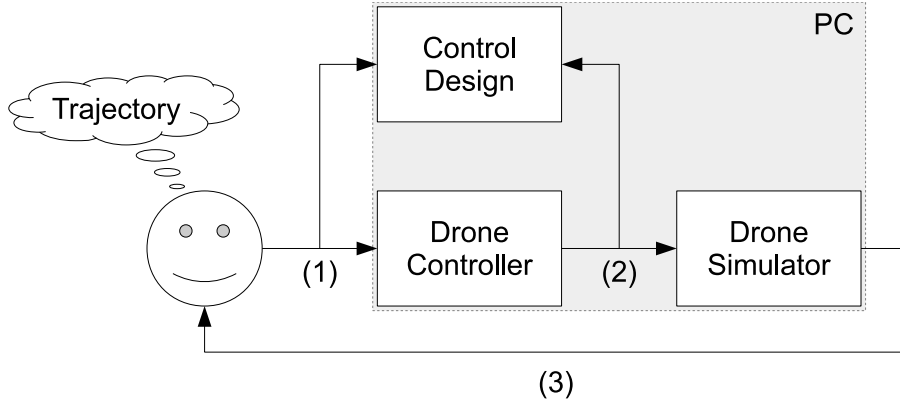


Figure 2.8 – Drone-control identification loop. A user applies an action (1) based on the trajectory he has in mind and the real trajectory of the drone (3). Then, a drone controller translates the action of the user into commands for the drone (2). To identify the controller needed, we measured the actions of the user and the commands applied to the drone.

The investigation aimed to find the control strategy; that is, the identification of upper-body gestures for the development of an intuitive control interface was done in a parallel study at TNE and results can be found in [108]. From my side, I investigated three different postures that could be suitable for a user to assume while using the proposed wearable drone controller (i.e., lying, sitting, and standing). In this regard, we considered both an empirical and an objective evaluation of the level of discomfort related to a particular posture assumed by the user. The empirical evaluation relies on a self evaluation provided by the research participants after each experimental session. On the other hand, the objective evaluation is based on the physiological response that I measured from the participants, with the aim of to detect any physiological form of stress that could be assigned to a discomfort experienced by the participants.

Being comfort a subjective sense of physical or physiological ease and not really measurable, stress metrics are chosen as suggested in [183]. Therefore, I measured Electrocardiogram (ECG) and Electrodermal Activity (EDA) to detect if there is any form of stress factor that could influence the normal execution of the task. All details about the experiment and sensors are given in sections 2.4.1 and 2.4.1, respectively. EDA measurements were eventually discarded, as the measurements were too noisy and did not show any significant response. From the ECG signal, I extracted the

## Chapter 2. Wearable Drone Controller

R-peaks, which are needed for the analysis of Heart Rate Variability (HRV). For that purpose, I used an efficient algorithm suitable for real-time applications [130]. Both time and frequency domain methods were considered for the HRV analysis as described in [174]. For the time domain, I computed features such as the Root-Mean Square of Successive Differences (RMSSD) and the mean value of the so-called normal-to-normal (NN) intervals, the intervals between two consecutive QRS peaks. For the frequency-domain analysis of HRV, I first estimated the Lomb-Scargle Power Spectral Density (PSD) of the NN intervals [143], and then I computed the power in the Low Frequency (LF) band (between 0.04 and 0.15 Hz), in the High Frequency (HF) band (between 0.15 and 0.4 Hz), and the ratio of the two, referred to as the LF/HF ratio. The experiment conducted in this regard is described in subsection 2.4.1 and results are shown in subsection 2.5.1.

### 2.3.3 Wearable Sensor Node for Drone Control: Hardware Design

A block diagram of the proposed wearable body sensor node is shown in Figure 2.9. The main components and functionalities are briefly described in the next paragraph, which aims to provide a general overview of the system. Then, in the following subsections, all components are detailed.

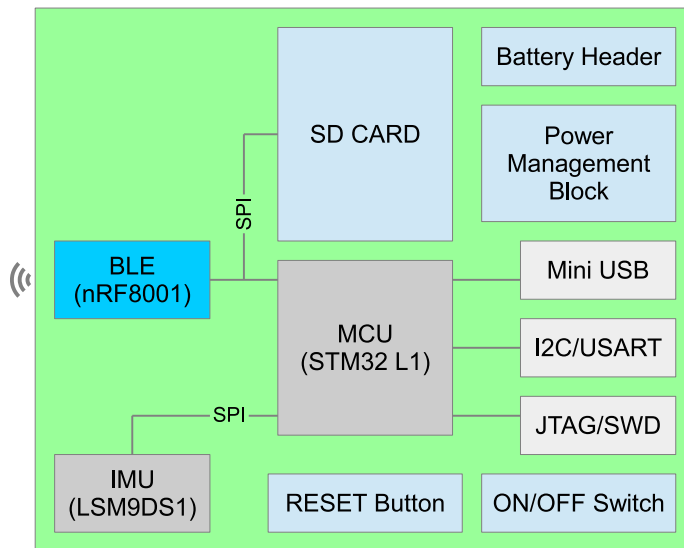


Figure 2.9 – Block diagram of the proposed wearable drone controller.

The real-time signal acquisition and processing is handled by a Microcontroller



### 2.3. Design of a Natural-User Interface for Drone Control

Unit (MCU). An IMU, connected to the MCU via Serial Peripheral Interface (SPI), is included for tracking the torso movements of the pilot. Finally, a Bluetooth Low Energy (BLE) module, connected as well to the MCU via SPI, is used to send the commands (i.e., pitch and roll) to an intermediate device that can transmit the commands to the drone. For instance, a smartphone or tablet can be used for this purpose [132]. Moreover, this intermediate device is needed to receive the commands from the glove, which is used to control the remaining DoF that are not controllable with the proposed wearable sensor node. The prototype of the proposed wearable body sensor node is showed in Figure 2.10.

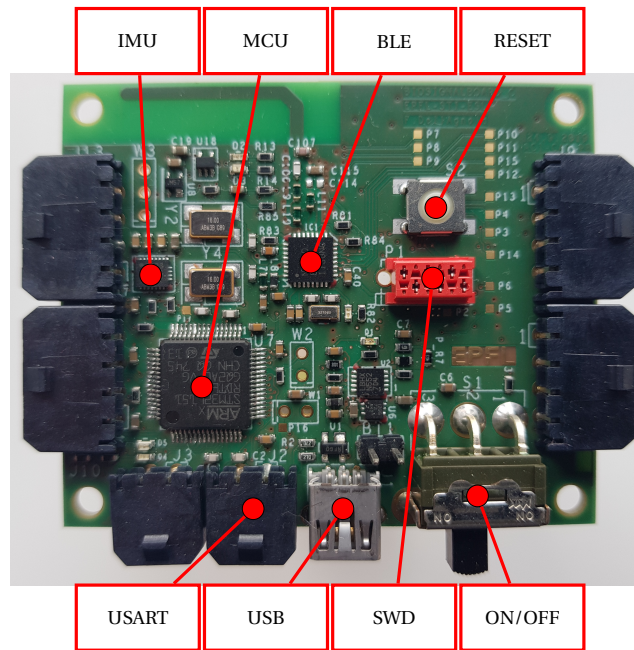


Figure 2.10 – Prototype of the proposed wearable drone controller.

In contrast to the solution proposed in [108], where data from eight cameras are sent to a computer and used to track the motion of the upper body, this solution requires fewer sensors. Moreover, the computing is done directly in the proposed wearable node, with low energy intercommunication between sensors and MCU. In other words, the upper body movements are acquired by the IMUs, transmitted to the MCU for decoding, and finally, only few commands are transmitted to the drone.

### Micro-Controller Unit

The core element of the proposed sensor node is a STM32L1 MCU from STMicroelectronics [166], which has been used in wireless sensors for medical sensing applications [169]. The STM32L1 is an ultra-low-power platform equipped with an Arm®Cortex®-M3 32-bit CPU from 32 kHz up to 32 MHz, 384 Kbytes of Flash memory, 48 Kbytes of RAM, 12 Kbytes of true EEPROM, and 128-byte backup register. Such computations and storage resources are not all required for the application presented in this chapter. Less resources would be enough to assure the usage capabilities required to control a drone. However, to ensure both monitoring and processing of physiological signals detailed in chapter 4, a microcontroller with additional resources is required.

Moreover, the STM32L1 includes analog peripherals; such as, three operational amplifiers, a 21-channel 12-bit ADC 1 Msps, a 2-channel 12-bit DAC with output buffers, two ultra-low-power-comparators, which can be used for instance to monitor input voltage and battery level. Furthermore, the STM32L1 includes 11 timers; namely, one 32-bit, eight 16-bit, and two watchdog timers; which assure the synchronization of the application. Finally, the STM32L1 includes a 12 channels DMA controller and 12 peripheral communication interfaces, namely, one Universal Serial Bus (USB) 2.0, five Universal Synchronous and Asynchronous Receiver-Transmitters (USARTs), three SPIs, two I2Cs, and one SDIO interface. These interfaces, SPIs in particular, are used to connect additional on-board modules (i.e., sensors, communication, and storage modules) to the microcontroller.

### Inertial and Magnetic Sensors

There are different valid solutions to measure acceleration, angular speed, and magnetic field with a single System on Chip (SoC). The most relevant solutions are reported in Table 2.1. All reported IMUs include a 3-axial accelerometer, a 3-axial gyroscope, and a 3-axial magnetometer, with similar characteristics in terms of measurement range, sampling frequency, and power supply, but different energy consumption. In terms of current consumption the new BMX160 sensor from Bosch Sensortec is definitely the best choice. Unfortunately, such a device was not available at the moment of the fabrication of the proposed system. The second best option, and the best option during design time, is the LSM9DS1 [165] from STMicroelectronics. Such a device was chosen because of its limited energy consumption, but also because of high performance and low cost. Other solutions, such as the modules of

### 2.3. Design of a Natural-User Interface for Drone Control

the MTi-1-T series including on-board sensor fusion (SF) algorithms were evaluated as well. Although on-board sensor fusion would limit the load on the application processor, such a solution was discarded because of the size, the current consumption, and the cost that are still considerable.

Sensor	Manufacturer	Current (mA) Consumption	Voltage (V) Range	Unit Price (CHF)
BMX160 ( <i>New</i> )	Bosch Sensortec	1.59	1.7 - 3.6	~6.50
LSM9DS1TR	STMicroelec.	1.9	1.9 - 3.6	~6.00
ICM-20948	InvenSense	3.1	1.7 - 3.6	~7.70
MPU-9250	InvenSense	3.7	2.4 - 3.6	Obsolete
BMX055	Bosch Sensortec	5	1.2 - 3.6	~6.40
MTi-1-T (+SF)	Xsens	>15	2.16 - 3.6	>180.00

Table 2.1 – Comparison of 9-axis IMUs with on-chip 16-bit Analog-to-Digital Converters (ADCs), combining a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer.

#### Communications

BLE connectivity is provided by nRF8001, a single-chip designed for low-power operation in the peripheral role (current consumption of 12.7 mA for transmission and 14.6 mA for reception). The module is particularly interesting for low-power applications due to its reduced current consumption of 2 mA and 0.5  $\mu$ A in idle state and sleep mode, respectively. The BLE module is directly connected via SPI to the STM32, which controls the nRF8001 operating modes using commands defined by the Application Controller Interface (ACI).

An auxiliary configurable interface, which can be either a master Inter-Integrated Circuit (I2C) bus or a USART, is included to access external devices, allowing the devices to gather additional sensor data or to communicate with a PC.

To allow data collection and facilitate data storage, the proposed platform also mounts a Micro SD Flash Socket, which can support microSD cards with capacities up to 2 Gbytes. This storage option has been included to avoid data streaming for applications related to data collection, where sending all sensed data over the radio to a remote device would be energetically inefficient.

A Serial Wire Debug (SWD) interface is included to communicate with the STM32

## Chapter 2. Wearable Drone Controller

---

microcontroller located on the platform through the ST-LINK/V2 debugger and programmer. Alternatively, Device Firmware Upgrade (DFU) utility can be used to interact with the STM32 system memory bootloader, running from the Flash of the microcontroller, thus allowing internal memory programming through USB, which is also used to supply the device and charge the battery.

### Power Management

A switch is used to control the power-on sequence of the node. When the node is powered down, the power regulators are disabled and the board reset is low. The node can also be reinitialized by pushing the reset button.

Lowering the duty cycle of interaction between the peripherals is a key aspect to extend battery life. To this aim, the peripherals (e.g., sensors, microSD card, USB module) are powered off by firmware when not in use. However, this is not feasible with applications that require high frequency data capture. In this regard, the proposed hardware provides features, such as timers and customized interrupt lines, to simplify application programming and enhance event-driven applications, which can be used to improve power management.

Moreover, the STM32L1 microcontroller offers different operational modes; namely, Standby mode (down to 305 nA), Stop mode (down to 0.475  $\mu$ A), Low-power run mode (11  $\mu$ A), and Run mode (230  $\mu$ A/MHz); and a wake-up time of 8  $\mu$ s, which can assure the low-power usage capabilities of the proposed sensor node.

### 2.3.4 Wearable Drone Controller: Software Design

The software running on the MCU is divided into four steps, namely, signal acquisition, sensor fusion, body movement decoding, and transmission. The signal acquisition step aims to acquire the motion of the IMU. The measurements are fused to estimate the orientation of the sensor, which being located on the torso of the pilot tracks the movements of the pilot's trunk. Then, a body movement decoding algorithm is applied to translate the movements into commands for the drone. Finally, the communication unit treats the commands and sends them to the drone.

### Signal Acquisition

The signals acquired from the IMU are acceleration, angular velocity, and magnetic field. The sampling frequency is chosen according to [12], where it is reported that 25 Hz is usually adequate for swimming; 50 Hz is often adequate for activities such as the tennis serve, if the impact of the ball and racket is not the focus of the study; and 100 Hz is often needed for quantitative analysis of activities as fast as a golf swing. Assuming a dynamic of the gestures needed to control a drone comparable with the movements in swimming, I selected a sampling frequency greater than 25 Hz. In particular, considering the options provided by the sensors, I have chosen a range of  $\pm 4$  g and sampling frequency at 119 Hz for the accelerometer; a range of  $\pm 500$  deg/s and sampling frequency at 119 Hz for the gyroscope; and a range of  $\pm 4$  Gauss and sampling frequency at 80 Hz for the magnetometer. Due to the physical characteristics of the sensors, the choice for a common sampling frequency was not possible. Therefore, I selected such a configuration to offer a sensor fusion based on samples that are updated within the worst case delay of only 25 ms, which is less than the human response time perceived as instantaneous [111].

### Sensor Fusion

The orientation of the torso is computed with a gradient descent algorithm [99] implemented on the MCU. The algorithm is based on the measurements of the IMU, which are fused to estimate the orientation of the sensor and consequently also the orientation of the torso of the pilot. In particular, I estimated the lateral bending angle, the sagittal bending angle, and the rotational angle [5, 195, 196]. The gradient descent algorithm was selected for its performance and ability to operate at low sampling rates, which significantly reduces the power consumption [99]. Since the dynamic of the movements is slow, the gradient descent algorithm runs at a frequency of 50 Hz, which is a good compromise between performance and power consumption.

### Body Movements Decoding

For the considered case study, where a drone is flying at a constant speed, only roll and pitch angles are controlled. These commands are computed from the orientation of the pilot's trunk, as suggested by the parallel study conducted at TNE and reported in [108]. The mapping of the trunk's movements into drone commands are defined

## Chapter 2. Wearable Drone Controller

---

as follow:

$$\begin{bmatrix} \text{Pitch} \\ \text{Roll} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} \text{Lateral Bending Angle} \\ \text{Sagittal Bending Angle} \\ \text{Rotational Angle} \end{bmatrix} \quad (2.1)$$

### Communication Unit

To decrease the energy consumption due to communications [150], I proposed a simple method that reduces the transmission rate and consequently also the energy consumption of the system. The pseudo-code of the applied algorithm is reported in Algorithm 1.

---

#### Algorithm 1 Communication

---

```

1: procedure TRANSMIT( $\mathbf{x}[k]$ )
2:    $X_{\text{prev}} \leftarrow \mathbf{x}[k-1], \dots, \mathbf{x}[k-N-1]$ 
3:    $X_{\text{curr}} \leftarrow \mathbf{x}[k], \dots, \mathbf{x}[k-N]$ 
4:   if  $\text{RMS}(X_{\text{curr}} - X_{\text{prev}}) > \text{threshold}$  then
5:      $y \leftarrow \text{mean}(X_{\text{curr}})$ 
6:     Send( $y$ )

```

---

Let  $\mathbf{x}[k]$  be a vector including both Roll[ $k$ ] and Pitch[ $k$ ], the commands that have to be transmitted to the drone at instant  $k$ . Given the finite sequence of  $N + 1$  previous commands  $X_{\text{prev}}$  defined as follows:

$$X_{\text{prev}} = \mathbf{x}[k-1], \dots, \mathbf{x}[k-N-1] \quad (2.2)$$

and the sequence  $X_{\text{curr}}$  including the current command  $\mathbf{x}[k]$  and the  $N = 9$  previous ones:

$$X_{\text{curr}} = \mathbf{x}[k], \dots, \mathbf{x}[k-N] \quad (2.3)$$

I calculated the Root Mean Square (RMS) of their difference as follows:

$$\text{RMS}(X_{\text{curr}} - X_{\text{prev}}) = \sqrt{|X_{\text{curr}} - X_{\text{prev}}|^2} \quad (2.4)$$

Finally, only if  $\text{RMS}(X_{\text{curr}} - X_{\text{prev}})$  is greater than a certain threshold, the mean value of the sequence  $X_{\text{curr}}$  is transmitted to the drone. Otherwise, nothing is sent, in other words, commands generated from small pilot's adjustments that are below the

threshold are not transmitted. The moving average approach, that is, the mean value of the sequence  $X_{\text{curr}}$ , is applied to remove high frequency noise [108]. Moreover, it attenuates any possible sudden correction of the trajectory of the drone that may result from an accumulation of not transmitted commands.

The threshold is chosen based on a trade-off between communication reduction and control accuracy. A high threshold will drastically reduce the transmission rate and consequently deteriorate the control accuracy. On the other hand, a small threshold will not reduce the transmission rate and transmit the entire sequence of commands without affecting the control accuracy. To quantify the control accuracy, I used the Root Mean Square Error (RMSE) between the transmitted signal and  $y$ , where  $y$  for this purpose is the mean value of the sequence  $X_{\text{curr}}$  computed at every iteration (i.e., before the if case of line 4 in Algorithm 1). The transmission reduction factor is calculated as the ratio between transmitted commands and the commands received by the Algorithm 1. The value of this threshold is reported in subsection 2.5.4 and obtained from experimental data collected as described in subsection 2.4.1.

## 2.4 Experimental Validation

The proposed wearable system was validated with three main experiments, which were approved by the École Polytechnique Fédérale de Lausanne Brain Mind Institute Ethics Committee for Human Behavioral Research and the Ethics Committee Geneva. In the first two experiments, a First-Person View (FPV) of the flight was shown through a Head-Mounted Display (HMD).

The first Virtual Reality (VR) based experiment was organized to collect data aimed to design the control strategy. Moreover, we collected data to evaluate different postures that a user could assume while controlling a drone (i.e., lying, sitting, and standing). The second experiment, again in a virtual environment, aimed at comparing the performance of the proposed system against an eight-camera motion capture system from Vicon [192]. Data collected from this experiment was used to fix the threshold introduced in Equation 2.3.4. Finally, the third experiment was designed to validate the proposed sensor node while controlling a real drone.

For the first experiment, we recruited 19 young healthy participants, 17 of which completed the experiment ( $23.7 \pm 1.1$  years old, one woman). The two participants who decided to interrupt the experiment suffered from VR sickness and their data were excluded from the analyses. Unfortunately, we were not able to recruit a gender bal-

anced sample of participants, as the contacted women did not manifest a particular interest in this study. However, considering that in 2018, only 9.9% of the firefighter in Switzerland were women [56], around 5% in England [69], and 7% in United States [33], the selected sample of participants is quite representative. In general, participants were not used to VR experiences. Participants reported considerable experience with videogames (i.e.,  $5.9 \pm 1.1$ ), rated from a scale between 1 (none) and 7 (regular use). However, the abilities acquired while playing videogames cannot be translated into the gestural interface used in this study, as this type of control interface is different than traditional keyboards or joysticks. Moreover, participants' experience with VR was rated only  $1.4 \pm 1$  out of 7. Therefore, participants' experiences with videogames should not have an impact on the use of the gesture-based control strategy under evaluation.

For the second experiment, I recorded data from three new participants ( $30.7 \pm 6.4$  years old), one woman and two men, who volunteered to participate in the study. The same three participants, who showed to have good skills in flying a drone in the VR simulator, were recruited for the third experiment as well. It is important to remind that in both the second and the third experiment, under evaluation is the implementation only, and not the control strategy as it was for the first experiment. Therefore, the fact that volunteers have particular skills is absolutely not relevant. The reason for this biased choice was simply to avoid as much as possible any risk of crashing the drone. All participants provided informed consent and volunteered to participate in the study. The details about the experiments are provided next.

### 2.4.1 Open-loop Experiment with Virtual Reality

To collect data for designing the control strategy and determining the recommended posture, we proceeded as sketched in Figure 2.11. Both a drone simulator for PC and an auto-pilot were developed at LIS, see [108] for details. The auto-pilot provided a sequence of maneuvers (Figure 2.11 (2)), which forced the drone simulator to follow a predefined flying trajectory (Figure 2.11 (3)). Then, the participants were asked to recognize (Figure 2.11 (a)) and follow the trajectory of the simulated drone by applying self-selected and flight-like upper-body movements (Figure 2.11 (b)). Finally, to design the control strategy and to evaluate the postures that a user could assume while controlling a drone (i.e., lying, sitting, and standing), we collected data from both participants and the simulator. From participants, we recorded upper-body movements, physiological signals (Figure 2.11 (1)), and position. Participants also filled a questionnaire aimed to evaluate different aspects of the experiment, such



## 2.4. Experimental Validation

as enjoyment, tiredness, audio-video quality, clarity of the provided instructions, and any lack of training time. More important, they evaluated the difficulty of performing the proposed tasks and ranked the different positions considering comfort, intuitiveness, and level of immersion. From the simulator, we stored the applied sequence of maneuvers. Being this work a collaboration between different labs, I reported in section 2.5.1, the results that I obtained from the physiological signals. For more details, please refer to [108, 109].

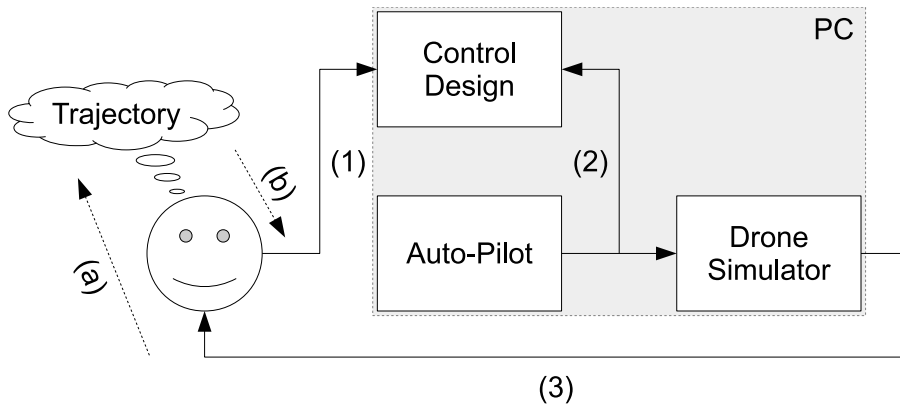


Figure 2.11 – Graphical representation of the open-loop approach applied for the design of the control strategy.

### Experimental setup

As shown in Figure 2.12, research participants controlled the FPV flight simulator while lying (on the left), sitting (in the center), and standing (on the right). They wore an HMD (Oculus Rift, Development Kit 2, Oculus VR, LLC), different reflective markers, and sensors.

The simulator was created using FlightGear with the YASim dynamic models and was running on a PC. Low-level controls, such as propeller thrust and flap inclination were regulated through PID controllers implemented in a C++ software running in parallel. This software also generated a randomized maneuver list before each flight sequence. For synchronizing the beginning of each maneuver, voltage pulses were sent to the motion capture system through an Arduino board.

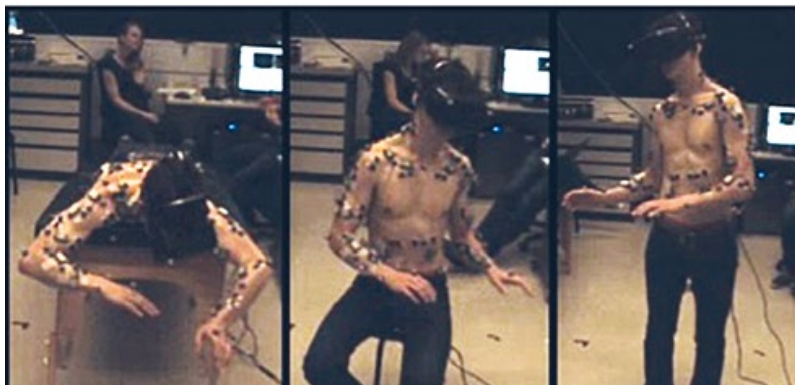


Figure 2.12 – Setup of the open-loop experiment. A participant, wearing a Head-Mounted Display (HMD), different reflective markers, and sensors; and performing turning maneuvers while lying (on the left), sitting (in the center), and standing (on the right). In the background, a PC running the drone simulator and the acquisition routines.

### Signal acquisition

To define the control strategy as well as the number and the position of the sensors that will be required to drive the drone, both muscular activity (32 muscles) and kinematics of the upper body were monitored [108]. For the muscular activity, we used two wireless transmission systems (Desktop DTS, Noraxon Inc., USA) and superficial Ag-AgCl electrodes (Kendall H124SG, EMG electrodes, 30x24 mm) located as shown in Figure 2.13 (green rectangles). The kinematic activity was measured with Vicon, an optical camera system for motion capture.

To evaluate the stress level that could be related to uncomfortable postures, I measured both ECG and EDA [183], using INYU, a wearable device from SmartCardia [169]. ECG, in particular lead II, was acquired from the thorax (red ellipses in Figure 2.13), while EDA was measured on the shoulder (blue circles in Figure 2.13). EDA is typically measured from fingers, but as we wanted to have sensors that could be integrated into a jacket, we opted for this unusual location, which has been demonstrated to be a valid alternative [191]. Both measurements were acquired with a sampling frequency of 250 Hz.

As an assessment, a questionnaire was used to collect a self-evaluation of both the applied self-selected control strategy (i.e., the flight-like upper-body movements) and the comfort level of the proposed postures (i.e., lying, sitting, and standing).

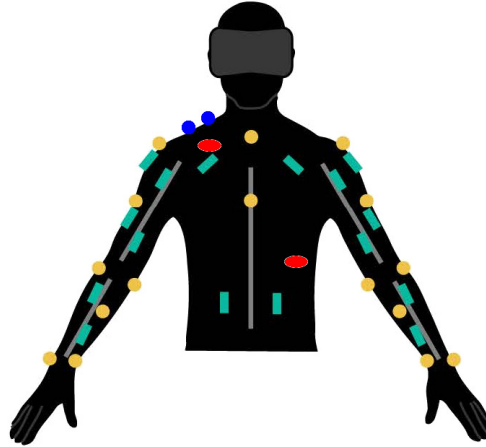


Figure 2.13 – Sensor placement for the open-loop experiment. Gray lines represent the upper-body segments, yellow circles the kinematic markers, green rectangles the Electromyogram (EMG) sensors, red ellipses the Electrocardiogram (ECG) electrodes, and blue circles indicate the Electrodermal Activity (EDA) electrodes.

### Study protocol

The beginning of the experiment is characterized by a setup phase, which includes a period for providing the necessary explanations to the participants and the time for the sensor placement. Then, the participants were shown an automatically controlled flight sequence in FPV through an HMD, and were instructed to follow the movements of the simulated drone using self-selected and flight-like upper-body movements. The proposed sequence consisted of ten alternations between 6-seconds constant forward (FW) motions (speed 12 m/s) and 7-seconds directional maneuvers, such as Right banked turn, Left banked turn, Upward pitch, and Downward pitch, presented in a randomized order. To assist the participant in recognizing the maneuvers they have to perform, each new maneuver was notified by a text indication one second prior to its start. After each sequence, a resting period is proposed to recover and to fill a questionnaire (Q). Each posture-related phase was repeated twice, one for each drone (i.e., quadcopter and fixed-wing drone). A graphical representation of the experimental protocol is shown in Figure 2.14. The use of two different types of drones aimed to verify if the control strategy was suitable for both types of drones. However, this analysis was performed in a parallel project (see [108, 109] for more details).

## Chapter 2. Wearable Drone Controller

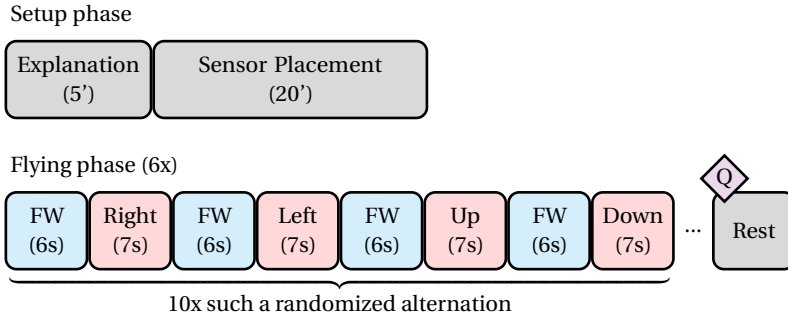


Figure 2.14 – Protocol of the open-loop experiment. The experiment starts with a setup phase including a period for explanations and sensor placement. Then, as a flying phase, we consider all combinations between drone (i.e., quadcopter and fixed-wing drone) and posture (i.e., lying, sitting, and standing). For each combination, presented in a randomized order, we propose ten alternations of constant forward (FW) motions and maneuvers, such as, Right banked turn, Left banked turn, Upward pitch, and Downward pitch, presented in a randomized order as well. After each sequence, a resting period is proposed to recover and to fill a questionnaire (Q).

### 2.4.2 Closed-loop Experiment with Virtual Reality

The closed-loop experiment principally aimed to validate the control strategy proposed in Equation 2.1 and developed based on the open-loop experiment (subsection 2.4.1). A second objective, but not less important, was to compare the performance in a virtual environment of the wearable system against the eight-camera motion capture system from Vicon. For this purpose, we used the same VR based flight simulator used in the open-loop experiment, but focusing on a simulated fixed-wing drone, flying with a constant speed of 12 m/s. Figure 2.15 shows a graphical representation of the setup used for this experiment.

During the experiment, participants were asked to follow a flight trajectory characterized by cloud-shaped waypoints. To this aim, they had to correct (Figure 2.15 (a)) the real flying trajectory (Figure 2.15 (3)) and follow the waypoints by applying (Figure 2.15 (b)) trunk movements. The upper-body kinematics activities (Figure 2.15 (1)) of the participants were acquired with both the proposed wearable system and the camera motion capture system. The proposed wearable system computed both roll and pitch commands (Figure 2.15 (2a)) as described in subsection 2.3.4 and transmitted without the optimization described in Equation 2.3.4 to a PC for comparison. Although the output commands of both systems were recorded for comparison, only the commands (Figure 2.15 (2b)) based on the camera motion capture system were

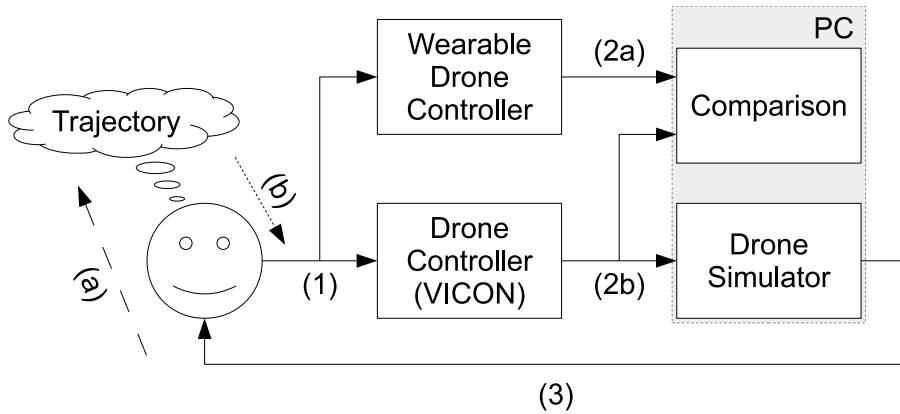


Figure 2.15 – Graphical representation of the control loop applied to compare the proposed wearable drone controller with a controller based on a camera motion capture system.

streamed to the control routine for controlling the simulator in soft real-time. Finally, the comparison of the outcomes of both gesture-based control systems was done offline on a PC.

### Experimental setup

Participants flew the simulator with the movement of their torso. To this aim, they wore an HMD (Oculus Rift, Development Kit 2, Oculus VR, LLC), different reflective markers, and sensors, as shown on the left side of Figure 2.16. A view of the VR environment seen by the participants through the HMD is shown on the right side of the figure. To provide visual feedback of the movements performed by the volunteers, an avatar was added to the scene instead of the drone. Such an avatar was also used at the beginning of the experiment to instruct the participants about the control strategy to adopt.

### Acquired signals

The proposed wearable sensor node was used to track the kinematics of the torso. As a reference, the camera motion capture system from Vicon was used to acquire the kinematics from the raw marker positions, which were imported into a custom Matlab routine using Vicon's DataStream SDK. The routine extracted the angular excursions of the torso, computed the corresponding pitch and roll angles for the

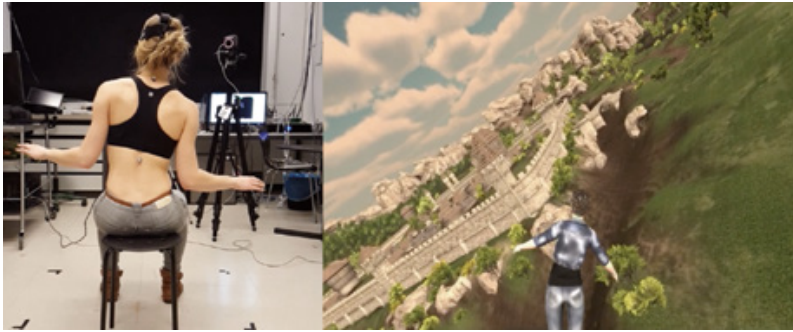


Figure 2.16 – Setup for drone control systems comparison in a virtual environment. On the left a participant flying in Virtual Reality (VR). On the right, a scene displayed in the Head-Mounted Display (HMD) worn by the participants.

virtual drone as described in Eq. 2.1, and transmitted the latter values to the simulator after applying a moving average filter to prevent instabilities.

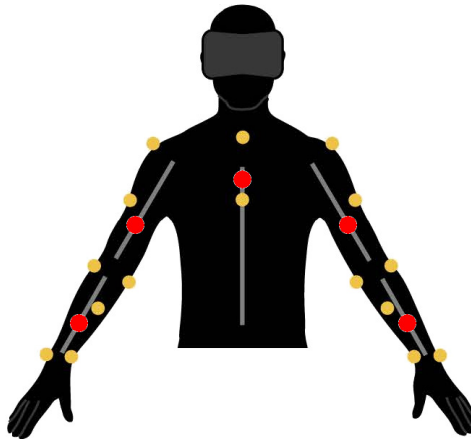


Figure 2.17 – Sensor placement for the closed-loop experiment. Gray lines represent the upper-body segments, yellow circles the kinematic markers, and red circles indicate the wearable sensor nodes.

### Study protocol

After a brief explanation about the goal of the study and the protocol of the experiment, participants were equipped with the sensors and HMD. Subsequently, they were first shown a one-minute demonstration sequence during which the simulated

## 2.4. Experimental Validation

aircraft was autonomously flying through a predefined trajectory, while an avatar displayed the control movements corresponding to the motion of the participant and consequently of the aircraft as well. The participants were then asked to fly along a randomly generated trajectory characterized by forty-two cloud-shaped waypoints (diameter 0.6 m) and distributed every 40 m. To follow the waypoints, the participants had to alternate five distinct manoeuvres (i.e., constant forward motion, right-banked turn, left-banked turn, upward pitch, and downward pitch). The participants repeated this flying sequence six times. A graphical representation of the experimental protocol is shown in Figure 2.18.

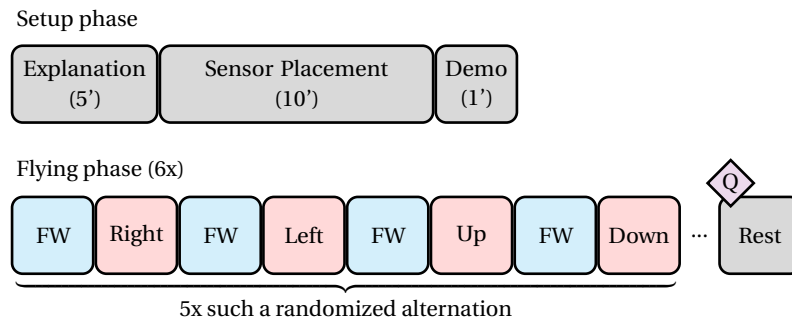


Figure 2.18 – Protocol of the closed-loop experiment. The experiment starts with a setup phase including a period for explanations, sensor placement, and a demo used to instruct the participants. Then, we propose six flying sequences consisting of five alternations of constant forward (FW) motions and maneuvers, such as, Right banked turn, Left banked turn, Upward pitch, and Downward pitch, presented in a randomized order. After each sequence, a resting period is proposed to recover and to fill a questionnaire (Q).

### 2.4.3 Experiment with a real drone

The third and last experiment presented in this chapter aimed to test the proposed wearable system for the teleoperation of a real drone (Figure 2.19). Moreover, I tested Algorithm 1 and applied it to reduce the communication rate, consequently, also the energy consumption.

During this experiment, participants controlled a real drone with upper-body gestures (Figure 2.19 (1b)), which were translated into low-level commands (Figure 2.19 (2b)) from the proposed wearable drone controller. For safety reasons, the proposed wearable drone controller was interfaced with the drone through a safety interface. Such a safety interface was there to prevent crashes of the drone. To this aim, an operator

## Chapter 2. Wearable Drone Controller

constantly supervised the trajectory of the drone (Figure 2.19 (3)) and was ready to take control and fly the drone with a traditional controller. In other words, with the safety interface, the operator selected which commands had to be forwarded to the drone (Figure 2.19 (2)); namely, those coming from the wearable drone controller (Figure 2.19 (2b)), thus, from the participant; or those coming from the traditional controller (Figure 2.19 (2a)), thus, the commands provided by the operator (Figure 2.19 (1a)).

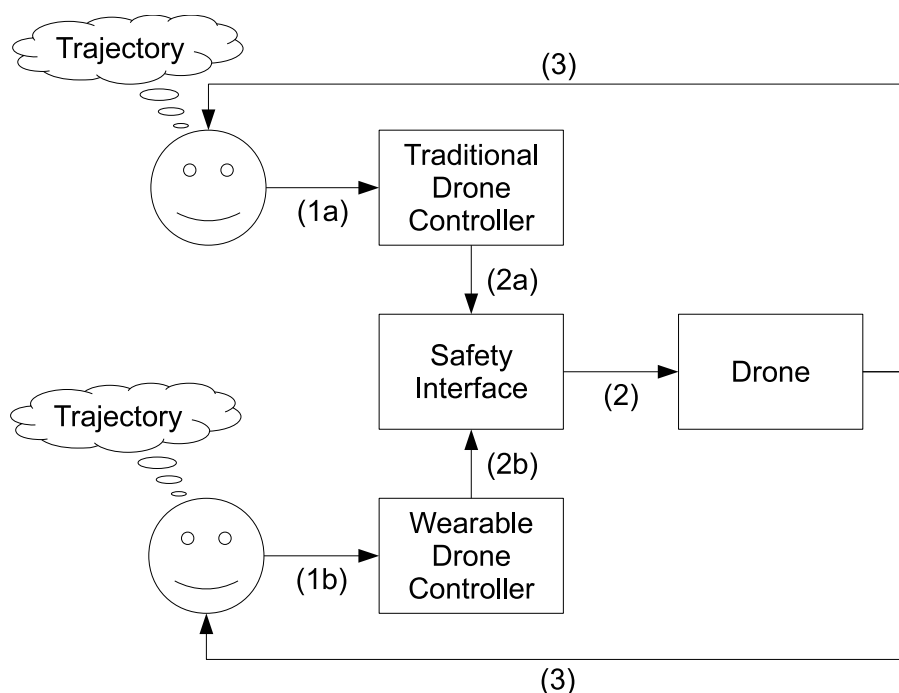


Figure 2.19 – Graphical representation of the control loop applied to fly a real drone.

### Experimental setup

For this experiment, the proposed wearable system was integrated in a soft exoskeleton called FlyJacket [153], as shown in Fig. 2.20. Participants wore the FlyJacket design with arm supports and a smart glove capable of detecting predefined finger gestures through capacitive sensors placed on each finger and the palm. The commands for the drone were computed with the proposed wearable system, sent to the safety interface (implemented on a PC), and further transmitted to the drone as detailed in [32].



## 2.4. Experimental Validation



Figure 2.20 – Setup used for flying a real drone with FlyJacket.

The gloves were used to set points of interest by pressing the middle, or the ring finger, against the thumb depending on the desired nature of the point to set. According to the regulation of the Swiss Firefighters, different colors can be used to describe the nature of the point of interest; that is, yellow to indicate rescue situations, red for fire, blue for both water sources and damages, and green for escape ways. As shown in Fig. 2.21, the points of interest were displayed on a map, which can be used to facilitate the planning of a SAR mission. The glove was also used to send high-level commands to the drone; namely, automatic takeoff, landing, and return home.



Figure 2.21 – Map showing the covered flight trajectory of the real drone and the identified points of interest (i.e., red and green circles).

The flight was performed with a Bebop 2, a quadcopter from Parrot [131], mimick-

## Chapter 2. Wearable Drone Controller

ing the flight dynamic of a fixed-wing drone. The drone streamed real-time video feedback of the fly to the goggles that participants were wearing. The trajectory of the drone was displayed on a computer, as shown in Fig. 2.21. Red and green dots are points of interest that a participant set during the flight. The point of interest also appeared in the center of the field of view of the drone (white cross in Fig. 2.22). The numbers above the recorded points indicate the estimated distance between the point and the drone in meters. All points of interest can be directly added and removed during the flight from both the map and the field of view displayed on the HMD.

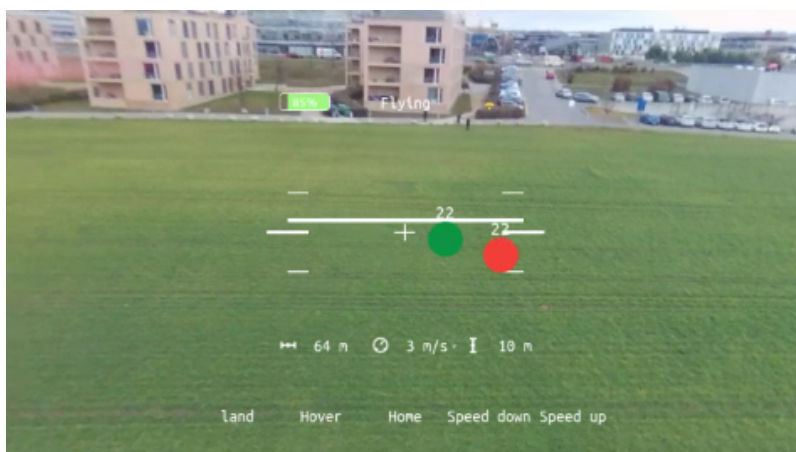


Figure 2.22 – View from the camera of the drone. The same view is shown to the pilot through the Head-Mounted Display (HMD).

### Study protocol

The experimental scenario was designed to reproduce a simulated and simplified SAR mission. In this context, participants were asked to take off and follow a figure-eight flight trajectory. Moreover, we asked to randomly geotag some points of interest, which could represent, for instance, injured people or dangerous areas.

## 2.5 Experimental Results

The three main results reported in this section are the following. First, I reported the analysis of the postures assumed by the participants while interacting with the drone simulator. Second, I reported a comparison between Flyjacket, the proposed new-

generation controller based on IMUs integrated in an upper-body soft exoskeleton, and an optical camera system for motion capture from Vicon. Finally, I evaluated the flight of a real drone controlled with the FlyJacket design, providing an estimation of the reduction of the energy consumption achieved with the method applied for optimizing the communication rate.

### 2.5.1 Body Posture Evaluation

A subjective evaluation provided by the research participants revealed that the body posture (i.e., sitting, standing or lying face down) does not affect the control strategy, or the subjective levels of comfort and immersion (Figure 2.23) [108, 109].

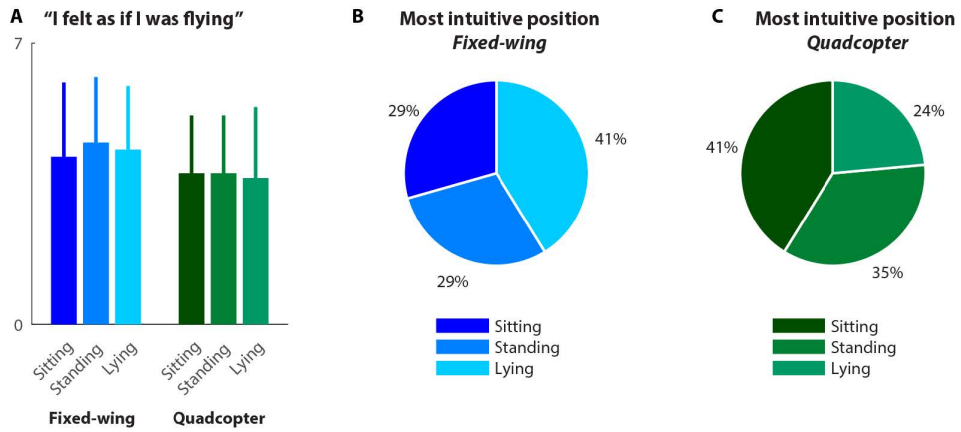


Figure 2.23 – Subjective evaluation of the control strategy (Figure from [109]). In particular, the impact of both the participant’s position and the flight style. Fig. A reports the sensation of flying on a 7-point scale (0: strongly disagree, 7: strongly agree). Fig. B shows the most intuitive body postures with the fixed-wing flight style. Fig. C shows the most intuitive body postures with the quadcopter flight style.

Although there is a little preference for the lying and the sitting positions to respectively control a fixed-wing drone and a quadcopter, there is no statistically significant difference revealing that the aircraft (simulated fixed-wing drone or quadcopter), or the participant’s position (sitting, standing or lying face down) is affecting the control strategy or the perceived levels of comfort and immersion [108, 109].

This result is confirmed by the physiological response of the participants shown in Figure 2.24. The only significant difference is observed in the RR-intervals extracted from ECG, which can be attributed to the physical effort made by the participants.

## Chapter 2. Wearable Drone Controller

In fact, it is well known that standing is physically more demanding than sitting. Therefore, the RR-interval is shorter while standing than sitting. Following the same line of argument, while lying the RR-interval should be higher than sitting and standing. However, to control a drone while lying, it implies to lift the torso from time to time, which is an activity that can be physically very demanding. Consequently, while lying, we noticed a short RR-interval, which is significantly shorter ( $p < 0.001$ ) than RR-intervals measured while sitting and standing (see Figure 2.24).

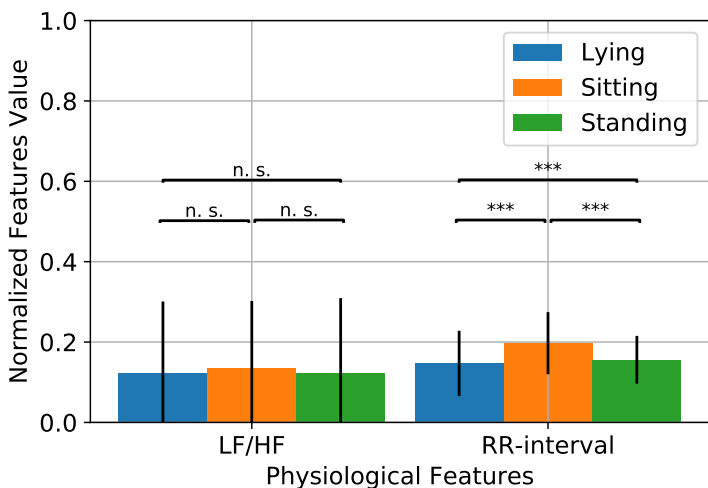


Figure 2.24 – Physiological response of the research participants. The error bars show the standard deviation, the three asterisks indicate a significant difference with  $p < 0.001$ , and n.s. means not significant.

Finally, to facilitate the translation towards the control of a real drone, we selected the sitting position, which requires only light equipment, yet is safer for the operator than standing upright. Moreover, as physical activities might affect cognitive measures, by minimizing the physical activity, we can reduce possible incertitude related to the cognitive workload monitoring.

Participants also reported a preference for a fixed-wing drone rather than a quadcopter. However, the young research participants (mostly students) mainly ignore the real needs of search and rescue missions. Therefore, this results should be considered with caution and as a general indication only.

### 2.5.2 FlyJacket performance vs. Vicon

As an evaluation of FlyJacket's performance, Figure 2.25 shows the absolute values of the difference between the angles acquired with FlyJacket and those measured with Vicon. The statistical relationship, or association, between the two systems, was evaluated with Pearson's correlation coefficients, which are 0.86, 0.96, and 0.90, for Lateral Bending Angle, Sagittal Bending Angle, and Rotational Angle, respectively. Moreover, I reported the RMSE, the RMS of the difference between the angles acquired with FlyJacket and those acquired with Vicon. The values are: 3.81, 2.24, and 6.50 degrees, for Lateral Bending Angle, Sagittal Bending Angle, and Rotational Angle, respectively.

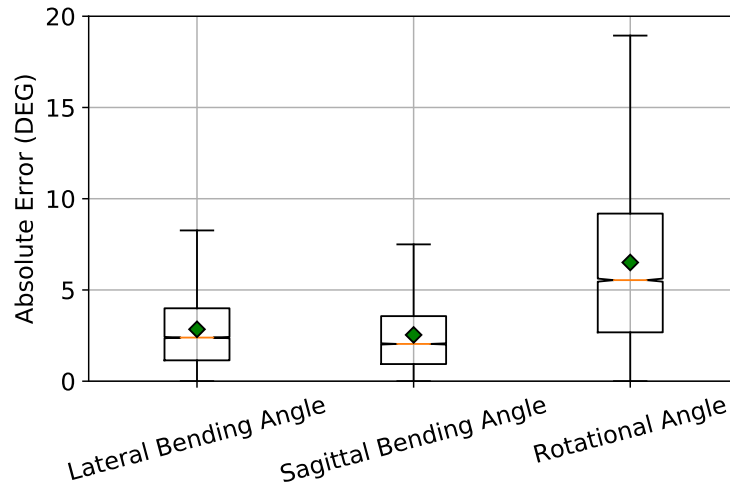


Figure 2.25 – FlyJacket vs. Vicon.

The difference between the systems comes from different aspects. An error is introduced by the implemented sensor-fusion algorithm, which can cause an RMSE up to 7° at 10 Hz [99]. Moreover, movement artifacts may also contribute to increase this error. To have a reliable reference, the upper-body trunk orientation was measured with adhesive reflective markers attached to both the sternum and the spinal column [108]. Therefore, neglecting the tiny skin layer that covers the bones in those locations, we can say that the reference was basically at the skeleton level of the participant. Instead, the measurements acquired with the proposed wearable sensor node were more on the surface level, as the node was attached to a belt and fixed around the chest of the participants. With such a light FlyJacket design, we noticed

that some movements, for instance of the arms, slightly affected the orientation of the node.

A parallel study showed that both intuitiveness and performance were more consistent when using the proposed system than when performing the same task with a traditional controller [153]. Therefore, we can conclude that although such a difference may be relevant in other fields, such as in rehabilitation, it does not seem to play a crucial role while controlling a drone. Moreover, this difference does not seem to be perceived by the user, but this conclusion has to be taken with a grain of salt, as it comes from informal observations made during public events and demos.

2.5.3 FlyJacket used to fly a real drone

The FlyJacket design has been tested for the teleoperation of a real drone. During this test, the supervisor took control of the drone only a few times during the training but let the user fly through the figure-eight trajectory while performing the test. Figure 2.26 shows the difference between the FlyJacket commands and the corresponding response of the drone orientation for both pitch and roll. The Pearson's

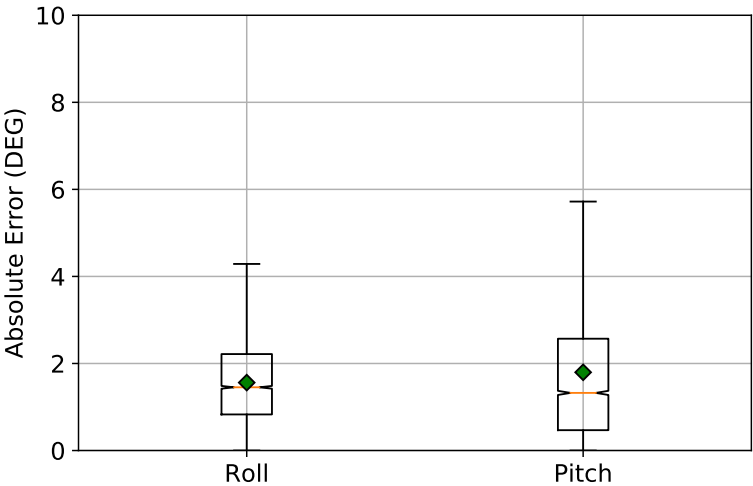


Figure 2.26 – Teleoperation of a real drone. Difference between the FlyJacket commands and the corresponding response of the drone orientation.

correlation coefficient between the commands provided by FlyJacket and the drone attitude is greater than 0.88. In particular, 0.93, and 0.88, for pitch and roll angles,

respectively. The RMS of the difference between the angles computed with FlyJacket and those assumed by the drone are 2.47 and 1.85 degrees, for pitch and roll angles, respectively. Such a difference can result from the method proposed for optimizing the communication, but it can also come from external disturbances such as wind. Indeed, wind can easily affect the orientation of the drone, which can cause part of the difference reported in Figure 2.26. Other aspects to consider are the dynamic of the drone and its onboard control-loop, which tries to maintain the stability of the drone and to follow the given reference. Because of its dynamics, the drone cannot instantaneously follow the desired trajectory. Therefore, we can expect to have a minimal transition time, in which both the desired and the assumed orientations do not necessarily match.

These results show that the drone can follow the commands provided by the pilot while using FlyJacket. Moreover, during the experiment, all participants were able to follow the figure-eight trajectory as requested without any particular problem and without crashing the drone. Therefore, we can conclude that the FlyJacket design is capable of controlling a simulator (based on a formal evaluation with 17 volunteers presented in section 2.4.2, but also based on our informal observations collected during various demonstrations, such as, NCCR-Robotics annual retreats and industry days, EPFL open days, etc.). Moreover, three short experiments of 10 minutes each (described in section 2.4.3) demonstrated the feasibility of controlling a real drone.

### 2.5.4 Power Saving by Reducing the Communication Rate

The method applied to reduce the communication rate is analysed in this subsection. To this aim, Figure 2.27 shows the trade-off between the transmission (Tx) rate (or transmission reduction) and the RMSE; that is, the difference between the resampled version of the transmitted (sparse) sequence of commands and the original sequence. This RMSE is used to quantify the quality of the transmitted sequence and it can also be seen as some kind of quantization error that increases or decreases as a function of the selected threshold. As explained in Section 2.3.4, this threshold is the maximum RMS difference between current and previous commands, and it defines when to send or not a command. From Figure 2.27 we can see that a high threshold yields a reduction of the communication rate, but on the other hand, this reduction is paid for by a deterioration of the quality of the transmitted signal.

The data used to plot Figure 2.27 was obtained from the recordings of a flight without applying any threshold. Then, I computed both RMSE and Tx Rate that would result

by selecting different thresholds. Finally, this plot gives an estimation of the effects on the quality of the transmitted sequence and the possible transmission reduction that can derive.

The choice of this threshold depends on the limitations imposed by the application, which defines how much the communication can be reduced and how big is the error that can be introduced. For this particular experiment, I arbitrarily selected a threshold equal to 1. This choice resulted to be a good compromise, as the communication was reduced by 50% and the RMSE was less than  $1^\circ$ , which was not perceived by the participants. However, it would be interesting to push this threshold up to the limit when a human starts to perceive the error introduced. Thus, to finalize the optimization of this communication reduction, new experiments should be considered.

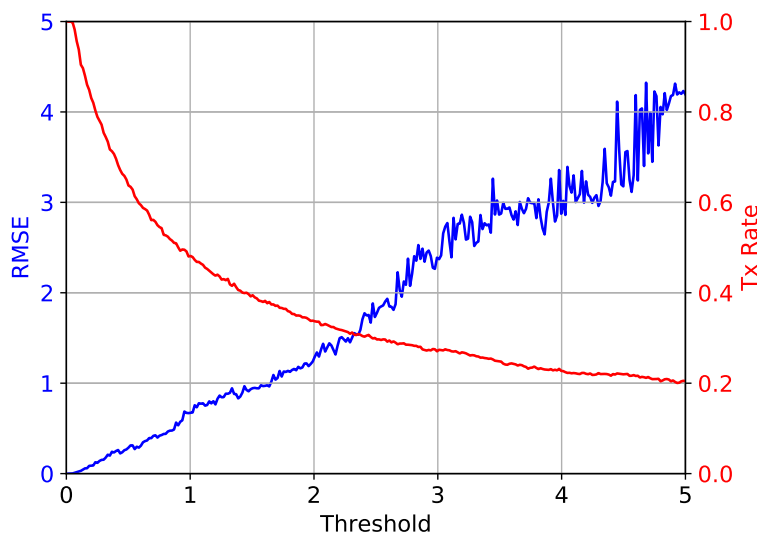


Figure 2.27 – Evaluation of both the transmission (Tx) rate and the introduced Root Mean Square Error (RMSE) as a function of different thresholds.

To quantify the communication reduction in terms of current consumption, we can refer to the measurements reported in Table 2.2. This table shows the average values of both current consumption ( $i$ ) and time ( $t$ ) of the proposed sensor node being in a particular state, namely, processing, sending, and sleeping. By choosing a threshold equal to 1, the transmission rate is reduced by 50% and the total current consumption of the node is reduced from 19.36 mA to 18.52 mA. Consequently, this



choice yields a 4.34% reduction of the energy consumption of the proposed wearable sensor node. Moreover, by turning off the BLE module while not in use, the average current consumption can be further reduced by 1.47 mA, which results in a total reduction of the energy consumption of 11.9%. Of course, as the minimal connection interval of the BLE module is 1.25 ms, the BLE module can be turned off even if the Algorithm 1 is not applied, but in this case the energy consumption is only reduced by 7.4%.

Tx Rate	Processing		Sending		Sleeping		Total
	$i_P$ (mA)	$t_P$ (ms)	$i_{Tx}$ (mA)	$t_{Tx}$ (ms)	$i_S$ (mA)	$t_S$ (ms)	$i_{Tot}$ (mA)
100%	25.7	1.68	38.4	4	17.4	44.3	19.36
50%	25.7	1.74	38.4	2	17.4	46.3	18.52
0%	25.7	1.74	-	-	17.4	48.3	17.68

Table 2.2 – Average current consumption ( $i$ ) and run time ( $t$ ) of the proposed sensor node; while being in a particular state, namely, processing, sending, and sleeping; and while sending the commands with an average transmission (Tx) rate of 100, 50, and 0%.

The 50% communication reduction is a result of the particular case of a pilot flying through a figure-eight trajectory, which could be applied in a SAR mission. However, different reduction factors could be obtained during a straight or aerobatic flights, where the reduction could be respectively more or less important. Moreover, these results are limited to the analysis of the proposed wearable sensor node while acquiring, processing, and sending data via BLE to a third device (e.g., PC, tablet, smartphone, or customized embedded system), which is constantly located in a range of one or two meters. Therefore, the variation of the energy consumption due to different ranges of operation is not considered in this work, as well as the energy consumption of the third device used to forward the commands to the drone.

## 2.6 Conclusion

In this chapter, I have proposed the design of a wearable embedded system for drone teleoperation in SAR missions. The control strategy is based on upper-body movements derived from a spontaneous representation of the interaction. Therefore, as demonstrated in [108, 153], the proposed wearable and intuitive interface can reduce the training time required to reach proficiency and improve the reliability of teleoperated activities. The proposed wearable embedded system has been developed and integrated into a soft exoskeleton, called FlyJacket. The system has

## Chapter 2. Wearable Drone Controller

---

been interfaced and demonstrated for the teleoperation of both a simulated and a real drone, during experiments and events, such as Cybathlon and EPFL Open Days. The use of FlyJacket leaves both arms and hands-free allowing the execution of complementary tasks, such as geotagging points of interests or sending high-level commands to the drone.

The accuracy of the motion capture has been compared with an optical camera system from Vicon, which is the leading developer of motion capture products and services for the life sciences, entertainment, virtual reality, and engineering industries. The results showed that the proposed wearable solution can track the movements required to drive a drone. Of course, the tracking accuracy can be improved, but still, a parallel study showed that participants' performance was more consistent when using FlyJacket than when performing the same task with a traditional controller [153].

Moreover, I proposed a method that reduces the communication rate and consequently also the energy consumption of the system. Compared to a typical system that is continuously streaming the commands, the proposed approach can reduce the transmission volume of the system up to 50%, which does not seem to affect the perception of the user. Such a drastic reduction of the transmission volume yields a reduction of the energy consumption, which is up to 11.9% for the proposed wearable drone controller, called FlyJacket. Being this conclusion drawn from only 30 minutes of fly, through a figure-eight trajectory, and based on a BLE communication unit, further analysis is needed to provide a more formal evaluation of the final gain and when the reduction of communication starts to be perceived by the user. Of course, the energy-saving really depends on the flying trajectory, a straight fly allows a higher energy saving, while an acrobatic flight will not allow saving much energy. Indeed, the energy-saving will be amplified while using a proper communication protocol, where the energy required to send commands to a drone is typically higher than the one required from BLE to send data to a nearby computer. Therefore, the results are a bit conservative, but they demonstrate that it is possible to save energy.

Another way to save energy would be to make sleep mode more efficient. As reported in table 2.2, the system is using 17.4 mA while doing nothing, and this is too much. One of the reasons for this high current consumption is that in some prototypes, I found resistors erroneously placed instead of capacitors, and this causes an undesired increase in the current consumption. This erroneous placement has been encountered a second time, where one prototype mounted the MCU rotated by 90

degrees. Although I carefully checked afterward, there could be a little component that still causes an undesired current consumption. Other sources of elevated current consumption are the linear regulators used for providing the different power supplies as well as the passive filters added to isolate the analogical part from the digital. Therefore, a redesign of these circuits could easily reduce energy consumption.

Although it has been shown that the proposed system can be used for the teleoperation of a real drone, this solution presents a limitation. The proposed wearable drone controller only controls two of the six DoF of a drone (i.e., roll and pitch). Other DoF, such as longitudinal (forward and backward) and vertical (upward and downward) movements, were controlled with a glove. Lateral (right and left) movements, as well as the rotation among the vertical axis (yaw movement), were not addressed in this work. Therefore, a further investigation, considering the use of additional sensors to track the movements of hands, arms, and head, is needed to include the mapping of the missing DoF. Such a study will provide a comfortable and complete solution that could be used in SAR missions with drones, and its use could be easily extended to different fields, machines, and populations, including leisure purposes.



## 3 Cognitive Workload Detection Method

### 3.1 Introduction

In search and rescue missions, rescuers have to simultaneously focus on multiple tasks. More precisely, on top of the already challenging task of flying a drone, rescuers have to recognize the situation and quickly take proper decisions to rescue victims that are frequently in danger of life. Moreover, rescuers have to deal with both scarcity of human resources and time pressure. Therefore, rescuers could potentially have to control multiple drones to simultaneously execute complementary or parallel tasks, or to cover big areas in less time.

The problem is that any multi-tasking operation is cognitively very demanding and high levels of cognitive workload can negatively affect human performance [103, 115, 175]. Consequently, operating under high cognitive workload levels can severely compromise the execution of the mission and lead to failure with catastrophic outcomes [194]. Therefore, to ensure an efficient execution of the missions, there is a need for an online tracking system that provides the necessary information about the cognitive workload of each of the rescuers (i.e., subject-specific cognitive workload monitoring). In other words, by detecting the cognitive overload of each rescuer in a subject-specific manner, actions (e.g., replacing the rescuer that is overloaded) could be taken to better distribute human resources and ensure the success of the mission.

To assess cognitive workload, researchers typically use surveys [75, 148], performance metrics to evaluate the execution of the task [38, 100], and information from physiological signals [23]. However, neither surveys nor performance metrics are suitable in search and rescue missions. Surveys only provide subjective and sporadic measure-

### **Chapter 3. Cognitive Workload Detection Method**

---

ments, and are not always reliable [2]. Moreover, rescuers cannot stop their mission to fill a questionnaire. Also, performance metrics have to be defined in advance, but since every rescue mission is unique, reliable performance metrics are difficult to set up. On the other hand, physiological signals can be acquired in a noninvasive way without disturbing the work of the rescuers. Therefore, the use of physiological signals seems to be the most promising solution to assess continuous monitoring of cognitive workload [23, 77, 146].

Several studies combine physiological signals with different machine learning algorithms for cognitive workload monitoring in different fields [21, 77]. However, to the best of my knowledge, I am the first to address the cognitive workload monitoring of pilots involved in search and rescue missions with drones.

#### **Contributions of this Chapter**

Non-invasive online cognitive workload monitoring from physiological signals in search and rescue missions with drones is the focus of the work in this chapter, which proposes the following contributions:

- To induce different levels of cognitive workload related to search and rescue missions, I modified a virtual-reality based drone simulator to include parallel tasks, such as objects recognition. The simulator was used for the acquisition of different physiological signals acquired from 34 participants involved in a cognitively demanding simulated, but immersive Search and Rescue (SAR) mission, 24 using a traditional controller and 10 using the proposed wearable controller, called FlyJacket.
- To characterize the physiological responses of cognitive workload, I performed an exhaustive investigation of relevant features extracted from physiological signals and I selected the most representative ones.
- To reduce both inter-subject and inter-day variability, I explored different feature normalization techniques showing that a normalization that considers both subject and day improves the classification results.
- To further reduce the inter-subject variability, I provided a new learning method based on Support Vector Machines (SVMs) suitable for a subject-specific optimization. This SVM-based method uses two regularization terms, one for

learning the general behaviour, and another one for tuning the model to fit the characteristics of a particular data subset.

- Finally, I proved the ability of my method to detect low and high levels of cognitive workload with both traditional controllers and new advanced controllers, such as the new FlyJacket design, achieving an accuracy of 87.3% and 91.2%, respectively. These results are obtained from a test set acquired from 34 subjects while flying a drone simulator and mapping a graphic representation of a damaged situation. My results are better than the latest state-of-the-art studies.

## Publications

This work yielded the following publications:

- F. Dell'Agnola, P.-K. Jao, A. Arza, R. Chavarriaga, J. d. R. Millán, D. Floreano, and D. Atienza. **Machine Learning Based Monitoring of Cognitive Workload in Rescue Missions with Drones.** *IEEE Transaction on Affective Computing*, In preparation.
- G. Masinelli, A. Arza, F. Dell'Agnola, and D. Atienza. **SPARE, SPectral peAk REcovery: full PPG pulsewave reconstruction.** *IEEE Journal of Biomedical and Health Informatics*, In preparation.
- P.-K. Jao, R. Chavarriaga, F. Dell'Agnola, A. Arza, D. Atienza, and J. d. R. Millán. **EEG Correlates of Difficulty Levels in Dynamical Transitions of Simulated Flying and Mapping Tasks.** *IEEE Transactions on Human-Machine Systems*, Accepted on October 2020.
- F. Dell'Agnola, N. Momeni, A. Arza, and D. Atienza. **Cognitive workload monitoring in virtual reality based rescue missions with drones.** *22nd International Conference on Human-Computer Interaction (HCII)*, Copenhagen, Denmark, 2020.
- N. Momeni, F. Dell'Agnola, A. Arza, and D. Atienza. **Real-Time Cognitive Workload Monitoring Based on Machine Learning Using Physiological Signals in Rescue Missions.** *41st International Engineering in Medicine and Biology Conference (EMBC)*, Berlin, Germany, 2019.

- F. Dell'Agnola, L. Cammoun, and D. Atienza. **Physiological Characterization of Need for Assistance in Rescue Missions with Drones.** *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, 2018.

### Chapter Outline

The rest of the chapter is organized as follows. Section 3.2 gives an overview of the related work in the field of cognitive workload monitoring from physiological signals. Section 3.3 describes my design approach. Section 3.4 details the filtering stage used to remove the noise from the signals. Section 3.5 describes the feature selection method applied for the characterization of cognitive workload from physiological signals. Section 3.6 introduces a classification method suitable for embedded systems. Section 3.7 describes the setup of the experiment and Section 3.8 reports my results. Finally, Section 3.9 presents the main conclusions of this chapter.

### 3.2 Cognitive Workload Monitoring: State of the Art

Cognitive workload characterization and estimation have been addressed by a large number of studies in different areas, which aim to characterize either the performance or the distress of a person involved in a particular task or situation [45, 86, 146]. In this section, I reviewed the state-of-the-art studies that apply machine learning techniques to detect cognitive workload induced by high cognitive tasks. In particular, I analyzed in detail the works that use unobtrusively measured physiological signals. Although Electroencephalogram (EEG) is one of the most used signals [15, 146], it is not considered in my analysis because of the intrusiveness of the electrodes in real-life applications [110].

Table 3.1 summarizes the most recent and significant studies. This table includes the performed task (used to induce different levels of cognitive workload), the measured physiological signals, the type of signal segmentation (i.e., window length and overlap), the applied machine learning methods, the targeted classes, and the results of the classification (i.e., Accuracy, Sensitivity, and Specificity).



Table 3.1 – Summary of the state-of-the-art studies addressing cognitive workload monitoring, including performed task and measured physiological signals, such as Respiratory Activity (RSP), Electrocardiogram (ECG), Photoplethysmogram (PPG), Skin Temperature (SKT), and Electrodermal Activity (EDA). Moreover, the table includes type of signal segmentation, applied machine learning methods, such as Extreme Gradient Boosting (XGB), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), Linear Discriminant Analysis (LDA), and Artificial Neural Network (ANN). Finally, the table also includes the number of targeted classes and classification results (i.e., Accuracy, Sensitivity, and Specificity).

Study	Performed Tasks	Physiological Signals	Window Length (Overlap)	Classifier (Classes)	Results Acc. Sens. Spec.
Momeni et al. [113]	Simulated SAR missions with drones	ECG, RSP, PPG, and SKT	60s (30s)	XGB (2)	86%* – –
Montesinos et al. [114]	Arithmetic tasks	ECG, PPG, RSP, SKT, and EDA	60s (30s)	RF (2)	84.13%* – –
Chen et al. [30]	Real car driving	ECG, RSP, and EDA	100s (90s)	SVM (3)	89.7% 88.5% 94.2%
Solovey et al. [161]	Driving in highway	ECG and EDA	30s (0s)	LR (2)	90% – –
Giakoumis et al. [62]	Video-game	ECG and EDA	25s (0s)	LDA (2)	94.96% 94.96% 94.96%
Tjolleng et al. [187]	Simulated car driving	ECG	100s (0s)	ANN (3)	82% 78% 91%
Gjoreski et al. [63]	Daily life activities	PPG, SKT, and EDA	300s (150s)	SVM (2)	98.96% 70.44% 99.88%

\* Results based on an unseen test set, all the other are limited to cross-validation.

### Chapter 3. Cognitive Workload Detection Method

---

My analysis identified, for most of the studies presented in the literature, the following common methodological steps: signal acquisition and preprocessing (filtering and segmentation), feature extraction, feature normalization, dimension reduction or feature selection, and classification or regression. However, although the methodology is well established, discrepancies are found in different steps. Hence, in the following, I reviewed these discrepancies.

First, significant differences have been observed on the physiological measures, which are Electrodermal Activity (EDA) [30, 62, 114, 161], Electrocardiogram (ECG) [30, 62, 113, 114, 161, 187], Photoplethysmogram (PPG) [113, 114], Respiratory Activity (RSP) [30, 113, 114], and peripheral Skin Temperature (SKT) [113, 114]. Although it has been shown that the use of multiple physiological signals can increase the detection accuracy [113], the type and number of signals, and in particular the features used to assess cognitive workload, often differ and strictly depend on the case study (e.g., the type of task used to induce different levels of cognitive workload) [23, 78, 98]. Thus, there is no clear definition of the best selection of signals and features to be used to assess cognitive workload in general.

Then, the applied segmentation used to extract the features from the signals also depends on the case study. In particular, the window lengths reported in Table 3.1 vary in a range from 25 to 300 seconds. Moreover, different window overlaps are applied either to increase the size of the dataset [113, 114] or to provide more frequent estimations in time [30, 63]. These differences can be explained by the fact that physiological methods do not provide a direct measurement of the workload, but rather they give information about how the individuals themselves respond to a particular load [23]. Therefore, a different signal segmentation may be applied depending on the dynamics of the physiological response induced by a particular cognitive workload.

An additional aspect that I observed in the literature review was that features are often normalized to standardize their ranges. Moreover, as suggested in [124], this normalization should be considered to reduce intra- and inter-subject variability caused by age, gender, time of day and other factors. However, not all studies reported whether or not a normalization was applied [124]. Moreover, it was not always clear how the normalization was done, and if it was applied to make any distinction between training and test sets. The point is that to properly emulate and test the behaviour of the system, data from the test set should be normalized based on the parameters obtained from the training set [124].

### 3.2. Cognitive Workload Monitoring: State of the Art

---

Moreover, other differences clearly exist in the choice of machine learning methods. The size of the training data and the specification of the system requirements (e.g., computational complexity, power and latency) may explain the different selections of machine learning algorithms. In fact, as most of the studies typically start with a limited amount of data, simple models like SVM [30, 63, 74], Linear Discriminant Analysis (LDA) [62, 76], Logistic Regression (LR) [161], and Decision Tree (DT) [114], are the most used machine learning techniques. In contrast, complex models such as Artificial Neural Network (ANN) [187], Random Forest (RF) [74, 114], and very recent models like Extreme Gradient Boosting (XGB) [113], have been less used so far. In any case, even if SVM has been the most used classifier in this field, there is no consistent indication whether it is the best model or not for different case studies.

Finally, my analysis shows that the highest levels of accuracy that have been obtained are in the range from 82 to 99%. This wide range is mainly due to the different experimental protocols, methodologies, and number of considered classes in each study. Moreover, the very high accuracy reported by different studies may be affected by overfitting since the evaluation of the proposed model is limited to the cross-validation stage [30, 62, 63, 161, 187]. In particular, only a few studies in this field evaluated the model in cross-validation as well as in an additional unseen test set (i.e., a set that has never been used in training) [113, 114], even though a proper estimation of the generalization power of a proposed model requires a final test on new observations.

In conclusion, there is a need to further investigate the contribution of each physiological signal, the impact of data normalization, and the performance of the selected classifier on unseen data in the context of rescue mission with drones, which are not appropriately covered in the literature.

Furthermore, it has to be considered that the workload is multidimensional [75] and the results from the aggregation of three broad aspects [23, 78, 98]. First, the workload depends on the type of task (mental or physical demand), as well as the amount of work and number of tasks to perform (load level). Second, it is affected by time, in particular by the duration of the interval of the temporal demand. Third, the subjective psychological experiences modulate the level of workload perceived by a subject (i.e., subject's capabilities, learning skills, and effort). Therefore, it is key to investigate cognitive workload in the particular field of interest, and subsequently consider the subjective workload level perceived by each person, as suggested in [27].

### 3.3 Cognitive Workload Detection Method

In this section, I described the general design of a machine learning algorithm suitable for the development of a wearable embedded system for online cognitive workload monitoring (Fig. 3.1, blocks with solid line). To design such a system, I applied different statistical pattern recognition methods (Fig. 3.1, blocks with dashed line) based on experimental data. The analysis was done off-line, but the final system was tested by emulating online processing (i.e. using causal spectral filters and computing the features only taking into account past information).

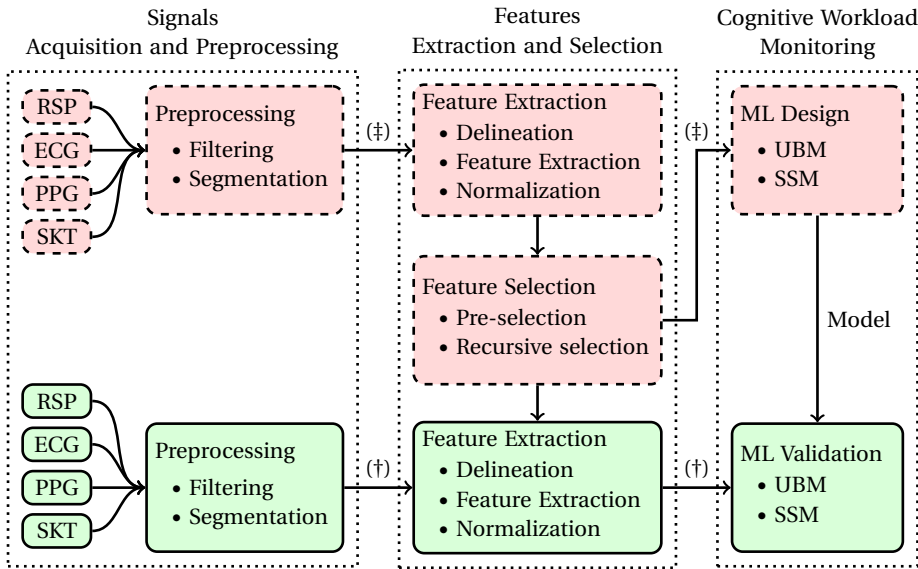


Figure 3.1 – Overview of the process for the design of a cognitive workload monitoring method. Blocs with dashed line represent the applied design/optimization methods based on training data (‡), while blocs with solid lines represent the final system evaluated with testing data (†). The method is based on multiple physiological signals, such as Respiratory Activity (RSP), Electrocardiogram (ECG), Photoplethysmogram (PPG), and peripheral Skin Temperature (SKT). In particular, the method is applied to select the features that better characterize the cognitive workload and can be used by a Machine-Learning (ML) algorithm, such as an Universal Background Model (UBM) or a Subject-Specific Model (SSM).

The system is divided into three main steps (Fig. 3.1, blocks with dotted line):

- Signals Acquisition and Preprocessing,
- Features Extraction and Selection,

- Cognitive Workload Monitoring.

The first step of the cognitive workload monitoring system is the signal acquisition. In this work, I collected experimental data for both design and evaluation of the proposed cognitive workload monitoring method. Then, once the data was collected, a preprocessing step was applied to remove artifacts from the signals. Moreover, a sliding window was applied for signal segmentation, which defined the time resolution of the workload monitoring system.

The features extraction phase is the following step, which included the generation of a feature vector that best represented the physiological response induced by different workloads. For an exhaustive investigation, I chose an exploratory approach in which I extracted a large number of different features in both time and frequency domains. Then, since physiological signals exhibited high intra- and inter-subject variability as a result of age, gender, time of day and other factors [124], I investigated different normalization methods. Next, I applied different features selection methods, which allowed us to define the best subset of features that should be used in the final system.

Finally, the cognitive workload monitoring step includes the prediction of a discrete cognitive workload level. For the design of the cognitive workload monitoring method, I considered the most common machine learning techniques based on pattern recognition algorithms suitable for implementation in embedded systems. Moreover, I considered a personalized learning approach to consider the highly person-dependent variance in the physiological response of an induced workload.

The performance of my method is then evaluated with the NASA Task Load Index (NASA-TLX), which is a subjective and multidimensional assessment tool that rates perceived workload [75].

## 3.4 Signals Acquisition and Preprocessing

For a thorough exploration of the physiological changes induced by cognitive workload, I measured RSP, ECG, PPG, SKT, and EDA, which could be measured with wearable units. These signals were the ones that are typically used in the literature [34, 123]. Their main physiological manifestations related to cognitive workloads were reported in Table 3.2 and described in Sec. 3.4.1.

### Chapter 3. Cognitive Workload Detection Method

Table 3.2 – Measurable physiological manifestations related to induced cognitive workloads and affected by the Sympathetic Nervous System (SNS), the Parasympathetic Nervous System (PSNS), and Hypotalai-Adrena (HPA) axis.

Signals	Physiological manifestation to workload response	Sensor location
RSP	SNS activation and PSNS counterbalance	Thorax
ECG	Both HPA axis and SNS activation, and PSNS counterbalance	Thorax
PPG	Neurohypophysis, HPA axis, and SNS activation, and PSNS counterbalance	Ear
SKT	Neurohypophysis and SNS activation	Finger

#### 3.4.1 Physiological Process behind Cognitive Workload

While performing a very demanding task, one of the effects driven by the Autonomic Nervous System (ANS) activation that involves both a Sympathetic Nervous System (SNS) activation and a Parasympathetic Nervous System (PSNS) counterbalance, is the need for more oxygen. This increased oxygen demand triggers faster and deeper respiration [10]. Therefore, RSP should be measured to track changes in cognitive workloads [30].

Another effect driven by ANS activation is the cardiac response, which is affected by the Hypotalai-Adrena (HPA) axis as well. This response is associated with variabilities in heart rate, defined as Heart Rate Variability (HRV), which can be obtained by monitoring the ECG signal. Consequently, the above relationship can explain the heart's ability to respond to multiple physiological and environmental stimuli [38].

The activation of the neurohypophysis, the HPA axis, and the ANS leads to changes in blood volume, peripheral blood vessels resistance, and cardiac response, which can be derived from the pulse wave. Therefore, features from the PPG are needed to detect those physiological changes induced by cognitive tasks [10, 63].

Moreover, it has been proved that a cognitive task causes peripheral vasoconstriction [10, 63], which are regulated by the vasoregulatory system and driven by both neurohypophysis and SNS. Therefore, SKT is needed to detect the variations in peripheral temperature that are associated with peripheral vasoconstriction.

#### 3.4.2 Signals Preprocessing

The first preprocessing step consists of removing the artifacts from the different signals with the following causal filters. To both ECG and PPG signals, I applied a baseline wander with cut-off frequency at 0.3 Hz [106]. Next, I applied as well a 32nd-order bandpass Finite Impulse Response (FIR) filter with a linear phase and Hamming window with cutoff frequencies at 0.3 and 30 Hz for ECG and cutoff frequencies at 0.1 and 5 Hz for PPG [10]. Moreover, I applied a 4th-order Butterworth Infinite Impulse Response (IIR) bandpass filter with cutoff frequencies at 0.03 and 0.9 Hz to the RSP signal. Nevertheless, because of the slow response time of the SKT thermistor (1.1 sec.), which avoided the high frequency noise to affect the signal, no filter is applied to the acquired SKT signal.

Finally, I applied a time-series segmentation of all the acquired physiological signals, which were thus divided into a sequence of samples in windows of 60 seconds.

### 3.5 Features Extraction and Selection

Following the methodology described in Section 3.3, I performed an offline investigation to select the features that have to be considered in the final system. To this aim, I first extracted from the segmented signals a complete set of features for an exhaustive assessment of the person's physiological response induced by cognitive workload. Then, I selected the best set of features that is rich in discriminatory information with respect to the physiological states induced by different levels of cognitive workload.

Finally, to emulate the online monitoring system, I limited the feature extraction step to the optimal set of features, which was then normalized and given as input to the developed machine learning algorithm for cognitive workload monitoring.

#### 3.5.1 Feature Extraction

For the design of the cognitive workload monitoring system, the feature extraction process included three main steps. First, I delineated the segmented signal to detect points of interest (e.g., signal onset, peak, offset, etc.). Second, I extracted physiological markers, which were a combination of different delineated points and provided information about the physiological state of the person (e.g., heart rate). Finally, I computed features in both time and frequency domains. For the time domain,

### Chapter 3. Cognitive Workload Detection Method

I used standard statistical features (i.e., mean, median, mode, standard deviation, variance, root mean square, and power), extracted either from the physiological markers or from the segmented signals directly. However, in the frequency domain, the features were computed specific to the characteristic of the physiology. Fig. 3.2 shows a schematic representation of the signal processing and feature extraction process.

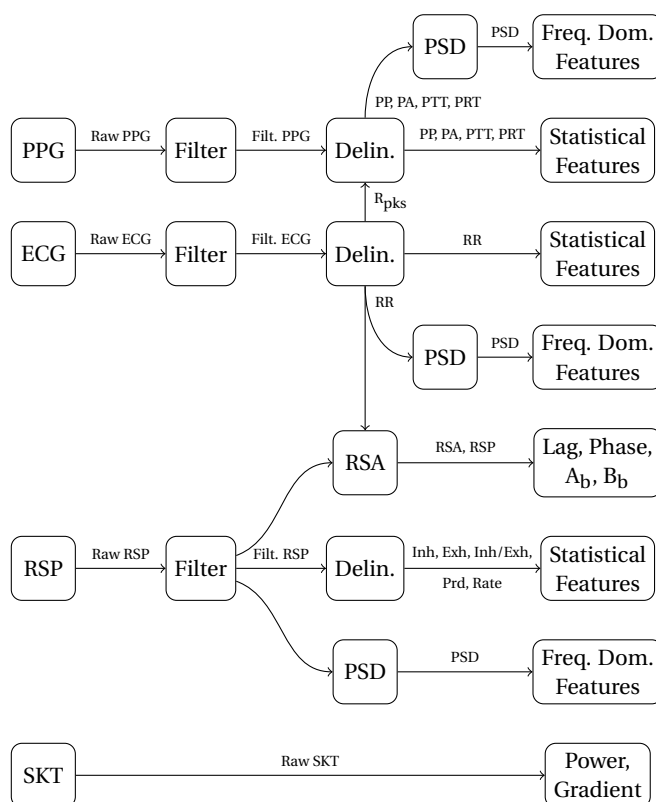


Figure 3.2 – Schematic representation of the signal processing and feature extraction processes. In particular the filtering and delineation of Photoplethysmogram (PPG), Electrocardiogram (ECG), Respiratory Activity (RSP), and Skin Temperature (SKT), to extract parameters, such as RR-interval, PP-interval, Pulse Amplitude (PA), Pulse Transit Time (PTT), Pulse Rising Time (PRT), Power Spectral Density (PSD), Respiratory Sinus Arrhythmia (RSA), RSP Period (Prd), RSP Rate, Inhalation (Inh) and Exhalation (Exh) time, etc. All these parameters are used to compute either statistical or frequency domain features.

Following an extensive literature review and by applying the experience from pre-



vious projects [38], I increased the number of both time and frequency domain analytical methods applied to a single segment of a physiological signal. I concluded upon 384 features: 127 from RSP, 38 from ECG, 190 from PPG, 2 from SKT, and 27 from Respiratory Sinus Arrhythmia (RSA). However, thanks to the applied feature selection method, my final system only used 25 features, 10 from RSP, 2 from ECG, 10 from PPG, 2 from SKT, and 1 from RSA. These 25 features are listed in Table 3.4 (page 93). Finally, as EDA did not show any significant correlated response with the induced cognitive workload, the signal was discarded. More details about the delineation and feature extraction for each considered signal are provided next.

#### Respiratory activity (RSP)

To extract the features from the RSP signal, I first delineated the signal based on the differences between adjacent samples of the filtered signal defined as:

$$\Delta x[k] = x[k] - x[k - 1] \quad (3.1)$$

Then, by applying a threshold, I detected from the sign of  $\Delta x$  the falling and rising edge, which coincided with the end of the inhalation (RSP-peaks) and the end of the exhalation (RSP-valleys), respectively. Then, all the pairs of peaks and valleys having a difference smaller than 20% of the mean respiration amplitude were removed as suggested by [74], because they may be assigned to artifacts instead of real respiratory cycles.

Next, from the delineated RSP, I extracted the following physiological markers: both Inhalation (Inh) and Exhalation (Exh) time, the ratio between the two (Inh/Enh), inhalation and exhalation amplitudes, respiratory period ( $RSP_{Prd}$ ), and respiratory rate ( $RSP_{Rate}$ ). Moreover, I computed their numerical differences using Eq. 3.1. Finally, I calculated the statistical features of the segmented respiratory signal, of its difference given by Eq. 3.1, and of all the aforementioned RSP physiological markers.

In the frequency domain, I computed the power of the segmented signal in four different bands of equal bandwidth, namely 0-0.25, 0.25-0.5, 0.5-0.75, and 0.75-1 Hz. Additionally, I considered the normalized band power as well, which was obtained by dividing each of the aforementioned band powers by the total power in the 0-1 Hz band.

### Chapter 3. Cognitive Workload Detection Method

---

#### Electrocardiogram (ECG)

From the filtered ECG signal, I computed the so-called Normal-to-Normal (NN) intervals, which are the intervals between normal QRS complexes that were detected with the delineation method described in [130]. Then, I computed features in the time domain describing the HRV [174]. More precisely, I computed the statistical features of the successive NN-intervals and the interval differences of successive NN-intervals. I also computed the number of interval differences of successive NN-intervals greater than 50 ms (NN50) and the proportion derived by dividing NN50 by the total number of NN-intervals (pNN50) within the processing window.

Additionally, I obtained several geometrical features from the Poincaré (or Lorenz) plot indicating vagal and sympathetic functions [188]. In particular, I extracted the length of the transverse axis ( $T$ ), which was vertical to the line  $NN_k = NN_{k+1}$ ; the length of the longitudinal axis ( $L$ ), which was parallel with the line  $NN_k = NN_{k+1}$ ; the Cardiac Sympathetic Index (CSI), defined as  $L/T$ ; the modified CSI ( $L^2/T$ ); and the Cardiac Vagal Index (CVI), defined as  $\log_{10}(LT)$  [174, 188].

Moreover, I extracted HRV features from the frequency-domain, as proposed in [174]. In particular, I computed the power in two frequency bands, namely, Low Frequency (LF) component (frequency between 0.04 and 0.15 Hz) and High Frequency (HF) component (frequency between 0.15 and 0.4 Hz). LF and HF powers were obtained from the estimation of the Lomb-Scargle Power Spectral Density (PSD) of the NN-intervals [143]. The power values were divided by the total power minus the Very Low Frequency (VLF) component (frequency  $\leq 0.04$  Hz). Moreover, I computed the power sum  $LF + 1/HF$  and the ratio  $LF/HF$ .

Furthermore, I extracted novel features from the HF band. The first one, called  $RR_{HF \text{ gauss}}$ , was the mean frequency of a Gaussian distribution used to fit the Lomb-Scargle PSD estimated in the HF band. This feature describes the shifting in frequency of the PSD in the HF band, where the shift is mainly caused by the respiratory activity [159]. The second one was called  $RR_{HF \text{ pond}}$  and was defined as:

$$RR_{HF \text{ pond}} = \frac{\sum_{f \in HF} f \text{PSD}\{RR[k]\}(f)}{\sum_{f \in HF} \text{PSD}\{RR[k]\}(f)} \quad (3.2)$$

Finally, I also computed the power of the HF divided into 5 sub-bands of equal length ( $RR_{HF \text{ sband } X_n}$ ), where the subscript index  $X = \{1, \dots, 5\}$ .

#### Photoplethysmogram (PPG)

According to [10], I delineated the PPG signal and I extracted the following physiological markers: the Pulse Period (PP), the time interval between two consecutive pulse peaks; the Pulse Amplitude (PA), the difference between the pulse peak and the pulse onset; the Pulse Transit Time (PTT<sub>M</sub>), the time interval between the R-Peak in the ECG signal and the instant when the PPG pulse reaches half of its onset-to-peak amplitude; the Pulse Rising Time (PRT), the time interval between the pulse onset and the pulse peak; and the Pulse Rising Speed (PRS), the ratio between amplitude difference and time interval computed from the points of the pulse wave located at 75% and 25% of the onset-to-peak amplitude, respectively.

From each of the aforementioned PPG physiological markers, I extracted features in both time and frequency domains, following the methodology applied to compute HRV from NN-intervals.

#### Peripheral Skin Temperature (SKT)

From the SKT signal, I directly extracted the SKT<sub>Gradient</sub> and the SKT<sub>Power</sub> of the signal. The SKT Gradient was computed as the mean of the difference between the portion of samples recorded during the first second of the window, acquired at a sampling frequency  $f_s$ , and the samples from the final one second of the window, namely:

$$\text{SKT}_{\text{Gradient}} = \frac{1}{f_s} \sum_{k=0}^{f_s} (\text{SKT}[k] - \text{SKT}[60f_s - k]) \quad (3.3)$$

Then, the SKT Power was the average power of the signal computed over the entire window of samples:

$$\text{SKT}_{\text{Power}} = \frac{1}{60f_s} \sum_{k=0}^{60f_s} \text{SKT}[k]^2 \quad (3.4)$$

#### Respiratory Sinus Arrhythmia (RSA)

Finally, I monitored Respiratory Sinus Arrhythmia (RSA), which is the natural variation in the heart rate that is associated with the respiratory cycle. Thus, it was measured from the ECG signal. RSA has been used as a noninvasive measure of

### Chapter 3. Cognitive Workload Detection Method

---

cardiac vagal tone, as a marker of PSNS tone [140] and thus, it could be used as a marker of the disruption of homeostasis induced by a highly demanding task.

Since RSA and cardiac vagal tone could dissociate under certain circumstances [71], I considered the hypotheses that these differences could come from external factors, such as a need to compensate for changes in cognitive workload.

RSA was estimated from the non-uniform time series of successive NN-intervals, which I interpolated using a linear function and I resampled at 2 kHz to be compared with the RSP signal. Then, I filtered the resulting uniform time series of successive NN-intervals with a 4th order band-pass Butterworth filter with cutting frequency at 0.15 and 0.4 Hz yielding an RSA. Indeed, this resampling frequency is quite unusual (normally around 4 Hz [71]). The reason is that I used a single sampling frequency for simplifying the synchronization of all acquired signals and the executed task, which required a sampling frequency of 2 kHz. Therefore, to avoid the interpolation of both RSA and RSP signals for comparison, I decided to work at 2 kHz.

From the computed RSA, I extracted features that aim to evaluate the agreement with the measured RSP signal, but first, both RSP and RSA signals were normalized to zero mean and unit variance. The first feature was the time delay of the RSA with respect to the RSP ( $RSA_{Lag}$ ), which was estimated by computing the cross-correlation of RSA and RSP. I also computed the phase shift between the two signals, which was given by Equation 3.5.

$$RSA_{Phase} = \cos^{-1} \left( \frac{RSP \cdot RSA}{\|RSP\| \cdot \|RSA\|} \right) \quad (3.5)$$

Subsequently, I extracted features based on the Tukey mean-difference plot, also called the Bland-Altman plot [19], to compare both RSA and RSP measurements. To this end, I computed the statistical features of the difference between the two signals and the mean of the two:

$$R_0 = RSP - RSA \quad (3.6)$$

$$A_0 = (RSP + RSA) / 2 \quad (3.7)$$

I also considered the statistical features of different log transformations of the mea-

surements, which yielded the following equations:

$$R_b = \log_b(\text{RSP}) - \log_b(\text{RSA}) \quad (3.8)$$

$$A_b = (\log_b(\text{RSP}) + \log_b(\text{RSA}))/2, \forall b = \{n, 2, 10\} \quad (3.9)$$

where the subscript  $b$  denoted the base of the logarithm (i.e.,  $n$ , 2, and 10).

#### 3.5.2 Features Normalization

Since the relative range of each feature varies widely, a normalization was applied so that each one contributed approximately equally to the classification problem. Hence, I applied a min-max normalization, which scaled the features within a 0-1 range.

Moreover, to address the problem related to both inter-subject and inter-day variability [38, 124], I investigated the following 3 different types of normalization. First, the total normalization (TN) consists of a normalization based on the full training set. Second, the subject dependent normalization (SN) consists of a normalization based on each training subset relative to a specific subject. Finally, the day and subject dependent normalization (DSN) affects each portion of the training set relative to both a specific day and a specific subject. Consequently, both the training and the test sets were scaled accordingly, using the parameters obtained only from the training set.

Finally, I selected the best normalization strategy that better emphasized the discriminant power of the features and their ability to classify the problem. In other words, I selected the method that gave the highest Fisher Discriminant Ratio (FDR) [185] of the normalized feature sets, which were obtained by applying one of the three proposed normalization methods (i.e., TN, SN, or DSN). Then, from the original training set, I created three different data sets, and each of them was normalized with one of the proposed normalization methods. Finally, I evaluated the classification performance of an SVM that used as input one of the three data sets, but with a common set of 14 features obtained after applying both feature pre-selection and Recursive Features Elimination (RFE), as described next (see subsection 3.5.3). This common set of features was chosen to avoid any bias, which can come from the use of a different type or number of features. The results were reported in Sec. 3.8.2.

### 3.5.3 Features Selection

Given the large number of features considered for the exhaustive characterization of cognitive workload, I divided the feature selection process into two main steps. First, as a pre-reduction to suppress the features that did not give any discriminatory information, I applied filter methods, which were particularly effective in computation time and robust to overfitting. Then, as a final reduction to consider the possible interactions between features, I applied embedded methods, which performed feature selection and classification simultaneously. Both feature selection steps were performed once with data from the training set.

For the pre-reduction of the feature space, I applied the following three methods. First, I applied a two-sample Student's t-test [185], which selected statistically discriminant features. As an alternative to Student's t-test, we could use Kruskal-Wallis [185], which does not require normality assumptions of the distribution. While using this alternative, this first feature reduction step is a bit more conservative, in the sense that it keeps a few more features. However, those extra features resulted to have the lowest discriminant power, which was visible on the following step of this feature selection process. Second, the discriminant features were ranked based on their FDR, which gave a score based on their ability to discriminate the problem. Lastly, I further reduced the feature space by removing the features that gave any redundant information. In particular, I removed the less discriminant features that were correlated with others with a Pearson's correlation coefficient above 0.95, which indicated a very strong correlation [3].

For the final reduction of the feature space, I applied RFE [72], which is an embedded method that uses an external estimator to assign weights to features. These weights were then used to prune the least important features from the current set. This procedure recursively pruned the selected features until all feature weights were different from 0. In this work, I applied RFE based on different classifiers; namely, LR, LDA, SVM, RF, and XGB; which I named: RFE-LR, RFE-LDA, RFE-SVM, RFE-RF, and RFE-XGB, respectively.

## 3.6 Cognitive Workload Monitoring

For the cognitive workload monitoring I explored the use of different machine learning algorithms. In particular, I investigated the use of linear models, namely LR, LDA, SVM, and Gaussian Naive Bayes (GNB), for a feasibility check. Then, I investigated

the use of non-linear models, such as k-Nearest Neighbors (k-NN), Quadratic Discriminant Analysis (QDA), SVM with a Radial Basis Function (RBF) as kernel, DT, RF, and XGB, with the aim of reducing the bias. The accuracy of each model in detecting high levels of cognitive workload was evaluated based on a 5-fold cross-validation. To this aim, the training set was randomly divided into folds of equal sizes (groups of samples). The prediction function was learned using four folds, and the fold left out was used for validation. The procedure was followed for each of the five folds. The performance measure reported by the cross-validation was then the average of the five evaluations.

Moreover, I considered a personalized learning approach to deal with the highly person-dependent variance. To this aim, I compared the performance of both a Universal Background Model (UBM) and a Subject-Specific Model (SSM).

#### 3.6.1 Model for Cognitive Workload Monitoring

To estimate cognitive workload I chose for both UBM and SSM a linear SVM, which has the following prediction model [18]:

$$y(x) = \mathbf{w}^T \mathbf{x} + b \quad (3.10)$$

where  $\mathbf{x}$  is the input vector,  $\mathbf{w}$  is the weight vector, and  $b$  is the offset. The corresponding optimal hyperplane separating the two classes is defined by the relation:

$$y(x) = \mathbf{w}^T \mathbf{x} + b = 0 \quad (3.11)$$

Therefore, an input vector  $\mathbf{x}$  was then assigned to class 1 if  $y(x) \geq 0$  and to class  $-1$  otherwise. Although the prediction model of both UBM and SSM was the same, the difference laid in the objective function. All the details are given in subsection 3.6.2 and subsection 3.6.3.

The parameters of both UBM and SSM were chosen based on a 5-fold cross-validation on the training set. For this validation I used a stratified split that preserves the same percentage for each target class as in the complete training set and also preserves the same percentage of data relative to the subject of interest. Then, the generalization of both models was tested on an unseen test set.

### Chapter 3. Cognitive Workload Detection Method

---

The performance of the models was evaluated based on: accuracy, the proportion of both true positives and true negatives results among the total number of cases; precision, or confidence, the proportion of predicted positive cases that are correctly real positives; recall, or sensitivity, the proportion of real positive cases that are correctly predicted positive; Receiver Operating Characteristic (ROC); and in particular, based on the F1-score, the weighted average of the precision and recall.

#### 3.6.2 Training of the Universal Background Model

The considered UBM was based on SVM with soft margins [18], which relaxed the condition for the optimal hyperplane (Eq. Equation 3.11) and allowed possible overlaps of the class-conditional distributions. As for a normal soft-margin SVM, the objective function of the UBM was defined as follows:

$$\begin{aligned} \arg \min_{w, b, \xi_i} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i \in D} \xi_i, \\ \text{subject to} \quad & t_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0; \quad (i \in D) \end{aligned} \quad (3.12)$$

where the regularization term  $C$  and the non-negative variables  $\xi_i$  relax the constraints of an otherwise hard-margin SVM. The data  $x$  in the training dataset  $D$  comprises  $N$  input vectors  $x_1, \dots, x_N$ , with corresponding target values  $t_1, \dots, t_N$ , and where  $t_i \in \{-1, 1\}$ . The parameter  $C$  is analogous to the inverse of a regularization coefficient because it controls the trade-off between minimizing training errors and controlling model complexity. A regularization term  $C = 0.1$  is chosen from a  $\log_{10}$  scale ranges from 0.001 to 1000 based on a stratified 5-fold cross-validation on the training set.

#### 3.6.3 Training of the Subject-Specific Model

As well as for the UBM, the considered SSM was based on a soft-margin SVM. However, in order to adapt the model to a specific subject, I modified the objective function of the original soft-margin SVM (Equation 3.12) to include two different soft-margins. The first soft-margin ( $C_s$ ) changes the degree of importance that is given to false estimations of samples coming from a particular subset of data, which can be a particular subject ( $S$ ). In other words, the term weighed by  $C_s$  allows a minimization of the errors ( $\xi$ ) for all the  $x$  in the training set related to a specific subject



( $x \in S$ ). Instead, the second soft-margin ( $C$ ) affects the rest of the dataset minimizing the errors  $\xi$  for all the  $x$  in the training set that are related to other subjects ( $x \notin S$ ).

Therefore, the final objective function for the SSM was defined as:

$$\begin{aligned} \arg \min_{w, b, \xi_i} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i \notin S} \xi_i + C_s \sum_{i \in S} \xi_i \\ \text{subject to} \quad & t_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0; \quad (i \in D) \\ & C_s > C \end{aligned} \tag{3.13}$$

With this model, I stated a preference for margins that classified the training data correctly, but I softened the constraints to allow for non-separable data with different penalties. To promote the minimization of the total sum of the penalties  $\xi_i \forall i \in S$ , despite the minimization of the total sum of the penalties  $\xi_i \forall i \notin S$ , I chose  $C_s$  to be greater than  $C$ . As usual, the regularization terms have to be large enough to avoid under-fitting, but not too much to avoid over-fitting as well. Based on a stratified 5-fold cross-validation on the training set, both regularization terms  $C = 0.001$  and  $C_s = 0.1$  are chosen from a  $\log_{10}$  scale in a ranges from 0.001 to 0.1 and from 0.1 to 100, respectively. Although both regularization terms seem to be bounded by the considered range, I kept the lower bounds as they are to avoid possible problems of under-fitting.

## 3.7 Experimental Setup

Collecting data in a real search and rescue mission is complex, because of the random frequency the events occur, but also because there are many variables that are still undefined. Therefore, to collect clean data for building a cognitive workload monitoring model and validate my approach, I used the simulator for search and rescue missions with drones, as used in [38], following two carefully designed study protocols (sections 3.7.2 and 3.7.3).

A first experiment was conducted to characterize cognitive workload levels in search and rescue mission with drones through physiological signals, to build a model for continuous monitoring, and to evaluate the contribution of a subject-specific approach. A second experiment was done to evaluate the quality of the system in case of the use of new advanced controllers, such as the FlyJacket design [153]. Considering that in previous experiments, participants reported some symptoms

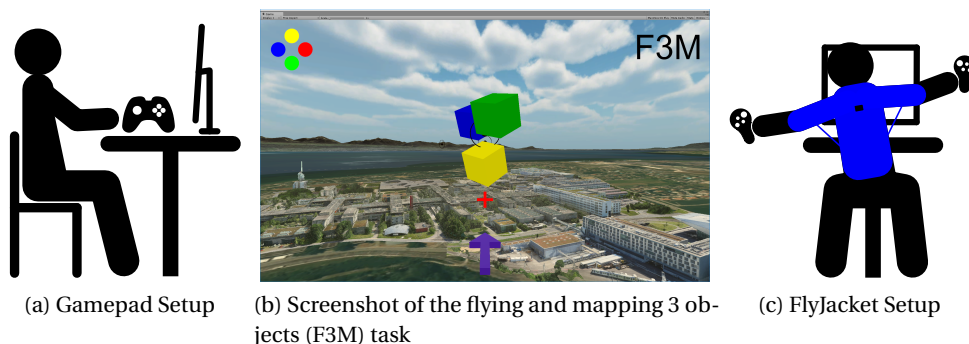


Figure 3.3 – The figure on the left (3.3a) shows the setup used to control the simulator with the gamepad while the figure on the right (3.3c) shows the use of FlyJacket. The figure in the middle (3.3b) shows a screenshot of the search and rescue drone simulator, where the layout of the buttons and the current task are displayed on the top-left and top-right corners, respectively. The purple arrow on the bottom indicates the direction of the next waypoint (black circles). The red cross above the arrow indicates the center of the drone. The cubes in the middle are the objects to be detected by pressing a button on the controller.

of sickness due to the use of Virtual Reality (VR) glasses, I decided to use for both experiments a screen instead of VR. This decision avoided, first, diseases between participants and, second, incomplete data collection caused by an eventual need to prematurely stop the experiment.

In contrast with the gamepad, where the movements were limited to the thumbs, the use of FlyJacket implies movements of both arms and torso. Therefore, when comparing tasks involving different type of movements, there is a risk of yielding in a performance overestimation. Thus, to avoid as much as possible any possible miss-classification caused by movement artifacts, I trained the machine learning algorithm with the data collected during the experiment with the gamepad. Then, to do a final tuning of the SSM, I only used the data of the first trial of the experiment with FlyJacket. Finally, the data of the second trial of the experiment with FlyJacket was used as test set.

All the signal processing, features extraction, machine learning design, and classification were done using Matlab R2016a [184].

#### 3.7.1 Search and Rescue Drone Simulator

As presented in [38], the simulator consists of a simplified search and rescue scenario, where the pilot of a drone has to deal with two different activities, namely, flying and mapping. The flying activity consists in flying a drone following a randomly generated trajectory depicted by spherical waypoints. Instead, the mapping activity consists in mapping the situation of a disaster area, which is represented by cubes of 4 different colors randomly distributed over the flying trajectory.

The colors are: yellow to indicate rescue situations, red for fire, blue for water damages, and green for accidents. The colors were chosen according to the regulation of the Swiss Firefighters [64]. An overview of the scenario is shown in Fig. 3.3.

Using the simulator, I modulated both flying and mapping activities to induce different levels of cognitive workload. The same principle was applied in [38] and in MATB-II [158], where different tasks were combined to induce different levels of workload. Here, flying and mapping activities were combined yielding in four different tasks: Baseline (B), Flying (F), Mapping 3 objects (3M), and Flying and Mapping 3 objects (F3M), which are described next.

##### **Baseline (B)**

As baseline I considered a flying sequence controlled by an auto-pilot. During this task, no special activity was required, the participants only had to watch the sequence. This sequence put the participants in a framework that was the same for the entire experiment, avoiding as much as possible changes of uncontrollable variables. This task had the lowest expected workload level of this study.

##### **Recovery (R)**

What we call the recovery (R) phase is nothing else than a repetition of the baseline. The only difference comes from the fact that R was proposed after a sequence of tasks, and not at the beginning as B. The use of this recovery phase is reserved for future studies, which aim to investigate how long the physiological signals need to return to their baseline values.

## **Chapter 3. Cognitive Workload Detection Method**

---

### **Flying (F)**

For the flying task, the participants were asked to pilot the drone flying as close as possible through the center of the waypoints, no mapping activity was required. A medium/high workload level was expected in this task.

### **Mapping 3 Objects (3M)**

This task consisted of mapping three objects that were randomly displayed on the screen at a time. The flight was carried out by the auto-pilot, as in B. During this task, the participants were asked to press the button on the controller that had the same color as the object displayed on the screen. The total number of objects to be mapped was 240 per session, i.e., 60 per color. As in F, a medium/high workload level was expected in this task.

### **Flying and Mapping 3 Objects (F3M)**

This task consisted of performing both F and 3M simultaneously. As multitasking skills were required to perform F3M, I considered that this task induced a high level of cognitive workload.

## **3.7.2 Study Protocol 1: Use of a Gamepad**

During this experiment, participants sat in front of a screen and controlled the simulator with a gamepad from Logitech [96], as shown in Fig. 3.3a. To avoid as much as possible false detection caused by artefacts, participants were asked not to talk and to avoid as much as possible any kind of unnecessary movements. Otherwise, they were free to rest and move during the resting periods.

The experiment started with a setup phase, in which after providing the study and protocol explanations to the participants, I placed the sensors for the physiological signals acquisition. Then, participants started with a warm-up phase to get familiar with the simulator. This warm-up sequence included a mix of both flying and mapping activities as presented in F3M. However, the mapping activity only involved the detection of one object at a time.

The rest of the protocol is shown in Fig 3.4. After warm-up, participants performed the first trial, which started with a baseline of five minutes and followed by a sequence

### 3.7. Experimental Setup

including F3M, 3M, and F, executed in a randomized order. A resting period of 3 minutes was enforced after each task. This period also allowed participants to fill a questionnaire (Q), based on the NASA-TLX procedure.

Finally, the participants performed two additional trials, namely Trial 2 and Trial 3. These two additional trials started with a baseline and continued with a randomized sequence of F3M, 3M and F. Both trials ended with R followed by a resting period, in which the NASA-TLX was filled again. Each task presented in Trial 2 and 3 lasted three minutes.

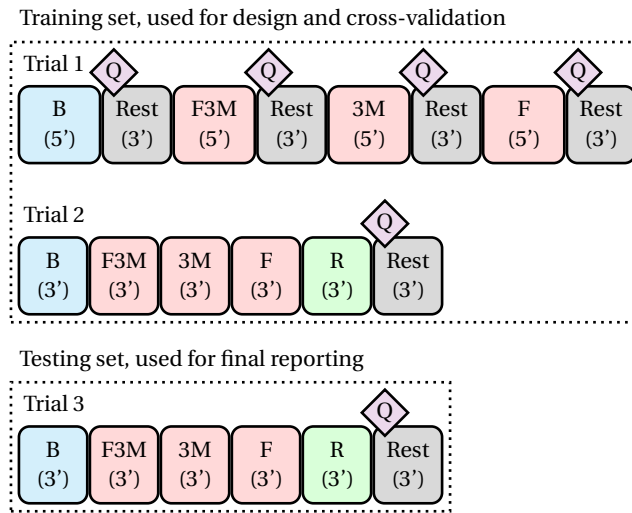


Figure 3.4 – Protocol of the experiment with the gamepad. The trials start with a baseline (B), which is followed by the tasks flying and mapping 3 objects (F3M), mapping 3 objects (3M), and flying (F), presented in a randomized order. Then, it includes a recovery (R) period, which is equal to B, but presented after a sequence of tasks. Finally, after each task or sequence, a resting period is proposed to fill a questionnaire (Q).

As shown in Fig. 3.4, I used all data acquired during both Trial 1 and Trial 2 as training set, and all data collected during Trial 3 as test set. I am aware that this split does not truly respect independent temporality of data because both training and test data sets are taken from the same day and not from a day that is not used for training (as it would be in a real application). Therefore, this choice may overestimate the generalization performance of the system. However, as I expected an inter-day variability of the physiological responses [38, 124], I assumed that a daily calibration of the system would be required, which implied a need for some data of the same

### Chapter 3. Cognitive Workload Detection Method

day. A further investigation over different days could potentially avoid the burden on such a daily calibration, but this analysis was left for a future study.

#### 3.7.3 Study Protocol 2: Use of FlyJacket

In this case, participants controlled the drone simulator with FlyJacket and mapped the disaster situation with the simultaneous use of both left and right Oculus Touch controllers [125], as shown in Fig. 3.3c.

The experiment started with a setup and a warm-up phase, as in the previous experiment (Sec. 3.7.2). Then, participants performed two trials as shown in Fig. 3.5, which started with a baseline of five minutes and followed by a sequence including F3M, and F executed in a randomized order. Again, a resting period of three minutes was enforced after each task, where the participants filled the questionnaire.

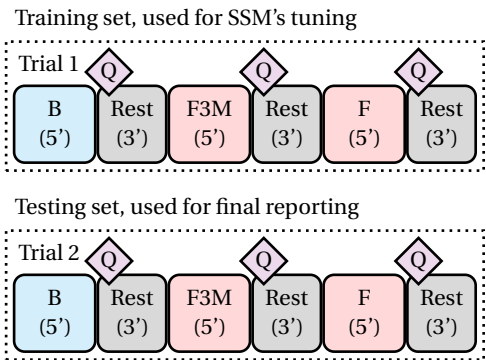


Figure 3.5 – Protocol of the experiment with FlyJacket. The trials start with a baseline (B), which is followed by the tasks flying and mapping 3 objects (F3M), and flying (F), presented in a randomized order. After each task, a resting period is proposed to fill a questionnaire (Q).

Being this second experiment only for proving the feasibility of detecting both low and high levels of the cognitive workload with the proposed method, I decided to present a reduced protocol to prevent participants from doing long, tedious and tiring experiments. Therefore, I considered this protocol with only two trials, with three tasks of five minutes each, and keeping the use F for future studies.

### **3.7.4 Acquired Signals**

I recorded RSP, ECG, PPG, SKT, and EDA in a noninvasive way through the Biopac MP160 data acquisition system [16] with a sampling frequency of 2 kHz. We also recorded EEG, but its use was reserved for a different study, which has been published recently [83]. EEG was not considered for this work due to the obtrusiveness of the sensors, which are not integrable into a jacket.

### **3.7.5 Research Participants**

The experiment with the gamepad was done by 24 participants (6 females and 18 males) aged between 21 and 39 years old ( $27.7 \pm 4.8$ ), who performed the study protocol twice in two sessions that took place on different days.

The experiment with Flyjacket was done by 10 additional participants (3 females and 7 males) aged between 22 and 30 years old ( $26.8 \pm 2.3$ ), who participated in the experiment on a single day session.

All 34 participants provided informed consent and volunteered to participate in the study. The participants were healthy, free of any cardiac abnormalities and were receiving no medical treatment. The ethical approval for this study was obtained from the Cantonal Ethics Commissions for Human Research Vaud and Geneva, ethical approval application number PB2017-00295.

## **3.8 Experimental Results**

Given the recorded data set of study protocol 1, I selected the best combination of normalization, feature selection, and classification methods suitable for cognitive workload monitoring. The methods were obtained based on the cross-validations workflow including 747 observations. Finally, I showed the performance of the proposed methods on two unseen test sets, including 260 and 57 observations from study protocols 1 and 2, respectively.

### **3.8.1 Self-perception of induced cognitive workload**

The reported overall workload on each task perceived by the 34 participants based on the NASA-TLX is shown in Fig. 3.6. A one-way Analysis of Variance (ANOVA) [59] conducted on the influence of the tasks confirms that participants have perceived

### Chapter 3. Cognitive Workload Detection Method

different levels of workload. Furthermore, a multiple pairwise comparison analysis using the Student's t-test with up to 164 samples revealed, except for 3M vs. F, statistical significant mean differences ( $p$ -value  $< 0.001$ ). The missing values in the data range were excluded from this multiple comparison. More precisely, all comparisons with the 3M task were limited to 144 samples, as the protocol with the FlyJacket setup was performed without executing this task.

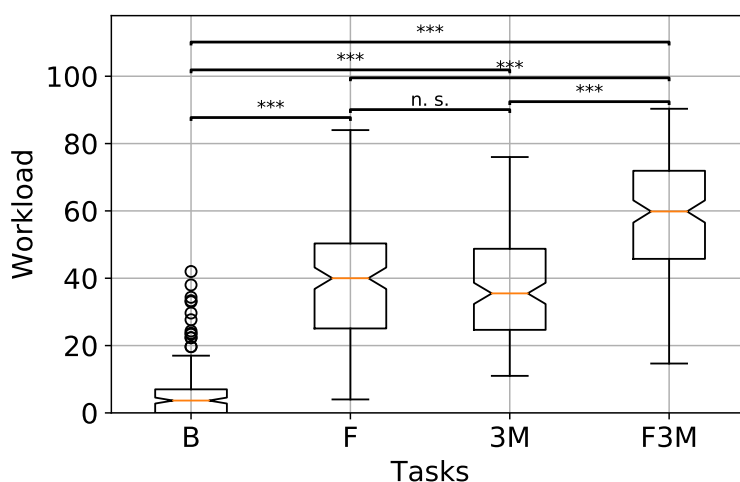


Figure 3.6 – Cognitive workload level perceived by the participants while performing baseline (B), flying (F), mapping 3 objects (3M), and both flying and mapping 3 objects (F3M). The results are based on the NASA-TLX. Student's t-tests with up to 164 samples were applied (n.s.: not significant, and \*\*\*:  $p < 0.001$ ).

However, as shown in Fig. 3.6, the perceived level of cognitive workload has a large variance. A two-way ANOVA revealed that such a large variance came from a significant ( $p < 0.001$ ) effect of task, day, and subject on the level of cognitive workload,  $F(3,414) = 1637.19$ ,  $F(1,414) = 28.70$ ,  $F(33,414) = 48.93$ , respectively. Therefore, the results of the NASA-TLX confirmed the need for both a day- and a subject-specific approach.

Although there was a significant difference in the perceived workload between most of the tasks, Fig. 3.6 shows that the distribution of both F and 3M presented a large overlap with F3M. Instead, the difference between tasks B and F3M was clear. Therefore, as I first wanted to check if I could detect low and high levels of cognitive workload, I focused on the extreme cases, which were induced by tasks B and F3M, respectively. The use of F and 3M was left for a future investigation, where a fine-



grained cognitive workload monitoring would be targeted.

#### 3.8.2 Features discriminant power emphasized by normalization

To reduce the variance introduced by the different participants as well as by the fact that they did the experiment on different days, I investigated different normalization approaches (i.e., TN, SN, and DSN) as described in Section 3.5.2. After applying each normalization approach, I firstly evaluated the discriminant power of the features based on their FDR. Results are shown in Fig. 3.7, where it can be seen that DSN better emphasises the discriminant power of the features. In comparison with TN, the FDR of the most important feature was emphasised by a factor of 80.9% or 166.9%, over SN or DSN, respectively.

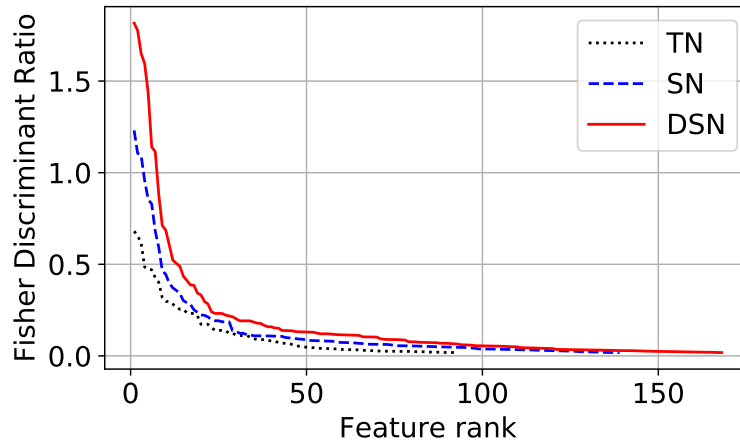


Figure 3.7 – Comparison of different normalization methods affecting the discriminant power of the features. A day and subject dependent normalization (DSN) shows a higher Fisher Discriminant Ratio (FDR) than a subject dependent normalization (SN) and a total normalization (TN).

Secondly, following the methodology presented in Sec. 3.3, I compared how each normalization approach contributed to the classification problem by using a linear SVM model. I noticed that the normalization affected the feature selection process, which selected 14 features after TN or SN, or 25 features after DSN. Therefore, to avoid biased results caused by the use of a different number of features, I used for this comparison the first 14 most discriminant features selected by RFE-SVM after TN, SN, or DSN normalization. Fig. 3.8 shows the ROC and the F1-score of the SVM combined with the different normalization methods, where it can be seen that once

again DSN outperforms both TN and SN.

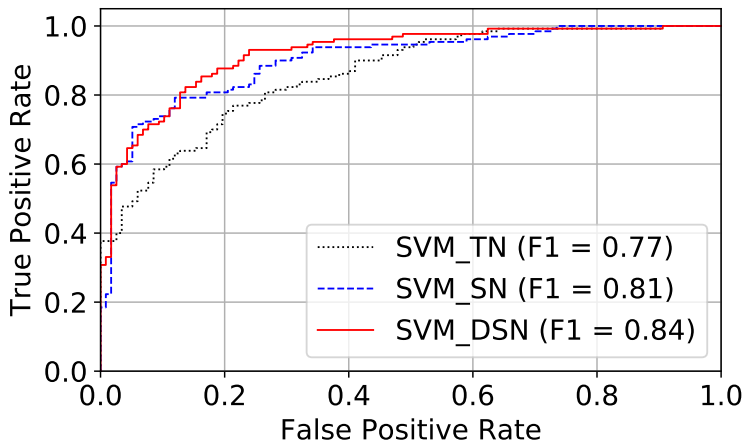


Figure 3.8 – Effect of different normalization methods on cognitive workload monitoring with Support Vector Machine (SVM). The use of a day and subject dependent normalization (DSN) shows a higher performance than a subject dependent normalization (SN) and a total normalization (TN).

My results showed that feature normalization plays an important role during both features selection and classification. DSN normalization gave better results (a bigger F1-score) compared to SN and TN. Similar trends were obtained by applying RFE with other classifiers, such as LR or LDA. Therefore, I selected DNS as normalization method.

### 3.8.3 Physiological featuring of cognitive workload

By applying the filter methods presented in Section 3.5.3, I eliminated 282 non-informative features from the normalized (based on DSN) 384 features initially considered for an exhaustive cognitive workload characterization. In particular, I reduced the dimension of the feature space from 384 down to 168 features with the two-sample Student's t-test, and down to 102 features by checking their linear correlation.

Although, the above pre-selection step drastically reduced the feature space, the use of such an amount of features requires models with high capacity and may lead to overfit if trained with a limited dataset like ours. Therefore, to obtain a reasonable feature set that can be used for cognitive workload monitoring, a further dimension

### 3.8. Experimental Results

Table 3.3 – Feature selection performance while combining both Recursive Features Elimination (RFE) and classification methods, such as Logistic Regression (LR), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Support Vector Machine (SVM), with both linear and Radial Basis Function (RBF) kernels, Gaussian Naive Bayes (GNB), k-Nearest Neighbors (k-NN), Decision Tree (DT), Random Forest (RF), and Extreme Gradient Boosting (XGB).

Embedded Features Selection (number of selected features)										
No RFE (102) RFE-XGB (5) RFE-LR (10) RFE-LDA (12) RFE-SVM (25)										
Accuracy of different classifiers										
Classifiers	Train	CV	Train	CV	Train	CV	Train	CV	Train	CV
LR	93	85	85	85	90	86	90	87	91	86
LDA	94	84	85	86	91	85	91	87	92	87
QDA	100	82	87	86	90	87*	92	87	95	88
SVM <sub>Lin</sub>	88	87	85	87*	89	86	87	86	89	88*
SVM <sub>RBF</sub>	92	87*	87	88*	91	86	92	88	93	87
GNB	84	85	85	88*	86	86	86	85	88	88
k-NN	91	85	90	87*	91	85	92	87*	92	87
DT	90	85	89	84	90	80	88	76	90	84
RF	90	87	90	85	90	85	90	85	92	85
XGB	94	87	90	85	92	84	91	85	93	85
(*) Best F1-score:	89		88		88		89		90	

reduction based on an embedded method was applied, as presented in Section 3.5.3.

The features space was reduced from 102 to 5, 10, 12 and 25 by applying RFE-XGB, RFE-LR, RFE-LDA, RFE-SVM, respectively. A consistent set of features was found by RFE based on LR, LDA, and SVM. For the case of RFE-XGB, I used a low-complex model to avoid overfitting and inconsistent results. In particular, I limited the model to 10 estimators and three as the maximum depth of each decision tree. Such a low-complex RFE-XGB showed a drastic lower selection compared to other methods. Without banning the ensemble methods from building complex models, RFE does not converge to the same result if executed several times. In contrast, by limiting the model complexity, RFE provides a reproducible result. However, this trick does not

### Chapter 3. Cognitive Workload Detection Method

---

help the RFE-RF method, which did not converge to a consistent solution. Indeed, the model always selected a different set of features every time the method was applied, even by reducing the model complexity (i.e. number of estimators and maximum depth of tree as I did for RFE-XGB). Therefore, such complex models are not suitable for small datasets, which oblige the use of low complex models.

The feature set obtained after applying both filter and embedded methods is shown in Table 3.4. Although the number of selected features varies between 5 and 25 depending on the applied embedded methods, a common subset of features was identified. I observed that the features obtained by RFE-LR, RFE-LDA and RFE-XGB were almost all included in the feature set obtained by RFE-SVM. In particular,  $RSP_{Rate\ Median}$  and  $SKT_{Power}$  were selected by all the four methods, followed by  $RSP_{Prd\ Median}$ ,  $SKT_{Gradient}$ ,  $RSA_{R2\ Std}$ ,  $PRT_{Median}$ ,  $RSP_{Rate\ Diff\ RMS}$  and  $PP_{Median}$ , which were selected by three methods out of four. Therefore, this result suggested that these eight features seem to be the most important ones in terms of cognitive workload characterization in the context of this experiment.

Additionally, I investigated the effect of using the different feature sets obtained with the considered RFE methods on different classification methods. Results are presented in Table 3.3, where I reported both the training and the cross-validation accuracy of each RFE and classification method combination.

A significant difference between training and cross-validation accuracy indicated a sign of overfitting (e.g., QDA with 102 features). Moreover, I reported the best cross-validation F1-score for each applied RFE method. Although there seem to be no significant differences across methods, the highest best F1-score, as well as the best cross-validation accuracy, were reached when linear SVM was applied on both RFE and classification. Therefore, RFE-SVM was the employed feature selection method hereafter.

#### 3.8.4 Classifiers for cognitive workload monitoring

A ROC curve was used to further evaluate the performance of the considered classifiers. I reported in Figure 3.9 the ROC curve of cognitive workload classification in cross-validation. In particular, for greater clarity of the illustration, I only reported the results of the best classifiers (Area Under the Curve (AUC)  $\geq 0.94$ ), namely LR, LDA, k-NN, linear SVM, and SVM with RBF kernel. My results showed that, with the amount of data available, the use of non-linear models did not really increase the

### 3.8. Experimental Results

Table 3.4 – Mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the most important features used to detect low and high levels of cognitive workload, induced by baseline (B) and by both flying and mapping 3 objects (F3M), respectively. The last column shows the p-value of the two-sample t-test. The features are grouped by physiological signals, namely Respiratory Activity (RSP), Electrocardiogram (ECG), Photoplethysmogram (PPG), and Skin Temperature (SKT).

Physiological Features	Task B $\mu \pm \sigma$	Task F3M $\mu \pm \sigma$	p-Val < $10^{-x}$ $x$
RSP <sub>Rate</sub> Mean <sup>1,4</sup>	0.28 ± 0.22	0.71 ± 0.23	107
RSP <sub>Rate</sub> Median <sup>1,2,3,4</sup>	0.28 ± 0.23	0.71 ± 0.23	106
RSP <sub>Prd</sub> Mean <sup>1,2,3</sup>	0.61 ± 0.25	0.22 ± 0.21	90
Inh <sub>Time</sub> Median <sup>1</sup>	0.53 ± 0.31	0.20 ± 0.23	52
Exh <sub>Time</sub> Median <sup>1</sup>	0.63 ± 0.29	0.32 ± 0.28	45
Inh <sub>Time</sub> Mean <sup>1</sup>	0.50 ± 0.31	0.22 ± 0.25	38
Inh <sub>Time</sub> RMS <sup>3,4</sup>	0.44 ± 0.30	0.19 ± 0.25	31
RSA <sub>R2</sub> Std <sup>1,3,4</sup>	0.40 ± 0.28	0.57 ± 0.28	17
RSP <sub>Pks</sub> Mode <sup>1</sup>	0.61 ± 0.29	0.50 ± 0.30	08
RSP <sub>Rate</sub> Diff RMS <sup>1,2,3</sup>	0.33 ± 0.29	0.42 ± 0.29	06
RSP <sub>PSD3n</sub> <sup>1</sup>	0.35 ± 0.29	0.43 ± 0.30	05
RSP <sub>PSD1n</sub> <sup>1</sup>	0.48 ± 0.35	0.41 ± 0.31	04
RR <sub>HF</sub> gauss <sup>1,3</sup>	0.32 ± 0.23	0.68 ± 0.26	74
RR <sub>HF</sub> sband 3n <sup>1</sup>	0.47 ± 0.32	0.30 ± 0.28	15
RR <sub>Lorenz</sub> L2 <sup>2</sup>	0.49 ± 0.30	0.35 ± 0.26	11
RR <sub>CVI</sub> <sup>2</sup>	0.54 ± 0.29	0.42 ± 0.28	10
PP <sub>HF</sub> sband 5n <sup>1</sup>	0.23 ± 0.25	0.46 ± 0.30	28
PA <sub>RMS</sub> <sup>1,2</sup>	0.53 ± 0.35	0.32 ± 0.27	20
PA <sub>Lorenz</sub> L <sup>1</sup>	0.44 ± 0.33	0.26 ± 0.25	17
PRS <sub>Mean</sub> <sup>2</sup>	0.38 ± 0.35	0.55 ± 0.31	13
PP <sub>CSI</sub> <sup>1</sup>	0.46 ± 0.29	0.33 ± 0.26	11
PA <sub>CSI</sub> modified <sup>2</sup>	0.40 ± 0.30	0.28 ± 0.27	09
PRT <sub>Median</sub> <sup>1,2,3</sup>	0.44 ± 0.31	0.56 ± 0.31	08
PTT <sub>M</sub> Mode2 <sup>1</sup>	0.50 ± 0.35	0.58 ± 0.28	05
PP <sub>Median</sub> <sup>1,2,3</sup>	0.55 ± 0.31	0.47 ± 0.28	05
PTT <sub>M</sub> HF pond <sup>1</sup>	0.47 ± 0.28	0.54 ± 0.29	05
PRT <sub>LFp1oHF</sub> <sup>1</sup>	0.38 ± 0.31	0.30 ± 0.27	05
PP <sub>Mode2</sub> <sup>1</sup>	0.55 ± 0.33	0.49 ± 0.28	04
SKT <sub>Power</sub> <sup>1,2,3,4</sup>	0.61 ± 0.35	0.37 ± 0.30	24
SKT <sub>Gradient</sub> <sup>1,2,3</sup>	0.57 ± 0.29	0.38 ± 0.26	20
Selected feature with: <sup>1</sup> RFE-SVM, <sup>2</sup> RFE-LDA, <sup>3</sup> RFE-LR, <sup>4</sup> RFE-XGB			

### Chapter 3. Cognitive Workload Detection Method

detection accuracy. Instead, non-linear models tended to introduce a larger variance between training and cross-validation accuracy.

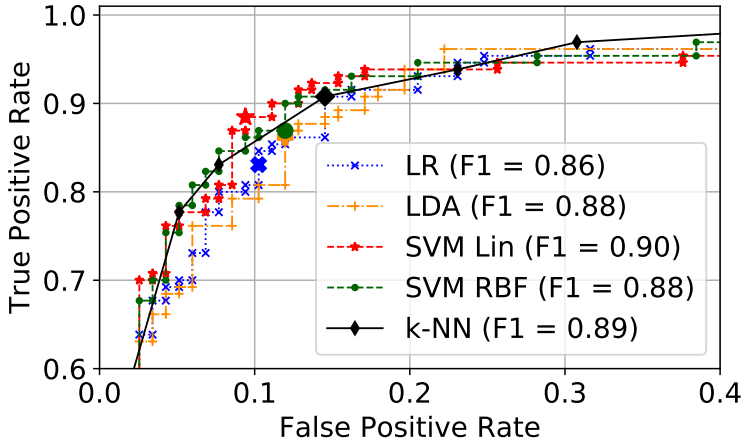


Figure 3.9 – Comparison between the best classifiers on cross-validation. Bigger markers denote the performance of the different models based on their corresponding cross-validated threshold or offset  $b$ .

As illustrated in Fig. 3.9, linear SVM shows a higher F1-score and a better ROC curve, in particular by comparing the bigger markers representing the performance of the models based on their corresponding cross-validated threshold or offset  $b$ . Therefore, a linear SVM was selected for my further investigation.

Although this choice was made to reach the highest classification accuracy, it may not be the optimal solution for embedded implementations. Other solutions considering fewer features may be preferred for implementations in low-power embedded systems, where the power consumption may play an important role. However, my results indicated a certain flexibility in terms of the number of features that have to be used. In fact, the best F1-score was quite similar for all the applied feature selection embedded methods. Moreover, except for DT, the cross-validation accuracy reported in Table 3.3 after RFE was in a restricted range, between 84 and 88%. This range variability seemed to be more dependent on the selected classifier (difference  $> 4.5\%$ ) rather than the number of selected features (difference  $< 3.5\%$ ). In fact, a linear SVM with an input of only five features could provide a reduced implementation complexity with a loss of only 1% of classification accuracy.

### 3.8.5 Classification improved with the SSM

Once I selected the set of features (i.e., 25 features with RFE-SVM) and the classification method, a linear SVM, I tested the contribution of a subject-specific approach compared to a use of a general model (i.e., SSM vs. UBM). First, I trained the models as described in Section 3.6. The regularization term  $C = 0.1$  of the UBM was selected based on a 5-fold cross-validation on the training set. However, for the SSM I selected  $C = 0.001$  and  $C_S = 0.1$ , which were the most common regularization terms found with a 5-fold cross-validation on the training set applied for all the 24 participants.

Finally, Table 3.5 reports the comparison between UBM and SSM, which were tested on an unseen test set emulating an online cognitive workload monitoring. The average accuracy of the UBM is 80.4% and is improved to 87.3% by the use of the SSM. Both the Wilcoxon rank-sum test [193] and the McNemar's test [42] over the

Table 3.5 – Performance of a Universal Background Model (UBM) vs. a Subject-Specific Model (SSM) on a test set collected from participants using a gamepad (Study 1, subsection 3.7.2).

	Model	class	precision	recall	F1-score	samples
UBM		B	0.81	0.76	0.79	123
		F3M	0.80	0.84	0.82	137
		avg	0.80	0.80	0.80	260
SSM		B	0.89	0.83	0.86	123
		F3M	0.86	0.91	0.88	137
		avg	0.87	0.87	0.87	260

260 samples indicated that the SSM showed a statistically significant improvement of the classification performance (p-value < 0.01).

Although an improvement was reached for all the participants on cross-validation while using the SSM, one participant over 24 did not show the expected improvement on the final test set. This result may be explained by the need for more training data that could be used to better fits the physiological response of that particular participant.

However, as shown in Table 3.6, the higher performance of the SSM compared to the UBM is confirmed on the test set acquired using FlyJacket (Study protocol 2, Sec 3.7.3). In fact, a global accuracy of 89.5% is reached by the UBM and is improved

### Chapter 3. Cognitive Workload Detection Method

to 91.2% by the use of the SSM. Both the Wilcoxon rank-sum test and the McNemar's test over the 57 samples indicated again that the SSM showed a statistically significant improvement of the classification performance (p-value < 0.05).

Table 3.6 – Performance of a Universal Background Model (UBM) vs. a Subject-Specific Model (SSM) on a test set collected from participants while FlyJacket (Study 2, subsection 3.7.3).

Model	class	precision	recall	F1-score	samples
UBM	B	0.87	0.93	0.90	29
	F3M	0.92	0.86	0.89	28
	avg	0.89	0.89	0.89	57
SSM	B	0.88	0.97	0.92	29
	F3M	0.96	0.86	0.91	28
	avg	0.91	0.91	0.91	57

The reason for the better performance of the SSM compared to the UBM is that the SSM takes advantage of all the observations with a different weight. The observations related to other participants contribute by letting the model learn the general behaviour, with a regularization term  $C$  that allows a higher misclassification of such observations. However, the observations related to the specific subject contribute by tuning the model with a regularization term  $C_S$  that stresses the margins between the classes for that particular subject. Therefore, a combination of both relaxed and stressed soft margins, introduced by the regularization term  $C$  and  $C_S$ , respectively, allows both the learning of the general behaviour and the tuning of the model that better suits a specific subject. In light of the above, I could conclude that the personalized model performed in general better than the universal model.

Moreover, the results obtained by the SSM were comparable with the state-of-the-art (See Table 3.1), in particular with the work presented in [113]. Although a similar accuracy was reached, my model was simpler and used a reduced number of features. Another important difference was found in how the test set was selected. The authors in [113] achieved an accuracy of 86% with a randomly selected test set. Instead, I selected as test set the data from the last trial performed by each subject, namely Trial 3. For any classification problem that broke the interchangeability hypothesis, such as the time dependent cognitive workload monitoring, a random training/test split should be avoided, as it yielded to a biased evaluation of the model. With a



random split, the model learnt from prospective data, which were normally not available when designing and training a prediction model. Moreover, the model was evaluated based on retrospective data, which were too similar to the training data. Consequently, the classifier tended to look better than it really was. Therefore, to estimate how well a model will work with new data, a time dependent training/test split should be considered.

Although the evaluation of my model was less biased compared with the literature, there was still a minor risk of performance overestimation. In fact, both training and test sets were taken from the same day and not only from a day that was not used for training. Therefore, this choice did not fully generalize how well a model would work with data collected on new days, as the proposed model required a daily calibration.

The slightly better performance achieved by the classifier while using FlyJacket instead of the gamepad was assigned to the increased amount of training data. In fact, for the case of FlyJacket, the weights of the classifier were tuned based on all data collected during study protocol 1 including Trial 1 of study protocol 2. A quick test considering less training data (ignoring Day 2 of study protocol 1) reduced the accuracy of the UBM from 86% to 82%.

A potential limitation while using FlyJacket is the possible effect of movement artifacts, which may differ from Task B to Task F3M and could potentially affect the classification accuracy. Therefore, for a better understanding of where movement artifacts, if any, could affect the physiological features in use, I performed multiple times the Wilcoxon rank-sum test [193] (i.e., one test per feature) comparing both scenarios, namely, data acquired from participants using the gamepad versus data acquired while using the FlyJacket setup. In this comparison, the Wilcoxon rank-sum test revealed that there is no evidence of stochastic dominance between most of the features ( $p\text{-val} > 0.05$ ), except for eight of them, namely,  $PP_{CSI}$ ,  $PP_{HF\ sband\ 5n}$ ,  $PA_{RMS}$ ,  $PTT_{M\ HF\ pond}$ ,  $RSP_{Prd\ Mean}$ ,  $RSP_{Rate\ Diff\ RMS}$ ,  $RSA_{R2\ Std}$ , and  $RSP_{Esp\ Time\ Median}$ . With these results, we can conclude that most of the features (about two-third) are robust to movement artifacts, but eight of them (about one-third) may be affected.

However, the Wilcoxon rank-sum test revealed a significant stochastic dominance in three features (i.e.,  $SKT_{Average\ Power}$ ,  $RSP_{Insp\ Time\ Mean}$ , and  $RSP_{Esp\ Time\ Median}$ ); while performing a within-group comparison; that is, a comparing among subjects who used only the gamepad. Similarly, the Wilcoxon rank-sum test revealed a significant stochastic dominance in seven features (i.e.,  $PP_{CSI}$ ,  $PP_{HF\ sband\ 5n}$ ,  $PA_{RMS}$ ,  $PRT_{Median}$ ,  $PTT_{M\ HF\ pond}$ ,  $RSP_{Insp\ Time\ Mean}$ , and  $RSP_{Rate\ Diff\ RMS}$ ), while performing a compari-

### Chapter 3. Cognitive Workload Detection Method

---

son among subjects who used only the FlyJacket setup. Therefore, being most of the significant dominance present in both within and between groups, it is difficult to conclude that movements are affecting the results, exception made for  $RSP_{Prd\ Mean}$  and  $RSA_{R2\ Std}$ .

However, the risk of biasing the results due to movement artifacts is minimized by the fact that 86.9% of the samples used to train the classifier comes from study protocol 1 (with the gamepad), in which the movements were minimal and limited to the thumbs. Moreover, all features, normalization coefficients, and regularization terms were also chosen based on data coming from study protocol 1 exclusively. Therefore, if any movement artifact is present, it should not significantly influence the classification.

#### 3.8.6 Emulated online cognitive workload monitoring

A visual representation of the emulated online cognitive workload monitoring of both UBM and SSM is shown in Fig. 3.10. Since the order of the tasks was randomized, I only reported the 76 samples of the sequences having consecutive transitions between B and F3M tasks. This analysis was based on the experiment performed with the gamepad (Study protocol 1, Trial 3). During the first 180 seconds, participants performed the B task, which had a low level of workload. For the last 180 seconds, participants perform the F3M task, which had a higher level of workload. The detection was done on the test set, where features were extracted from a 60 seconds sliding window and no overlap. Negative and positive scores denoted low and high workloads, respectively. A Wilcoxon rank-sum test with 76 samples indicated that the scores before and after 180 seconds were significantly different ( $p\text{-value} < 10^{-8}$ ).

Another interesting aspect that turns out from Fig. 3.10 is the contradictory difference between the averaged predicted scores of the UBM and SSM. As the SSM is performing better than the UBM, I would expect to see a bigger absolute value of the averaged score of the SSM than to the one of the UBM. However, the upper margin of the standard deviation of the predicted score reported in the interval between 60 and 180 seconds (Task B) and the lower margin in the interval between 240 and 360 seconds (Task F3M) seem to be similar for both UBM and SSM. This behaviour may be explained by the attempt of the SVM to choose the hyperplane that maximized the distance from it to the nearest data point on each side. Thus, as the SVM tends to maximize the margins, the performance of the SVM based SSM may be limited to a consistent but marginal improvement.

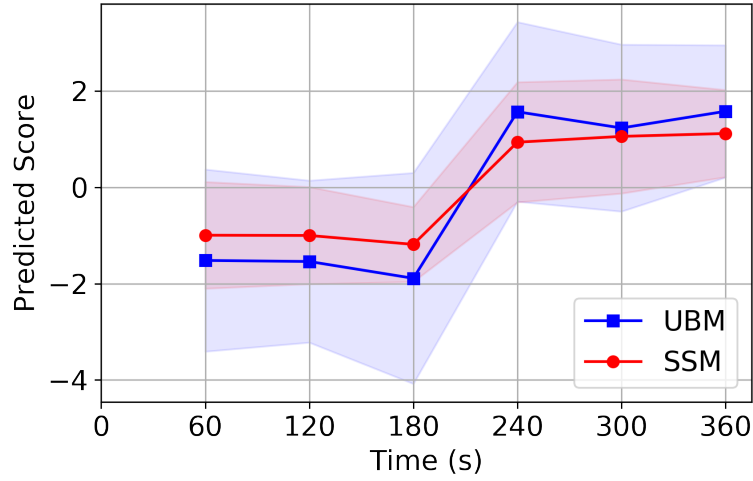


Figure 3.10 – Comparison of a Universal Background Model (UBM) vs. a Subject-Specific Model (SSM) on a simulated online cognitive workload monitoring (60 seconds sliding window and no overlap). A negative score denotes low workloads and a positive score denotes high workloads. Participants performed the B task for 180 seconds, followed by the F3M task for other 180 seconds.

By comparing Fig. 3.6 and Fig. 3.10, we can see that both perceived and detected cognitive workload were affected by a large variance. However, as shown in Fig. 3.10, such a variance was partially reduced by the use of the SSM, which contributed to better fit the physiological response of a single subject.

### 3.9 Conclusion

In this work, I proposed a reliable subject-specific machine learning algorithm for continuous cognitive-workload monitoring in search and rescue missions with drones. My multi-modal cognitive workload monitoring model combines the information of features extracted from physiological signals (i.e, RSP, ECG, PPG, and SKT) acquired in a non-invasive way.

I performed an exhaustive investigation involving up to 384 features and concluded that only 25 are required to get the highest classification accuracy. In addition, I explored different feature normalization techniques to reduce both subject and day inter-variability. My results showed that a combination of both day and subject

### Chapter 3. Cognitive Workload Detection Method

---

normalization improved the detection accuracy. Moreover, I introduced a novel SVM based learning method suitable for subject-specific optimizations. In particular, this model could distinguish between low and high cognitive workload with an average accuracy of 87.3%, on an unseen test set. Furthermore, I tested my model on 10 new participants using the flight simulator with an advanced controller, FlyJacket [153], and the proposed model reached an average accuracy of 91.2%.

In conclusion, my model can be used to monitor cognitive workload while driving a drone in real-life applications, with both traditional and advanced drone controllers, such as FlyJacket. Moreover, information of the cognitive workload perceived by the user, which can be reliably obtained with the proposed model in real time, can be used to improve shared-control systems [25], by modulating the human-robot interaction and dynamically adapt the level of assistance. An adaptive interaction between rescuers and drones is, in fact, a key factor to ensure an efficient execution of the missions.

## **4 Wearable and Multimodal Cognitive Workload Monitoring System**

### **4.1 Introduction**

Cognitive workload affects operators' performance, mainly in the case of high-risk and high-demanding situations [103, 115, 175]. Thus, an unobtrusive and real-time cognitive workload monitoring system could provide important feedback about the operator's state and resulting performance in decision-making instances. Therefore, such a system could adaptively support the operators according to their specific needs, improve their performance, and potentially decrease hazards.

In the last years, the assessment of cognitive workload monitoring from physiological signals has been addressed in different studies [23, 77, 146], providing numerous solutions that are suitable for real-time applications. A combined use of different physiological signals seems to be the most reliable approach, as due to its multi-dimensional nature, cognitive workload provokes multiple physiological reactions, which are visible in Respiratory Activity (RSP), Electrocardiogram (ECG), pulse wave through Photoplethysmogram (PPG), Skin Temperature (SKT) and Electrodermal Activity (EDA) [10, 30, 113]. Thanks to the recent explosion of commercial wearable devices, different studies have started to use them for the detection of cognitive workload [30, 62, 63, 114]. However, the use of such wearable devices is limited to data acquisition and storage, either onboard [30, 62] or on a cloud [63, 114], and the inference typically relies on a post processing approach. The reason is that onboard multimodal data processing is often not allowed by the manufacture or limited by the computational resources offered by these wearables. Therefore, to address the online cognitive workload monitoring, there is a need for new wearable embedded systems that can do both physiological signals acquisition and edge processing.

## **Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System**

---

The design of such a real-time wearable embedded system presents two main challenges. First, stringent processing and memory constraints of an embedded device can limit the performance and the execution of the algorithms, which should be completed within a specified time. Dealing with such constraints is indeed a challenge, as a multimodal cognitive workload monitoring system requires the acquisition and the processing of a large amount of data. Therefore, optimizations need to be considered to avoid delays and potentially large memory requirements, which can affect both functionality and performance.

Second, energy consumption should be maintained as low as possible to gain in size and battery life. Therefore, efficient and smart algorithms are needed to guarantee high performance in terms of energy consumption, accuracy, and robustness. In this regard, it should be possible to dynamically adapt the level of complexity by addressing the trade-offs between the required accuracy and the available energy of the system. A previous work targeting emotion recognition [58] motivates this assumption. The authors showed that a complex classifier typically reaches high detection performance, but it frequently lacks in energy efficiency. On the other hand, a simple classifier cannot always guarantee high detection performance, but it is energetically more efficient. Therefore, they presented an algorithm that continues increasing the model complexity until the system is confident about its detection. Then, it finally provides the result. That solution has the advantage of assuring the best detection performance all the time. However, it obliges to acquire and store all physiological signals required for eventual further processing, which is not optimal in terms of energy efficiency. Therefore, instead of switching to the next model until the system is confident about its current detection, I propose to provide the first obtained result, and if needed, to select a new model only for the next detection cycle.

To verify these hypotheses, in this chapter, I presented a new wearable embedded system design for online cognitive workload monitoring in search and rescue missions with drones. On the hardware side, it includes a multi-channel physiological signals acquisition and a low-power processing platform, which is suited for cognitive workload monitoring. On the software side, the proposed wearable embedded system includes novel energy-aware bio-signal processing and the application of a self-aware concept for scalable energy consumption. To this aim, different embedded machine learning algorithms and methods are used for online cognitive workload monitoring, exploiting the trade-offs between the required accuracy and the available energy of the system.

### Contributions of this Chapter

The realization of a wearable system for non-invasive online cognitive workload monitoring from physiological signals in Search and Rescue (SAR) missions with drones is the focus of the work in this chapter, which proposes the following contributions:

- To achieve non-invasive online cognitive workload monitoring and explore the trade-offs between the required accuracy and the energy consumption. To this purpose, I present a novel wearable system for physiological signal acquisition and online processing.
- The proposed multi-channel signal-acquisition and processing platform includes embedded algorithms and methods that have been validated for online monitoring of low and high levels of cognitive workload, achieving an accuracy of 75%.
- To optimize energy consumption, I propose a self-aware approach that exploits a scalable machine-learning method with different power-saving levels, which yields an increase of 78% of the battery life. The approach shows an acceptable accuracy loss with respect to the best universal background model presented in chapter 3 (i.e., from a theoretical 80.32% to 77.65%).

### Publications

This work yielded the following publications:

- F. Dell'Agnola, U. Pale, A. Arza, and D. Atienza. **Wearable Embedded System for Multimodal Cognitive Workload Monitoring**. *IEEE Transactions on Biomedical Circuits and Systems*, In preparation.
- V. Montesinos, F. Dell'Agnola, A. Arza, and D. Atienza. **Multi-Modal Acute Stress Recognition Using Off-the-Shelf Wearable Devices**. *41st International Engineering in Medicine and Biology Conference (EMBC)*, Berlin, Germany, 2019.

### Chapter Outline

The rest of the chapter is organized as follows. Section 4.2 gives an overview of the state of the art, covering the field of cognitive workload monitoring from physiologi-

cal signals, the wearable device that has been used for that purpose, and the main applications applying the concept of self-awareness. Section 4.3 gives an overview of the proposed system. Section 4.4 and Section 4.5 provide the details of both hardware and software design, respectively. Section 4.6 describes the validation steps applied to obtain the results reported in Section 4.7, together with a discussion of the validation approach. Finally, Section 4.8 presents the main conclusions of this chapter.

### **4.2 Cognitive Workload Monitoring Systems: State of the Art**

The mental effort required to perform a single or a combination of tasks is called cognitive workload [23]. Its monitoring has been addressed in different studies, allowing the development of different algorithms and methods. Compared to surveys [75, 148] and performance metrics [38, 100], the use of physiological signals is the most promising method to assess a continuous and unobtrusive monitoring [23]. Moreover, due to its multi-modal physiological response, many studies showed that the cognitive workload can be detected from a combination of different physiological signals, such as RSP, ECG, PPG, SKT, EDA, and Electroencephalogram (EEG) [23, 77, 146]. However, most of the proposed methods mainly rely on bulky setups [16, 186] used for data acquisition and on computers for offline post-processing [161, 187].

As reported in Table 4.1, in more recent studies, researchers started using portable devices [30, 62, 63] as well as wearable sensors [14, 65, 114]. However, most of these wearable devices are simple acquisition systems, such as Empatica E4 [46] and Everion® [17] wristbands, where data are collected and stored locally or on a cloud, for visualization or post-processing on a computer. Although this type of devices is interesting for data acquisition, the limited accessibility of onboard data processing and software personalization is often a weakness, especially in terms of energy consumption. In fact, the streaming and the storage of such a significant volume of data is energetically not efficient, in particular, if data storage is on a cloud instead of locally. Therefore, to reduce data streaming and consequent energy consumption, edge processing has to be considered, that is, the processing has to be done as much as possible on the sensor node.



Table 4.1 – Summary of the state-of-the-art studies on multimodal cognitive workload monitoring applications that make use of wearables. The table includes the performed task and measured physiological signals, such as Respiratory Activity (RSP), Electrocardiogram (ECG), Photoplethysmogram (PPG), Skin Temperature (SKT), Electrodermal Activity (EDA), and Electroencephalogram (EEG). Moreover, it shows the type of Microcontroller Unit (MCU), segmentation window, battery lifetime, and name of the platform used in the study.

Application	Processing	Physiological Signals	Seg. Win	Battery Lifetime	Sensor Platform
Stress detection [62]	PC	ECG, EDA	25s	-	Procomp5 Infiniti
Driving stress detection [30]	PC	RSP, ECG, EDA	100s	-	FlexComp
Stress monitoring [63]	PC	PPG, SKT, EDA	30-300s	-	Empatica E4
Stress in working environments [14]	PC	ECG, EDA, EEG	-	-	Zephyr, Shimmer, MindWave
Stress monitoring [65]	PC	PPG, EDA	20s	-	In-house portable device
Cognitive workload detection [114]	PC	RSP, ECG, PPG, SKT, EDA	60s	-	Shimmer, Empatica E4
Workload during manual labor [105]	PC	RSP, ECG, PPG, SKT, EDA	60s	-	Shimmer, Empatica E4
Workload detection in rescue missions (This work)	Cortex-M3	RSP, ECG, PPG, SKT	60s	22.6h (420 mAh)	In-house wearable device

### 4.2.1 Bio-signal acquisition with fitness devices

This type of approach, namely, edge processing, has been already applied in many fields, such as in fitness monitoring. Since the late 20th century, the development of new wearable fitness tracking devices has grown considerably. Initially, such wearable devices were simple pedometer or heart rate monitoring, and now, a variety of fitness parameters are directly computed in the new generation of wearables. Few examples are Suunto [172], Garmin [60], Polar [139], Fitbit [55], and Jawbone UP [84]. Although many wearable devices exist to monitor physical exercise and life style daily activity, they are mainly designed to target that specific application only. Thus, they are not easily adaptable to other applications, such as cognitive workload monitoring. More flexible solutions are instead offered by smartwatches (e.g. from Apple [9], Huawei [80], and Samsung [156]), which can be fully programmable by experts. However, these watches still lack sensors, such as RSP, ECG and SKT. Moreover, the access to raw data is often not allowed. In fact, as the measurements are particularly susceptible to degradation, signal conditioning is typically applied by hardware or by proprietary low-level algorithms, which are not accessible to the users. For instance, in such devices, the heart rate is traditionally computed as a windowed average of RR intervals in terms of beats per minute, which makes the measurement more robust to noise due to motion artifacts, environment, and user error [67]. However, such a conditioning step can remove important information of the signal and affect the computation of features such as Heart Rate Variability (HRV), as the variability is attenuated by averaging the RR intervals. Therefore, the extraction of reliable features with fitness-oriented devices is a problem.

### 4.2.2 Bio-signal acquisition with healthcare devices

The development of reliable wearable devices including both signal acquisition and online processing has been addressed as well in the field of healthcare, where detecting a disease with a wearable device is particularly interesting for two main reasons. First, it saves lives, as an early detection of critical abnormal situations allows a prompt reaction from the medical service. In this regard, many diseases has been addressed in the literature, such as the detection of myocardial infarction [163], paroxysmal atrial fibrillation [37], and epilepsy [58], with platforms such as INYU [169] and e-Glass [197]. The second reason focuses on reducing the costs of close and continuous medical supervision and care of particular diseases, such as sleep apnea [168], and episodes of arrhythmia [151]. The data analysis required by a continuous supervision of a patient suffering from such diseases is unsustainable for

## 4.2. Cognitive Workload Monitoring Systems: State of the Art

---

traditional healthcare delivery systems, as specialists have to process a lot of data to find sometimes only few abnormal localised episodes. Therefore, wearable devices have been widely used to offer large-scale and cost-effective solutions to this problem. The use of wearable and miniaturized sensors, able to continuously measure and report abnormal and unforeseen situations, can indeed provide the ubiquitous, long-term and even real-time monitoring required by the patients without hospitalization [152]. However, although this technology has transformed healthcare by allowing continuous monitoring of patients, the type and number of sensors are limited, and some physiological signals are not available on a single platform. For instance, INYU [169] obtains vital signs such as ECG, RSP and EDA, but not PPG, nor SKT. The SHIMMER baseboard allows physiological measurements such as ECG and EDA, but not RSP, PPG, nor SKT. To including these measurements, the systems require external expansion platforms, which will affect the communication rate and increase energy consumption. Therefore, there is a need to develop a new wearable device for physiological signals acquisition and processing that can be used for online cognitive workload monitoring.

### 4.2.3 Embedded systems and the concept of self-awareness

The lack of reliable and open platforms is also limiting the investigation and the development of new techniques targeting optimizations in terms of energy consumption. In this regard, different studies demonstrated the efficiency of application-oriented optimization methods, which address the trade-off between the required accuracy of the embedded application and the available energy of the system. These techniques rely on the concept of self-awareness, introduced to describe the knowledge about the system itself and the environment in which it operates [90, 94]. Self-awareness has three main phases that are translated into an Observe-Decide-Act (ODA) loop. First, observations are collected and evaluated to determine the possible actions to fulfill the system's objectives (e.g., reduce energy consumption and increase detection performance). Then, a decision is made based on the best use of the available actions and applied to meet the system's goals [79]. Self-awareness is indeed the key to designing new intelligent wearable systems and the topic has been the subject of different studies [31, 90, 94]. In some studies, researchers started considering this concept in embedded systems [6] and even in System on Chip (SoC) [44, 82], enabling correct functionality within desired constraints despite the presence of highly dynamic changes in both the application and the environment [7, 48, 142].

Recently, the concept of self-awareness has been applied to biomedical applications

## **Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System**

---

and, in particular, in the context of medical wearable systems such as seizure detection systems [58], where different conditions of both the patient and the environment can significantly affect the quality of the output of the system. In [173], the authors measured different parameters such as the system's confidence in order to improve the observation process of emotion recognition and get a high-quality description of the system from raw data. In [8], the authors adopted situation-awareness and personalized data (e.g., gender and age) to increase the accuracy of remote health monitoring from wearable sensors. Although a wide range of studies applied the concept of self-awareness, it has not been considered yet for cognitive workload monitoring.

Moreover, current methods mainly rely on the awareness of environmental situations, but recent approaches have started to put the focus on the classification performance [58, 129]. However, these recent methods mainly focus on the optimization of the machine-learning stage only, that is, without fully extending the concept to the acquisition layer. Extending the optimization to the acquisition layer is very important, as this layer is often responsible for the highest energy consumption of an embedded system [8, 151, 152]. Therefore, further investigations are needed.

In conclusion, previous studies in cognitive workload monitoring have mainly focused on offline approaches using bulky setups or multiple wearable systems. Therefore, a further effort has to be made to target online approaches and minimize the invasiveness of current systems in the context of cognitive workload monitoring. Moreover, in the literature of biomedical systems, the concept of self-awareness has started to be explored, but its use has never been considered for energy-efficient designs of systems monitoring cognitive workload. Therefore, the limitations of wearable systems, in terms of computational resources and energy consumption, need to be further investigated to exploit the concept of self-awareness, while also considering its impact on real-time cognitive workload detection based on physiological signals.

### **4.3 Self-Aware and Energy-Scalable System for Cognitive Workload Monitoring**

This section provides a high-level description of the proposed real-time wearable detection system. As an overview, Figure 4.1 shows a graphical representation of the main architectural blocks designed to detect cognitive workload with this system.

### 4.3. Self-Aware and Energy-Scalable System for Cognitive Workload Monitoring

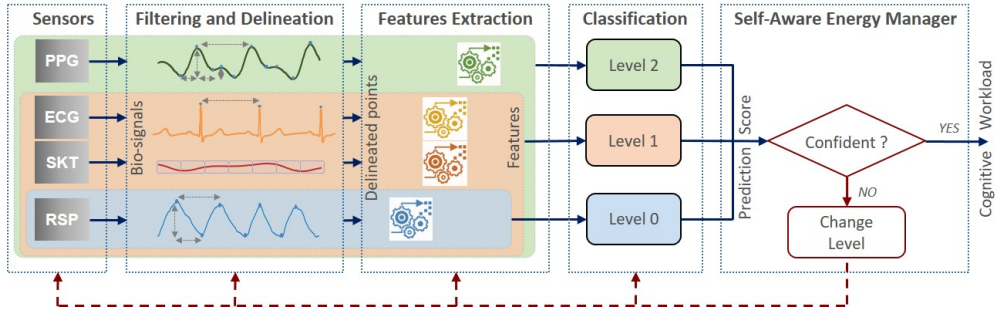


Figure 4.1 – Overview of the proposed self-aware and energy-scalable wearable system for online cognitive workload monitoring.

The main architecture of the applied embedded machine-learning technique includes few standard steps, namely, preprocessing (i.e., filtering and delineation), feature extraction, and classification. These steps are the ones presented in Chapter 3. However, for the classification, I used a linear Support Vector Machine (SVM) without any subject-specific optimization, that is, I used a single model for all subjects. The main reasons motivating this choice were the followings: This simple SVM achieves a good performance compared to other classifiers, it avoids overfitting due to its low capacity, and is particularly interesting for embedded implementations. Therefore, it is the proper candidate to verify the feasibility of an online cognitive workload detection with an embedded device. More details about the embedded machine-learning technique applied to target cognitive workload monitoring are given in Sec. 4.3.2.

To improve the energy efficiency of the system, and consequently, the battery life while maintaining high detection accuracy, I present a novel self-aware energy-management (Section 4.3.1). This energy-management is an application of the self-aware concept for scalable energy consumption. This application relies on an additional strength of the SVM, which provides a score indicating the likelihood that a label comes from a particular class. This score is used as a measure of self-awareness, that is, an estimation of the detection confidence. Then, based on the estimated confidence, the model selects the desired level of complexity to detect the cognitive workload exploiting the trade-offs between the required accuracy and the available energy of the system. For this particular case, I arbitrarily chose three levels of complexity, which resulted in the use of three SVM models, one for each level. Although the proposed self-aware approach relies on SVMs, this approach is not restricted to this particular type of classifier. However, it applies to any binary

classification technique that provides a non-binary score (e.g., likelihood, detection probability, and similarity score).

### 4.3.1 Self-Aware Energy Management

The original concept of self-awareness is usually applied to adapt the model to specific situations [90, 94]. Similarly, this concept finds an application in cognitive workload monitoring, as a wide range of studies showed the need for different combinations of physiological signals, which depend on the situation [23, 77, 146]. However, this is not the purpose of this work. Instead, it should be possible to dynamically adapt the level of the model complexity (e.g., by adjusting the number of input features) to improve energy efficiency while keeping the highest detection performance. A work targeting emotion recognition [129] motivates this assumption. There, the authors showed that a simple classifier, although energetically efficient, cannot always guarantee high detection performance. On the other hand, a complex classifier can achieve high detection performance, but it cannot provide high energy efficiency. Therefore, I addressed this trade-off by proposing an application of the self-aware concept for online energy-scalable embedded machine-learning algorithms in the field of cognitive workload monitoring.

### Proposed Self-Aware Energy-Scalable Model

The main idea behind this self-aware energy-management technique is to start providing a detection with the simplest machine-learning model, which shows an acceptable detection performance and minimum energy consumption. Then, based on how confident this detection is, the model is kept or replaced. Figure 4.2 shows the overall flow of a cycle of the proposed energy-scalable embedded machine-learning algorithm.

The work in [58] presents a similar approach used for epileptic seizures detection. In brief, that algorithm continues increasing the model complexity until the system is confident about its detection. Then, it finally provides the result. That solution has the advantage of assuring the best detection performance during the complete considered time. However, it obliges to acquire and store all physiological signals required for eventual further processing, which is not optimal in terms of energy efficiency. Therefore, instead of switching to the next model until the system is confident about its current detection, I propose to provide the first obtained result,

### 4.3. Self-Aware and Energy-Scalable System for Cognitive Workload Monitoring

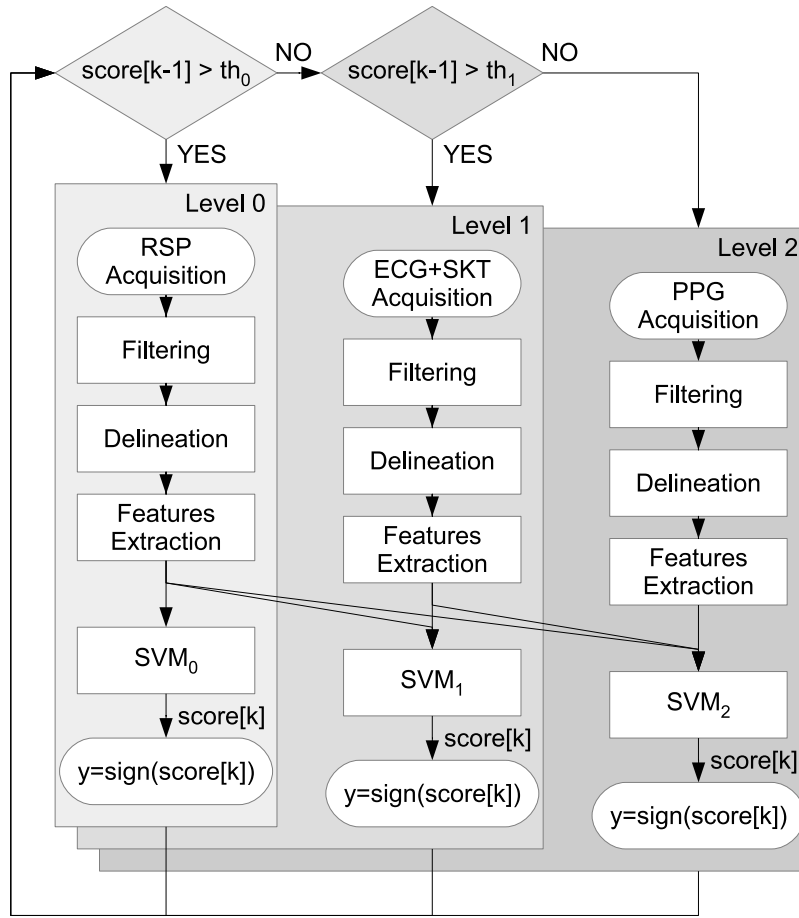


Figure 4.2 – Self-aware concept of energy-scalable embedded machine-learning algorithms and methods for online cognitive workload monitoring.

and if needed, to select a new model only for the next detection cycle.

Here, I extended the concept of self-awareness and energy scalable machine-learning system down to the acquisition level, proposing a solution energetically more efficient. On the other hand, the confidence detection of this solution relies on an evaluation delayed by one cycle. In other words, the confidence level depends on the score provided by the previous detection cycle (i.e., 60 seconds before the current one), which may differ from the confidence of the current detection. Therefore, the proposed solution may pay the price of the improvement in energy efficiency with a possible drop in detection performance.

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

---

To control the trade-off between detection performance and energy consumption, I had to evaluate both strengths and constraints of the proposed cognitive workload detection method. Moreover, to design this multi-level approach, I had to identify the possible levels that this energy-scalable machine-learning system should have. For this purpose, I first analyzed the sensitivity of the system to identify the importance of each physiological signal. Then, I evaluated the complexity of the algorithms running on the microcontroller. This evaluation aimed to identify the computational effort needed to process the acquired signals and its consequent energy consumption. Finally, I measured the average energy consumption of the system switching between the levels to estimate the battery lifetime.

### Levels of Energy-Scalable

To validate this approach, I proposed a case study with the following three levels of scaled-energy embedded machine-learning algorithms based on a linear SVM.

**Level 0** relies on features extracted from a single physiological signal, namely, the RSP signal. This choice was motivated by the results reported in Section 4.7, where I showed that the features of this signal were the most important and enough to get an accurate estimation of the cognitive workload. The use of a single sensor drastically reduced the energy consumption of the system while preserving an acceptable classification accuracy.

**Level 1** avails from the features extracted from RSP, ECG, and SKT, that is, all features except those from PPG. This choice was encouraged by the power-saving obtained by removing PPG. As reported in Section 4.7, both PPG signal acquisition and processing were the most expensive in terms of energy consumption compared to the other signals. Moreover, as reported in Section 4.7, the classification accuracy was lightly affected by the removal of the PPG features. Therefore, Level 1 offered an acceptable classification accuracy while it drastically reduced the energy consumption of the system.

**Level 2** benefits from the features extracted from all signals, which allowed accurate detection of different levels of cognitive workload. Although Level 2 is the most accurate in terms of classification accuracy, the energy consumption of this model is the highest, as both the number of active sensors and the computation effort required in Level 2 are the most important.

Although I proposed only three levels of energy scalability, the method also applies to



### 4.3. Self-Aware and Energy-Scalable System for Cognitive Workload Monitoring

a general case of multiple levels ( $L$ ) of machine-learning models, which can address different trade-offs between detection performance and energy consumption.

#### Proposed Self-Aware Manager

To switch between the three different levels, I used the likelihood score provided by the SVM. As shown in Figure 4.3, when the absolute value of this score is lower than a certain threshold (i.e., close to the separation hyperplane), the classification is considered inaccurate. Consequently, as previously explained, the system needs to select a model with a higher level of complexity for the next detection cycle. On the other hand, when the absolute value of the score is high, the classification is considered accurate, and a simple model can be used for the next detection.

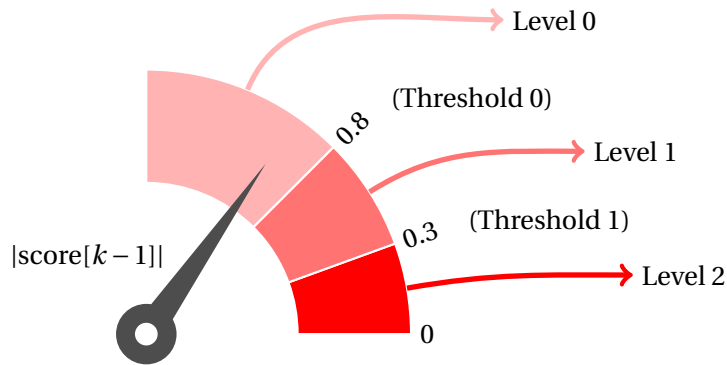


Figure 4.3 – Switching principle of the Self-Aware Manager

In this specific case of three levels of scalability, we need to define two thresholds. Intuitively, we could set these thresholds equal to 1, as it is the hard margin of SVM. However, to generalize the approach to cases in which the data are not linearly separable, I selected the thresholds from values in a range between 0 and the maximum absolute value of the resubstitution score; that is, the score derived by applying a model to the training data from which it was learned. The obtained thresholds are 0.3 and 0.8, which split Level 1 from Level 2 and Level 0 from Level 1, respectively. These thresholds were validated with a development set randomly extracted from the training data collected during the experiments described in Chapter 3.

### 4.3.2 Embedded Machine-Learning Algorithm

As shown in Algorithm 2, the training phase of the proposed classification procedure includes two main steps. The first step consists of training the classifiers of each level ( $l$ ), with the selected training set ( $\mathbf{X}_{tr}$ ) and the corresponding ground-truth labels ( $\mathbf{y}_{tr}$ ) (Lines 2 and 3). The second step aims to define the  $(L - 1)$  required thresholds needed to switch between the levels (Lines 4 to 15). To this aim, the algorithm iterates the possible threshold candidates ( $th$ ) in a range from 0 to  $th_{Max}$  with an increment of 0.05, where  $th_{Max}$  is the maximum absolute resubstitution score (Lines 5 to 15). Then, if the absolute value of the previous score is smaller than the threshold candidate, the score is calculated with the model of level  $l$ , otherwise, it is calculated with a more complex model, that is, the model of level  $l + 1$ . Next, the sign of the current score yields the target, that is, the estimated class associated with the current  $k$  observation of the development ( $dv$ ) set (Line 11). Lines 6 to 11 apply to all  $N_{dv}$  observations of the development set. Once all targets are available, the algorithm computes the accuracy (Line 12). Finally, if the accuracy reaches a plateau (Line 13), the algorithm assigns the obtained value to threshold  $l$  (Line 14), and it stops the iteration (Line 15). The algorithm repeats Lines 5 to 15 for each needed threshold.

---

**Algorithm 2** Self-Aware Classification - train phase
 

---

```

1: procedure TRAIN( $\mathbf{X}_{tr}, \mathbf{y}_{tr}, \mathbf{X}_{dv}, \mathbf{y}_{dv}, L, th_{Max}$ )
2:   for  $l \leftarrow 0$  to  $L - 1$  do
3:     SVM $_l$  Train( $\mathbf{X}_{tr}, \mathbf{y}_{tr}$ ) ▷ Train all SVMs
4:   for  $l \leftarrow 0$  to  $L - 2$  do ▷ Search for  $L - 1$  thresholds
5:     for  $th \leftarrow 0$  to  $th_{Max}$  do ▷ Iterate possible values for threshold  $l$ 
6:       for  $k \leftarrow 1$  to  $N_{dv}$  do ▷ Iterate data in development set
7:         if  $|score[k - 1]| > th$  then ▷ Check good confidence
8:           score $[k] \leftarrow$  SVM $_l$  Predict( $\mathbf{X}_{dv}[k]$ ) ▷ Compute score
9:         else
10:          score $[k] \leftarrow$  SVM $_{l+1}$  Predict( $\mathbf{X}_{dv}[k]$ ) ▷ Use complex model
11:          target $[k] \leftarrow$  sign(score $[k]$ ) ▷ Compute target
12:          accuracy $[th] \leftarrow \sum_{k=1}^{N_{dv}} (target[k] == \mathbf{y}_{dv}[k]) / N_{dv}$  ▷ Compute accuracy
13:          if  $|accuracy[th] - accuracy[th - 1]| < \xi$  then ▷ Accuracy plateau
14:            threshold $[l] \leftarrow th$  ▷ Store threshold  $l$ 
15:            break ▷ Stop searching for threshold  $l$ 
16:   return SVM, threshold
  
```

---

Once the energy-scaled models of each level are trained, and the thresholds indicating the confidence bounds are defined, the full self-aware machine-learning model

is ready to be used for inference. To this end, Algorithm 3 shows the classification procedure. The algorithm starts by selecting level 0 (Line 2) and checks if the absolute value of the previous score is higher than threshold 0 (Line 3). If it is not the case, it means that the previous inference was not confident enough; that is, the absolute value of the score was too close to 0. In this case, the algorithm repeats the process from Line 2 by selecting a higher level. If instead, the previous inference provided a high score, the algorithm uses the model corresponding to the selected level to process the observation  $\mathbf{X}_{te}[k]$  of the test set and calculate the score (Line 4). Finally, it exits the loop (Line 5), computes the target (Line 6), and returns the values of both score and target (Line 7).

---

**Algorithm 3** Self-Aware Classification - test phase

---

```

1: function TEST( $\mathbf{X}_{te}[k]$ , SVM,  $L$ , score[ $k-1$ ], threshold)
2:   for  $l \leftarrow 0$  to  $L-1$  do
3:     if  $|\text{score}[k-1]| > \text{threshold}[l]$  or  $l == L-1$  then           ▷ Check confidence
4:       score[ $k$ ]  $\leftarrow$  SVM $_l$  Predict( $\mathbf{X}_{te}[k]$ )                     ▷ Compute score
5:       break                                                         ▷ Stop iteration
6:   target[ $k$ ]  $\leftarrow$  sign(score[ $k$ ])                               ▷ Compute target
7:   return score[ $k$ ], target[ $k$ ]

```

---

#### 4.4 Multi-Sensor Hardware Design

To assess cognitive workload monitoring from physiological signals, I designed the wearable system presented in this section. This wearable platform has been introduced in Chapter 2, where I presented the section related to the control of a drone. In this section, I provide a description of the remaining components and functionalities, which are intended for cognitive workload monitoring.

Figure 4.4 shows the complete block diagram of the proposed wearable system, where I highlighted the principal components. An Ultra-Low Power (ULP) 32-bit Microcontroller Unit (MCU) Arm Cortex-M3 is the core of this system, and it is responsible for both online computing and power management. For the signal acquisition, the system connects a validated set of commercial off-the-shelf components, intended for the acquisition of different physiological signals. Moreover, an Inertial Measurement Unit (IMU), an additional module including a 3-axis accelerometer, gyroscope, and magnetometer, is intended for gesture recognition as introduced in chapter 2. However, the measurements of this IMU can serve as well for signal artifacts removal. Furthermore, the platform includes different communication

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

technologies allowing interconnections with external devices, such as a PC, tablets, smartphones, or even smartwatches. Finally, the platform includes a Micro SD Card socket for onboard data storage. Figure 4.5 shows the prototype of the proposed wearable system, and in particular a picture of both front ( 4.5a) and rear ( 4.5b) views.

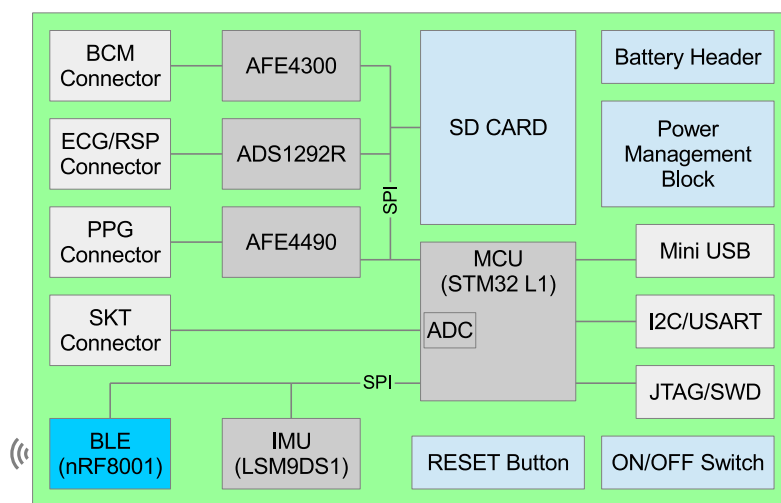


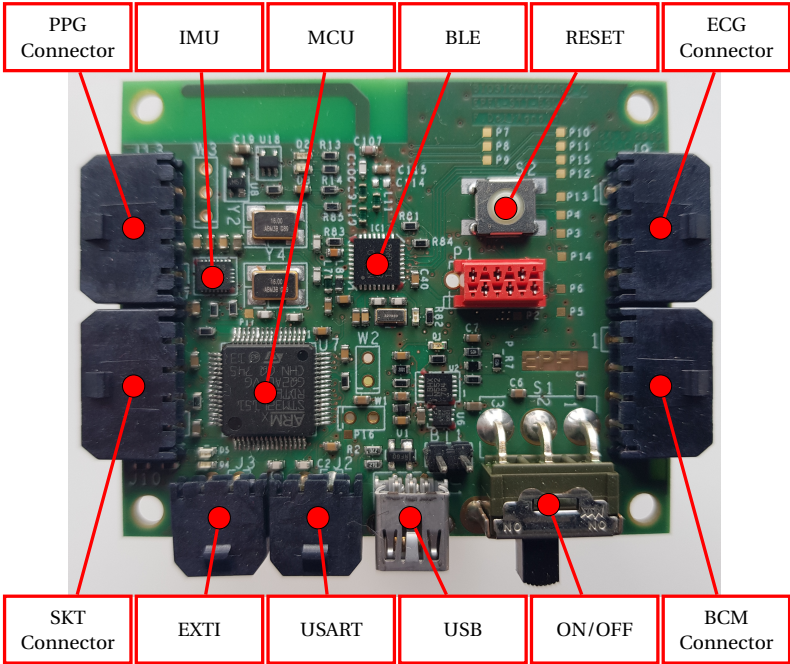
Figure 4.4 – Block diagram of the proposed wearable system.

#### 4.4.1 Micro-Controller Unit

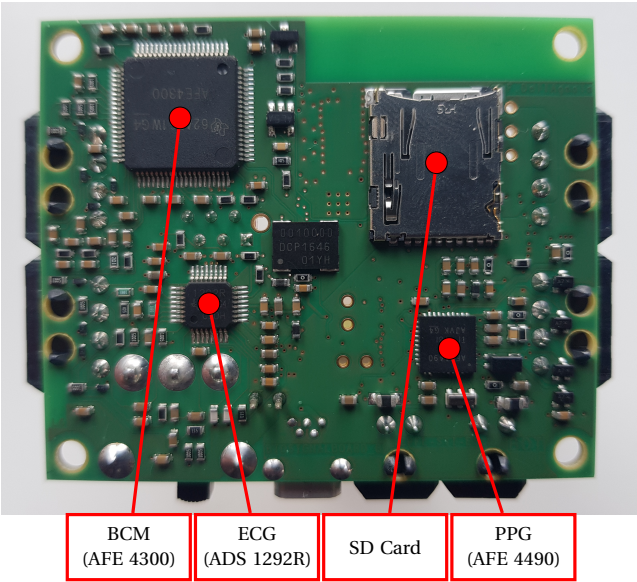
The core element of the proposed sensor node is a STM32L1 MCU from STMicroelectronics [166], which has been used in wireless sensors for medical sensing applications [169]. The STM32L1 is an ultra-low-power platform equipped with an Arm®Cortex®-M3 32-bit CPU from 32 kHz up to 32 MHz, 384 Kbytes of Flash memory, 48 Kbytes of RAM, 12 Kbytes of true EEPROM, and 128-byte backup register. Among all these characteristics, the crucial ones are the CPU and the memory, namely, the Flash and the RAM. This MCU offers a good compromise between computational capabilities, storage, and energy consumption. The use of a Cortex-M0 would allow a further reduction of energy consumption, but the computational resources would be insufficient for this application. On the other hand, a Cortex-M4 has much more resources than a Cortex-M3, but it suffers from elevated energy consumption.

The STM32L1 also includes analog peripherals, such as three operational amplifiers,

4.4. Multi-Sensor Hardware Design



(a) PCB front view.



(b) PCB back view.

Figure 4.5 – Prototype of the proposed wearable system.

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

a 21-channel 12-bit ADC 1 Msps, a 2-channel 12-bit DAC with output buffers, two ultra-low-power-comparators, which can be used for instance to monitor input voltage and battery level. Moreover, the STM32L1 includes 11 timers, namely, one 32-bit, eight 16-bit, and two watchdog timers to assure the synchronization of the application. Finally, the STM32L1 includes a 12-channel DMA controller and 12 peripheral communication interfaces, namely, one Universal Serial Bus (USB) 2.0, five Universal Synchronous and Asynchronous Receiver-Transmitters (USARTs), three Serial Peripheral Interfaces (SPIs), two I2Cs, and one SDIO interface. These interfaces, SPIs in particular, are used to connect additional on-board modules (i.e., sensors, communication, and storage modules) to the MCU.

### 4.4.2 Bio-Sensing Modules

For the acquisition of the physiological signals, different bio-sensing modules are connected to the MCU through an SPI. Table 4.2 summarizes the selected modules and their possible configurations. These modules include both voltage and optical sensors as follows.

Table 4.2 – Bio-sensors characteristics

Signal	Component	Sampling frequency	Range	Precision
ECG	ADS1292 [179]	8 kHz	$\pm 2.42$ V	24 bits
RSP	ADS1292 [179]	8 kHz	2-10 k $\Omega$	24 bits
PPG	AFE4490 [181]	5 kHz	$\pm 1.2$ V	22 bits
SKT	LMT70 [182]	1 MHz	20 – 42°	12 bits
BCM	AFE4300 [180]	860 Hz	0-2.5 k $\Omega$	16 bits

A Low-Power, 2-Channel, 24-Bit Analog Front-End is mounted for both ECG and RSP measurements. The signal acquisition rate can be up to 8 kHz, but a sampling frequency of 125 Hz is more than enough for the targeted application. The input range of  $\pm 2.42$  V and a precision of 24 bits allow to detect extremely weak signals ranging from 0.5 mV to 5.0 mV (such as ECG), which is affected by a DC component of up to  $\pm 300$  mV (resulting from the electrode-skin contact), and a common-mode component resulting from the potential between the electrodes and the ground (up to 1.5 V).

An additional module is connected for PPG and Pulse Oximetry, that is, the AFE4490

Analog Front-End from Texas Instruments [181]. This module comprises a receiver channel with a 22-bit Analog-to-Digital Converter (ADC) and a LED transmit section (with an integrated LED driver). The sampling frequency is up to 5 kHz and it is very configurable in terms of timings, which allows complete control of the acquisition of the signal.

Moreover, an analog temperature sensor (i.e. LMT70 from Texas Instruments [182]) is connected to the 12-bit ADC of the MCU for SKT measurements. The sensor has an accuracy of 0.05°C from 20°C to 42°C. The sampling frequency can be up to 1 MHz, which is the one of the ADC integrated in the MCU.

Finally, the system also includes an additional 16-bit Analog Front-End (i.e. AFE4300) [180] for body impedance measurements, which can be used to measure Body Composition Monitor (BCM), Impedance Cardiogram (ICG), EDA, and RSP. The sampling rate is up to 860 Hz.

##### 4.4.3 Communications

As explained in Section 2.3.3, Bluetooth Low Energy (BLE) connectivity is provided by nRF8001, a single-chip designed for low-power applications (current consumption of 12.7 mA for transmission, 14.6 mA for reception, 2 mA in idle state and 0.5  $\mu$ A in sleep mode). The BLE module is directly connected via SPI to the MCU, which controls the nRF8001 operating modes utilizing commands defined by the Application Controller Interface (ACI).

For accessing external devices, allowing data gathering from other sensors or communications with a PC, the device offers an auxiliary configurable interface, which can be either through an Inter-Integrated Circuit (I2C) bus or a USART.

The proposed platform also mounts a Micro SD Flash Socket supporting microSD cards with capacities up to 2 Gbytes. This option allows quick data collection and onboard storage without the need for streaming data over the radio to a remote device, which would be energetically inefficient.

For programming the MCU, there are two options. First, through a Serial Wire Debug (SWD) interface and the ST-LINK/V2 debugger and programmer. Alternatively, it is possible to program the MCU through USB. In this regard, the Device Firmware Upgrade (DFU) utility has to be used to manage the interaction with the STM32 system memory bootloader, running from the Flash of the MCU, thus allowing

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

internal memory programming. USB is also used to supply the device and charge the battery.

### 4.4.4 Power Management

A switch controls the power-on/off sequence of the device; that is, it enables the power regulators and releases the reset line of the MCU. If needed, the reset button can pull down again the reset line, causing a reinitialization of the MCU. To assure the low-power usage capabilities of the proposed sensor node, the power management of the proposed system involves the control of both the MCU and its peripherals. To this aim, and to achieve the best compromise between low energy consumption, short startup time, and available wakeup sources, the MCU offers different operational modes (see Table 4.3).

Table 4.3 – Power modes of the Microcontroller Unit (MCU) (from Run/active down to Standby mode), with the corresponding current consumption and wakeup time to Run mode.

Operational mode	Current consumption	Wakeup time
Run mode (from Flash)	230 $\mu\text{A}/\text{MHz}$	0 $\mu\text{s}$
Sleep mode (from Flash)	43 $\mu\text{A}/\text{MHz}$	0.4 $\mu\text{s}$
Low-power run mode	11 $\mu\text{A}$	3 $\mu\text{s}$
Low-power sleep mode	4.4 $\mu\text{A}$	46 $\mu\text{s}$
Stop mode with RTC	1.35 $\mu\text{A}$	<8 $\mu\text{s}$
Stop mode without RTC	0.475 $\mu\text{A}$	<8 $\mu\text{s}$
Standby mode with RTC	1.15 $\mu\text{A}$	58 $\mu\text{s}$
Standby mode without RTC	0.305 $\mu\text{A}$	58 $\mu\text{s}$

Run mode is the normal operating mode, where all the required resources of the MCU are active. The current consumption of the MCU in this mode can be down to 230  $\mu\text{A}/\text{MHz}$ , and about 10.5 mA with a code executed from the Flash at 32 MHz. In Sleep mode, only the Central Processing Unit (CPU) is stopped, while all resources of the MCU continue to operate and can wake up the CPU when an interrupt occurs. Sleep mode power consumption is about 2 mA at 32 MHz with all peripherals off. The MCU can be woken up in 0.4  $\mu\text{s}$  by any interrupt or event. Low-Power Run mode is achieved by reducing the clock frequency and the number of enabled resources.



#### 4.5. Real-Time Software Design and Implementation

---

Moreover, the internal regulator goes in low-power mode to minimize its operating current. Low-Power Sleep mode is achieved by entering Sleep mode with the internal voltage regulator in Low-power mode. In Low-power sleep mode, both the clock frequency and the number of enabled resources are limited. The system reverts to the run mode with the regulator on, when the wakeup sequence is triggered by an event or an interrupt. Stop mode achieves the lowest power consumption while retaining the Random Access Memory (RAM) and register contents and Real-Time Clock (RTC). To reach this mode, all clocks are stopped, exception made for the RTC, which can remain active. The internal voltage regulator is in low-power mode. The MCU can be woken up in 8  $\mu$ s by interrupts or events on specific external lines. Standby mode achieves the lowest power consumption. All clocks are stopped, apart from the RTC, which can continue being active. The internal voltage regulator is switched off so that the entire VCORE domain is powered off. After entering Standby mode, the RAM and register contents are lost except for registers in the Standby circuitry. The MCU exits Standby mode in 60  $\mu$ s when an external reset (NRST pin) or a rising edge on one of the three WKUP pins occurs. Otherwise, if the RTC is still active, the MCU can be woken up by an IWDG reset, an RTC alarm (A or B), an RTC tamper event, an RTC timestamp event, or an RTC Wakeup event.

Last but not least, the MCU is also responsible for the power management of the peripherals (e.g., sensors, SD card, BLE module). Table 4.4 shows the power modes of the different peripherals, with the corresponding current consumption and wakeup time. Indeed, lowering the duty cycle of the interaction between peripherals is an essential aspect to extend battery lifetime. Although this is not feasible with applications that require high-frequency data capture, the proposed application still has room for maneuver to enhance power management, where peripherals can be powered off by firmware when not in use. In this regard, the MCU provides features, such as timers and customized interrupt lines, that simplify the programming allowing the implementation of event-driven applications.

#### 4.5 Real-Time Software Design and Implementation

The SW architecture proposed for the implementation of the cognitive workload detection is shown in Fig. 4.1 of Sec. 4.3. The processing flow of this architecture includes five main layers, namely, signal acquisition, filtering, delineation, feature extraction, and classification. This flow applies to all signals (i.e., PPG, ECG, RSP, and SKT). To allow a future parallelization, the processing of each signal is managed

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

Table 4.4 – Power modes of the different peripherals, with the corresponding current consumption and wakeup time.

Front-End	Current consumption		Wakeup time
	(Active)	(Standby)	
ADS1292 [179]	0.223 mA	0.053 mA	10 ms
AFE4300 [180]	0.97 mA	0.1 mA	1 ms
AFE4490 [181]	0.1 mA + LED	0.1 mA	1000 ms
LMT70 [182]	9.2 $\mu$ A	50 nA	0.6 ms
nRF8001 [122]	12.7 mA <sup>(1)</sup>	1.6 mA	2-10 ms

(1) Current consumption while transmitting.

independently to each other. The objective is also to create an independent and modular implementation, providing a flexible solution that is suitable for applications different than cognitive workload detection. To this aim, the different layers of the software are distributed into separate routines, at least one for each layer and signal. The execution of these routines is activated by Interrupt Requests (IRQs), triggered either by sensors, starting the corresponding Interrupt Service Routine (ISR) of the acquisition layer, or by timers, which start the other ISRs. An overview of the IRQ sources (i.e., timers and interrupt lines) that trigger them is shown in Table 4.5, together with the corresponding activation frequency.

Table 4.5 – Distributed processing using interrupts and timers

Processing step	SKT	ECG	RSP	PPG
Acquisition	ADC1 (1 s)	EXTI2 (8 ms)	EXTI2 (0.2 s)	EXTI3 (15.625 ms)
Filtering	-	EXTI2 (8 ms)	TIM7 (3 s)	TIM7 (3 s)
Delineation	TIM10 (1 s)	TIM6 (1.75 s)	TIM7 (3 s)	TIM7 (3 s)
Feature extraction and classification	TIM5 (60 s)	TIM5 (60 s)	TIM5 (60 s)	TIM5 (60 s)

To balance system throughput versus interrupt latency, we assigned different Interrupt Priority Levels (IPLs). For instance, to avoid any loss of data, all IRQs of the acquisition layer need to be treated more quickly than others. Moreover, as the amount of processing in the acquisition layer is limited, it makes sense to assign a higher priority to that kind of interrupt. Therefore, the acquisition has the highest

## 4.5. Real-Time Software Design and Implementation

---

IPL than others. Then, to assure a real-time detection, a second level of priority is assigned to the layer executing both feature extraction and classification. Finally, a third level of priority is assigned to the middle layer, that is, the layer in charge of delineating the acquired signals.

### 4.5.1 Signal Acquisition

The first layer of the proposed system architecture is the signal acquisition. This layer represents the lowest level of the full architecture, that is, the interface with the sensors. In this layer, the samples of each signal are acquired, converted, and stored in a circular buffer for later use from the upper layers.

As previously introduced, the signal acquisition is handled by ISRs, which are triggered by IRQs coming from the corresponding sensors every time a sample is available, that is, at a predefined sampling rate. All the ISRs of the acquisition layer have equal priority, but the highest one compared to the rest of the system. This means that the system acquires any available samples regardless of what the system is currently processing. The only exception is made when the system is already acquiring another signal. In this case, the new acquisition will be executed right after the one in execution.

Once a sensor triggers its corresponding ISR, the MCU reads the input sample from the registers of the sensors. As the ADCs of the sensors have different precision (i.e., different numbers of bits), the system truncates the sample and keeps only the 16 Most Significant Bits (MSBs). This truncation is applied to standardize the precision and obtained a uniform input datatype. This standardization is applied to allow a multiple utilization of functions that are common for most of the signals. For instance, the function for computing the Power Spectral Density (PSD) is needed by RSP, ECG, and PPG. Therefore, the use of different datatype would require three different implementations of that function. Obviously, such a truncation causes a loss of information. However, there is a considerable gain in both processing time and memory usage.

Optimizing memory usage is indeed a must, as the acquisition layer saves a considerable amount of data. This storage involves the use of circular buffers, which sizes depend on length of the window and both input and output rates, respectively, the sampling frequency and the time required to process the stored data. Therefore, being the processing time quite important, there is a need for allocating circular

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

Table 4.6 – Selected sampling frequency, processing window length, and buffer size used at the acquisition layer for each signal.

Signal	Sampling frequency (Hz)	Window length (s)	Buffer size (bytes)
ECG	125	1.75	1000
PPG	64	3	1200
RSP	5	3	300
SKT	5	1	2

buffers with significant size to prevent data overwriting. Table 4.6 shows the selected sampling frequency, the processing window length, and the buffer size used at the acquisition layer for each signal. To determine the size of each buffer we multiply the sampling frequency with the window length, the number of bytes per sample (i.e., two, as the data type is an unsigned short), and an additional factor to avoid overwriting, that is, between two and three.

### 4.5.2 Filtering and Delineation

Once a desired sequence of samples is acquired, a preprocessing is applied. This process is managed by timers, which trigger the ISRs of the filtering and delineation layers, to respectively remove the noise and extract the points of interest (e.g., signal onsets and peaks) from the corresponding discrete-time signal.

To this purpose, different digital filters are applied. The ECG signal is processed using a relative energy filter as described in [126]. This filter is applied at the acquisition layer, as it can filter sample by sample. However, both RSP and PPG are filtered using optimized functions from the CMSIS-DSP library and the process applies every 3 seconds. In particular, the RSP signal is filtered using a 10-order bandpass Finite Impulse Response (FIR) filter with cutting frequencies at 0.03 and 0.5 Hz. Instead, the PPG signal is filtered using a 32-order bandpass FIR filter with cutting frequencies at 0.1 and 5 Hz. Filtered data is stored in circular buffers so that it is accessible by the delineation layer at any time. The size of these buffers is the same as the one reported in Table 4.6. The only exception is the case of the ECG signal, where each sample is directly filtered at the acquisition layer. Therefore, it requires only one buffer.

The delineation layer applies to detect relevant fiducial points from the different

## 4.5. Real-Time Software Design and Implementation

signals. For instance, the delineation applied to the ECG signal yields the R peaks; RSP delineation yields both peaks and valleys; and PPG delineation yields signal onsets, peaks, and the points with the maximum slope between onsets and peaks. As reported in Table 4.7, these fiducial points are the source of different physiological parameters.

The successive difference of the fiducial points of the ECG results in the RR-intervals. From RSP, inhalation and exhalation time ( $INH_{time}$  and  $EXH_{time}$ , respectively), as well as the respiration rate ( $RSP_{rate}$ ) and period ( $RSP_{period}$ ) are calculated. Parameters related to the PPG signal are Peak-to-Peak interval (PP), Pulse Rising Time (PRT), and Pulse Amplitude (PA). Figure 4.6 shows a graphical overview of these fiducial points and parameters extracted from the different signals.

Table 4.7 – Delineated points and bio-parameters for each physiological signal

Signal	Delineated points	bio-parameters
ECG	R-peaks	RR-intervals
PPG	Pulse onset, slope, and peak	PA, PP, and PRT
RSP	Pulse onset and peak	$INH_{time}$ , $EXH_{time}$ , and $RSP_{period}$

Finally, the physiological parameters are stored in circular buffers as well. Table 4.8 shows the size of these buffers, where the size of each buffer is obtained by multiply the maximum frequency with the number of parameters per period, the number of bytes per parameter (i.e., two, as the data type is again an unsigned short), and an additional factor to avoid overwriting, that is, 1.5.

Table 4.8 – Maximum points per minute, number of parameters per period, and buffer size used at the acquisition layer for each signal.

Signal	Maximum frequency	Number of parameters	Buffer size (bytes)
ECG	200 Beats per min.	1	600
PPG	200 Pulses per min.	3	1800
RSP	40 Breaths per min.	3	360
SKT	-	2	4

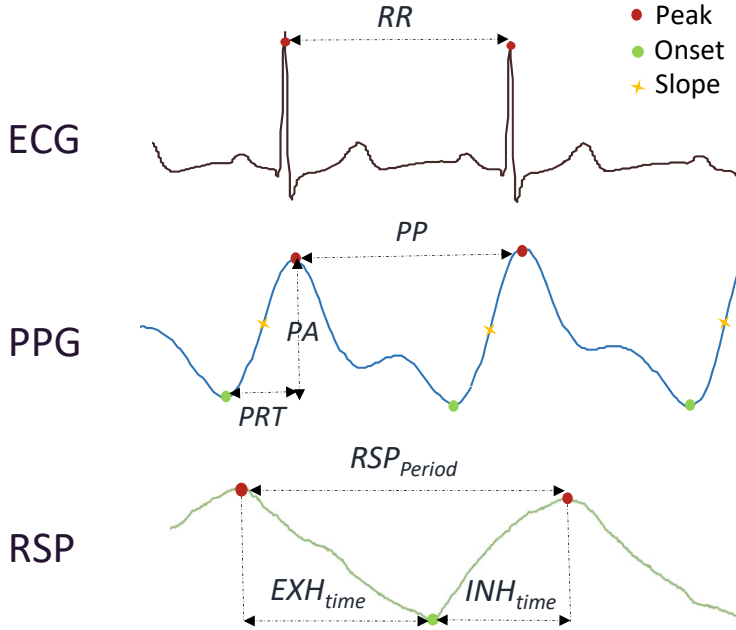


Figure 4.6 – Delineated points and parameters extracted from each physiological signal.

### 4.5.3 Feature Extraction and Classification

In the last layer, the physiological parameters are read from the respective buffers and used to compute the features needed to detect cognitive workload. This process extracts both time and frequency domain, including simple statistical features, such as mean, median, and Root Mean Square (RMS); as well as more complex ones, such as normalised power in specific frequency bands. Most of the functions used to compute these features have been ported to C from a Matlab version, which was implemented in the investigation presented in chapter 3. However, to compute the PSD, the approach involving a linear interpolation and the Fast Fourier Transform (FFT) with functions from the CMSIS-DSP library were preferred to the Lomb-Scargle periodogram implementation. The use of the CMSIS-DSP library resulted in a considerable gain in computational time.

Once all features are calculated using the parameters extracted from the physiological signals, a normalisation step is applied using pre-selected values. Finally, the classification is performed using the pre-trained SVM, leading to a workload classification every 60 seconds. For the computation of all features, the algorithm requires

---

## 4.5. Real-Time Software Design and Implementation

708 bytes, which are equally divided between physiological features, parameters used for the normalization, and weights of the SVM.

### 4.5.4 Synchronization

At the highest layer, TIM5 guarantees the synchronization required by the application, that is, the cognitive workload monitoring. The idea is the following. At the lowest layer, independent ISRs carry out the acquisition of the different signals and stores the sample in separate buffers. Then, at the middle level, other ISRs process the acquired signals to delineate the fiducial points, which are stored into additional buffers. Finally, the ISR at the highest layer marks the instant indicating the fiducial points that have to be processed, compute the features from these points, and classifies the cognitive workload. In this last step, two options of synchronization are possible.

The first option is that whenever TIM5 triggers an interrupt, the corresponding ISR compute all the featured and classifies the cognitive workload from the fiducial points delineated up to that moment. This option is the simplest one. Thus, it is the approach selected for validating the feasibility of classifying cognitive workload with a wearable device. As with most of the simple solutions, this one presents as well an inconvenience that is important to mention. Once TIM5 triggers an interrupt, the ISRs of the middle layer may not have finished with the delineation of the signals. These missing points let the application computing the cognitive workload from windows that, in the worst case, it may be shifted from 3 seconds for both PPG and RSP and from 1.75 seconds for ECG. Therefore, this solution can present a synchronization error of up to 5%. However, considering the slow dynamics of the signals, the introduced error can be neglected as a first approximation.

To avoid this error, I propose a second approach, which is the following. Once TIM5 triggers the interrupt, let's say at time  $t$ , the ISR of the highest layer should let the ISRs of the middle layer finishing the delineation of samples acquired up to that moment. In this case, all windows are synchronized with each other. However, as we need to wait for all the pending delineations to be completed, this approach may introduce a variable delay of up to 15% before providing the estimation of cognitive workload. Moreover, this approach is a bit more complex to handle because there is a need to mark the samples acquired up to time  $t$ , recognize the fiducial points that characterize the signals before that time, use them for the computation of all required features, classify the cognitive workload, and leave for the next cycle the fiducial

points that were delineated, but characterize the signals after time  $t$ . Therefore, this approach is left for future optimizations.

## 4.6 Setup and Validation Process

This section describes both the setup and the approach used to validate the proposed system. As shown in Fig. 4.7, the validation of the proposed platform includes both hardware (HW) and software (SW) validation. On the HW side, I mainly checked the signal acquisition, and in particular, the quality of the signals compared to a certified acquisition system from Biopac [16] used as a reference. Whereas, on the SW side, I checked the execution of the implemented algorithms at different layers. To this aim, I compared the embedded implementation with the original Matlab version running on a PC. Finally, at the end of this section, I described the validation of the proposed

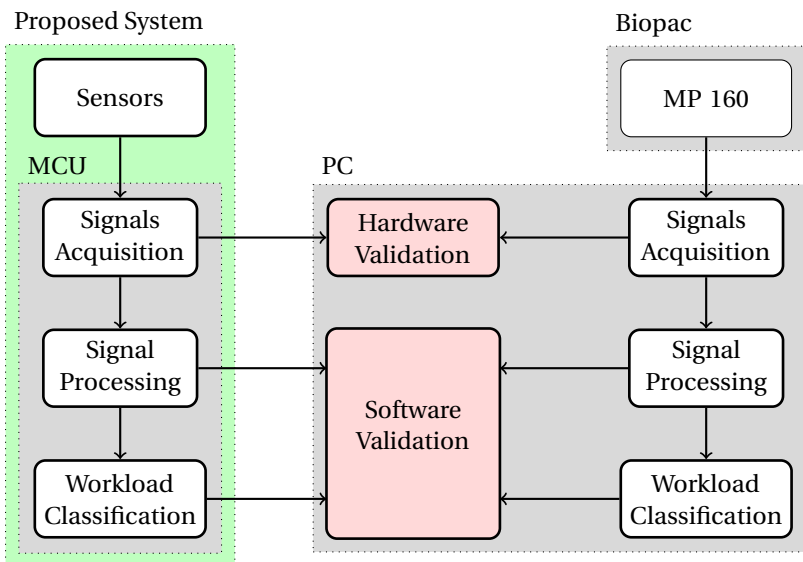


Figure 4.7 – Validation Method.

self-aware energy-scalable optimization.

### 4.6.1 Hardware Validation Process

The hardware validation is performed at different levels of abstraction from behavioral, down to layout. As a first step, we tested the functionalities of the different components embedded in the platform as well as all communication interfaces of



the system, namely, SPI, USART, and BLE. Second, we tested the acquisition layer, which is crucial for the performance of the final application.

Although checking all functionalities and communication is essential, in this context, it is worth focusing on the process applied to validate the acquisition layer. First of all, the use of validated off-the-shelf sensing components already provides the basis for a reliable signal acquisition. Moreover, a visual evaluation of the physiological signals acquired with the proposed platform provided satisfying results during different tests and demos. However, a formal validation of the proposed platform relies on the comparison with a certified acquisition system. To this aim, I measured the signal quality to quantitatively evaluate the acquisition layer.

As reported in [127, 128], the definition of signal quality differs depending on the application. There, the authors defined the signal quality in two ways:

- **Basic quality:** The main peaks (e.g., R-peaks of ECG and pulse peaks of PPG) are clearly identifiable. For instance, in the case of ECG and PPG, a basic quality is fundamental to guarantee a reliable Heart Rate (HR) extraction as well as the detection of some types of arrhythmias. Therefore, depending on the physiological signal, it should allow either HRV or Pulse Rate Variability (PRV) analysis.
- **Diagnostic quality:** Both main peaks and secondary waveforms are clearly identifiable. For ECG, P (if present), QRS, and T waveforms are clear. For PPG, the pulse wave waveforms are clean with systolic and diastolic waves visible. In this case, the signals allow clinical diagnosis.

In the context of this work, a basic quality is enough, that is, the ability to delineate the main peaks, which are the ones used to compute the features describing the physiological response generated by cognitive workloads. Therefore, I considered the ability to delineate these points of interest as a metric for evaluating the signal quality. To this aim, I delineated the physiological signals acquired with both the proposed system and Biopac with the method described in subsection 4.5.2. The full process was done offline on a computer and the result of this signal quality evaluation is shown in subsection 4.7.1.

### **4.6.2 Software Validation Process**

The functionality of the system is verified by emulating an online cognitive workload detection from a given input data stored into the SD Card. This testing data consists of physiological signals collected during the experiment described in chapter 3. For this validation, the signals were down-sampled and scaled to meet the characteristics of the acquisition layer.

For emulating the real-time acquisition, all sensors were active to let them trigger an interruption once a sample was ready. In this way, instead of reading the measurements from the sensors, we took the corresponding measurement from the SD Card. Then, the algorithm processes the data and provides a result of the estimated cognitive workload. This estimation, as well as the relevant values extracted from the layers characterizing the onboard processing, are stored into the SD Card for an offline comparison with the original Matlab implementation running on a PC. This approach was selected to provide a reliable evaluation of the implemented algorithms by taking advantage of the available dataset.

The analysis is divided into three different steps. First, we analyze the filtering and the delineation of the fiducial points extracted from RSP, ECG, and PPG. This analysis does not apply to SKT, as we directly extracted the features without a delineation. To this aim, we evaluated the number of matched fiducial points detected on board, and those detected offline with the Matlab implementation. The quality of the delineation is evaluated with sensitivity, predictivity, and mean error as the distance between matched points. Second, we validated the features extraction layer, where features were calculated from a 60-seconds window of parameters computed from the fiducial points. In this case, we reported the measurements of the relative error and the Pearson's correlation coefficient for assessment of feature changes. Finally, at the application level, we evaluated the full system by comparing the cognitive workload prediction computed in real-time by our system versus the PC based analysis.

### **4.6.3 Validation of the Self-Aware and Energy-Scalable Method**

The multi-level approach for energy scalability and self-awareness allows controlling the trade-off between detection performance and energy consumption according to the constraints of the proposed application, namely, online cognitive workload detection. To evaluate the benefit of optimizing this trade-off, we first determined the

complexity of the embedded algorithms. To this aim, the computation complexity was determined by measuring the time required to process each physiological signal, that is, the time required for both filtering and delineation, as well as the time required to extract both time and frequency domain features.

Second, I analyzed the sensitivity of the system by considering different configurations. To this aim, I used several SVMs, where each model has as input a particular subset of features extracted from either all or only a limited amount of physiological signals. This analysis aimed to identify the importance of each physiological signal and quantified the impact of using a scaled model. The accuracy achieved from these models was the metric used to characterize the signal importance. These results, combined with the computational complexity, yielded the definition of the energy-scalable levels of operation, as introduced in Section 4.3.1.

Third, I measured the energy consumption of the system considering the configurations required from each energy-scalable level. In particular, I first measured the total supply current of the system, while all components were in a low-power mode. Then, I measured again the current while switching on and off the main components, namely, MCU, RSP/ECG sensor, PPG sensor, and BLE module. With these measurements, we could estimate the supply current required for each level of operation.

Finally, I computed the theoretical maximum accuracy of the system and its estimated battery life, while considering the benefit of the proposed energy-scalable machine-learning method.

## **4.7 Results**

In this section, I provide an evaluation of the proposed system. First, I reported the results of the hardware validation, which aimed to evaluate the quality of the hardware described in Section 4.4, and in particular the acquisition layer. Second, I reported the results of the software integration, including an evaluation of each layer of the software described in Section 4.5. These results came from an emulation of an online cognitive workload monitoring with data stored in the SD Card. This approach was chosen to have the same input data for both the embedded implementation and the reference implementation running on a PC. Then, I presented a summary of the computation complexity, followed by an evaluation of the sensitivity of the system. The analysis of both computation complexity and system sensitivity was crucial and

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

validated the choice of the selected energy scalable operation levels described in Section 4.3. Right after, I reported the energy consumption of each level, obtained by measuring the contribution of each function individually. Finally, I showed the benefits of the proposed self-aware energy-scalable machine-learning model. In this regard, I presented the theoretical detection performance as well as an estimation of the gain in battery life.

### 4.7.1 Multi-Channel Acquisition Platform: Hardware Validation

One of the principal aspects of the hardware validation is the evaluation of the acquisition layer, which is responsible for acquiring the physiological signals that are necessary for cognitive workload detection. To validate this layer, I compared the signals acquired through the proposed system with those obtained from a validated acquisition system from Biopac. As explained in Section 4.6.1, the metric used for this comparison is the capability to detect the main fiducial points. In particular, Table 4.9 reports the results of this ability (evaluated over 23 segmented windows of 60 seconds), which was evaluated with sensitivity, positive predictive value, mean error (distance in seconds between the peaks detected from both signals), and standard deviation of this distance.

Table 4.9 – Evaluation of the fiducial points delineated from the physiological signal acquired with the proposed system. The metrics are sensitivity, positive predictive value, mean error, and standard deviation of the mean error.

Delineated points	Sensitivity (p.u.)	Predictivity (p.u.)	Mean error (sec)	Standard dev. (sec)
ECG R-peaks	$0.964 \pm 0.087$	$0.976 \pm 0.036$	$0.012 \pm 0.004$	$0.003 \pm 0.005$
PPG peaks	$0.848 \pm 0.129$	$0.859 \pm 0.112$	$0.041 \pm 0.037$	$0.025 \pm 0.033$
PPG onsets	$0.853 \pm 0.129$	$0.865 \pm 0.113$	$0.040 \pm 0.037$	$0.024 \pm 0.033$
RSP peaks	$0.771 \pm 0.222$	$0.931 \pm 0.151$	$0.525 \pm 0.334$	$0.320 \pm 0.226$
RSP onsets	$0.787 \pm 0.194$	$0.940 \pm 0.117$	$0.518 \pm 0.333$	$0.327 \pm 0.187$

The results are promising. The quality of ECG is enough for this application, as the ability to detect the fiducial points is essentially equivalent in both cases. The quality of PPG accuses a little degradation, but in principle, it should not affect the final classification more than 1-2%. This estimation was obtained by perturbing the inputs of the classifier with errors of such amplitude, and the classifier showed a minimal

variation of the result. Regarding RSP, we can draw a similar conclusion. Here, both sensitivity and mean error indicates some discrepancies in the signals. These discrepancies may result from the fact that RSP was measured from the variation of the thoracic impedance, while with Biopac, we used a strap stretched around the chest to capture both expansion and contraction of the rib cage.

#### 4.7.2 Online Cognitive Workload Monitoring: Software Validation

As explained in subsection 4.6.2, the first step of the software validation consists of an evaluation of the delineation layer, which is responsible for detecting the fiducial points of the physiological signals. To validate this layer, I compared the number of onboard detected peaks that match those detected with the Matlab implementation running on a PC. In principle, there should be no difference, as the method implemented in the MCU is the same as the one implemented in Matlab. However, a lack of precision is expected due to the limited computational resources of the MCU, such as the absence of a floating-point unit, as well as the need to use 16-bit operations instead of 32-bit to mitigate the limited storage capabilities.

Table 4.10 reports the results evaluated over 87 segmented windows of 60 seconds, including sensitivity, positive predictive value, mean error (distance in seconds between the peaks detected with the embedded implementation and the reference Matlab implementation running on a PC), and standard deviation of this error. The results are promising, the mean error is less than 0.14, while sensitivity and predictivity are greater than 84% and 85%, respectively. Considering that these delineated points are used to compute features over a 60-seconds window, which includes many of them, such a little error seems not affecting the final result.

The second validation step involves the feature extraction layer. In this regard, Figure 4.8 shows a box-plot representing a distribution of the overall feature similarities. In particular, it shows the measurements of the relative distance between the reference implementation and the Spearman's correlation coefficient. The results presented here quantify the quality of the features extracted from the fiducial points every 60 seconds. Ideally, we should have a relative distance close to 0 and a correlation coefficient close to 1. However, a quick test revealed that the classification accuracy accuses only 1-2% points of reduction associated with an upward 5% shift in the mean of the input parameter distribution. Therefore, even if some features do not match exactly the values from the Matlab implementation, there is an acceptable 5% margin of error.

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

Table 4.10 – Evaluation of the fiducial points delineated from each physiological signal. The metrics are sensitivity, positive predictive value, mean error, and standard deviation of the mean error.

Delineated points	Sensitivity (p.u.)	Predictivity (p.u.)	Mean error (sec)	Standard dev. (sec)
ECG R-peaks	$0.844 \pm 0.221$	$0.855 \pm 0.241$	$0.066 \pm 0.029$	$0.039 \pm 0.012$
PPG peaks	$0.844 \pm 0.134$	$0.897 \pm 0.176$	$0.009 \pm 0.015$	$0.006 \pm 0.005$
PPG onsets	$0.851 \pm 0.109$	$0.908 \pm 0.156$	$0.012 \pm 0.015$	$0.009 \pm 0.008$
RSP peaks	$0.898 \pm 0.123$	$0.916 \pm 0.121$	$0.093 \pm 0.056$	$0.065 \pm 0.029$
RSP onsets	$0.854 \pm 0.155$	$0.921 \pm 0.154$	$0.139 \pm 0.072$	$0.100 \pm 0.039$

In general, most of the features have high similarity and correlation values, such as  $PP_{\text{Median}}$ ,  $PA_{\text{RMS}}$ ,  $PRT_{\text{Median}}$ ,  $RSP_{\text{Rate Mean}}$ ,  $SKT_{\text{Gradient}}$ , and  $SKT_{\text{Power}}$ , while others are less similar, such as  $RR_{\text{HF Band 3n}}$ ,  $PP_{\text{CSI}}$ ,  $PP_{\text{HF Band 5n}}$ ,  $PA_{\text{Lorenz L}}$ ,  $RSP_{\text{PSD1n}}$ , and  $RSP_{\text{PSD3n}}$ . From these listed features, we can see that the lack of similarities comes from the frequency domain features, which are the most complex to compute. Once again the limited computational resources of the embedded system, such as the absence of double-precision and a floating-point unit, affect the exactness of the feature computation. Although the feature extraction implementation has a non-negligible margin of improvement, the errors introduced at this layer do not seem to significantly affect the results of the classification. Therefore, these results are sufficient, even if I was expecting a much better quality of the features.

Finally, the last validation step applies to the classification layer, which provides an estimation of the cognitive workload touching a human. The attained results show that the embedded implementation of the model used for Level 2 (i.e., the model using all available features) achieves an accuracy of 75%, which is 5 percentage points lower than the 80.32% obtained from the reference implementation running on a PC. As I already mentioned, we have to consider that compared to PCs, embedded systems have limited processing and storage resources. These limitations affect the computation of various features, which are in part responsible for this drop in performance.

The feature normalization step is responsible as well, where a drastic truncation of the coefficients used to normalize the features can yield a reduction of up to 2 percentage points of the accuracy. From these results, we can conclude that,

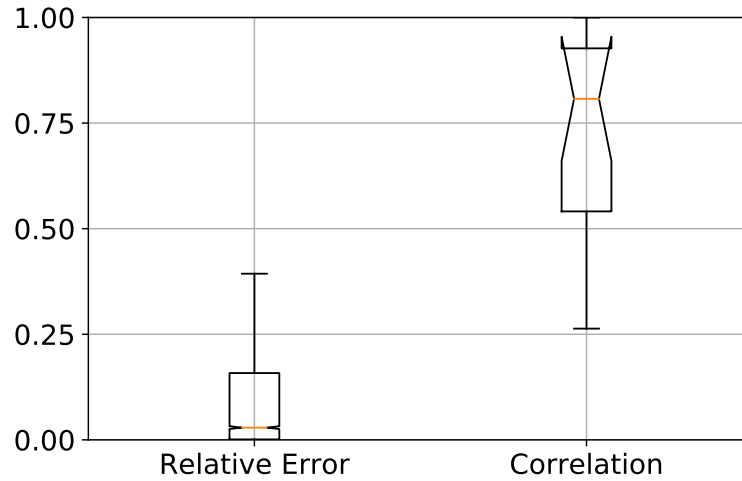


Figure 4.8 – Feature extraction evaluation. The metrics are the relative error related to the reference implementation and the Spearman's correlation coefficient.

in terms of accuracy, the impact of the errors produced by the feature extraction layer is comparable to the ones generated by feature normalization. However, as the computation effort of the normalization layer is almost negligible, compared to the processing required by feature extraction, the impact of a little error at this normalization level is very high. Therefore, we can conclude that the normalization requires high precision, while feature extraction can benefit from some constraints relaxation.

From these results, I can draw the first conclusion by saying that the embedded implementation of all different used functions to process all the signals and estimate cognitive workload works. Therefore, the proposed embedded solution is suitable for cognitive workload monitoring. Now, with this result in mind, we can evaluate the individual computational cost of all the implemented functions.

### 4.7.3 Evaluation of Computation Complexity

The computation complexity of the implemented functions used to extract the different set of features is presented here. In this regard, Table 4.11 reports the time required by the MCU to process a 60-second segmentation window of data. The measurement are divided per signal and per processing layer, namely, preprocessing (i.e.,

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

filtering and delineation) and feature extraction, in both time and frequency domain. Finally, these features are processed by the final layer, that is, the classification or inference layer, which provides an estimation of the cognitive workload level in less than 0.04 ms. The first conclusion we can draw from these results is that since the

Table 4.11 – Computation time (ms) of a 60-second signal window processing.

Signals	Preprocessing	Features Extraction		Total
	(Filtering + Delineation)	(Time domain)	(Freq. domain)	
SKT	-	0.325	-	0.325
ECG	130.580	24.666	9795.080	9950.326
PPG	134.471	5.189	39199.017	39338.677
RSP	6.580	1.506	2280.228	2288.314
				51577.642

required processing time does not exceed 60 seconds, all computations can be done on a low-power MCU Arm Cortex-M3 meeting the real-time constraints needed to provide a cognitive workload estimation every 60 seconds.

Moreover, we also observe that the computation of frequency-domain features is very high. Such a significant computation effort is due to the need for computing a Lomb-Scargle PSD estimation of a successive sequence of delineated points. This applies to the case of ECG and PPG, where PSD is computed once for each sequence of RR-intervals extracted from ECG and four times for the different series of delineated points extracted from PPG. For the case of RSP, the computation is faster, as the spectrum is computed only once and directly from the filtered signal with a normal FFT. The main problem here is that as a first attempt, to compute all required spectra and to make the system working, we used existing and standard functions, which are not at all optimized.

The optimization of the approach applied to calculate the PSDs is in the process. In this regard, we expect to have a drastic reduction in computation, that is, up to 200 times faster than the current implementation. As proof of this claim, a preliminary result showed that the use of functions from the CMSIS-DSP library drastically reduced the computation time of the RSP spectrum, where an optimized function computed its FFT and all related features in less than 10 ms. Of course, the computation of the PSD of both ECG and PPG is trickier. In this case, the required spectra come from an unevenly-spaced sequence of samples (e.g., RR-intervals), and therefore, the com-



putation requires a different approach. However, although the frequency-domain processing of both ECG and PPG is a bit more complex than the processing of RSP, we expect a comparable improvement, by interpolating the unevenly-spaced samples to allow the use of an optimized FFT for these signals as well. Although I expect a drastic reduction in the computation effort of the frequency-domain features, still it will not be lower than the computation cost of the time-domain features. Therefore, we could expect that by removing the frequency-domain features, we could save in computation, and consequently, also in energy consumption.

By checking again the results reported in Table 4.11, we can also see that features extracted from PPG are the ones requiring the most significant computational effort, that is, 65.5% of the available processing time (i.e., 60 seconds). It follows the processing of ECG and RSP, which require 16.5% and 3.8%, respectively. The processing of SKT is instead negligible. Therefore, we can also conclude that, if we want to gain in energy efficiency, PPG is a possible candidate to be discarded.

To summarize, a horizontal evaluation of the results reported in Table 4.11 shows that, in terms of computational effort, frequency-domain features are a weakness. Moreover, a vertical evaluation shows that PPG suffers from a significant computational complexity. Therefore, if we only examine the computational effort, I could suggest discarding the use of frequency-domain features to save energy consumption. However, if we consider the expected improvements in computation, the impact of removing these features will be minor and limited to the processing level. By removing frequency-domain features, it indeed only affects the computation and does not allow to power-off the sensors, as they still have to remain active to acquire the signals needed to compute the time-domain features. Therefore, I do not consider this option in this work. However, it could be considered to build an additional intermediate energy-scaled level. Alternatively, by adopting the approach of removing an unnecessary signal, we can extend the concept of energy-scalable machine-learning down to the acquisition layer. Therefore, we can benefit from both processing reduction and optimized use of sensors.

#### 4.7.4 Evaluation of System Sensitivity and Energy-Scalable Levels

Although the evaluation of the computation complexity relies on measurements obtained from the embedded implementation, the estimation of the system sensitivity is the result of an analysis using the method and models implemented on a PC. This analysis aims first, to determine both the most and less discriminant signals, and

# Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

then, define the energy-scalable levels by addressing the trade-off between the contribution of the features from different physiological signals and their computation complexity.

In this regard, Table 4.12 reports both the theoretical classification accuracy (values from the Matlab implementation) and the computational costs (measurements of the algorithms running on the MCU) of different reduced models, namely, models designed to only use a particular set of physiological signals. From the upper part of this table, we can see a considerable drop in accuracy (down to 83.51% in cross validation), if the model cannot benefit from RSP. In contrast, by removing ECG, SKT, or PPG, the accuracy drops down only to 87.44%, 88.18%, and 86.92%, respectively. Two factors drive these drops, specifically, the quantity and the quality of the features. By removing the RSP signal, we first affect the quantity, as we almost remove half of the features. However, from the lower part of the table, we can see that the drop is less significant if we exclude the other half of the features, that is, by keeping only the ten features from RSP. In this case, the accuracy is 84.02%, which is higher than the 83.51% obtained with 13 features from PPG, ECG, and SKT. Thus, by removing the RSP signal, we also affect the quality of the input set of features. Therefore, we can conclude that RSP is the most relevant physiological signal compared to PPG, ECG, and SKT all together. This trend is confirmed in the test set.

Table 4.12 – Accuracy and computational costs for different scaled models.

Active signals				Features	Accuracy (%)		Computation (%)
					CV	Test	
RSP	PPG	ECG	SKT	22	88.33	80.32	85.96 <sup>(L2)</sup>
RSP	PPG	ECG	-	20	88.18	72.87	85.96
RSP	PPG	-	SKT	19	87.44	77.66	69.38
RSP	-	ECG	SKT	14	86.92	68.09	20.40 <sup>(L1)</sup>
-	PPG	ECG	SKT	13	83.51	58.51	82.15
RSP	-	-	-	9	84.02	62.23	3.81 <sup>(L0)</sup>
-	PPG	-	-	8	72.45	53.19	65.56
-	-	ECG	-	3	77.30	54.26	16.58
-	-	-	SKT	2	71.65	50.53	<0.01

(L0), (L1), and (L2) indicate the signals chosen for Level 0, 1, and 2, respectively.

Table 4.12 reports as well the computational effort relative to the available processing

time, that is, the percentage of the time the system has to be active to compute an estimation of the cognitive workload. The computational effort of each scenario is computed by summing up the corresponding values of Table 4.11 divided by 60 seconds. These results show that by using only one signal, that is, RSP, we can ensure a nonrandom cognitive workload detection with a very minimal computational effort. Once again, this is the reason why I chose this configuration for the model of Level 0.

Moreover, Table 4.12 shows that, by adding the contribution of ECG and SKT on top of RSP, it is possible to improve the accuracy without increasing too much the computational effort. This configuration characterizes the model selected for Level 1. This choice benefits from the fact that SKT acquisition has a negligible cost and its use shows an increase in classification accuracy. Moreover, both RSP and ECG comes from the same sensor, which means that while acquiring RSP, we get the measurements of ECG almost for free. Therefore, with the model selected for Level 1, we gain in accuracy by paying only in computational effort.

Finally, the model based on all physiological signals is the one selected for Level 2, as it reaches the highest performance, at least in terms of classification accuracy. The computational effort of this model is instead very high, but this is the price to pay to attain such accuracy. However, based on the original hypothesis, the idea is to limit the use of such a complex model only when strictly necessary. To quantify this need for complexity, I show, in the next section, an evaluation of the use of these three proposed levels based on the dataset collected during the experiment presented in Chapter 3. However, first, let's see how much is the current consumption of each level.

#### **4.7.5 Evaluation of the Energy-Scalable Machine-Learning Method**

Each level has its acquisition configuration and computation complexity, which results in the different current consumptions, as shown in Table 4.13. These values are computed from a weighted average of the current consumption measured from each module (i.e., sensors, MCU, BLE, etc.), considering the time while they are either active or in low-power mode.

It is worth to mention that the final active time of the MCU also includes a switching period of 0.3%, which comes from the fact that the MCU needs some time to wake up from or to enter in low-power mode. For the application presented in this work, I

## Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System

Table 4.13 – Average current consumption of the different components operating under the proposed energy-scalable levels. In bracket the values while always acquiring all signals.

Module	Current consumption (mA)		
	Level 0	Level 1	Level 2
MCU	14.3	16.1	23.5
RSP/ECG	1.0	1.0	1.0
PPG	0.4 (8.4)	0.4 (8.4)	8.4
Bluetooth	0.014	0.014	0.014
Total	15.7 (23.7)	17.5 (25.5)	32.9

considered two low-power modes, namely, Sleep mode and Stop mode, which have a wake-up time of  $0.4 \mu s$  and up to  $8 \mu s$ , respectively. Thus, considering the wake-up sources and frequencies presented in Table 4.5, the MCU respectively spends 10 ms and 184 ms switching from/to Sleep and Stop modes to/from Active mode, every minute. Therefore, as the total time of activity (i.e., computation and switching time) does not exceed 100%, both modes are suitable for this application. However, due to its energy efficiency, Stop mode is preferred, as it allows a higher increase in battery life.

A further gain in battery life is also achieved by extending the concept of self-awareness and energy-scalable machine-learning method down to the acquisition level. As proof of this claim, Figure 4.9 shows a comparison of the weighted current consumption for each level, while always acquiring the full set of signals, and while optimizing the acquisition by sampling only the required signals. In the case of a full signal acquisition, the reduction of the current consumption is limited to the processing stage. However, by optimizing the acquisition, we can further reduce the current consumption by 24 percentage points. In the case of full signal acquisition, the reduction of the current consumption is limited to the processing stage. However, by optimizing the signal acquisition as well, we can further decrease the current consumption by 24 percentage points. This gain comes entirely from the PPG sensor usage optimization, where the current consumption of the sensor is reduced from 8.4 mA down to 0.4 mA, when not in use. This non-null current consumption is because, in Level 0 and Level 1, the sensor is not fully powered down, instead it goes in a low-power mode allowing to retain all configurations. This mode was chosen to

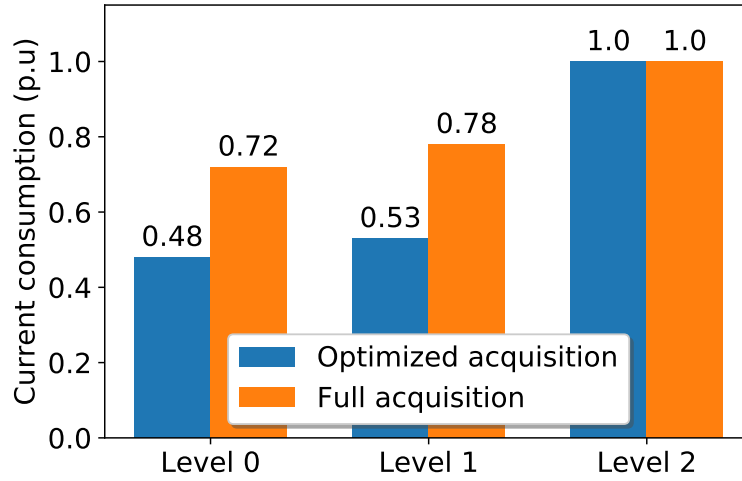


Figure 4.9 – Current consumption of the proposed levels while always acquiring all signals and acquiring only the required ones.

avoid the need for reconfiguring the sensor all the time.

Although both Table 4.13 and Figure 4.9 show a considerable reduction in current consumption, the system cannot work full time in Level 0. Therefore, to produce a reliable classification, the system has to continuously switch between the three different levels of energy-scalable machine-learning models. In this regard, Table 4.14 shows an estimation of the expected average current consumption, which is the result of an emulated online cognitive workload detection running on a PC and based on data from the experiment described in Chapter 3. The results show that based on the test set, which is composed of 188 minutes collected from 24 subjects, the system runs 114 minutes in Level 0, 47 minutes in Level 1, and 27 minutes in Level 2. A similar result is obtained while acquiring all signals, that is, 111 minutes in Level 0, 52 minutes in Level 1, and 25 minutes in Level 2. Therefore, the proposed self-aware approach reduces the average current consumption from 32.9 mA to 25.4 mA, while acquiring all signals, and down to 18.6 mA, while extending the concept of self-awareness down to the acquisition level.

As highlighted in Figure 4.10, the current consumption of the proposed strategy is almost half compared to a standard approach. Indeed, the system shows a decrease of the classification accuracy, but this reduction is less than 0.3 percentage points, which is acceptable considering the important gain in energy efficiency.

Table 4.14 – Average current consumption of the different energy-scalable models of each level. In bracket the values while always acquiring all signals.

	Time (min)	Unit current (mA)	Total current (mA)
Level0	114 (111)	15.7	9.5 (14.0)
Level 1	47 (52)	17.5	4.4 (7.1)
Level 2	27 (25)	32.9	4.7 (4.4)
Total	188	-	18.6 (25.4)

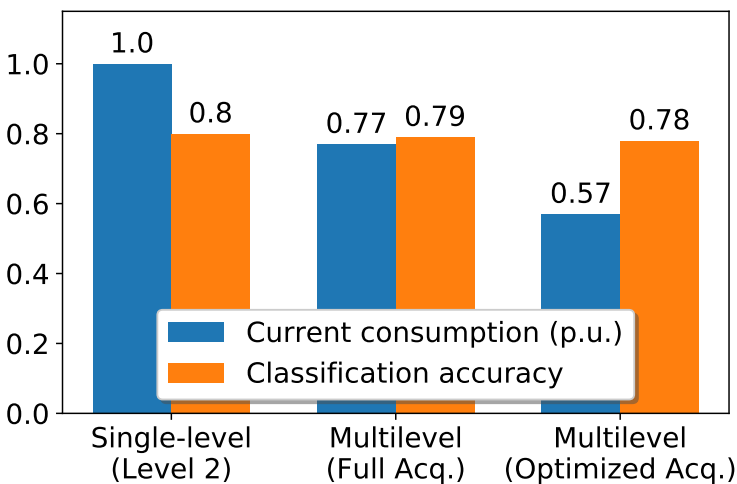


Figure 4.10 – Current consumption and classification accuracy for the different approaches, namely, a single-level model, a multi-level model acquiring all signals, and the proposed multilevel model acquiring only the required signals.

As a recap, Table 4.15 shows the main results achieved with the application of the concept of self-awareness and energy-scaled machine-learning. In particular, we can see that the battery life can be increased from 12.7 to 22.6 hours without really affecting the classification accuracy. The theoretical accuracy of the proposed self-aware system exploiting three different levels of energy-scalable machine-learning is in fact 77.65%, which is still comparable to the 80.32% reached while using the full model, that is, the model used for Level 2. The results also show that both multilevel approaches (i.e., the multilevel models with full and optimized acquisitions) spend more or less the same amount of time in the different levels. This result suggests that most of the time, the system is correct with its prediction about the level to use. Therefore, we can conclude that adapting the level with a delay of 60 seconds is not a problem.

Table 4.15 – Comparison of the different approaches in terms of accuracy, time spent in the various levels, average current consumption, and estimated battery life.

Approach	Accuracy	Time (%) (L0, L1, L2)	Avg. current (mA)	Battery life (h) (420 mAh)
Single level (Full Acq.)	80.32	0, 0, 100	32.9	12.7
Multilevel (Full Acq.)	78.72	59, 28, 13	25.4	16.5
Multilevel (Opt. Acq.)	77.65	61, 25, 14	18.6	22.6

## 4.8 Conclusions

Rescuers often operate in stressful conditions, which increases in cognitive workload so that it could compromise the outcome of a mission. Therefore, there is a need to detect high levels of cognitive workload to assist them and prevent a possible drop in performance. To address this problem, in this chapter, I proposed the hardware/software co-design of a wearable embedded system for monitoring the cognitive workload of humans engaged in drone teleoperations during search and rescue missions.

On the hardware side, the proposed system includes different sensors and a micro-controller allowing both non-invasive physiological signals acquisition and online

## **Chapter 4. Wearable and Multimodal Cognitive Workload Monitoring System**

---

processing. On the software side, the proposed system includes novel energy-aware bio-signal processing algorithms and an application of the self-aware concept for scalable energy consumption. To this aim, different embedded machine-learning algorithms and methods are used for online cognitive workload monitoring, exploiting the trade-offs between the required accuracy and the available energy of the system.

The results showed that the proposed multi-channel signal-acquisition and processing platform, which included embedded algorithms and methods, were validated for online monitoring of low and high levels of cognitive workload, thus achieving an accuracy comparable to the reference Matlab implementation running on a PC. Moreover, the proposed optimization scheme, based on a self-aware approach that exploited a scalable machine-learning method with different power-saving levels, allowed an increase of 78% of the battery lifetime. Compared to the best universal background model, the approach also showed an acceptable accuracy lost (i.e., from 80.32% to 77.65%).

In conclusion, I proposed and validated a wearable system for the particular case of cognitive workload monitoring. However, the multi-channel physiological signals acquisition platform together with the implemented energy-aware modular signal processing architecture are also suitable for different bio-monitoring applications. For instance, the proposed platform could be used to detect stress induced by fatigue [21], pain [97], or emotional problems [93], such as depression, anxiety, and anger. The proposed system yields to lay the foundations for the development of novel share-control strategies, which could improve the teleoperation of drones or distant robots in general. In this regard, the information of the human state, and in particular, the intensity of cognitive workload, is the key to dynamically adapt the level of autonomy based on the real needs of the operator.



## 5 Conclusions and Perspectives

To conclude this thesis, I would like to highlight hereafter the main contributions of my research work together with some research directions that could be addressed in future works.

### 5.1 Summary and Conclusions

Thanks to recent enhancements in both artificial intelligence and share-control techniques, both human and robot abilities can be combined. These merging skills can ensure high performance during unique and high cognitive demanding tasks, such as in search and rescue missions. However, to bring robot teleoperation to such a level, there is a need to develop control solutions that are both portable and intuitive to use. Moreover, to dynamically adapt human-robot interaction, there is a need to consider the information of the human state.

In this thesis, I first developed a new wearable embedded system for drone control, which tracks upper-body movements and translates them into commands for a drone. The proposed solution is fully portable, it allows both indoor and outdoor deployments, and compared to traditional controllers, it is more intuitive to use. Then, I investigated and proposed a method for cognitive workload monitoring from physiological signals, which is suitable for embedded solutions. Finally, I enhanced both hardware and software design of the wearable embedded system for drone control to integrate the online cognitive workload monitoring as well. My work paves the way for new personalized and adaptive share-control techniques. In particular, thanks to the continuous and noninvasive monitoring of the human state, the level

## Chapter 5. Conclusions and Perspectives

---

of control can be dynamically adapted based on the real needs of the pilot.

In the following subsections, I provide a more detailed summary of the contributions described in this thesis.

### 5.1.1 Wearable Drone Controller

A control strategy, based on upper-body movements derived from a spontaneous representation of human-drone interactions, was implemented in a wearable embedded system. Such a system offers an intuitive and effective interface, which can reduce the training and improve the reliability of teleoperated activities. Moreover, the proposed system was embedded into a soft exoskeleton, which compared to the state of the art, provided a more portable and intuitive solution. Therefore, my system could be used by different populations to control various types of machines in different fields, including Search and Rescue (SAR) missions with drones, but also for leisure purposes. Beyond the implementation, I analyzed the system to reduce energy consumption. Knowing that wireless communication is often one of the weaknesses of embedded systems in terms of energy consumption, I proposed a method that drastically reduced the transmission volume, and consequently also the energy consumption. Compared to a typical system that is continuously streaming the commands, the proposed solution showed a reduction of the transmission volume up to 50%, yielding up to 11.9% gain of the energy consumption. Finally, the proposed wearable embedded system, called FlyJacket, was interfaced and demonstrated for the teleoperation of both a drone simulator and a real drone during different experiments, but also in public events, such as Cybathlon and École Polytechnique Fédérale de Lausanne (EPFL) Open Days.

### 5.1.2 Cognitive Workload Detection Method

To address continuous cognitive-workload monitoring in search and rescue missions with drones, I proposed a reliable subject-specific machine-learning algorithm based on Support Vector Machine (SVM). My multi-modal model fused the contributions of some important features to assess cognitive workload monitoring. These features came from different physiological signals (i.e., RSP, ECG, PPG, and SKT) acquired in a non-invasive way. Then, to reduce both subject and day inter-variability, I explored different feature normalization techniques. The results showed that a combination of both day and subject normalization improved the detection accuracy. Finally, the

proposed SVM model was tested on a new test set, demonstrating the capability of detecting both low and high cognitive workload levels with an average accuracy of 87.3% and 91.2%, by using a traditional controller and my wearable embedded system based on torso movements, respectively.

### 5.1.3 Wearable Cognitive Workload Monitoring System

To the best of my knowledge, online cognitive workload monitoring has not been addressed yet in the field of search and rescue. Therefore, to fill this gap, I proposed a hardware/software co-design of a wearable embedded system for online cognitive workload monitoring from physiological signals. On the hardware side, this wearable system included a multi-channel physiological signals acquisition and a low-power processing platform that was suited for both drone control and cognitive workload monitoring. On the software side, the proposed system included novel energy-aware bio-signal processing and the application of a self-aware concept for scalable energy consumption. The proposed multi-channel signal-acquisition and processing platform was validated for online monitoring of low and high levels of cognitive workload, achieving accuracy comparable to the reference Matlab implementation running on a PC. Moreover, the proposed optimization scheme, based on a self-aware approach that exploits a scalable machine-learning method with different power-saving levels, allowed an increase of 78% of the battery lifetime. Compared to the best universal background model, the approach showed an acceptable accuracy lost, that is, from 80.32% to 77.65%.

## 5.2 Future Work

In this section, I provide some possible research directions that could be taken based on the research findings described in this thesis. In particular, I highlighted some of the short- and long-term possible research lines.

### 5.2.1 Suggestions for Drone Control Improvements

In this work, I designed and presented a wearable device to controls both roll and pitch commands of a drone. To control the remaining Degrees of Freedom (DoF), instead, we used a glove. The problem is that this glove is limiting the use of the hands, which should remain free to handle any possible parallel task. Therefore, further investigation is needed to integrate the control of all DoF into the jacket.

## Chapter 5. Conclusions and Perspectives

---

In this regard, new experiments following the guidelines and methods presented in [108] should take place, but now, the methods should be applied to identify the way to control the remaining DoF.

By integrating the control of all DoF into a single system, it will avoid the need for an additional device, which collects all the commands and forward them to the drone. Instead of an additional device, the central node could manage the transmission of these commands, but for this purpose, the node should provide the appropriate communication interface, which is a long-range RF transmitter. Then, the method applied to reduce the communication ratio should be adapted to consider the impact of the distance. In this regard, the communication rate should not fall below a minimum acceptable value, which will prevent the loss of control. Therefore, this communication rate should increase while operating at long distances, and decrease otherwise.

Another possible research line involves the translation of the pilot's gesture into steering commands, which relies on linear transformations with weights that were maintained constant across participants. The choice of these weights is critical, as high values can yield to overshoot, and little values typically cause slow responses. During the experiments, I noticed that there seems to be some correlation between movement amplitude and flying expertise. Novices usually behave quite a lot, while experts tend to control the drone with small, smooth, and precise movements. Thus, it is preferred to use little weights for novices to attenuate bumpy movements, while higher weights are more appropriate for experts. Therefore, to find the proper compromise avoiding both overshoots and too slow responses, there is a need to address the trade-off between control weights and pilots expertise. A system with variable weights could fit a more extended range of the population.

### 5.2.2 Perspectives in Cognitive Workload Characterization

In this work, I presented a method that detects low and high levels of cognitive workload. Preliminary results, which we recently published in [39], showed that an Extreme Gradient Boosting (XGB) was able to detect high, medium, and low cognitive workload levels, with an accuracy of 62.9%. If we consider the little difference between the induced levels of cognitive workload, the results are promising and demonstrate that fine-grain detection is feasible, even in the field of SAR missions. However, the results of the three-class XGB present a sign of overfitting, which is due to the limited amount of training data used to fit the parameters of such a high

capacity model. Therefore, there is a need to investigate the problem of multi-class cognitive workload detection more deeply. In particular, I suggest to collect more data and better tune the parameters of that model. Additional data from different cognitive workload conditions would allow fine-grain detection, and most likely, also a regression, which could also represent the coupling between cognitive workload and performance.

Although cognitive workload detection has already been covered in different fields, the main problem is always the same, that is, finding a ground truth and derive a model. However, the fact that both environment and human expertise affect the perception of cognitive workload can change everything. Therefore, as this experiment is based on a simulated and controlled environment, the subjects were not exposed to the same stressful conditions as they would be in the case of real SAR missions. In particular, no real drone was employed, there was no fire, and no one was in real danger. Therefore, there is a need to further investigate unexpected physiological changes in the field during real-life rescue missions with drones, to estimate the possible benefits of person-specific thresholds indicating the need for assistance. Moreover, there is a need to quantify cognitive workload in different fields to define a unified metric that can be used as a standard reference. To this aim, I think that we should include a context coherence adaptation and define how a particular feature change based on external factors or activity.

### 5.2.3 Future of Online Cognitive Workload Monitoring

A possible research line would be the investigation of the use of less discriminating features, which are correlated with the selected ones. In the process of feature reduction, I selected the most discriminant features and discarded all the correlated ones, without considering the computational cost. This choice was made to achieve the best performance in terms of classification accuracy. However, to improve the hardware/software co-design, we should now consider both the computational cost as well as the discriminatory power of the features. In this way, we could select additional power-saving levels and better address the trade-off between classification accuracy and energy consumption. In this regard, the first possible approach could be to consider new energy-saving levels by differentiating between time and frequency domain features. Then, we could build supplementary levels by replacing the features with high computational costs with those that are correlated and require a lower computational effort.

## Chapter 5. Conclusions and Perspectives

---

Before going in this direction, the software (SW) implementation needs to be optimized. In this regard, indeed, there is still a considerable margin for improvement, and in particular, in the computation of the frequency-domain features, which are responsible for the main computational effort. Although the CMSIS-DSP library provides a collection of optimized functions for signal processing, which can drastically reduce the computation time, it would be interesting to further optimize the computation of these frequency-domain features by extracting only the necessary frequency bands. While using the function from the CMSIS-DSP library, we compute the full spectra, which contain frequency bands that are useless in this specific application. Therefore, an optimization of such functions can further reduce both the computation of these frequency-domain features and memory usage.

An additional aspect that should be optimized is the filtering layer, which suffers from a similar problem. To lower the computational effort, we used functions from the CMSIS-DSP library. However, such functions need a fixed amount of data as input and oblige the storage of raw samples, causing a waste of memory that is almost twice higher than what it is needed. Therefore, instead of combining the filtering process with the delineation, I would suggest integrating this part at the acquisition level. By implementing the filtering process at the acquisition level, it would be possible to process every sample one by one. In this way, the storage of raw samples would be limited to a number given by both order and type of filter. Of course, this approach should be implemented and optimized to achieve performances comparable to the functions provided by the CMSIS-DSP library.

A new interesting research line could be the use of a different sampling approach, namely, the event-based sampling. This method has been validated for the acquisition of Electrocardiogram (ECG) [170], but not yet for other signals. Therefore, it would be interesting to extend this approach to signals such as Respiratory Activity (RSP), Photoplethysmogram (PPG), and Skin Temperature (SKT), and apply the methods in applications such as the one presented in this work. From preliminary studies performed at Embedded System Laboratory (ESL), we observed that for measurements, such as PPG, RSP, and Inertial Measurement Unit (IMU), where at least some portions of the signal present a low dynamic, event-based sampling works pretty well. The idea is simple. While using standard sampling, we approximate a signal with a concatenation of short linear interpolations. Now, by extending this concept of linear interpolation, we can acquire fewer samples and save energy while still describing the signals with enough precision. I think that such an approach could further reduce the energy consumption of the system, but it involves a significant

redesign of both hardware and software. Indeed, from the hardware side, the acquisition layer would need a redesign as current sensors are not designed to support such a sampling method. Similarly, the software side would need to be adapted for the processing of such unevenly sampled data. To this aim, new methods are needed, such as the one proposed to delineate ECG [198].

To further reduce energy consumption, we could further reduce both the supply voltage and the operating frequency [151]. Moreover, it would be interesting to consider the use of a multi-core Microcontroller Unit (MCU) instead of a single-core [22]. In this regard, most of the processing is already parallelized, that is, the processing for each signal. However, further efforts should be made to address the parallelization of the complex functions, which could allow an additional energy-saving when running on different cores. With such a multi-core architecture, we could turn on and off the cores based on the needs, as I do with the sensors. In this way, we can extend the concept of self-awareness, not only at the acquisition and the processing layers, but also at the hardware level of the MCU. Therefore, we could further reduce the energy consumption while operating at different energy-scalable levels.

Finally, for the development of the proposed wearable drone controller with an embedded unit for cognitive workload monitoring, I considered a design that was not too fragile and also suitable for research purposes. Therefore, I used commercially available off-the-shelf components of medium size. These components were the best options at the design time. However, some of those components are now available in a smaller size, and others, such as MCUs and Bluetooth Low Energy (BLE) modules, are now embedded in single Application-Specific Integrated Circuits (ASICs), providing solutions that are energetically more efficient. Therefore, an update of the design should be addressed before bringing the product to the market. Moreover, to further reduce both size and energy consumption, the integration of the full system into a single ASIC would be an option to consider.

### 5.2.4 Human-State: An Input for New Share-Control Techniques

In this work, I validated the use of the proposed wearable system for the particular case of cognitive workload monitoring. However, the cognitive workload is not the only information that characterizes the human state. For instance, other stress factors induced by fatigue [21] or emotional problems [93], such as depression, anxiety, and anger, affect the human state. Moreover, both motivation and the

feeling of self-confidence also play a crucial role [147]. These factors affect the effort that an operator will put into the execution of a difficult task. Therefore, further investigations are necessary to include supplementary knowledge of the human state.

As I already mentioned, the proposed system paved the way for the development of novel share-control strategies, which could improve the teleoperation of distant robots. Indeed, the information about the human state is the key to dynamically adapt the level of autonomy based on the operator's needs. To enhance current share-control techniques, I think we could use such information to adapt the control weights of the drone. In particular, we could adapt both the reactivity and speed of the drone by using the information about the human state. In this way, we could also address the fact that pilots have different abilities. However, further investigations are required to identify how to use such information in a more precise manner.

### 5.3 Retrospective

Designing experiments to collect data is always challenging. There are many factors to consider, and for sure, something can go wrong. As a perfectionist, quite a positive person, and sometimes maybe a bit too ambitious, I designed my first experiments by directly trying to address the complex problem. Finally, as there was no much to conclude from these experiments, I did not report them in my thesis. However, with these tests, I had the chance to learn that sometimes you have to do it wrong, but not on purpose, just by accepting that it is not precise as it should be. This process is part of the design of the experiment. If you want to learn about something, you have to take measurements. I remember my thesis co-director, Prof. Diego Barrettino, gave me this advice at some point, and indeed, he was right. With these lines, I want to say that it is crucial to do preliminary essays to understand what you need for the final experiments. To recap, I recommend to start with simple cases, simplistic problems, get experience, and move to more complex dilemmas with good knowledge and the proper equipment.

Designing embedded systems is a fascinating activity. You need to analyze the application to identify both requirements and constraints. Then, you set the specifications, define the system architecture, design block diagrams, data flow graphs, etc. Next, you can design and develop both hardware and software. Finally, you test the system at different levels for checking the functionalities of both hardware and software and ensure that the system is without errors and defects. What I reported in my thesis



was the result of this process, including the main decisional and validation steps. What I did not mention was that designing an embedded system is already quite laborious, and this process is even worst when the system is intended for research purposes. Often, in this case, many requirements and specifications are unknown at the beginning. On the other hand, you can relax some constraints (e.g., aesthetic finishes, dimension reduction, etc.) because the system is not supposed to go to the market, or at least not in the immediate future.

For the design of the proposed system, I considered different options, such as the number and the type of sensors. The idea was to extend the design targeting other possible applications addressed in our lab, such as the prediction of atrial fibrillation [37] and the detection of epileptic seizures [58]. Therefore, because of these additional and sometimes not well-defined requirements, the design process took longer than expected. For the hardware design, I had the chance to benefit from a solid starting point, that was, a consolidated prototype of a wearable device developed at ESL from previous students. Thanks to my predecessors, I was able to reuse part of the design of such a system, adapt it to my needs, and add the missing elements. The prototyping we did it at EPFL, where I had all the support I needed. The software design was instead more complex. On one hand, I had the chance to benefit from the help of many students, who helped me implementing and testing different functionalities. But, on the other hand, I also had bad luck with some of them who did something that was not utilizable in the final implementation. A bunch of such little works had to be reviewed, corrected, cleaned, and adapted. At some point, Napoleone Bonaparte said: "If you want a thing done well, do it yourself." With this sentence, I don't want to say that I don't make mistakes. To err is human. However, I wonder sometimes if instead of delegating some work to students, who occasionally were not that motivated, I would not have done it faster by myself. Anyway, I guess this is part of a Ph.D. student's life, where you still have to learn how to learn, and you have to learn how to teach.



# Bibliography

- [1] D. A. Abbink, M. Mulder, and E. R. Boer. Haptic shared control: smoothly shifting control authority? *Cognition, Technology & Work*, 14:19–28, 2011.
- [2] B. Ahmed, H. M. Khan, J. Choi, and R. Gutierrez-Osuna. ReBreathe: A calibration protocol that improves stress/relax classification by relabeling deep breathing relaxation exercises. *IEEE Transactions on Affective Computing*, 7(2):150–161, Apr 2016.
- [3] H. Akoglu. User's guide to correlation coefficients. *Turkish journal of emergency medicine*, 18(3):91–93, 2018.
- [4] V. Alvarez-Santos, R. Iglesias, X. Pardo, C. Regueiro, and A. Canedo-Rodriguez. Gesture-based interaction with voice feedback for a tour-guide robot. *Journal of Visual Communication and Image Representation*, 25(2):499 – 509, 2014.
- [5] K. Aminian and B. Najafi. Capturing human motion using body-fixed sensors: Outdoor measurement and clinical applications. *Computer Animation and Virtual Worlds*, 15(2):79–94, 2004.
- [6] A. Aminifar. *Analysis, design, and optimization of embedded control systems*, volume 1746. Linköping University Electronic Press, 2016.
- [7] A. Aminifar, P. Tabuada, P. Eles, and Z. Peng. Self-triggered controllers and hard real-time guarantees. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 636–641, 2016.
- [8] A. Anzanpour, I. Azimi, M. Götzinger, A. M. Rahmani, N. TaheriNejad, P. Liljeberg, A. Jantsch, and N. Dutt. Self-awareness in remote health monitoring systems using wearable electronics. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 1056–1061, 2017.

## Bibliography

---

- [9] Apple. Watch. <https://www.apple.com/watch>.
- [10] A. Arza, J. M. Garzón-Rey, J. Lázaro, E. Gil, R. Lopez-Anton, C. de la Camara, P. Laguna, R. Bailon, and J. Aguiló. Measuring acute stress response through physiological signals: towards a quantitative assessment of stress. *Medical and Biological Engineering and Computing*, pages 1–17, Aug 2018.
- [11] F. Augugliaro, S. Lupashin, M. Hamer, M. Male, Cason Hehn, M. W. Mueller, J. Willmann, F. Gramazio, M. Kohler, and R. D’Andrea. The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine*, 34(4):46 – 64, 2014.
- [12] R. Bartlett. *Introduction to Sports Biomechanics*. Routledge, 2 edition, 2008.
- [13] T. Baudel and M. Beaudouin-Lafon. Charade: Remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, July 1993.
- [14] S. Betti, R. M. Lova, E. Rovini, G. Acerbi, L. Santarelli, M. Cabiati, S. D. Ry, and F. Cavallo. Evaluation of an integrated system of wearable physiological sensors for stress monitoring in working environments by using biological markers. *IEEE Transactions on Biomedical Engineering*, 65(8):1748–1758, 2018.
- [15] S. Betti, R. Molino Lova, E. Rovini, G. Acerbi, L. Santarelli, M. Cabiati, S. Del Ry, and F. Cavallo. Evaluation of an integrated system of wearable physiological sensors for stress monitoring in working environments by using biological markers. *IEEE Transactions on Biomedical Engineering*, 65(8):1748–1758, Aug 2018.
- [16] Biopac. MP160 Data Acquisition Systems. <https://www.biopac.com/product/mp150-data-acquisition-systems>.
- [17] Biovotion. Everion. <https://www.biovotion.com/everion/>.
- [18] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, Jan 2006.
- [19] J. M. Bland and D. G. Altman. Statistical methods for assessing agreement between 2 methods of clinical measurement. *Lancet*, 8476:307–310, 1986.
- [20] O. Blanke. Multisensory brain mechanisms of bodily self-consciousness. *Nature Reviews Neuroscience*, 13:556–571, 2012.

- [21] G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni. Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. *Neuroscience & Biobehavioral Reviews*, 44:58–75, Jul 2014.
- [22] R. Braojos, A. Dogan, I. Beretta, G. Ansaloni, and D. Atienza. Hardware/software approach for code synchronization in low-power multi-core sensor nodes. In *Design, Automation, and Test in Europe (DATE) Conference and Exhibition*, pages 1–6, 2014.
- [23] B. Cain. A Review of the Mental Workload Literature. Toronto. *Defence Research and Development Canada*, 07 2007.
- [24] C. Carignan, J. Tang, and S. Roderick. Development of an exoskeleton haptic interface for virtual task training. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 369, pages 3697–3702, 12 2009.
- [25] T. Carlson, R. Leeb, R. Chavarriaga, and J. D. R. Millán. Online modulation of the level of assistance in shared control systems. In *IEEE International Conference on Systems, Man and Cybernetics*, 2012.
- [26] T. Carlson and J. d. R. Millán. Brain-controlled wheelchairs: A robotic architecture. *IEEE Robotics and Automation Magazine*, 20(1):65–73, 2013. The original accepted preprint was entitled: "The Robotic Architecture of an Asynchronous Brain-Actuated Wheelchair".
- [27] D. Carneiro, P. Novais, J. C. Augusto, and N. Payne. New methods for stress assessment and monitoring at the workplace. *IEEE Transactions on Affective Computing*, 10(2):237–254, Apr 2019.
- [28] J. Casper and R. R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2003.
- [29] J. Y. C. Chen, M. J. Barnes, and M. Harper-Sciarini. Supervisory control of multiple robots: Human-performance issues and user-interface design. *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews)*, 41(4):435–454, Jul 2011.
- [30] L.-l. Chen, Y. Zhao, P.-f. Ye, J. Zhang, and J.-z. Zou. Detecting driving stress in physiological signals based on multimodal feature analysis and kernel classifiers. *Expert Systems with Applications*, 85:279–291, Nov 2017.

## Bibliography

---

- [31] T. Chen, F. Faniyi, R. Bahsoon, P. R. Lewis, X. Yao, L. L. Minku, and L. Esterle. The handbook of engineering self-aware and self-expressive systems. *ArXiv*, abs/1409.1793, 2014.
- [32] A. Cherpillod, S. Mintchev, and D. Floreano. Embodied Flight with a Drone. *CoRR*, abs/1707.0, 2017.
- [33] J. Chua. *Recruiting female firefighters: Closing the gender gap*, 2018 (accessed October 12, 2020).
- [34] B. Cinaz, B. Arnrich, R. La Marca, and G. Tröster. Monitoring of mental workload levels during an everyday life office-work scenario. *Personal and Ubiquitous Computing*, 17(2):229–239, Feb 2013.
- [35] T. Crouch, N. Air, and S. Museum. *Lighter Than Air: An Illustrated History of Balloons and Airships*. Johns Hopkins University Press, 2009.
- [36] T. Crouch, S. N. Air, and S. Museum. *Wings: A History of Aviation from Kites to the Space Age*. Smithsonian National Air and Space Museum, 2003.
- [37] E. De Giovanni, A. Aminifar, A. Luca, S. Yazdani, J.-M. Vesin, and D. Atienza Alonso. A patient-specific methodology for prediction of paroxysmal atrial fibrillation onset. *Computing in Cardiology*, 44, 2017.
- [38] F. Dell’Agnola, L. Cammoun, and D. Atienza. Physiological characterization of need for assistance in rescue missions with drones. In *IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, Jan 2018.
- [39] F. Dell’Agnola, N. Momeni, A. Arza, and D. Atienza. Cognitive workload monitoring in virtual reality based rescue missions with drones. In *12th International Conference on Virtual, Augmented and Mixed Reality, Copenhagen, Denmark*, July 2020.
- [40] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano, L. M. Gambardella, R. Siegwart, and D. Scaramuzza. The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36:1167–1269, October 2019.
- [41] Y. Deng and P. Wang. *Ancient Chinese Inventions*. Cultural China series. China Intercontinental Press, 2005.
- [42] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

- 
- [43] G. Dudek, J. Sattar, and A. Xu. A visual language for robot control and programming: A human-interface study. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2507–2513, 2007.
  - [44] N. Dutt, A. Jantsch, and S. Sarma. Toward smart embedded systems: A self-aware system-on-chip (soc) perspective. *ACM Transactions on Embedded Computing Systems (TECS)*, 15(2):1–27, 2016.
  - [45] F. T. Eggemeier, G. F. Wilson, A. F. Kramer, and D. L. Damos. Workload assessment in multi-task environments. In *Multiple Task Performance*. Taylor & Francis, 1991.
  - [46] Empatica E4. Real-time physiological data streaming and visualization. <https://www.empatica.com>.
  - [47] D. C. Engelbart and W. K. English. A research center for augmenting human intellect. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 395–410, 1968.
  - [48] F. Faniyi, P. R. Lewis, R. Bahsoon, and X. Yao. Architecting self-aware software systems. In *2014 IEEE/IFIP Conference on Software Architecture*, pages 91–94, 2014.
  - [49] P. Fankhauser, M. Bloesch, and M. Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters*, 3(4):3019–3026, Oct 2018.
  - [50] B. Fasel, J. Spörri, J. Kröll, E. Müller, and K. Aminian. Using inertial sensors for reconstructing 3d full-body movement in sports - possibilities and limitations on the example of alpine ski racing. In *33rd International Conference on Biomechanics in Sports*, Poitiers, 2015.
  - [51] A. R. Fayjie, A. Ramezani, D. Oualid, and D. J. Lee. Voice enabled smart drone control. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 119–121, 2017.
  - [52] S. S. Fels and G. E. Hinton. Glove-talk: a neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks*, 4(1):2–8, 1993.
  - [53] R. A. S. Fernández, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy. Natural user interfaces for human-drone multi-modal interaction.

## Bibliography

---

- In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1013–1022, 2016.
- [54] R. Finn and S. Scheduling. *Developments and Challenges for Autonomous Unmanned Vehicles: A Compendium*, volume 3. Springer-Verlag Berlin Heidelberg, 01 2010.
- [55] Fitbit. Smartwatches and trackers for fitness and lifestyle. <https://www.fitbit.com>.
- [56] G. FKS. *Feuerwehrstatistik 2018*, 2018 (accessed October 12, 2020).
- [57] D. Floreano and R. J. Wood. Science, technology and the future of small autonomous drones. *Nature*, 521:460–466, 05 2015.
- [58] F. Forooghifar, A. Aminifar, L. Cammoun, I. Wisniewski, C. Ciumas, P. Ryvlin, and D. Atienza. A self-aware epilepsy monitoring system for real-time epileptic seizure detection. *Mobile Networks and Applications*, 2019.
- [59] D. A. Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2 edition, 2009.
- [60] Garmin. Sport watch. <https://www.garmin.com>.
- [61] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280 – 2292, 2014.
- [62] D. Giakoumis, D. Tzovaras, and G. Hassapis. Subject-dependent biosignal features for increased accuracy in psychological stress detection. *International Journal of Human-Computer Studies*, 71(4):425–439, Apr 2013.
- [63] M. Gjoreski, M. Luštrek, M. Gams, and H. Gjoreski. Monitoring stress with a wrist device using context. *Journal of Biomedical Informatics*, 73:159–170, Sep 2017.
- [64] D. Goepfert, R. Karlen, M. Hartmann, G. Stäheli, S. Enz, H. Cina, J. Signer, M. Thalmann, M. Knöri, H. Benz, and P. Zurkirchen. *Reglement Einsatzführung*. Feuerwehr Koordination Schweiz FKS, Bern, 2015.
- [65] A. Golgouneh and B. Tarvirdizadeh. Fabrication of a portable device for stress monitoring using wearable sensors and soft computing algorithms. *Neural Computing and Applications*, pages 1–23, 6 2019.



- 
- [66] R. Gomez, K. Nakamura, T. Kawahara, and K. Nakadai. Multi-party human-robot interaction with distant-talking speech recognition. In *7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 439–446, Mar 2012.
- [67] L. Gonzalez, T. Paniagua, N. Starliper, and E. Lobaton. Signal quality for rr interval prediction on wearable sensors. In *41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2525–2528, 2019.
- [68] M. A. Goodrich and A. C. Schultz. Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, 1(3):203–275, 2007.
- [69] Gov.uk. *Fire and rescue authorities operational statistics*, 2017 (accessed October 12, 2020).
- [70] B. Gromov, L. M. Gambardella, and A. Giusti. Robot Identification and Localization with Pointing Gestures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [71] P. Grossman and E. W. Taylor. Toward understanding respiratory sinus arrhythmia: Relations to cardiac vagal tone, evolution and biobehavioral functions. *Biological Psychology*, 74(2):263–285, Feb 2007.
- [72] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.
- [73] J. Han and N. Gold. Lessons learned in exploring the leap motion™ sensor for gesture-based instrument design. In *International Conference on New Interfaces for Musical Expression*. Goldsmiths University of London, 2014.
- [74] L. Han, Q. Zhang, X. Chen, Q. Zhan, T. Yang, and Z. Zhao. Detecting work-related stress with a wearable device. *Computers in Industry*, 90:42–49, Sep 2017.
- [75] S. G. Hart and L. E. Staveland. *Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research*, volume 52 of *Advances in Psychology*, pages 139–183. Elsevier, 1988.
- [76] J. A. Healey and R. W. Picard. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, Jun 2005.

## Bibliography

---

- [77] J. Heard, C. E. Harriott, and J. A. Adams. A survey of workload assessment algorithms. *IEEE Transactions on Human-Machine Systems*, 48(5):434–451, Oct 2018.
- [78] T. Heine, G. Lenis, P. Reichensperger, T. Beran, O. Doessel, and B. Deml. Electrocardiographic features for the measurement of drivers’ mental workload. *Applied Ergonomics*, 61:31–43, May 2017.
- [79] H. Hoffmann, M. Maggio, M. Santambrogio, A. Leva, and A. Agarwal. Seec: A framework for self-aware computing. *Computer Science and Artificial Intelligence Lab (CSAIL), Technical report*, 11 2010.
- [80] Huawei. Watch. <https://consumer.huawei.com/en/wearables/watch2>.
- [81] Hypersuit. Hypersuit. <https://www.hypersuit.fr>.
- [82] A. Jantsch, N. Dutt, and A. M. Rahmani. Self-awareness in systems on chip—a survey. *IEEE Design & Test*, 34(6):8–26, 2017.
- [83] P.-K. Jao. *Decoding Cognitive States under Varying Difficulty Levels*. EPFL, Lausanne, 2020.
- [84] Jawbone. UP. <https://jawbone.com/up>.
- [85] G. Jones, N. Berthouze, R. Bielski, and S. Julier. Towards a situated, multimodal interface for multiple uav control. In *2010 IEEE International Conference on Robotics and Automation*, pages 1739–1744. IEEE, 2010.
- [86] D. Kahneman. *Attention and effort*. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [87] D. M. Kaushik and R. Jain. Gesture based interaction nui: an overview. *arXiv preprint arXiv:1404.2364*, 2014.
- [88] A. Kay. User interface: A personal view. *The art of human-computer interface design*, pages 191–207, 1990.
- [89] M. N. H. Khan and C. Neustaedter. An exploratory study of the use of drones for assisting firefighters during emergency situations. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.
- [90] S. Kounev, J. O. Kephart, A. Milenkoski, and X. Zhu. *Self-Aware Computing Systems*. Springer International Publishing, 1st edition, 2017.

- [91] C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides, and C. Panayiotou. Drones: Augmenting our quality of life. *IEEE Potentials*, 38(1):30–36, 2019.
- [92] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester. A survey on wireless body area networks. *Wireless Networks*, 17(1):1–18, 2011.
- [93] B. G. Lee, T. W. Chong, B. L. Lee, H. J. Park, Y. N. Kim, and B. Kim. Wearable Mobile-Based Emotional Response-Monitoring System for Drivers. *IEEE Transactions on Human-Machine Systems*, 47(5):636–649, 2017.
- [94] P. R. Lewis, A. Chandra, S. Parsons, E. Robinson, K. Glette, R. Bahsoon, J. Torresen, and X. Yao. A survey of self-awareness and its application in computing systems. In *Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 102–107. IEEE, 2011.
- [95] Y. Lim, S. Ramasamy, A. Gardi, T. Kistan, and R. Sabatini. Cognitive Human-Machine Interfaces and Interactions for Unmanned Aircraft. *Journal of Intelligent & Robotic Systems*, pages 1–20, Oct 2017.
- [96] Logitech. Gamepad F310. <https://support.logi.com/hc/en-us/articles/360024326793>.
- [97] D. Lopez-Martinez and R. Picard. Multi-task neural networks for personalized pain recognition from physiological signals. In *Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 181–184, 2017.
- [98] R. J. Lysaght, S. G. Hill, a. O. Dick, B. D. Plamondon, P. M. Linton, W. W. Wierwille, a. L. Zaklad, a. C. Bittner Jr, and R. J. Wherry. Operator workload: Comprehensive review and evaluation of operator workload methodologies. *United States Army Research Institute for the Behavioral Sciences, Technical Report*, 1989.
- [99] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. *IEEE International Conference on Rehabilitation Robotics*, 2011.
- [100] H. Mansikka, K. Virtanen, and D. Harris. Dissociation between mental workload, performance, and task awareness in pilots of high performance aircraft. *IEEE Transactions on Human-Machine Systems*, 49(1):1–9, Feb 2019.

## Bibliography

---

- [101] T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. New generation of human machine interfaces for controlling UAV through depth-based gesture recognition. In R. E. Karlsen, D. W. Gage, C. M. Shoemaker, and G. R. Gerhart, editors, *Unmanned Systems Technology XVI*, volume 9084, pages 93 – 103. International Society for Optics and Photonics, SPIE, 2014.
- [102] D. Mantegazza, J. Guzzi, L. M. Gambardella, and A. Giusti. Vision-based control of a quadrotor in user proximity: Mediated vs end-to-end learning approaches. In *International Conference on Robotics and Automation (ICRA)*, 2018.
- [103] A. Marinescu, S. Sharples, A. Ritchie, T. S. López, M. McDowell, and H. Morvan. Exploring the relationship between mental workload, variation in performance and physiological parameters. *IFAC-PapersOnLine*, 49(19):591–596, 2016.
- [104] A. Mashood, H. Noura, I. Jawhar, and N. Mohamed. A gesture based kinect for quadrotor control. In *2015 International Conference on Information and Communication Technology Research (ICTRC)*, pages 298–301. IEEE, 2015.
- [105] G. Masinelli, F. Forooghifar, A. Arza, A. Aminifar, and D. Atienza. Self-aware machine learning for multimodal workload monitoring during manual labor on edge wearable sensors. *IEEE Design Test*, pages 1–1, 2020.
- [106] C. R. Meyer and H. N. Keiser. Electrocardiogram baseline noise estimation and removal using cubic splines and state-space computation techniques. *Computers and Biomedical Research*, 10:459–470, 1977.
- [107] Microsoft Azure. Kinect. <https://azure.microsoft.com/services/kinect-dk>.
- [108] J. Miehllbradt, A. Cherpillod, S. Mintchev, M. Coscia, F. Artoni, D. Floreano, and S. Micera. Data-driven body-machine interface for the accurate control of drones. *Proceedings of the National Academy of Sciences*, 115(31):7913–7918, Jul 2018.
- [109] J. C. Miehllbradt. *Body-machine interfaces for non-homologous human-machine interactions*. EPFL, Lausanne, 2019.
- [110] V. Mihajlović, B. Grundlehner, R. Vullers, and J. Penders. Wearable, wireless EEG solutions in daily life applications: What are we missing? *IEEE journal of biomedical and health informatics*, 19, Jun 2014.
- [111] R. B. Miller. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*,

- AFIPS '68 (Fall, part I), page 267–277, New York, NY, USA, 1968. Association for Computing Machinery.
- [112] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico, K. Karydis, N. Atanasov, G. Loianno, D. Scaramuzza, K. Daniilidis, C. J. Taylor, and V. Kumar. Fast, autonomous flight in gps-denied and cluttered environments. *Journal of Field Robotics*, 35:101–120, January 2018.
  - [113] N. Momeni, F. Dell’Agnola, A. Arza, and D. Atienza. Real-time cognitive workload monitoring based on machine learning using physiological signals in rescue missions. In *41th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2019.
  - [114] V. Montesinos, F. Dell’Agnola, A. Arza, A. Aminifar, and D. Atienza. Multi-modal acute stress recognition using off-the-shelf wearable devices. In *41th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2019.
  - [115] N. Moray. *Mental workload : its theory and measurement*. Published in coordination with NATO Scientific Affairs, Plenum Press, 1979.
  - [116] E. Mueggler, N. Baumli, F. Fontana, and D. Scaramuzza. Towards evasive maneuvers with quadrotors using dynamic vision sensors. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–8, Sep. 2015.
  - [117] M. Muller, S. Lupashin, and R. D’Andrea. Quadrocopter ball juggling. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 5113–5120. IEEE, 2011.
  - [118] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. M. Erkmen. Search and Rescue Robotics. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1151–1173. Springer, Berlin, Heidelberg, 2008.
  - [119] T. Nef, M. Mihelj, G. Kiefer, C. Perndl, R. Muller, and R. Riener. Armin - exoskeleton for arm therapy in stroke patients. In *IEEE 10th International Conference on Rehabilitation Robotics*, pages 68–74, June 2007.
  - [120] W. S. Ng and E. Sharlin. Collocated interaction with flying robots. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 143–149. IEEE, Jul 2011.

## Bibliography

---

- [121] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee. *Telerobotics*, pages 1085–1108. Springer handbooks. Springer International Publishing AG, Switzerland, 1 2016.
- [122] Nordic Semiconductor. nRF8001. [https://infocenter.nordicsemi.com/topic/struct\\_nrf8000/struct/nrf8001\\_ps.html](https://infocenter.nordicsemi.com/topic/struct_nrf8000/struct/nrf8001_ps.html).
- [123] D. Novak, B. Beyeler, X. Omlin, and R. Riener. Workload estimation in physical human-robot interaction using physiological measurements. *Interacting with Computers*, 27(6), 2015.
- [124] D. Novak, M. Mihelj, and M. Munih. A survey of methods for data fusion and system adaptation using autonomic nervous system responses in physiological computing. *Interacting with Computers*, 24(3):154–172, May 2012.
- [125] Oculus. Touch Controllers. <https://www.oculus.com/rift/accessories>.
- [126] L. Orlandic, E. d. Giovanni, A. Arza, S. Yazdani, J. Vesin, and D. Atienza. Reward: Design, optimization, and evaluation of a real-time relative-energy wearable r-peak detection algorithm \*. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3341–3347, 2019.
- [127] C. Orphanidou. *Quality Assessment for the Electrocardiogram (ECG)*, pages 15–40. Springer International Publishing, Cham, 2018.
- [128] C. Orphanidou. *Quality Assessment for the Photoplethysmogram (PPG)*, pages 41–63. Springer International Publishing, Cham, 2018.
- [129] M. Padmanabhan, S. Murali, F. Rincón, and D. Atienza. Energy-Aware Embedded Classifier Design for Real-Time Emotion Analysis. In *37th IEEE Annual International Conference of the Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy, 2015.
- [130] J. Pan and W. J. Tompkins. A Real-Time QRS Detection Algorithm. *IEEE transactions on bio-medical engineering*, BME-32(3):230–236, 1985.
- [131] Parrot. Bebop 2. <https://www.parrot.com/us/drones/parrot-bebop-2>.
- [132] Parrot. FreeFlight Pro. <https://www.parrot.com/global/freeflight-pro>.

- [133] J. M. Peschel and R. R. Murphy. On the human-machine interaction of unmanned aerial system mission specialists. *IEEE Transactions on Human-Machine Systems*, 2013.
- [134] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy. Computer vision based general object following for gps-denied multirotor unmanned vehicles. In *2014 American Control Conference*, pages 1886–1891, 2014.
- [135] L. Peternel, J. Babič, E. Oztop, T. Inamura, and D. Zhang. *Human-in-the-Loop Robot Control and Learning*. Frontiers Media SA, Lausanne, 2020.
- [136] P. Petrides, P. Kolios, C. Kyrkou, T. Theocharides, and C. Panayiotou. *Disaster Prevention and Emergency Response Using Unmanned Aerial Systems*, pages 379–403. Springer International Publishing, Cham, 2017.
- [137] K. Pfeil, S. L. Koh, and J. LaViola. Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles. In *International conference on Intelligent User Interfaces (IUI)*, 2013.
- [138] P. Phamduy, M. DeBellis, and M. Porfiri. Controlling a robotic fish via a natural user interface for informal science education. *IEEE Transactions on Multimedia*, 17(12):2328–2337, 2015.
- [139] Polar. Heart rate monitors, fitness trackers and gps sport watches. <https://www.polar.com>.
- [140] S. W. Porges. Cardiac vagal tone: A physiological index of stress. *Neuroscience & Biobehavioral Reviews*, 19(2):225–233, Jun 1995.
- [141] E. Prassler, G. Lawitzky, A. Stopp, G. Grunwald, M. Hägele, R. Dillmann, and I. Iossifidis. *Advances in Human-Robot Interaction*. Springer, Berlin, Heidelberg, 2014.
- [142] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis. The benefits of self-awareness and attention in fog and mist computing. *Computer*, 48(7):37–45, 2015.
- [143] W. H. Press and G. B. Rybicki. Fast Algorithm for Spectral Analysis of Unevenly Sampled Data. *Astrophysical Journal, Part 1*, 338:277–280, 1989.
- [144] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner. Networked UAVs as aerial sensor network for disaster management applications. *Elektrotechnik und Informationstechnik*, 127(3):56–63, 2010.

## Bibliography

---

- [145] M. Quigley, M. A. Goodrich, and R. W. Beard. Semi-autonomous human-uav interfaces for fixed-wing mini-uavs. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2457–2462. IEEE, 2004.
- [146] M. Ranchet, J. C. Morgan, A. E. Akinwuntan, and H. Devos. Cognitive workload across the spectrum of cognitive impairments: A systematic review of physiological measures. *Neuroscience & Biobehavioral Reviews*, 80:516–537, Sep 2017.
- [147] P. Rani, C. Liu, N. Sarkar, and E. Vanman. An empirical study of machine learning techniques for affect recognition in human-robot interaction. *Pattern Analysis and Applications*, 9(1):58–69, 2006.
- [148] G. B. Reid, S. S. Potter, and J. R. Bressler. Subjective Workload Assessment Technique (SWAT): A User’s Guide. *Distribution*, page 115, 1989.
- [149] L. Ricci, D. Formica, L. Sparaci, F. R. Lasorsa, F. Taffoni, E. Tamilia, and E. Guglielmelli. A new calibration methodology for thorax and upper limbs motion capture in children using magneto and inertial sensors. *Sensors (Basel, Switzerland)*, 14(1):1057–1072, 2014.
- [150] F. J. Rincon, L. Gutierrez, M. Jimenez, V. Diaz, N. Khaled, D. Atienza, M. Sanchez-Elez, J. Recas, and G. De Micheli. Implementation of an automated ecg-based diagnosis for a wireless body sensor platform. *Proceedings of the International Conference on Biomedical Electronics and Devices (BIODEVICES 2009)*, 1:88–96, 2009.
- [151] F. J. Rincón, M. Paselli, J. Recas, Q. Zhao, M. Sánchez-Elez, D. Atienza, J. Penders, and G. De Micheli. Os-based sensor node platform and energy estimation model for health-care wireless sensor networks. In *Proceedings of the Conference on Design, Automation and Test in Europe, DATE '08*, page 1027–1032, New York, NY, USA, 2008. Association for Computing Machinery.
- [152] F. Rincón, J. Recas, N. Khaled, and D. Atienza. Development and evaluation of multilead wavelet-based ecg delineation algorithms for embedded wireless sensor nodes. *IEEE Transactions on Information Technology in Biomedicine*, 15(6):854–863, 2011.
- [153] C. Rognon, S. Mintchev, F. DellAgnola, A. Cherpillod, D. Atienza, and D. Floreano. Flyjacket: An upper body soft exoskeleton for immersive drone control. *IEEE Robotics and Automation Letters*, 3(3):2362–2369, Jul 2018.



- [154] S. Saeedi, T. Carlson, R. Chavarriaga, and J. d. R. Millán. Making the most of context-awareness in brain-computer interfaces. *2013 IEEE International Conference on Cybernetics (CYBCONF)*, pages 68–73, 2013.
- [155] P. Salvini, M. Nicolescu, and H. Ishiguro. Benefits of human - robot interaction [tc spotlight]. *Robotics & Automation Magazine, IEEE*, 18:98–99, 12 2011.
- [156] Samsung. Galaxy Watch. <https://www.samsung.com/us/mobile/wearables/smartwatches>.
- [157] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri. A kinect-based natural interface for quadrotor control. *Entertainment Computing*, 4(3):179–186, Aug 2013.
- [158] Y. Santiago-Espada, R. R. Myer, K. A. Latorella, and J. R. Comstock. The Multi-Attribute Task Battery II ( MATB-II ) Software for Human Performance and Workload Research : A User’s Guide NASA/TM–2011-217164. Technical Report July, NASA, 2011.
- [159] F. Shaffer and J. P. Ginsberg. An overview of heart rate variability metrics and norms. *Frontiers in public health*, 5(258), Sep 2017.
- [160] H. Sharp, J. Preece, and Y. Rogers. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2019.
- [161] E. T. Solovey, M. Zec, E. A. Garcia Perez, B. Reimer, and B. Mehler. Classifying driver workload using physiological and driving performance data: two field studies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4057–4066. ACM, 2014.
- [162] Somniacs SA. Birdly. <https://birdly.com>.
- [163] D. Sopic, A. Aminifar, A. Aminifar, and D. Atienza Alonso. Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems. *IEEE Transactions on Biomedical Circuits and Systems*, 12(5):982–992, 2018.
- [164] H. I. Stern, J. P. Wachs, and Y. Edan. Designing hand gestures vocabularies for natural interaction by combining psycho-physiological and recognition factors. *Int. Journal of Semantic Computing, Special Issue on Gesture in Multi-modal Systems*, 2(1):137–160, 2008.

## Bibliography

---

- [165] STMicroelectronics. LSM9DS1. <https://www.st.com/en/mems-and-sensors/lsm9ds1.html>.
- [166] STMicroelectronics. STM32L151. <https://www.st.com/en/microcontrollers-microprocessors/stm32l151rd.html>.
- [167] A. Stoica, F. Salvioli, and C. Flowers. Remote control of quadrotor teams, using hand gestures. *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 296–297, 2014.
- [168] G. Surrel, A. Aminifar, F. J. Rincon Vallejos, S. Murali, and D. Atienza Alonso. Online obstructive sleep apnea detection on medical wearable sensors. *IEEE Transactions on Biomedical Circuits and Systems*, 12(4):762–773, 2018.
- [169] G. Surrel, F. Rincon, S. Murali, and D. Atienza. Design of ultra-low-power smart wearable systems. In *16th Latin-American Test Symposium (LATS)*, pages 1–2, 2015.
- [170] G. Surrel, T. Teijeiro, M. Chevrier, A. Aminifar, and D. Atienza. Event-triggered sensing for high-quality and low-power cardiovascular monitoring systems. *IEEE Design & Test*, 2019.
- [171] I. E. Sutherland. Sketchpad a man-machine graphical communication system. *Simulation*, 2(5):R–3, 1964.
- [172] Suunto. Sport watch. <https://www.suunto.com>.
- [173] N. TaheriNejad, A. Jantsch, and D. Pollreis. Comprehensive observation and its role in self-awareness; an emotion recognition system example. In *FedCSIS*, pages 117–124, 2016.
- [174] Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology. Heart rate variability: Standards of measurement, physiological interpretation, and clinical use. *Circulation*, 93(5):1043–1065, Mar 1996.
- [175] K. H. Teigen. Yerkes-dodson: A law for all seasons. *Theory & Psychology*, 4(4):525–547, Nov 1994.
- [176] L. Teixeira, I. Alzugaray, and M. Chli. Autonomous aerial inspection using visual-inertial robust localization and mapping. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*, 2017.

- [177] M. Terzi, A. Anastasiou, P. Kolios, C. Panayiotou, and T. Theocharides. Swifters: A multi-uav platform for disaster management. In *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–7, 2019.
- [178] N. Tesla. Method of and apparatus for controlling mechanism of moving vessels or vehicles. <http://www.google.com/patents/US613809>, U.S. Patent US1898684934, Jul. 1898. Online; accessed 12 May 2020.
- [179] Texas Instruments. ADS1292R. <https://www.ti.com/lit/gpn/ADS1292R>.
- [180] Texas Instruments. AFE4300. <http://www.ti.com/lit/gpn/AFE4300>.
- [181] Texas Instruments. AFE4490. <http://www.ti.com/lit/gpn/afe4490>.
- [182] Texas Instruments. LMT70. <https://www.ti.com/lit/gpn/lmt70>.
- [183] J. F. Thayer, F. Ahs, M. Fredrikson, J. J. Sollers III, and T. D. Wager. A meta-analysis of heart rate variability and neuroimaging studies: Implications for heart rate variability as a marker of stress and health. *Neuroscience and Biobehavioral Reviews*, 36(2):747–756, 2012.
- [184] The MathWorks Inc., Natick, Massachusetts. Matlab R2016a. <https://www.mathworks.com>.
- [185] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras. *Introduction to Pattern Recognition*. Academic Press, Boston, 2010.
- [186] Thought Technology Ltd. Biofeedback, neurofeedback and psychophysiological instrument manufacturer. <http://thoughttechnology.com>.
- [187] A. Tjolleng, K. Jung, W. Hong, W. Lee, B. Lee, H. You, J. Son, and S. Park. Classification of a driver’s cognitive workload levels using artificial neural network on ecg signals. *Applied Ergonomics*, 59:326–332, Mar 2017.
- [188] M. Toichi, T. Sugiura, T. Murai, and A. Sengoku. A new method of assessing cardiac autonomic function and its comparison with spectral analysis and coefficient of variation of R-R interval. *Journal of the Autonomic Nervous System*, 62(1-2):79–84, Jan 1997.
- [189] L. Tonin, R. Leeb, M. Tavella, S. Perdikis, and J. d. R. Millán. The role of shared-control in bci-based telepresence. *Ieee International Conference On Systems, Man And Cybernetics (Smc 2010)*, pages 1462–1466, 2010.

## Bibliography

---

- [190] Ultraleap. Leap Motion. <https://www.ultraleap.com/product/leap-motion-controller>.
- [191] M. van Dooren, G.-J. de Vries, and J. H. Janssen. Emotional sweating across the body: comparing 16 different skin conductance measurement locations. *Physiology & behavior*, 106(2):298–304, may 2012.
- [192] Vicon. Optical camera system for motion capture. <https://www.vicon.com>.
- [193] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1(6):80–83, 1945.
- [194] G. F. Wilson. An analysis of mental workload in pilots during flight using multiple psychophysiological measures. *The International Journal of Aviation Psychology*, 12(1):3–18, Jan 2002.
- [195] G. Wu and P. R. Cavanagh. ISB Recommendations in the Reporting for Standardization of Kinematic Data. *Journal of Biomechanics*, 28(10):1257–1261, 1995.
- [196] G. Wu, F. C. T. Van Der Helm, H. E. J. Veeger, M. Makhssous, P. Van Roy, C. Anglin, J. Nagels, A. R. Karduna, K. McQuade, X. Wang, F. W. Werner, and B. Buchholz. ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion - Part II: Shoulder, elbow, wrist and hand. *Journal of Biomechanics*, 38(5):981–992, 2005.
- [197] R. Zanetti, A. Aminifar, and D. Atienza Alonso. Robust epileptic seizure detection on wearable systems with reduced false-alarm rate. *42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Montréal, Québec, CA, page 4, 2020.
- [198] S. Zanolli, T. Teijeiro, F. Montagna, and D. Atienza Alonso. An event-based system for low-power ecg qrs complex detection. *Design, Automation, and Test in Europe (DATE) Conference and Exhibition*, page 6, 2020.

# Acronyms

CNBI	Center for Neuroprosthetics and Bioengineering Institute. ii
ESL	Embedded System Laboratory. ii, 28, 150, 153
LIS	Laboratory of Intelligent Systems. ii, 28, 40
TNE	Translational Neural Engineering. ii, 28, 31, 37
SPAI	Scuola Professionale Artigianale Industriale. ii
EPFL	École Polytechnique Fédérale de Lausanne. i–iii, 17, 23, 28, 146, 153
ETHZ	Eidgenössische Technische Hochschule Zürich. ii
SSST	Scuola Specializzata Superiore di Tecnica. i, ii
SUPSI	Scuola Universitaria Professionale della Svizzera Italiana. i, ii
UNIL	Université de Lausanne. i
ACI	Application Controller Interface. 119
ADC	Analog-to-Digital Converter. xxi, 35, 119, 123
ANN	Artificial Neural Network. 65, 67
ANOVA	Analysis of Variance. 87, 88
ANS	Autonomic Nervous System. 70
ASIC	Application-Specific Integrated Circuit. 151
AUC	Area Under the Curve. 92
BCM	Body Composition Monitor. 119
BLE	Bluetooth Low Energy. 33, 35, 57, 58, 119, 121, 129, 131, 151

## Acronyms

---

CPU	Central Processing Unit. 120
CSI	Cardiac Sympathetic Index. 74
CVI	Cardiac Vagal Index. 74
DFU	Device Firmware Upgrade. 36, 119
DoF	Degree of Freedom. 4, 30, 33, 59, 147, 148
DT	Decision Tree. 67, 79, 91, 94
ECG	Electrocardiogram. 31, 42, 43, 51, 65, 66, 68–75, 87, 93, 99, 101, 104, 105, 107, 112, 118, 121, 123–125, 127, 129–132, 136–139, 150, 151
EDA	Electrodermal Activity. 31, 42, 43, 65, 66, 69, 73, 87, 101, 104, 105, 107, 119
EEG	Electroencephalogram. 64, 87, 104, 105
EMG	Electromyogram. 26, 42, 43
Exh	Exhalation. 72, 73
FDR	Fisher Discriminant Ratio. 77, 78, 89
FFT	Fast Fourier Transform. 126, 136, 137
FIR	Finite Impulse Response. 71, 124
FPV	First-Person View. 39, 41, 43
GNB	Gaussian Naive Bayes. 78, 91
GUI	Graphical User Interface. 8
HF	High Frequency. 32, 74
HMD	Head-Mounted Display. 39, 41–43, 45, 46, 50
HMI	Human-Machine Interface. 4, 5, 21–24, 27
HPA	Hypotalai-Adrena. xxi, 70
HR	Heart Rate. 129
HRV	Heart Rate Variability. 32, 70, 74, 75, 106, 129
HW	hardware. 128
I2C	Inter-Integrated Circuit. 35, 119
ICG	Impedance Cardiogram. 119
IIR	Infinite Impulse Response. 71

IMU	Inertial Measurement Unit. xxi, 29, 30, 33–37, 51, 115, 150
Inh	Inhalation. 72, 73
IPL	Interrupt Priority Level. 122, 123
IRQ	Interrupt Request. 122, 123
ISR	Interrupt Service Routine. 122–124, 127
k-NN	k-Nearest Neighbors. 79, 91, 92
LDA	Linear Discriminant Analysis. 65, 67, 78, 90–92
LF	Low Frequency. 32, 74
LR	Logistic Regression. 65, 67, 78, 90–92
MCU	Microcontroller Unit. xxi, 32–34, 36, 37, 105, 115, 116, 118–121, 123, 131, 133, 135, 136, 138–140, 151
ML	Machine-Learning. 68
MSB	Most Significant Bit. 123
NASA-TLX	NASA Task Load Index. 69, 85, 87, 88
NN	Normal-to-Normal. 74
NUI	Natural User Interface. xvii, 9–11, 17, 24–27, 29
PA	Pulse Amplitude. 72, 75
PP	Pulse Period. 75
PPG	Photoplethysmogram. 65, 66, 68–73, 75, 87, 93, 99, 101, 104, 105, 107, 112, 118, 121, 123–125, 127, 129–132, 136–138, 140, 150
Prd	Period. 72
PRS	Pulse Rising Speed. 75
PRT	Pulse Rising Time. 72, 75, 125
PRV	Pulse Rate Variability. 129
PSD	Power Spectral Density. 32, 72, 74, 123, 126, 136
PSNS	Parasympathetic Nervous System. xxi, 70, 76
PTT	Pulse Transit Time. 72
QDA	Quadratic Discriminant Analysis. 79, 91, 92

## Acronyms

---

RAM	Random Access Memory. 121
RBF	Radial Basis Function. 79, 91, 92
RF	Random Forest. 65, 67, 78, 79, 91, 92
RFE	Recursive Features Elimination. xxi, 77, 78, 89–92, 94, 95
RMS	Root Mean Square. 38, 53, 55, 126
RMSE	Root Mean Square Error. xviii, 39, 53, 55, 56
RMSSD	Root-Mean Square of Successive Differences. 32
ROC	Receiver Operating Characteristic. 80, 89, 92, 94
RSA	Respiratory Sinus Arrhythmia. 72, 73, 75, 76
RSP	Respiratory Activity. 65, 66, 68–73, 76, 87, 93, 99, 101, 104, 105, 107, 112, 118, 119, 121, 123–125, 127, 130, 131, 133, 136–139, 150
RTC	Real-Time Clock. 120, 121
SAR	Search and Rescue. 1–6, 10, 11, 13, 14, 16, 17, 24, 26, 27, 49, 50, 57, 59, 62, 65, 103, 146, 148, 149
SKT	Skin Temperature. 65, 66, 68–73, 75, 87, 93, 99, 101, 104, 105, 107, 112, 119, 121, 130, 137–139, 150
SNS	Sympathetic Nervous System. xxi, 70
SoC	System on Chip. 34, 107
SPI	Serial Peripheral Interface. 33–35, 118, 119, 129
SSM	Subject-Specific Model. xviii, xxi, 68, 79–82, 95, 96, 98, 99
SVM	Support Vector Machine. 17, 62, 65, 67, 77–80, 89–92, 94, 95, 98, 100, 109, 112, 113, 126, 127, 131, 146
SW	software. 128, 150
SWD	Serial Wire Debug. 35, 119
UAV	Unmanned Aerial Vehicle. 2, 4, 7, 8, 10, 21
UBM	Universal Background Model. xviii, xxi, 68, 79, 80, 95–99
ULP	Ultra-Low Power. 115



USART	Universal Synchronous and Asynchronous Receiver-Transmitter. 34, 35, 118, 119, 129
USB	Universal Serial Bus. 34, 36, 118–120
VLF	Very Low Frequency. 74
VR	Virtual Reality. 8, 39, 40, 44–46, 82
XGB	Extreme Gradient Boosting. 65, 67, 78, 79, 91, 92, 148



# Fabio Dell'Agnola

6760 Faido, Switzerland  
fabio.dellagnola@ieee.org

Swiss Citizen  
Born on December 10, 1979



## Profile

---

Biography	I received both my M.Sc. and Ph.D. degrees in electrical engineering from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2013 and 2020, respectively. Before that, I received a B.Sc. degree in electrical and electronic engineering from the Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), in 2010.
Expertise	I have expertise in System Dynamics and Control, Embedded Machine Learning, Embedded Systems Design, Edge Computing Design, Low-Power Hardware and Software Co-Design, Wireless Body Sensor Networks (WBSN), and Internet of Things (IoT).
Awards	For the outstanding dissertation in electrical engineering and the excellent academic results achieved at SUPSI (i.e., best B.Sc. student) in 2010, I received the <b>Electrosuisse-Prize</b> and " <b>Premio dell'economia ticinese</b> ", respectively. Moreover, I received the <b>AMETI-Prize</b> for completing the Scuola Specializzata Superiore di Tecnica (SSST) in electronics as the 3rd best student in 2008.
Languages	Italian (native), French (fluent), English (fluent), and German (fluent).

## Professional experiences

---

2003 - 2008	<b>Electromechanical Technician</b> <i>Centro manutenzione autostradale, ala Munda, Camorino, Switzerland.</i> I participated in the regular maintenance of the electromechanical installations of the Gotthard road tunnel. Moreover, I managed the remote control system of the electromechanical installations located throughout the entire canton Tessin.
1999 - 2003	<b>Bus and truck driver</b> <i>La Posta, by Domenico Barenco, via Cantonale, Faido, Switzerland.</i>
1998 - 1999	<b>Electrician</b> <i>Azienda Elettrica Ticinese, Rodi, Switzerland.</i> I participated in the regular maintenance works of the hydroelectric power plant and worked on the realization of the new cable car.

1994 - 1998	<b>Electrical apprenticeship</b> <i>Ellettronord SA</i> , by Andrea Maffezzini, Faido, Switzerland. I worked on building construction, on the realization of electrical distribution panels, in troubleshooting and repair services.
-------------	--

## Education

---

2015 - 2020	<b>Ph.D. in Electrical Engineering</b> <i>École Polytechnique Fédérale de Lausanne (EPFL)</i> , Switzerland. Thesis title: <i>Wearable and Self-Aware Machine Learning System for Online Cognitive Workload Monitoring and Drone Control</i> . Supervision: Prof. D. Barrettino and Prof. D. Atienza
2010 - 2013	<b>M.Sc. in Electrical Engineering, Specialization in Energy</b> <i>École Polytechnique Fédérale de Lausanne (EPFL)</i> , Switzerland. Thesis title: <i>Development of a real-time wind turbine emulator system</i> . Supervision: Dr. S. Mariéthoz and Prof. C. N. Jones
2008 - 2010	<b>B.Sc. in Electrical Engineering, Specialisation in Mechatronics</b> <i>Scuola Universitaria Professionale della Svizzera Italiana (SUPSI)</i> , Manno, Switzerland. Thesis title: <i>Long-term stabilization of a Fabry-Perot resonator for positronium spectroscopy</i> . Supervision: Dr. P. Crivelli and Prof. S. Balemi
2005 - 2008	<b>Tech. dipl. SSS in Electrical Engineering</b> <i>Scuola Specializzata Superiore di Tecnica (SSST)</i> , Bellinzona, Switzerland. Thesis title: <i>Gestione Impianti Strade Cantonali</i> . Supervision: S. Snozzi
1994 - 1998	<b>Electrical apprenticeship</b> <i>Scuola Professionale Artigianale Industriale (SPAI)</i> , Biasca, Switzerland.

## Teaching Experiences

---

2015 - 2020	<b>Research Assistant, Embedded System Laboratory,</b> <i>École Polytechnique Fédérale de Lausanne (EPFL)</i> , Switzerland. Teaching assistance as well as projects supervision.
2013 - 2014	<b>Research Assistant, Automatic Control Laboratory,</b> Eidgenössische Technische Hochschule Zürich (ETHZ), Switzerland. Experimental investigations of the dynamic coupling that takes place between wind turbine rotors and electrical generators.
2011 - 2012	<b>Student Assistant, Electrical Machines Laboratory,</b> <i>École Polytechnique Fédérale de Lausanne (EPFL)</i> , Switzerland. Supervised students during their exercise sessions for the course <i>Electrical Machines</i> .

## Publications

---

Journals	<p>F. Dell'Agnola, U. Pale, A. Arza, and D. Atienza. <b>Wearable Embedded System for Multimodal Cognitive Workload Monitoring</b>. <i>IEEE Transactions on Biomedical Circuits and Systems</i>, In preparation.</p> <p>F. Dell'Agnola, P.-K. Jao, A. Arza, R. Chavarriaga, J. d. R. Millán, D. Floreano, and D. Atienza. <b>Machine Learning Based Monitoring of Cognitive Workload in Rescue Missions with Drones</b>. <i>IEEE Transaction on Affective Computing</i>, In preparation.</p> <p>G. Masinelli, A. Arza, F. Dell'Agnola, and D. Atienza. <b>SPARE, SPectral peAk REcovery: full PPG pulsewave reconstruction</b>. <i>IEEE Journal of Biomedical and Health Informatics</i>, In preparation.</p> <p>P.-K. Jao, R. Chavarriaga, F. Dell'Agnola, A. Arza, D. Atienza, and J. d. R. Millán. <b>EEG Correlates of Difficulty Levels in Dynamical Transitions of Simulated Flying and Mapping Tasks</b>. <i>IEEE Transactions on Human-Machine Systems</i>, Accepted on October 2020.</p> <p>C. Rognon, S. Mintchev, F. Dell'Agnola, A. Cherpillod, D. Atienza and D. Floreano. <b>FlyJacket: an Upper-Body Soft Exoskeleton for Immersive Drone Control</b>. <i>IEEE Robotics and Automation Letters</i>, Vol. 3, No. 3, pp. 2362-2369, July 2018. Video: <a href="https://youtu.be/L0FTPYkLKHl">youtu.be/L0FTPYkLKHl</a></p>
Conferences	<p>F. Dell'Agnola, N. Momeni, A. Arza, and D. Atienza. <b>Cognitive workload monitoring in virtual reality based rescue missions with drones</b>. <i>22nd International Conference on Human-Computer Interaction (HCII)</i>, Copenhagen, Denmark, 2020.</p> <p>N. Momeni, F. Dell'Agnola, A. Arza, and D. Atienza. <b>Real-Time Cognitive Workload Monitoring Based on Machine Learning Using Physiological Signals in Rescue Missions</b>. <i>41st International Engineering in Medicine and Biology Conference (EMBC)</i>, Berlin, Germany, 2019.</p> <p>V. Montesinos, F. Dell'Agnola, A. Arza, and D. Atienza. <b>Multi-Modal Acute Stress Recognition Using Off-the-Shelf Wearable Devices</b>. <i>41st International Engineering in Medicine and Biology Conference (EMBC)</i>, Berlin, Germany, 2019.</p> <p>F. Dell'Agnola, L. Cammoun, and D. Atienza. <b>Physiological Characterization of Need for Assistance in Rescue Missions with Drones</b>. <i>IEEE International Conference on Consumer Electronics (ICCE)</i>, Las Vegas, USA, 2018.</p>

Patents	<p>J. Miehlsbradt, F. Dell'Agnola, A. Cherpillod, M. Coscia, F. Artoni, S. Mintchev, D. Floreano, D. Atienza, S. Micera. <b>Teleoperation with a wearable sensor system</b>. <i>UK Patent Application</i>, No. 1810285.5, filed on June 22, 2018.</p> <p>C. Rognon, D. Floreano, S. Mintchev, A. Concordel, F. Dell'Agnola, D. Atienza. <b>Jacket for Embodied Interaction with Virtual or Distal Robotic Device</b>. <i>International Patent Application</i>, No. PCT/IB2017/055410, filed on September 8, 2017.</p>
---------	---

## Other Experiences

---

since 1997	Sergeant Major of the Faido Fire Department, Certified Swiss Federal Instructor on Basic Training, Self-Contained Breathing Apparatus (SCBA), and Technical and Tactical Intervention.
since 2013	Member of the Swiss Alpine Club (SAC), organizing different activities, such as ski touring, mountaineering, and climbing.
since 2014	International Snow Sports Instructor and J+S Ski Expert

## Annexes

---

References available upon request.

Last updated: November 5, 2020

