UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

University of Modena and Reggio Emilia

Department of Sciences and Methods for Engineering

Doctoral school of Industrial Innovation Engineering

XXXVIII cycle

Social acceptance in robotics: from biosignal-based user understanding to intelligent social navigation

---

PhD Candidate:
Andrea Ruo

Supervisors:
Valeria Villani
Lorenzo Sabattini

PhD Coordinator:
Franco Zambonelli

March 2026

Reggio Emilia, Italy, March 2026

To my family

Acknowledgements

Diverse ragioni mi hanno spinto a intraprendere questo percorso:

- *capire come oggi affrontiamo e risolviamo i problemi, e con quali tecnologie;*
- *approfondire la differenza tra il "si fa così" e il "perché si fa così";*
- *non dimenticare un consiglio ricevuto anni fa: "hai tutta la vita per fare un lavoro che non ti piacerà fare".*

A pochi anni dalla mia Laurea Magistrale desidero ringraziare ancora una volta i miei tutor, Prof.ssa Valeria Villani e Prof. Lorenzo Sabattini, per avermi guidato, supportato e formato durante questo percorso, trascorso purtroppo fin troppo velocemente.

Ringrazio i miei genitori, Marco e Marina, per i sacrifici, la dedizione e il sostegno costante che mi hanno permesso di raggiungere questo nuovo traguardo personale e professionale.

Ringrazio mia sorella, Valentina, per i consigli, la vicinanza e il supporto che non sono mai mancati in tutti questi anni.

Un ringraziamento speciale a Marianna, per avermi supportato e sopportato, e per la gioia e la serenità che ogni giorno porta nella mia vita.

Ringrazio Vincenzo per i mesi condivisi, intensi e pieni di emozioni; per le lunghe chiacchierate notturne e per i consigli che hanno arricchito questo cammino.

Ringrazio Jonathan per avermi fatto sentire più a mio agio in un periodo in cui ero lontano da casa, per chiedermi come stavo ogni mattina, per le risate e per le competenze professionali che mi ha trasmesso.

Infine, grazie a tutti gli amici che hanno condiviso con me momenti di leggerezza e incoraggiamento, fondamentali per arrivare fino a qui.

Abstract

The integration of robotic systems into environments shared with people requires that robots are not only safe and functional, but also socially acceptable. Social acceptance is defined as the extent to which users feel safe, understood, and comfortable when interacting with a robot, and depends on factors such as trust, transparency, compliance with social norms, and adaptability.

This thesis addresses the topic by first proposing a reference model based on three components: definitions, requirements, and architecture. The requirements include both functional and non-functional aspects, such as usability, perceived usefulness, safety, privacy, and cultural alignment. From an architectural perspective, a socially acceptable robot must integrate multimodal perception, awareness of the surrounding context, safety, and adaptive control.

Within this model, the research presented in this work focuses on two main aspects: user understanding and social navigation.

User understanding concerns the ability of the robot to detect and interpret cognitive and emotional states, in order to personalize interaction and enhance user comfort and trust. To this end, classification methods based on multimodal sensors and biosignal analysis (EEG, ECG, GSR, facial expressions) were developed to estimate mental workload and stress levels in human-robot collaboration scenarios. The studies confirmed the validity of physiological indices as predictors of cognitive and emotional states and led to the creation of a public multimodal dataset dedicated to collaborative robotics. These tools allow the robot to dynamically adapt its operational parameters, thereby improving the quality of interaction.

Social navigation, which is crucial for mobile robots operating in dynamic and populated environments, was addressed through the development of a control framework capable of integrating spatial and temporal constraints to ensure socially acceptable behavior. The approach combines signal temporal logic (STL) with control barrier functions (CBFs), guaranteeing compliance with specifications and safety of interaction even under uncertainty and in the presence of dynamic obstacles. The framework was extended with ad-

ditional constraints to maintain the user within the robot's field of view and with robust formulations that incorporate filtering techniques, ensuring reliability even in complex real-world scenarios.

In parallel, a cognitive architecture was proposed that integrates large language models (LLMs) to interpret heterogeneous inputs (emotional state, intentions, social context) and translate them into navigation parameters. This solution enables the robot to dynamically and contextually adapt its movement strategy, leading to the concept of intelligent social navigation. In addition, a lightweight architecture was developed for social guidance tasks, combining facial recognition, gesture detection, and skeleton tracking, in order to guarantee transparent and reactive behavior in public environments.

Overall, the thesis demonstrates that social acceptance is not a static property of the robot, but an emergent phenomenon resulting from interaction with the user. The proposed solutions for user understanding and social navigation lay the groundwork for robots capable of acting effectively, safely, and adaptively, and ultimately being accepted and trusted by people in real-world contexts.

Sommario

L'integrazione dei sistemi robotici negli ambienti condivisi con le persone richiede che i robot non siano soltanto sicuri e funzionali, ma anche socialmente accettabili. L'accettazione sociale viene definita come il grado in cui gli utenti si sentono al sicuro, compresi e a proprio agio nell'interazione con un robot, e dipende da fattori quali fiducia, trasparenza, rispetto delle norme sociali e adattabilità.

La tesi affronta tale tema proponendo inizialmente un modello di riferimento fondato su tre componenti: definizioni, requisiti e architettura. I requisiti comprendono aspetti funzionali e non funzionali, quali usabilità, utilità percepita, sicurezza, privacy e allineamento culturale; dal punto di vista architetturale, un robot socialmente accettabile deve integrare percezione multimodale, consapevolezza del contesto che lo circonda, sicurezza e controllo adattivo.

All'interno di questo modello, il lavoro di ricerca di questo elaborato si concentra su due aspetti principali: comprensione dell'utente e navigazione sociale.

La comprensione dell'utente riguarda la capacità del robot di rilevare e interpretare stati cognitivi ed emotivi, per personalizzare l'interazione e incrementare comfort e fiducia con l'utente. A tale scopo, sono stati sviluppati metodi di classificazione basati su sensori multimodali e analisi di biosignali (EEG, ECG, GSR, espressioni facciali), finalizzati a stimare carico mentale e livelli di stress in scenari di collaborazione uomo-robot. Gli studi hanno evidenziato la validità di indici fisiologici come predittori dello stato cognitivo ed emotivo, e hanno portato alla realizzazione di un dataset multimodale pubblico dedicato alla robotica collaborativa. Questi strumenti consentono al robot di adattare dinamicamente parametri operativi migliorando la qualità dell'interazione.

La navigazione sociale, fondamentale per i robot mobili in ambienti dinamici e popolati, è stata affrontata attraverso lo sviluppo di un framework di controllo capace di integrare vincoli spaziali e temporali assicurando che il robot abbia comportamenti socialmente accettabili. L'approccio combina Signal Temporal Logic (STL) con Control Barrier Functions (CBFs), garantendo il rispetto delle specifiche e la sicurezza dell'interazione anche in condizioni di incertezze e presenza di ostacoli dinamici. Il framework è stato esteso con

vincoli addizionali per il mantenimento del campo visivo dell'utente e con formulazioni robuste che integrano tecniche di filtraggio, assicurando affidabilità anche in scenari reali complessi.

Parallelamente, è stata proposta un'architettura cognitiva che integra modelli linguistici di grandi dimensioni (LLMs) per interpretare input eterogenei (stato emotivo, intenzioni, contesto sociale) e tradurli in parametri di navigazione, arrivando così al concetto di navigazione sociale intelligente. Questa soluzione abilita il robot ad adattare la propria strategia di movimento in modo dinamico e contestuale. Inoltre, è stata sviluppata un'architettura leggera per compiti di guida sociale, basata su riconoscimento facciale, rilevamento gestuale e tracciamento scheletrico, in grado di garantire un comportamento trasparente e reattivo in ambienti pubblici.

Nel complesso, la tesi dimostra che l'accettazione sociale non è una proprietà statica del robot, ma un fenomeno emergente dall'interazione con l'utente. Le soluzioni proposte per la comprensione dell'utente e la navigazione sociale pongono le basi per robot capaci di agire in modo efficace, sicuro e adattivo, ma soprattutto accettato e apprezzato dalle persone nei contesti reali di utilizzo.

Contents

List of Figures

List of Publications

Journal papers

[1] A. Ruo, L. Sabattini, and V. Villani, "Deployment of a CBF-based STL social navigation framework on omnidirectional mobile robots," in *Submitted to: Robotics and Automation Practice (RA-P)*, 2025

[2] A. Ruo, L. Bernardi, L. Campanelli, M. Grespan, D. Trane, R. Sedoni, D. Angeli, L. Sabattini, and V. Villani, "A low-cost 3D printed electromagnetic gripper for robotic arms," *Mechatronics*, vol. 110, p. 103374, 2025

[3] S. Borghi, A. Ruo, L. Sabattini, M. Peruzzini, and V. Villani, "Assessing operator stress in collaborative robotics: A multimodal approach," *Applied Ergonomics*, vol. 123, p. 104418, 2025

Conference papers

[4] A. Ruo, D. Napolitano, and V. Villani, "A wearable stereo vision-based obstacle detection system for visually impaired individuals," in *Workshop on Agents and Robots for reliable Engineered Autonomy*. Springer, 2025, pp. 35–50

[5] A. Ruo, J. Cacace, M. Dalmau-Moreno, L. Sabattini, and V. Villani, "An LLM-based architecture for socially intelligent robot navigation based on social cues," in *2025 34th IEEE international conference on robot and human interactive communication (RO-MAN)*. IEEE, 2025

[6] A. Ruo, L. Sabattini, and V. Villani, "Robust CBF-based STL motion planning for socially responsible robot navigation in the presence of measurement noise," in *2025 American Control Conference (ACC)*, 2025

[7] A. Ruo, S. Arreghini, L. Capra, R. De Chiara, V. Di Pasquale, A. Giusti, C. Iani, A. Paolillo, D. Petrak, A. Plaum, M. Quamara, L. Sabattini, V. Schmuck, P. Servillo, F. Zurolo, and V. Villani, "Systematic analysis of requirements for socially acceptable service robots," in *17th International Workshop on Human-Friendly Robotics*, 2024

[8] A. Coppari, S. Proia, A. Ruo, F. Favali, L. Sabattini, C. Secchi, V. Villani, M. Piazzola, and L. Capra, "A large language model-based motion planning for human-

robot interaction: An experimental case study," in *17th International Workshop on Human-Friendly Robotics*, 2024

[9] M. Quamara, L. Capra, V. Villani, C. Carlevaro, O. Celiktutan, A. Peretti, M. Piazzola, A. Ruo, L. Sabattini, V. Schmuck, *et al.*, "Towards a modular architecture for extended reality systems," in *8th International Conference on Artificial Intelligence and Virtual Reality (2024)*. Springer, 2024

[10] A. Ruo, L. Sabattini, and V. Villani, "CBF-based motion planning for socially responsible robot navigation guaranteeing STL specification," in *2024 European Control Conference (ECC)*, 2024, pp. 122–127

[11] ——, "CBF-based STL motion planning for social navigation in crowded environment," in *European Robotic Forum*, 2024

[12] ——, "Follow me: an architecture for user identification and social navigation with a mobile robot." in *European Robotic Forum*, 2024

[13] S. Borghi, F. Zucchi, E. Prati, A. Ruo, V. Villani, L. Sabattini, and M. Peruzzini, "Unlocking human-robot dynamics: Introducing sensecobot, a novel multimodal dataset on industry 4.0," in *2024 19th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2024, pp. 880–884

[14] A. Ruo, V. Villani, and L. Sabattini, "Use of EEG signals for mental workload assessment in human-robot collaboration," in *Human-Friendly Robotics 2022*, P. Borja, C. Della Santina, L. Peternel, and E. Torta, Eds. Cham: Springer International Publishing, 2023, pp. 233–247

Chapter 1

Introduction

The increasing integration of robotic systems into human-shared environments has underscored the importance of ensuring that robots behave in a way that is not only safe and functional, but also **socially acceptable**. As robots become part of everyday life, guiding users in public spaces, assisting the elderly, or supporting workers in industrial settings, they must navigate both physical and social environments while respecting human norms, expectations, and comfort.

In this context, the **European project Socially-Acceptable Extended Reality Models and Systems** (**SERMAS**) [15] plays a pivotal role. SERMAS aims to develop innovative, formal, and systematic methodologies and technologies for modeling, developing, analyzing, testing, and studying the use of socially acceptable advanced **eXtended Reality** (**XR**) **systems**. Importantly, the project is not limited to virtual or augmented environments: the methodologies and architectures are conceived to encompass a spectrum of XR agents, ranging from purely virtual avatars to embodied physical agents such as robots.

Building on the objectives of SERMAS, this thesis addresses the multifaceted concept of **social acceptance in robotics**, as shown in Fig. 1.1, proposing a structured perspective grounded in three pillars: **definitions**, **requirements**, and **architectures**. Social acceptance is defined as the extent to which users feel safe, understood, and comfortable in their interaction with robots, including trust, transparency, ethical behavior, and adaptability. Requirements for social acceptance are grouped into functional and non-functional aspects and include usability, perceived usefulness, cultural alignment, safety, privacy, and compliance with social norms [7]. Architecturally, a socially acceptable XR system must be equipped with multimodal perception capabilities, context awareness, human modeling, and adaptive control [9].

The study of this topic has produced the following publications:

[7] **Systematic analysis of requirements for socially acceptable service robots**
A. Ruo, S. Arreghini, L. Capra, R. De Chiara, V. Di Pasquale, A. Giusti, C. Iani, A. Paolillo, D. Petrak, A. Plaum, M. Quamara, L. Sabattini, V. Schmuck, P. Servillo, F. Zurolo, and V. Villani
17th International Workshop on Human-Friendly Robotics (HFR), 2024.

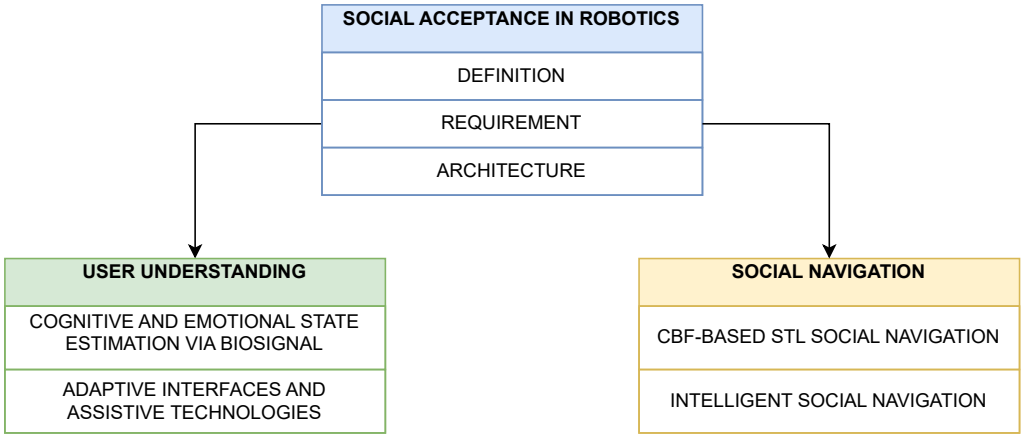


Figure 1.1: Overview of the thesis structure, highlighting the main research pillars and their interconnections.

[9] **Towards a modular architecture for extended reality systems**
M. Quamara, L. Capra, V. Villani, C. Carlevaro, O. Celiktutan, A. Peretti, M. Piazzola, A. Ruo, L. Sabattini, V. Schmuck, et al.
8th International Conference on Artificial Intelligence and Virtual Reality, 2024.

Among the various factors that contribute to the social acceptance of a robot, this thesis focuses on two key aspects: **user understanding** and **social navigation**, which are addressed in parallel to ensure socially intelligent behavior, as shown in Fig. 1.1.

The first key component of social acceptance is **user understanding**, which focuses on how a robot can perceive and interpret the cognitive and emotional states of its users. By gaining awareness of individual characteristics and behavioral patterns, the robot can personalize its behavior and improve interaction fluency, comfort, and trust.

To this end, this thesis explores user understanding through **multimodal sensing** and **biosignal analysis**. In particular, we leverage physiological and behavioral data to estimate user workload, stress levels, and interaction strategies. Such information is essential to adapt the robot's control parameters such as speed, distance, and trajectory in a socially intelligent manner. We first investigated mental workload using electroencephalographic (EEG) signals, comparing cognitive responses in various tasks including a human-robot collaboration (HRC) scenario. EEG-based indices such as *Theta*, *Alpha*, and *Beta* power, *Beta*/*Alpha* ratio, and *Theta*/*Alpha* ratio proved to be reliable indicators of mental effort, especially in tasks involving attention and decision-making during collaboration [14]. These findings provide a foundation for real-time monitoring of cognitive state during HRC. Building on this, we introduced SenseCobot, a publicly available multimodal dataset for collaborative robotics, which includes synchronized recordings of EEG, electrocardiogram (ECG), Galvanic skin response (GSR), facial expressions, and task performance from 21 users involved in progressive cobot programming tasks. Through this dataset, we extracted several stress-related indicators and demonstrated strong correlations between biosignal metrics (e.g., GSR recovery time, EEG Alpha asymmetry, HRV parameters) and perceived workload levels as assessed via NASA-TLX questionnaires [13]. A complementary study validated these results in a

realistic setting, proposing a multimodal methodology for real-time stress classification during robot programming, enabling the assessment of operator stress in HRC scenarios, and demonstrating how EEG, ECG, GSR, and facial expression parameters can be interpreted to detect human stress conditions in HRC [3]. Finally, recognizing the need to adapt interaction not only cognitively but also physically, we developed some **adaptive interfaces** and **assistive technologies** such as a low-cost electromagnetic robotic gripper, optimized for collaborative settings. Experimental validation showed that the proposed gripper can successfully handle a range of object sizes and weights [2]. In parallel, we explored assistive technologies for visually impaired users, implementing a wearable stereo vision system capable of detecting and verbally indicating obstacles in real-time, enhancing user autonomy and safety [4].

The study of these topics has produced the following publications:

[14] **Use of EEG signals for mental workload assessment in human-robot collaboration**
A. Ruo, V. Villani, and L. Sabattini
Human-Friendly Robotics (HFR), 2022.

[13] **Unlocking human-robot dynamics: Introducing SenseCobot, a novel multimodal dataset on industry 4.0**
S. Borghi, F. Zucchi, E. Prati, A. Ruo, V. Villani, L. Sabattini, and M. Peruzzini
19th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2024.

[3] **Assessing operator stress in collaborative robotics: A multimodal approach**
S. Borghi, A. Ruo, L. Sabattini, M. Peruzzini, and V. Villani
Applied Ergonomics, vol. 123, p. 104418, 2025.

[2] **A low-cost 3D printed electromagnetic gripper for robotic arms**
A. Ruo, L. Bernardi, L. Campanelli, M. Grespan, D. Trane, R. Sedoni, D. Angeli, L. Sabattini, and V. Villani
Mechatronics, vol. 110, p. 103374, 2025.

[4] **A wearable stereo vision-based obstacle detection system for visually impaired individuals**
A. Ruo, D. Napolitano, and V. Villani
Fifth International Workshop on Agents and Robots for reliable Engineered Autonomy, 2025.

The second key component of social acceptance, particularly for mobile robots, addressed in this thesis is **social navigation**, i.e. the ability to move through dynamic, populated environments while respecting human comfort, personal space, and behavioral expectations. In such contexts, navigation is no longer a purely geometric task but becomes inherently social, requiring the robot to comply with both safety and social navigation constraints. Beyond safety, however, effective interaction in human-shared spaces also demands **intelligent social navigation**, where the robot is capable of interpreting contextual cues, reasoning about user intent, and dynamically adapting its behavior. This requires a synergy between low-level control methods, that enforce formal guarantees of safety and task satisfaction, and high-level reasoning mechanisms, that can translate

social and affective information into navigation parameters. As illustrated in Fig. 1.1, these considerations provide the foundation for the chapter on social navigation.

To address this challenge, this thesis proposes a control framework for **socially responsible robot navigation** (**SRN**) grounded on **Signal Temporal Logic** (**STL**) and **Control Barrier Functions** (**CBFs**). STL provides a formalism to express spatio-temporal specifications such as task timing, spatial constraints, and social rules, while CBFs ensure real-time adherence to these constraints by enforcing forward invariance of safe sets in the control law. We first introduced a method to encode STL constraints into smooth CBFs, enabling the robot to perform motion planning that guarantees the satisfaction of the specifications under dynamic and uncertain conditions. This formulation supports **nonlinear velocity constraints** and accounts for **both static and dynamic obstacles**, making it suitable for complex environments [10]. We further enhanced this approach with an **angular constraint**, ensuring that the user remains within the robot's field of view during companion-like tasks [11]. Recognizing that real-world sensors introduce noise and uncertainty, we extended the methodology by integrating a **robust CBF-based STL** formulation with an **Unscented Kalman Filter** (**UKF**). This allows the robot to estimate state accurately and maintain safety guarantees even in the presence of imperfect localization or perception [6]. Simulation results demonstrated that this robust version could successfully satisfy **temporal constraints** and **collision avoidance** requirements in scenarios with varying operational modes and obstacle densities. Moreover, a **practical real-world validation** of the CBF–based STL social navigation framework for omnidirectional mobile robots in ROS2 Humble has been proposed in [1].

To further enhance the robot's capabilities for social navigation, we also considered how information in human-robot interaction (HRI) is inherently fragmented, ranging from physiological signals and environmental conditions to the robot's internal state, user characteristics, and task constraints. While humans intuitively integrate these diverse cues to guide their behavior, robots require an explicit mechanism to interpret and act on such complexity [8]. **Large Language Models** (**LLMs**) provide a promising solution by transforming heterogeneous inputs into a unified representation of the social context. Leveraging this capability, we proposed a two-layer architecture for socially intelligent navigation. In this architecture, a LLM is used to interpret human intent, social context, and affective cues such as user category, emotional state, and confidence to infer navigation parameters like speed limits, social distance, and navigation mode (e.g., follow, accompany, reach). These parameters are then enforced at the control level using a CBF framework [5] based on our works [6, 10, 11]. The system allows the robot to dynamically adjust its motion strategy based on **contextual and environmental cues**, demonstrating good accuracy in interpreting human requests and generating appropriate behavior. In addition, we developed a lightweight architecture that integrates user identification and behavior adaptation for social guidance tasks. By combining facial recognition, gesture detection, and skeleton tracking, the robot can recognize a user and guide them while continuously monitoring their proximity. In particular, the navigation is suspended when the user is no longer within the expected distance range, ensuring reactive and transparent guidance behavior in public environments [12].

The study of this topic has produced the following publications:

[10] **CBF-based motion planning for socially responsible robot navigation guaranteeing STL specification**
A. Ruo, L. Sabattini, and V. Villani
European Control Conference (ECC), 2024.

[11] **CBF-based STL motion planning for social navigation in crowded environment**
A. Ruo, L. Sabattini, and V. Villani
European Robotic Forum, 2024.

[6] **Robust CBF-based STL motion planning for socially responsible robot navigation in the presence of measurement noise**
A. Ruo, L. Sabattini, and V. Villani
American Control Conference (ACC), 2025.

[1] **Deployment of a CBF-based STL social navigation framework on omnidirectional mobile robots**
A. Ruo, L. Sabattini, and V. Villani
Submitted to: Robotics and Automation Practice (RA-P), 2025.

[8] **A large language model-based motion planning for human-robot interaction: An experimental case study**
A. Coppari, S. Proia, A. Ruo, F. Favali, L. Sabattini, C. Secchi, V. Villani, M. Piazzola, and L. Capra
17th International Workshop on Human-Friendly Robotics, 2024.

[5] **An LLM-based architecture for socially intelligent robot navigation based on social cues**
A. Ruo, J. Cacace, M. Dalmau-Moreno, L. Sabattini, and V. Villani
34st IEEE international conference on robot and human interactive communication (RO-MAN). IEEE, 2025.

[12] **Follow me: an architecture for user identification and social navigation with a mobile robot**
A. Ruo, L. Sabattini, and V. Villani
European Robotic Forum, 2024.

In conclusion, this thesis demonstrates that social acceptance in robotics is a systemic property that emerges from the interplay of multiple factors, including user understanding and social navigation. By enabling robots to understand users through classification and biosignal analysis, and to navigate socially through CBF-based STL motion planning and LLM reasoning, we lay the foundations for robotic systems that are not only effective, but also trusted, accepted, and embraced by people in real-world contexts.

1.1 Reference case study

Among the different contexts in which socially acceptable robotic systems can operate, a recurring reference throughout this thesis is the **receptionist agent**. This case study serves as a representative and integrative scenario, as it involves several aspects addressed in the following chapters: from the system's **social acceptance** requirements to the **user understanding** and **social navigation** capabilities necessary to ensure intuitive and trustworthy interaction.

A receptionist agent can be a mobile **service robot** designed to operate within public environments such as banks, hotels, airports, or university buildings, where it interacts with multiple users. Its main functions include welcoming visitors, providing information, identifying and authenticating users, and escorting them to specific destinations. To perform these tasks effectively, the robot must integrate **perception**, **communication**, and **navigation** capabilities while satisfying social, functional, and ethical constraints. For instance, it should be able to detect the user's intention to interact, adapt its behavior to different categories of people (e.g., adults, children, elderly, or impaired users), comply with safety and privacy regulations, and move through dynamic environments populated by humans without causing discomfort or intrusion.

Due to these multidisciplinary requirements, the receptionist agent constitutes an ideal benchmark for studying social acceptance in robotics. It combines elements of **multimodal perception**, **adaptive contro**l, and **context awareness**, representing a realistic case in which social, cognitive, and navigational dimensions converge.

The design of such an agent thus provides a unifying framework linking the concepts discussed in this thesis:

- in **Chapter 2**, it is used to illustrate how definitions, requirements, and architectures jointly contribute to the development of socially acceptable systems;
- in **Chapter 3**, its interactive capabilities motivate the need for user understanding, enabling the robot to perceive cognitive and emotional states for adaptive interaction;
- in **Chapter 4**, its guidance behavior exemplifies social navigation, where motion planning must respect both safety and social constraints.

In our works presented in this thesis, the receptionist agent evolves from a conceptual model for requirements analysis to a modular implementation within the SERMAS architecture and, finally, to practical demonstrations of social navigation and human guidance. Therefore, it serves as a consistent **reference** throughout the thesis, illustrating how the proposed methodologies and architectures can be instantiated in real and socially meaningful scenarios.

1.2 Thesis organization

This thesis is organized into four main chapters, each addressing a specific aspect of social acceptance in robotics.

Chapter 2 – Social Acceptance. This chapter introduces the foundations of social acceptance in robotics, structured around three pillars: definitions, requirements, and architectures. It reviews functional and non-functional requirements that a socially acceptable system should satisfy, and discusses modular architectures that enable scalability, security, and adaptability across different domains, including service robots and XR systems.

Chapter 3 – User Understanding. This chapter focuses on the first key component of social acceptance: the ability of robots to perceive and interpret the cognitive and emotional states of their users. It presents methods based on multimodal sensing and biosignal analysis (EEG, ECG, GSR, facial expressions) for workload and stress estimation, introduces the SenseCobot multimodal dataset, and describes the development of adaptive interfaces and assistive technologies such as a low-cost electromagnetic gripper and a wearable obstacle detection system for visually impaired users.

Chapter 4 – Social Navigation. Among the various facets of social acceptance, this chapter focuses on social navigation, which becomes particularly relevant for mobile robots. It introduces a control framework for socially responsible robot navigation based on STL and CBFs, ensuring compliance with safety and spatio-temporal specifications. It further extends this approach to intelligent social navigation, where contextual and affective cues are integrated through LLMs to infer navigation parameters and enable adaptive, context-aware robot behavior in human-shared environments.

Chapter 5 – Main Findings and Future Directions. The final part of the thesis summarizes the main contributions and results, highlighting how user understanding and social navigation jointly contribute to social acceptance in robotics. It also outlines directions for future research,

For clarity, throughout the thesis the terms **service robots** and **robotic agents** are used interchangeably to indicate embodied **robotic systems** designed to interact with humans in social environments.

Chapter 2

Social Acceptance

Work automation is entering a new phase in which robotic systems are expected to support a wide range of tasks, extending beyond traditional industrial applications. Over time, robots have evolved from single-function machines into intelligent and versatile systems, profoundly influencing various types of occupations [16]. Current interest in deploying robots in service tasks that require more interaction with humans has directed the focus on a new generation of **service robots**. Examples of such applications include healthcare [17], where receptionist agents assist patients with booking appointments and guiding them to specific offices; museums [18], where they enhance the visitor experience by offering guided tours and interactive exhibit information; or airports [19], where they can speed up the check-in process, provide real-time flight updates, and assist passengers with directions and general inquiries, all of which play a crucial role in enhancing user experience and operational efficiency. For instance, Stricker et al. [20] introduced TOOMAS, an interactive reception robot designed for shopping environments. This robot is capable of autonomously approaching potential customers, navigating the marketplace, and guiding customers to their selected products, thereby offering a fully accompanied shopping experience. In [21], the authors, as part of the EU-funded MuMMER project, developed a social robot designed to interact naturally and flexibly with users in public spaces such as shopping malls. The EU-funded project SPENCER [19] developed a reception robot designed to assist, inform, and guide passengers in large and crowded airports. This robot integrates map representation, laser-based people and group tracking, activity and motion planning to efficiently manage passenger assistance. Moreover, in [12], we presented an architecture for user identification and social navigation using a mobile robot. This architecture leverages computer vision, machine learning, and artificial intelligence algorithms to identify and guide users within a social navigation context, thereby providing an intuitive and user-friendly experience with the robot.

To enhance the diffusion of such systems, it is fundamental to guarantee **safety** and **smooth interaction** with the user. As regard safety, the robot must not cause any harm to the user. This topic has been largely investigated in the literature. For example, in the case of collaborative robots, the safety standards and collaborative modes have been considered in [22]. In navigation tasks, collisions with the users must be avoided [10, 23]. However, to guarantee smooth interaction, treating humans simply as obstacles

may not be sufficient. In [24], the authors have considered issues such as human-oriented perception, natural human-robot interaction, readable social cues, and real-time performance, presenting a taxonomy of design methods and system components used to build socially interactive robots. In [25] the authors have provided an analysis of users intentions, robot behaviors, and their impact on the interaction, building upon the works in [26] and [27]. Additionally, it is important to guarantee the acceptance of robots. In this regard, it is important that the user feels comfortable interacting with the robots and perceives the system as understandable, pleasant, easy to use, and useful [28]. Wirtz et al. [29] have presented the Service Robot Acceptance Model (sRAM) to explain how and why consumers accept and use service robots. The authors have argued that consumer acceptance of service robots depends on how well these robots can meet user's needs and achieve role congruence. The sRAM identifies three categories of elements that influence robot acceptance: functional elements (such as perceived ease of use and usefulness), social-emotional elements (such as perceived humanness and social interactivity), and relational elements (such as trust and rapport). In [30, 31], studies have been conducted to evaluate the quality of service based on real data. Specifically, [30] has found that customers' highest priorities for robot service quality are assurance and reliability, while tangibility and empathy are not as important. Meanwhile, [31] has found that customers in hospitality and restaurant scenarios have exhibited more unethical behavior when excluded by human staff rather than robots under service exclusion, but more unethical behavior when served by robots compared to human staff in inclusive scenarios.

These challenges are not limited to service robots. Indeed, similar issues also emerge in the broader domain of XR systems, as highlighted in the SERMAS project [15]. In this context, virtual agents and immersive technologies must be designed to ensure trust, transparency, and socially acceptable. This underscores the need for architectures that can be generalized across both physical and virtual agents. In particular, XR is an umbrella term encompassing three primary constituents, Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), each extending the boundaries of human perception by offering a different interactive and immersive experience for human users. XR systems employ a spectrum of technologies like computer vision, motion tracking, and advanced displays that blend the physical (or real) world and the virtual world (or digital content) [32]. The scope of XR is extensive, driven by ongoing technological advancements (like improved hardware and software, wireless connectivity, Artificial Intelligence (AI) and machine learning integration) and cross-platform compatibility that continuously push its boundaries and possibilities. As XR continues to evolve, its implications are evident in diverse settings, like healthcare [33], Industry 4.0 [34], Internet of Things (IoT) [35], and industrial training [36].

To realize complex XR systems, a few architectures have been proposed in the literature, ranging from those addressing inter-device communication in industrial AR applications [37] to those using distributed data transmission models [38] and mobile device integration [39] in MR-based applications and services. However, these proposals are limited in addressing the challenges of expanding or upgrading the existing XR systems to accommodate more users, devices, or interaction modalities based on the systems' deployment contexts. Modular approaches, being adaptable, can be applied across various

domains to improve the systems' scalability and offer the flexibility of incorporating new hardware and software components. However, existing modular architectures like [40] do not provide an execution environment where the XR systems based on these architectures can support the integration of routines for diverse languages, human perception, and human-system interaction behaviors. Implementing such systems is even more technically challenging due to conflicts, inconsistencies, etc., across different deployment contexts and applications.

In order to increase the social acceptance in the design process of service robots and XR systems, in our work [9] we proposed a **modular architecture** that allows for the applicability of XR systems across applications. The architecture comprises hardware, software, and algorithmic modules derived from abstracting the **system requirements**. We implemented this architecture relying upon **cloud infrastructure resources**, supported with edge computing frameworks and shared communication protocols, to allow for **scalability** towards new application scenarios. XR systems built on this architecture can address the additional requirements introduced by these scenarios by instantiating the corresponding module. The architecture allows for fully distributed scenarios, irrespective of where the involved runtime is located. Another relevant trait of our architecture is that it facilitates building **socially acceptable XR and robotic systems** that can be used in diverse social contexts. Social acceptance influences the user experience of XR systems for their wide-scale adoption. Thus, we aim to ensure that the systems will satisfy the requirements and the needs of end users in compliance with the social context in which they are deployed, emphasizing features like **security** and **privacy**.

Parallelly, to satisfy the conditions for acceptance, we identify a set of requirements, which are systematically analyzed and presented in [7]. These requirements are expected to elicit **acceptability** (i.e., an a priori positive evaluation by users when confronted with the system), which in turn can lead to general **acceptance** (i.e., a positive, long-term retrospective evaluation) [41] of service robots.

In particular, we provide a **detailed discussion** of the requirements that guide the design of socially acceptable service robots. Leveraging these requirements in the **design process** allows human–robot interaction to: i) satisfy the goals and needs of the user; ii) comply with the social context in which the system is deployed; and iii) be transparent, safe, secure, explainable, and trusted by the user.

The remainder of this chapter is organized as follows:

- Section 2.1 introduces the concept of social acceptance in robotics, reviewing the different connotations found in the literature;
- Section 2.2 presents a systematic analysis of the functional and non-functional requirements that a service robot must satisfy to be considered socially acceptable, including a case study of a receptionist agent;
- Section 2.3 discusses existing and proposed system architectures, highlighting the role of modularity, scalability, and security in the design of socially acceptable service robots and XR systems;
- Section 2.4 provides concluding remarks and an outlook, linking the discussion on social acceptance to the subsequent chapter on user understanding.

2.1 Definitions

In the literature, the concept of **acceptance** has been widely discussed in relation to technical systems in general, taking on various connotations [42, 43]. It is important to underline that in this context, acceptance is defined at an individual level, i.e., regarding the single user, and not at a societal level. A systematic review of the literature is presented in [16], discussing how **social acceptance** of robots has been studied in different occupational settings and what kinds of attitudes the studies have uncovered about robots as workers. In [41], the authors define acceptance as the positive evaluation that results in the motivation and eventual act of using the technology for the task it is designed to support. In particular, consider the case of a receptionist agent whose task is to provide information to users and guide them within a social environment. In this scenario, the agent should exhibit specific behaviors that are user-friendly and socially acceptable, to ensure an optimal user experience. In the case of service robots, social acceptance may also refer to the capability of the technology to be used in different social contexts in such a way that it does not make users feel uncomfortable or out of place. Indeed, social acceptance may be defined as a user feeling comfortable with an artificial agent as a conversational partner, finding its social skills credible, and accepting social interaction as a way to communicate with it [43]. In any use case, there are different aspects that must be considered to ensure the acceptance of a service robot.

The social acceptance of robotic agents in social spaces, following the connotations defined in [42] and summarized in Table 2.1, can be divided into different aspects. Firstly, **physical acceptance** refers to how users perceive the agent based on its appearance and physical aspect. For instance, a robot designed with a friendly appearance is more likely to be accepted by users. Secondly, **behavioral acceptance** involves user's perceptions of the agent's use of the space and its non-verbal communication. Effective non-verbal cues, such as gestures and facial expressions, can make the interaction appear more believable, fluent, and natural, thus enhancing user experience. **Functional acceptance** relates to the agent's practicality, innovativeness, and ease of use. Users are more likely to accept a robot that is perceived as useful, accurate, and innovative in its functionality. This includes the robot's ability to perform tasks effectively and efficiently. In terms of **social acceptance**, users must perceive the robot as a social entity capable of engaging in social behaviors. The satisfaction of users is also influenced by the opinions of others regarding human-machine interactions. The union between functional and social acceptance gives rise to trust, that is, the belief that the agent acts with integrity and reliability. **Cultural acceptance** addresses how well the agent aligns with the user's cultural norms and values. This includes factors like educational values and the general tech-savviness of the target audience. An agent that respects and reflects cultural aspects is more likely to be accepted by users. Lastly, **representational acceptance** is about users viewing the agent positively overall. This involves a general perception of the robot's role and presence in their environment.

Table 2.1: Acceptance connotations [42]

Requirement	Description/Rationale
Physical acceptance	Users perceive the agent as likable and credible based on its physical aspect
Behavioral acceptance	Users perceive the agent's non-verbal communication as believable and the interaction as fluent, natural, and pleasant
Functional acceptance	Users perceive the agent as easy-to-use, useful, accurate, and innovative
Social acceptance	Users perceive a social entity in the agent, consider it capable of performing social behavior, and are satisfied with what other people think of human-machine interaction
Cultural acceptance	Users accept the agent because it complies with their culture in general (e.g.: its educational values and tech-savviness)
Representational acceptance	Users consider the agent in a positive way

2.2 Requirements

Building upon these definitions, in [7] we have analyzed the requirements that may guide the design of socially acceptable robotic systems. Specifically, we have divided the requirements into functional and non-functional:

- **Functional requirements** (**FRs**) capture the intended behavior of the system. This behavior may be expressed as a service or function which the system is required to perform;
- **Non-Functional requirements** (**NFRs**) specify the criteria used to judge the operation of a system, rather than its specific behavior.

The requirements presented in this section cover all types of acceptance, as schematized in Table 2.1. In addition, a service robot must also be designed to meet the additional requirements schematized in Table 2.2. Alongside these requirements, robotic agents may need additional technical NFRs, which depend on the specific use case.

The requirement analysis proposed in [7] follows an approach guided by real-world use cases, bottom-up principles, and user-centered design derived from the SERMAS [15]. From the use cases of SERMAS, user's needs have been identified, leading to the derivation of requirements. To clarify the analysis, we can group all requirements into one or more of the following categories: *Interaction*, *Operability*, *Perception*, *Environment*, *Privacy*, *Regulations* and *Safety*.

Table 2.2: Additional Requirements

Requirement	Description/Rationale
Compliance	The agent correspond with the rules and the regulation imposed by the use cases
Ethics	The agent is civil, respectful, privacy-friendly, transparent, trustworthy
Performance	The agent is fast, safe, reliable, robust, scalable
Economics	The agent is innovative, realizable on designated budgets, and potentially marketable
Sustainability	The agent is sufficiently documented, generally compatible and modular
Eco-Friendliness	The agent is non-toxic, as energy-efficient and resource-saving as possible, durable, recyclable

2.2.1 Interaction

This category includes requirements related to how users interact with the service robot, encompassing input methods like text input, touchscreen interaction, voice commands, and gestures. Effective interaction mechanisms are crucial for ensuring user-friendly experiences, enabling users to communicate their needs and commands efficiently. The system should support multiple interaction modes to accommodate diverse user preferences and ensure accessibility for individuals with varying abilities.

2.2.2 Operability

Operational requirements in this category pertain to the functionality and stability of hardware and software components, ensuring smooth operation and reliability of the service robot. This encompasses the system's ability to perform tasks consistently over time without failures, as well as its capacity for regular maintenance and updates. High operability is essential for maintaining user trust and ensuring continuous, uninterrupted service.

2.2.3 Perception

This category includes requirements concerning the sensory inputs used by the service robot, such as computer vision, sensors, and other data sources for interpreting the environment. Accurate perception is vital for the system to understand and appropriately respond to its surroundings. This involves recognizing objects, interpreting human actions and emotions, and making informed decisions based on real-time data from various sources.

2.2.4 Environment

This category encompasses requirements addressing the environment in which the service robot operates, including both physical and digital environments. The agent must demonstrate adaptability to different environmental conditions and seamless integration with existing infrastructures. These environmental conditions may include factors such as lighting, noise, object arrangement, and so on.

2.2.5 Privacy

Privacy requirements involve measures to protect sensitive data, including personal and confidential information, ensuring proper handling and compliance with privacy regulations. The agent must implement robust data security protocols to prevent unauthorized access and misuse of information. Ensuring privacy protection is fundamental to building user trust and complying with legal standards.

2.2.6 Regulations

Regulatory requirements encompass adherence to company policies, industry standards, and legal regulations governing the use and deployment of service robots. Compliance with these regulations is crucial to avoid legal repercussions and ensure ethical use of technology.

2.2.7 Safety

Safety requirements ensure that the agent does not pose risks or hazards to users, minimizing the potential for injury or trauma during operation. The system should be designed with safety features to prevent accidents and respond effectively to emergencies.

2.2.8 Receptionist robot: a case study

The proposed requirement analysis allows the development of service robots that are socially acceptable, through the selection of specific FRs or NFRs, as explained in Sec. 2.2. Particularly, the system should meet FRs and NFRs, as shown in Fig. 2.1, with the NFRs depending on the particular use case in which the system is deployed.

Within a social environment, a receptionist agent should be able to receive and interact with users based on conversational activities, provide them with information, and guide them to certain destinations safely and acceptably in all its connotations. Furthermore, the agent should include an identification and authorization process to admit users to the building.

To achieve this, we can define the following FRs and NFRs:

- **Interaction FRs:**

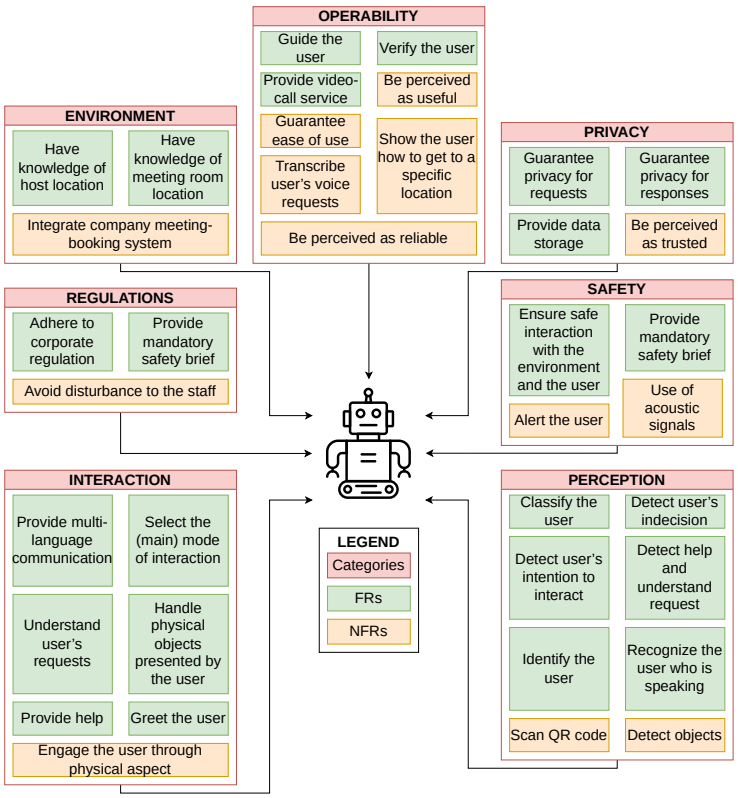


Figure 2.1: The requirements necessary for a socially acceptable receptionist agent are categorized into FRs, depicted in green, and NFRs, depicted in orange. They can be grouped into various categories, shown in red.

- **Provide multi-language communication**: users will have the possibility to choose their preferred language for communications, enabling interaction adaptation based on the user's language preference.
- **Greet the user**: the service robot should greet the user using the most appropriate interaction style for the identified user class.
- **Understand user's requests**: the agent should be able to understand various requests, such as accessing the building, locating specific areas within the building, accessing security information, or receiving directions to particular locations within the building, all aimed at meeting user needs.
- **Provide help**: the agent should be designed to offer useful suggestions and guidance for resolving common issues or assisting users when they encounter difficulties. Additionally, it should present a list of services it can provide assistance with, such as accessing a building for a meeting, offering directions to a designated meeting room or office, and providing safety information.
- **Select the (main) mode of interaction**: the system should offer users a choice of text/touch, voice, and/or gesture interaction methods to tailor interaction to individual preferences.
- **Handle physical objects presented by the user**: the agent should have the ability to interact with objects presented by the user to enhance user interaction and engagement.

- **Interaction NFRs:**
  - **Engage the user through physical aspect**: the service robot should incorporate its own on-screen representation to enhance user engagement. The agent representation will be designed to create a pleasant experience, communicating through voice messages synchronized with facial expressions and body movements. The objective is to convey attentiveness and helpfulness in providing service, all aimed at facilitating effective communication and user engagement.

The receptionist agent must be able to guide the visitor within the social environment until they reach the destination. During this task, the service robot should be able to orient itself, follow the best route to the goal pose, and alert the visitor if he or she has taken the wrong path.

To achieve this, we can define the following FRs and NFRs:

- **Operability FRs:**
  - **Provide video-call service**: if required by the application context, the service robot should have the capability to notify the host(s) of a visitor's arrival and initiate a video call (preferably utilizing a tool tailored to the company's requirements) to get approval for access.
  - **Verify the user**: the agent should integrate a user verification procedure (e.g., email link, SMS, etc.).
  - **Guide the user**: the service robot should be able to accompany visitors within the environment, retrieve the target destination, orient itself, select the optimal route, and alert visitors if they deviate from the route, prioritizing safety by avoiding obstacles.

- **Operability NFRs:**
  - **Guarantee ease of use**: the user should perceive the system easy to use and comprehend, aiming to enhance user satisfaction and minimize errors.
  - **Be perceived as useful**: the user must perceive the agent as useful, to increase the likelihood of its use.
  - **Be perceived as reliable**: the user should have the perception of reliability of the service robot, to increase the probability of its use.
  - **Show the user how to get to a specific location**: the agent should be able to display a map and provide guidance to the user on navigating from point A to point B to assist users in navigating the social environment.
  - **Transcribe user's voice requests**: the service robot should possess the capability to convert user's spoken commands into written text, utilizing functionalities such as speech-to-text (STT), to facilitate user interaction and data processing.

Furthermore, the receptionist agent should be able to autonomously detect a user approaching with the intention to interact with the system. Additionally, the service robot should be able to classify the user based on significant features extracted from the detected approaching visitor. This is useful for the adaptation of social interaction.

To achieve this, we can define the following FRs and NFRs:

- **Perception FRs:**
  - **Detect user's intention to interact**: the agent should autonomously detect approaching visitors, defined as individuals who move toward and stop in front of it, excluding passing visitors, to initiate interaction and ensure secure building access.
  - **Classify the user**: the service robot should be able to classify a visitor based on key characteristics extracted from the detected approaching visitor (e.g., age), facilitating the adaptation of social interaction to the specific user.
  - **Detect user's indecision**: the agent should be able to perceive the user's indecision during use and should possess a mechanism to suggest an assistance function to the user so as to help him or her.
  - **Detect help and understand request**: the service robot should recognize when a visitor explicitly asks for assistance in understanding his or her request, whether it is expressed in language, or through gestures, to improve user understanding.
  - **Identify the user**: the system should be capable of recognizing the presence of a user to initiate interaction, following the procedure outlined in corporate regulations.
  - **Recognize the user who is speaking**: the service robot will have the capability to identify the user who is speaking, particularly in scenarios where multiple users are present, to initiate interaction with the appropriate user.

- **Perception NFRs:**
  - **Scan QR code**: if required by the application context, the system should be capable of scanning the QR code provided by the user and verifying the contained information.
  - **Detect objects**: the service agent should identify whether the user is holding an object (such as a parcel), and the interaction is initiated only at the user's request, to offer support when requested.

In addition, during the tasks performed by the service robot within the case study, it is necessary for the receptionist agent to be able to adapt to different environmental conditions with existing infrastructures and be familiar with the surrounding environment.

For this reason, we can define the following FRs and NFRs:

- **Environment FRs:**
  - **Have knowledge of host location**: the service robot should have a mapping of the building employees and their location.
  - **Have knowledge of meeting room location**: if required by the application context, the agent should have a mapping of the meeting rooms and their location.

- **Environment NFRs:**
  - **Integrate company meeting-booking system**: if required by the appli-

cation context, the system should be capable of verifying meeting approval.

Potential weaknesses in the design or implementation of the service robot could be exploited by malicious actors or regular human users during human-agent interactions, leading to security breaches or agent compromise. Additionally, users may raise objections during the case study execution phase, fearing issues related to the insecure management of collected data.

For this reason, we can define the following FRs and NFRs:

- **Privacy FRs:**
  - **Guarantee privacy for requests**: the agent should ensure privacy for requests, aligning with both legal mandates and company policies.
  - **Guarantee privacy for responses**: the service robot should ensure privacy for responses, aligning with both legal mandates and company policies.
  - **Provide data storage**: data used for body tracking, facial landmarks, or any other information classified as "personal data" must be anonymized and not retained, in line with legal and corporate regulations.

- **Privacy NFRs:**
  - **Be perceived as trusted**: the user must be assured that interactions with the agent are safe with regard to the processing of personal data or, at the very least, the perceived risk associated with inappropriate data handling must be minimal in order to increase the likelihood of its use.

In addition, the service robot must adhere to corporate security and safety regulations to meet legal and company requirements.

To achieve this, we can define the following FRs and NFRs:

- **Regulations FRs:**
  - **Adhere to corporate regulation**: the service robot must adhere to corporate security and safety regulations to meet legal and company requirements.
  - **Provide mandatory safety brief**: the agent should provide mandatory information and/or company regulation on safety that are necessary for access to the building, before enabling the visitor to access.

- **Regulations NFRs:**
  - **Avoid disturbance to the staff**: the system must not disrupt or intrude upon the tasks performed by reception employees, in accordance with company requirements.

Lastly, the service robot, during the execution of tasks within the case study, must ensure safety.

To achieve this, we can define the following FRs and NFRs:

- **Safety FRs:**
  - **Ensure safe interaction with the environment and the user**: the service robot must be designed to interact safely with both its surroundings and the

people it serves, minimizing the risk of accidents or injuries.
  - **Provide mandatory safety brief**: the agent should provide visitors with essential information on security and/or business regulations required for access to the building before granting them entry, ensuring compliance with organizational guidelines.

- **Safety NFRs:**
  - **Alert the user**: the system should have the capability to alert human operators to abnormal environmental parameters, aiming to enhance environmental quality.
  - **Use of acoustic signals**: the service robot should be equipped with an audible signal or similar measure that activates during movement to signal its presence to people.

2.3 Architectures

Architectures make it possible to link abstract requirements such as safety, transparency, privacy, and trust to the concrete capabilities of a system, including perception, interaction, planning, and control. In this section, we present the architecture proposed in our work [9], which is designed to support social acceptance regardless of the final application domain or the specific system employed. Particularly, Table 2.3 provides a comparative summary of recent architectures that have been proposed for different types of systems.

Table 2.3: Comparative summary of related works and our proposed architecture.

Ref. (Year)	System type	Concern of interest	Applicability	Modularity	Security
[44] (2014)	MR	Mobile device support	Mobile applications	Not supported	No
[37] (2018)	AR	Device communication	Industry 4.0 shipyard	Not supported	No
[38] (2018)	MR	Sensor data transmission	Real-time collaborative systems	Not supported	No
[39] (2019)	MR	Device integration to IoT	IoT services	Not supported	Not detailed
[40] (2020)	XR	Human-system interaction	Industrial manufacturing	Supported	No
[45] (2021)	AR	Application deployment	Web browsers	Not supported	Yes
Our work	XR	Human-system interaction	Application-agnostic	Supported	Yes

In [37], Fernandez et al. proposed an architecture to streamline real-time device communication in industrial AR applications. The architecture relies on fog computing to reduce latency and offload cloud systems, as well as cloudlets to address fog gateways' limited computing power for computationally intensive services. For MR applications, some specific architecture variations exist for supporting mobile MR devices [44], device integration to IoT services [39], and distributed sensor data transmission based on client-server and peer-to-peer models [38].

Serras et al. [40] proposed an Interactive XR (IXR) architecture for enhancing non-savvy operators' capabilities in the Industry 4.0 paradigm. IXR references the Spoken Dialogue System's architecture to implement different AR mechanisms by integrating modules for voice-enabled human-system interaction. Despite IXR [40] focusing on the usability goals, it is abstract, lacks articulation of modules' technicalities, and is limited to verbal modes of interaction.

From the application deployment viewpoint, Pereira et al. [45] presented an AR-based architecture, ARENA, for building and hosting multi-user interactive XR applications on WebXR-capable browsers based on the user's location. This is achieved by capturing system resources' states in scenarios to reduce latency and for runtime management.

XR and robotic systems often involve intricate hardware and software components and need to scale to meet application-specific requirements. For social acceptance of these systems, it is required that they adapt to how non-specialized users perceive and engage with them in a social context. An architecture is thus called for that, at the same time, (i) can provide a structured framework to capture hardware and software components for engineering such complex systems that are applicable in different contexts, (ii) supports modularity to allow scaling up and extensions of the systems, (iii) enables social acceptance of the systems. Unfortunately, none of the existing architectures provides all these features in unison. More specifically, existing architectures do not provide a holistic system addressing human-system interaction, and they are limited in combining cloud and edge computing to address functional and non-functional requirements like security and privacy, given the social context of the system, regardless of the application domain.

To realize a system that combines hardware, software, and algorithmic modules to provide an experience that enables socially acceptable interactions with non-specialized users and adapts to the deployment context, we need to adopt a modular architecture that supports different interaction modalities (e.g., verbal and non-verbal, like user's facial expressions) across applications, ensures general applicability by being application-agnostic (i.e., not applicable only to a specific kind of setting like IoT), and is open to the integration of external inputs. In Sec. 2.3.1, we discuss our proposal for such an architecture. The designed modules incorporate requirements and end users' needs, specifically to support a large set of languages and enable new interaction forms to avoid bias whilst ensuring accessibility and security, which, in the present context, we consider representative features of the system's social acceptance.

2.3.1 Proposed architecture

The central concept of our system architecture proposed in [9] is **modularity**. The system is composed of separate but highly interconnected **modules**, where each module performs specialized tasks and can interact with other modules through two well-known architectural styles: **Representational State Transfer Application Programming Interface** (**RESTful API**) [46] and **publish/subscribe** (**PUB/SUB**) **events** [47]. The collective set of modules facilitates proper system functioning.

The overall architecture, shown in Fig. 2.2, enables fully distributed scenarios, regardless of the location of the involved runtime. System runtimes can leverage remote resources transparently to enable complex service compositions and a mesh of different runtimes, facilitating multi-user and multi-domain interaction models. This can significantly enhance social acceptance of XR and robotic systems by providing versatile and inclusive experiences that cater to diverse user needs and preferences in various contexts.

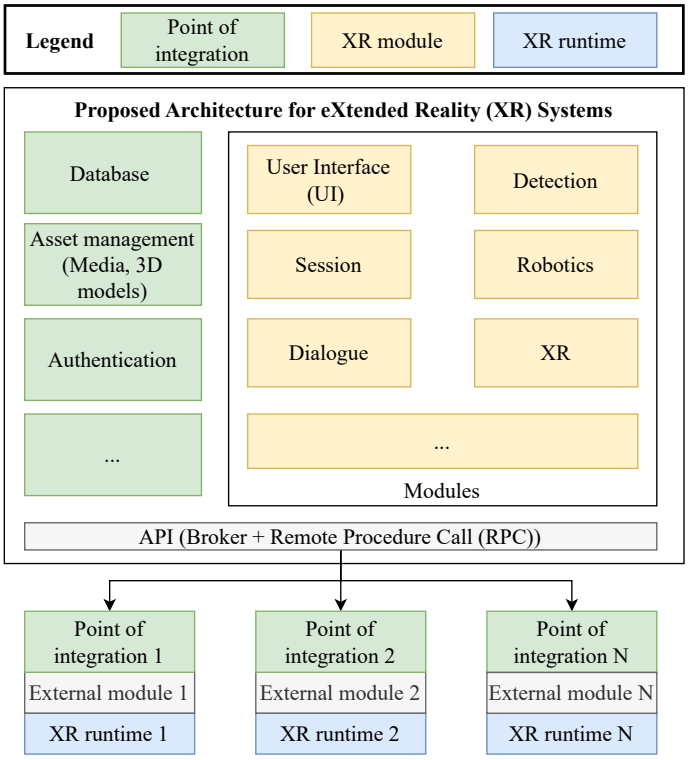


Figure 2.2: Proposed architecture for socially acceptable XR and robotic systems.

2.3.2 Organization into modules

Modules are integrated within the platform via integration points, which provide connections across various interfaces. The modules are defined to distribute responsibilities for implementing the required functionalities and simplifying the implementation of new requirements. To achieve this, we used a requirement-driven approach to identify the necessary modules for the system, starting from the requirements (e.g., functional, non-functional, security, privacy) and, where possible, consolidating and abstracting them to identify the corresponding and required functionality. Consequently, a module represents a piece of software and/or hardware of the system that implements a set of requirements or a single requirement in some cases.

We considered this approach for the following advantages. Firstly, it enables a clear identification of the responsibilities of each module, reducing the risk of leaving some requirements unaddressed. Secondly, it allows scalability towards new application scenarios and use cases. The system can address any additional requirements introduced by a new use case by implementing a corresponding module that does not conflict or overlap with existing ones. This can be facilitated by an architecture that supports straightforward integration of new modules and implements a shared communication protocol among the modules. Hence, adding, changing, or deleting a (third-party) component without affecting the rest of the system requires less effort, unlike with a non-modular approach.

Specifically, each module can be defined according to the following specifications: i) *functionalities*; ii) *hardware specifications*; iii) *software specifications*; iv) *input/output*

from/to other modules; v) *addressed requirements*. The list of modules required to make the system socially acceptable will be discussed in Section 2.3.3.

Communication model

To facilitate communication among modules, a unified communication model at the module or application level is proposed, as illustrated in Fig. 2.3. This model encompasses the following exchange patterns:

- **Request/Response Model over HTTP and JSON:** this communication model provides request/response patterns, wherein a client "calls" a server by sending an HTTP request, typically in JSON format, and the server responds with an HTTP response, also often in JSON format. A straightforward use case is retrieving information by initiating an instantaneous synchronous task (e.g., turning on a light or opening a gate).
- **Event-driven Messaging Model via MQTT:** this communication approach offers a PUB/SUB model and is suitable for message-passing events and near real-time notifications from the system. The process is generally asynchronous and is best suited for use cases where there is no specific timing between data exchanges or where the amount of data sent does not require explicit feedback from the receiver(s) (e.g., when a motion sensor detects an activity, it sends messages to the subscribed devices, allowing lights to activate and cameras to record real-time events).

Having a dual mode of communication is advantageous in terms of flexibility, scalability, and integration capabilities. For instance, by incorporating PUB/SUB capabilities, the system can better handle large volumes of events or messages, which can be challenging to achieve with only RESTful APIs. Also, PUB/SUB systems excel at real-time communication. When combined with RESTful APIs, these systems can address both real-time and request-response communication. Further, each module shares Swagger [48] for RESTful API and PUB/SUB events. Swagger is a set of open-source tools and specifications that provide a standardized way to describe, document, and consume RESTful APIs and PUB/SUB events. It does not specify the method for exchanging information with the API module but helps define and document the API structure.

Interoperability among modules

Interoperability among modules and applications is achieved by using two open standards, viz., OpenAPI [49] and AsyncAPI [50], which provides a formalized way to define data models (including message payloads) and endpoint interfaces. OpenAPI offers tools and standardized definitions for APIs based on the request-response communication model. AsyncAPI is a recent initiative supported by the Linux Foundation, which aims to provide means to describe the PUB/SUB communication model. From the specifications of the considered use case or application scenario, other tools can create wrappers with client or server code for use during development, formal validation of sent and received

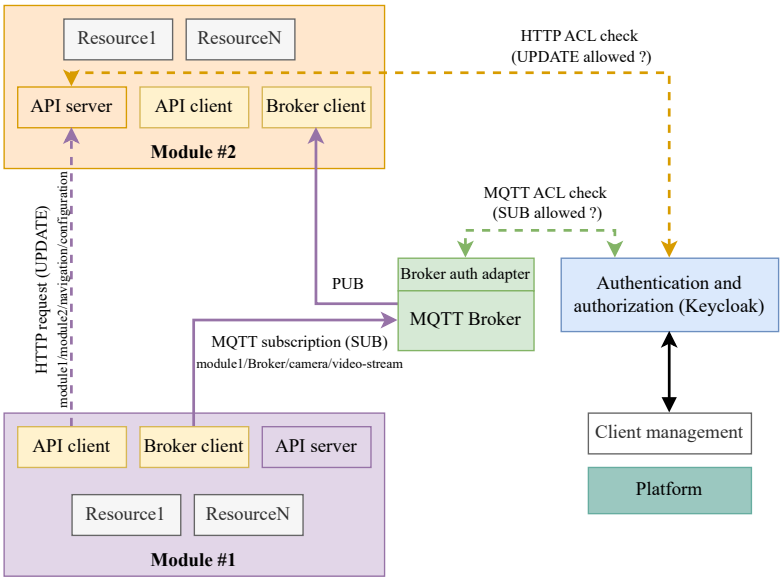


Figure 2.3: Model for unified communication among modules.

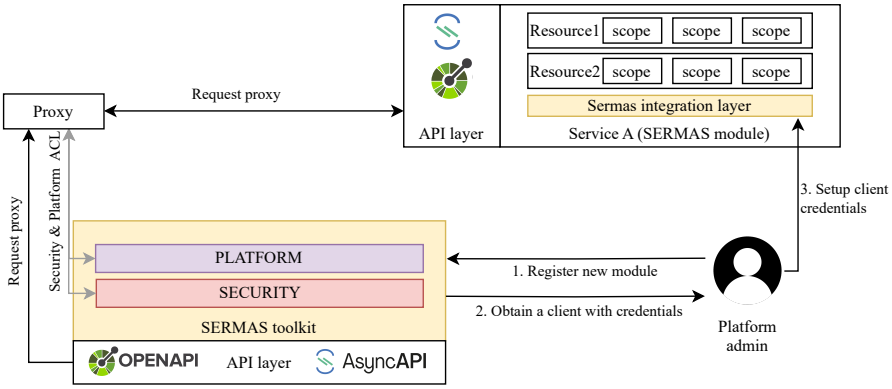


Figure 2.4: Modular approach of the toolkit to add new modules.

messages, generation of user interfaces used as documentation and, in some cases, interaction with server-side endpoints. For modules running on the same system platform (e.g., a game engine), communication among modules is managed within the platform itself, without the need for wrapping.

Security

We also incorporate solutions to address security, as it is one of the representative features of social acceptance in the present context. Establishing security requires treating the system as a socio-technical system, in which technology and human users interact by exchanging messages and data, and in which both technology and humans contribute to the security or can cause vulnerabilities. In this case, we used a message-signing approach based on the JSON Web Token (JWT) [51] to ensure the integrity of messages exchanged between modules. The signed JWT tokens allow to check whether the information contained within the token has been tampered with during communication. Likewise, inspired by Zero Trust Architecture (ZTA) [52], a centralized component in charge of authentication and authorization can identify and decide which user can retrieve information or perform action on protected resources. This will help in mitigating

concerns related to the safeguarding of user information, unauthorized use of interaction data, and manipulation of system experiences, which can contribute to social acceptance by ensuring the integrity of the technology.

2.3.3 Description of the modules

To implement the requirements discussed in the previous section, we identified a set of functional modules that together realize the proposed architecture for socially acceptable systems. Each module is designed to address specific socio-technical requirements and can be independently developed and deployed.

Below, we describe the main modules and, for each of them, provide examples of how they can be implemented.

- **Detection module:** this module supports the perception of the world through different sensors, prioritizing on-edge processing to limit the transmission and potential diffusion of user data. As hardware specification, it requires a camera and a microphone. It carries out tasks such as user recognition (the user is standing in front of the camera), determining user intention (the user's intention to interact with the system), user characterization in terms of mood (sentiment analysis), user age estimations, and user sentiment extraction from speech.
  We applied different technologies to achieve these functionalities. For example, for user detection and characterization, we used the Faceapi Javascript [53] library. This library allows the detection and recognition of user faces for web applications by extracting emotions from face analysis and age estimations. It is implemented on top of Tensorflow.js, a Google AI library for Javascript language.
  Specifically, we used three pre-trained AI models to extract the information about the user:
  - *Tiny face detector* based on TinyYolov2, to detect a face in front of the camera;
  - *Face expression net*, to detect user expression (i.e., happy, fearful, sad, disgust, angry, surprise, and neutral);
  - *Age gender net*, to estimate the age of the user in front of the camera.

  Furthermore, this module can perform object detection through the camera. Regarding the sentiment extraction from the user speech, we used the SpeechBrain [54] toolkit. It is an open-source, all-in-one speech toolkit based on PyTorch [55]. It can perform different tasks, such as speech recognition, speaker recognition and diarization, speech enhancement and separation, multi-microphone signal processing, end-to-end spoken language understanding, and language modeling.
- **Robotics module:** this module enables seamless interaction between the physical and XR-based virtual worlds. It is an interface between the architecture and components based on ROS2 [56], an open-source robotic operating system. It allows communication with several robot applications, such as robotic arms.
  Having a module that is a gateway towards ROS2 applications opens the XR architecture to have extensive interoperability with a range of robotic systems, as ROS2 is widely used in robotics. This component enables communications within the distributed system and allows direct exchange of events from robots to the

other modules. The type of robot depends on the specific application scenario. For instance, if there is a need for the robot to move within an indoor environment, a mobile base would be required. Conversely, if there is a need to grasp or hand over objects, a robotic arm is needed.

- **Dialogue module:** this module enables natural language interaction with the XR or robotic systems and performs speech-related tasks. It can get user speech from a microphone to perform tasks, such as converting speech to text, sending text to a LLM, and getting the response.
  Currently, two different services, viz., whisper [57] and speech-to-text by Google [58] are integrated for converting speech to text and can be used alternatively. The module gets the text from the user's speech and injects it into an LLM (LLM prompting). So far, the LLMs integrated inside the architecture are chatGPT [59] and google-flan-t5 [60]. The LLM is configured to get the responses in chunks to speed up its real-time response.
- **User Interface (UI) module:** this module offers points of interaction to enable integrated multi-modal UI features. It allows manipulating the system's graphical interface, such as creating buttons, forms, and cards. Depending on the application contention, it is required to choose the way to represent the UI and the type of screen to use.
- **Session module:** this module provides functionality for handling different types of sessions inside the system. One session type keeps track of the agent's status. An agent is a high-level functionality which uses one or more modules to perform specific tasks. Through this session, it is possible to recognize system errors for debugging. Another session type is a user session, which can track the user's interaction with the system from start to end. For example, when the user approaches the camera, the session starts to couple user interaction with the system until the interaction ends. By using sessions, the system can manage multiple interactions or agents simultaneously.
- **XR module:** this module handles XR environment settings. Currently, it is possible to add, delete, or update markers. A marker is an object, picture, or QR code to track static or dynamic objects in the user environment. Based on the marker type, the user's position in the environment can be recognized—for instance.
  Moreover, this module can provide information about the occlusion of a 3D object. For example, given the user's position and a 3D asset model, the module indicates whether a physical obstacle blocks the 3D object.

By leveraging OpenAPI and AsyncAPI specifications, tools can i) generate code to ease development and integration, ii) facilitate formal validation and generation of user interfaces, and iii) ease documentation. By using these standards, the level of abstraction is high, making the modules implementation-agnostic in terms of programming languages. Each module has its codebase and can be developed, deployed, and maintained independently. In such a case, if a module fails, it may not affect the entire application.

2.4 Outlook

The analysis conducted in this chapter has highlighted that social acceptance in XR and robotic systems cannot be reduced to a single dimension. Instead, it emerges from the interplay of requirements, both functional and non-functional, ensuring safety, transparency, privacy, and sustainability, and from architectural choices, such as modularity, scalability, and secure communication, which guarantee adaptability across heterogeneous application scenarios. The two perspectives are complementary: while requirements define what a socially acceptable system should achieve, modular architectures provide the means to achieve it in practice.

Nevertheless, meeting requirements and adopting a modular architecture alone is not sufficient. Social acceptance ultimately depends on the interaction between the robot and its users, who may differ in age, cultural background, experience with technology, or emotional state. This observation calls for systems that are capable of recognizing, classifying, and understanding users, so that interaction strategies and control parameters can be adapted accordingly.

This motivates the transition to the next chapter, which focuses on user understanding. By leveraging biosignals and behavioral data, it becomes possible to detect mental workload, stress, and individual preferences. These capabilities provide robots with the awareness necessary to tailor their behavior, thus bridging the gap between abstract requirements for social acceptance and their concrete realization in human-centered interaction.

Chapter 3

User Understanding

One of the key pillars of social acceptance in robotics is the capability of a system to understand its users. While the previous chapter has highlighted how definitions, requirements, and architectures contribute to building socially acceptable systems, these factors alone may not be sufficient to guarantee a user-friendly interaction. Social acceptance is not only a matter of functional compliance or technical reliability; it is also strongly determined by how effectively the system perceives and adapts to the individual human it interacts with. This capability is generally referred to as **user understanding**, encompassing the detection, modeling, and prediction of users' cognitive states, emotional conditions, intentions, and specific physical or social needs.

Indeed, robots capable of adapting their behavior to a user's workload, stress level, or physical limitations are more likely to be perceived as transparent, supportive, and trustworthy. Conversely, systems that ignore the user's state risk appearing rigid, unsafe, or socially inadequate. In this sense, user understanding acts as a bridge between abstract design principles of social acceptance, such as usability, perceived usefulness, and cultural alignment, and their concrete realization in real-world applications.

High cognitive workload can reduce performance, increase error rates, and generate frustration, while sustained stress can negatively affect trust and willingness to collaborate with robots. On the other hand, adaptive systems that can sense and respond to the user's **psychophysiological condition** foster smoother collaboration, reduce discomfort, and improve overall acceptance [61]. Therefore, designing systems capable of estimating cognitive and emotional states in real time is essential for safe, efficient, and user-centered interaction.

Among the different approaches to user understanding, **biosignals** [62] offer objective and reliable indicators of the user's psychophysiological condition. Measurements such as EEG, ECG, GSR, and facial expressions provide valuable information on workload, stress, and emotional stimulation. In our preliminary work [14], EEG signals were first investigated as a means to classify workload during human–robot collaboration. The results demonstrate that specific frequency bands, as well as derived index ratios, correlate strongly with cognitive load during decision-making and attention-intensive activities. These findings have established EEG-based metrics as valid indicators for online work-

load estimation, providing a foundation for adaptive control strategies in collaborative settings.

Despite growing interest in biosignal-based user modeling, one of the main limitations in the field is the lack of publicly available, high-quality datasets collected in realistic HRI scenarios. Without such resources, it is difficult for the community to compare algorithms, validate models, and develop generalizable solutions. To fill this gap, we have developed **SenseCobot** [13], a multimodal dataset specifically designed for collaborative robotics contexts. The dataset includes synchronized EEG, ECG, GSR, facial expressions, and task performance from 21 users engaged in progressive collaborative robot (cobot) programming tasks, together with subjective workload evaluations obtained via NASA Task Load Index (NASA-TLX). Building upon this dataset [13], we extended our research with a **multimodal stress evaluation** [3]. Experimental results showed that stress-related characteristics extracted from these modalities, for example, heart rate variability parameters, GSR recovery time, or EEG Alpha asymmetry, correlate robustly with subjective stress reports, confirming the importance of multimodal monitoring for user modeling.

User understanding goes beyond psychophysiological monitoring: it also involves adapting **physical interfaces** and **assistive functions** to individual needs. For instance, in collaborative robotics, we developed a low-cost 3D-printed electromagnetic gripper in [2], that combines affordability and functionality, enabling secure grasping of objects of varying shapes and weights in shared workspaces. By lowering technological and economic barriers, such devices make cobots more accessible and adaptable to user-specific requirements. Similarly, in assistive robotics, we designed a wearable stereo vision system for visually impaired users in [4], enabling real-time obstacle detection and verbal guidance. This project illustrates how tailoring systems perception and interaction strategies to the specific needs of vulnerable users can improve **autonomy**, **safety**, and **inclusivity**, which are fundamental aspects of social acceptance.

Taken together, these contributions demonstrate a **multidimensional approach** to user understanding, ranging from biosignal-based estimation of cognitive and emotional states, to the creation of multimodal datasets, to the design of adaptive physical interfaces and assistive devices. By integrating these perspectives, robots can be equipped with the capability to classify users and dynamically adapt their behavior.

The remainder of this chapter is organized as follows:

- Section 3.1 presents studies on biosignal-based estimation of mental workload and stress, focusing on EEG analysis and multimodal classification;
- Section 3.2 describes adaptive and assistive technologies, including the electromagnetic gripper for collaborative tasks and the wearable stereo vision system for visually impaired individuals;
- Section 3.3 summarizes the insights gained and discusses how user understanding serves as a prerequisite for socially acceptable robot navigation, which will be the focus of the next chapter.

3.1 Cognitive and emotional state estimation via biosignals

A core element of user understanding is the estimation of cognitive and emotional states through **biosignals**, which provide objective and continuous indicators of the user's psychophysiological condition. Compared to subjective self-reports, physiological signals allow real-time monitoring and adaptation, enabling robots to dynamically adjust their interaction strategies.

Stress is a complex physiological and psychological response that involves intricate interactions between various hormonal, neurological, and behavioral factors [63]. So, to detect stressing conditions with the utmost reliability and comprehensive detail, employing a multimodal approach that incorporates metrics derived from various biological signals is essential. According to [64] there are several involuntary responses induced by stress, mediated by the autonomic nervous system that can create characteristic patterns in some psycho-physiological signals, such as ECG, GSR, EEG, and facial expression.

As evidenced by the literature, **ECG** is a physiological measure that can be used to evaluate stress [65]. The cardiac signal is characterized by peaks denoted with the letters P, Q, R, S, and T [66]. R-peaks indicate ventricular depolarization of the heart and are the most prominent: several analyses exploit the distribution of these peaks and the distances between successive R peaks, also known as RR intervals (RRi). This parameter, which shows an inverse correlation with stress, and its inverse, Heart Rate (HR), are the most widely adopted and straightforward measure to estimate stress level [64]. HRV, another ECG derived parameter, is the distribution of RR intervals [RRi, RRi+1,...] over a given time interval, and reflects the activity of the sympathetic and vagal components of the autonomic nervous system and is related to stress [67].

GSR, also known as ElectroDermal Activity (EDA) of the skin is another signal that can be monitored in a simple and non-invasive way to evaluate stress [68]. It measures the electrical conductance of the skin, which is affected by the activity of sweat glands, activated during states of emotional arousal and anxiety, and found to be positively correlated with stress. GSR signal can be decomposed into two main components: a slower and non-specific tonic component due to the basic activity of the sweat glands and a faster stress-specific phasic phase [69]. A stress-induced stimulus creates a peak in the phasic component: significant features can be derived from this, such as rise time [70], which is inversely proportional to stress, the amplitude of the peak [71] and recovery time [70], as shown in Fig. 3.1.

Also, signals obtained from the electrical activity of the brain are useful for stress evaluation and stress monitoring. Brain signals obtained from **EEG** can be divided into different frequency bands: *Delta* (0.5-4 Hz), *Theta* (4-8 Hz), *Alpha* (8-12 Hz), *Beta* (12-30 Hz), and *Gamma* (>30 Hz). Each of these corresponds to different activity of the brain [72, 73]. Stress can disrupt the balance of these brainwaves, often leading to an overproduction of *Beta* [74] and *Theta* wave activity [14, 75] and a decrease in *Alpha* waves [76]. The literature [77] suggests that the *Alpha* activity in EEG signals may be reduced during stress conditions, while *Beta* activity may be increased: it follows that

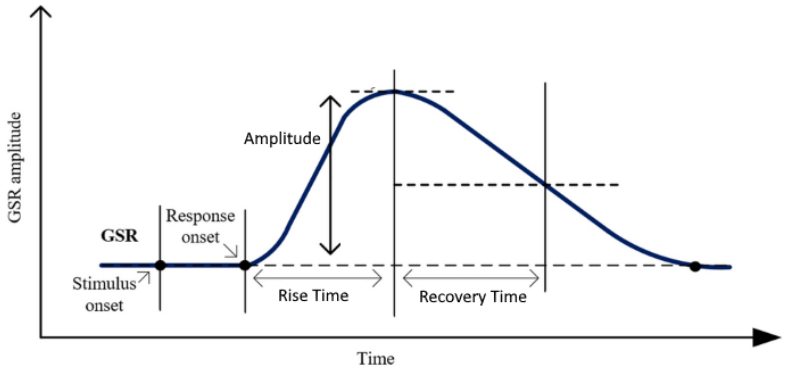


Figure 3.1: GSR Phasic stress peaks with some of the main metrics reported: the **Rise Time**, that is the time interval between the beginning of the peak and its maximum point, the **Recovery Time**, namely the time of descent from the peak and the height of the peak or **Amplitude**.

the $\frac{\beta}{\alpha}$ power ratio [78] is considered as a measure of cognitive load associated with the arousal dimension and stress. Furthermore, EEG Alpha Asymmetry, which refers to the difference in power or amplitude of EEG signals in the *Alpha* frequency band between the two frontal lobe hemispheres, is a stress-related metric. In particular, an increase in *Alpha* activity of the left frontal hemisphere respect to the right is positively associated with stress [79].

Furthermore, studies [80] have shown that **facial expressions** can be used to detect stress and anxiety levels: for example, the corrugator supercilii muscle, which is responsible for frowning, is activated during stress conditions, while the activity of the zygomaticus major muscle, responsible of smiling, is decreased during stress conditions.

While objective parameters obtained from the analysis of physiological signals can provide analytical information, they may not capture the individual's subjective experience accurately: the two dimensions are complementary and cannot be separated from each other. **Individual assessment questionnaires** are essential to capture this aspect. The NASA-TLX [81] is a popular tool used to evaluate mental workload and stress: it is a subjective, multidimensional assessment tool that rates perceived stress to assess the effectiveness of a task, system, or team, as well as other aspects of performance. Other questionnaires widely used for the evaluation of subjective stress and mental workload are: the Workload Profile (WP) [82], the Subjective Workload Assessment Technique (SWAT) [83], and others [84, 85, 86].

Recently, many works have been carried out, demonstrating the importance of stress assessment in the field of collaborative robotics. The study of Mariscal *et al.* [87], mainly focusing on pupil diameter, aimed to assess occupational risks in terms of mental stress to determine whether a worker experiences greater stress when working in collaboration with a cobot rather than with another person while performing the same production-line process. The work of Zakeri *et al.* [88] discussed the assessment of the mental workload of factory workers in an HRC environment, using EEG signals and subjective questionnaires. In [89], the researchers evaluated stress experienced by cobot programmers, analyzing EDA. Furthermore, the study of Pollak *et al.* [90] focused on the levels of psychological (primary and secondary stress appraisal) and physiological (heart rate) stress in human operators working in two different cobot modes (i.e., manual and autonomous). In a

Figure 3.2: Niryo Ned robot.

recent study by Bussolan *et al.* [91], a method was used to assess stress levels and cognitive load in operators engaged in collaborative robotics tasks, by analyzing cardiac, muscular and skin activity.

3.1.1 Electroencephalography for workload estimation

The study presented in our preliminary work [14] represented an initial step toward understanding how brain activity reflects cognitive workload during human–robot collaboration. Specifically, the contribution lies in a pilot study that compares two types of biosignals, EEG and EDA, with the aim of measuring the **workload** of participants during different types of tests, including a collaborative robotic experiment using the Niryo Ned robot[1], shown in Fig. 3.2. The EEG signals were recorded using the Neuroelectrics Enobio20 Helmet[2], illustrated in Fig. 3.3, and were processed with an automatic algorithm that allows pre-processing operations (data removal, downsampling, rereference, filter application, bad channels removal) and application of independent component analysis (ICA) algorithm for artifact removal. In particular, to facilitate the identification of ocular artifacts, two Sticktrode electrodes were used for the acquisition of electroculographic (EOG) signals. EDA activity was recorded by means of a BITalino (R)Evolution device, which is shown in Fig. 3.4, using a bipolar montage from the palmar sites of the hand. In this way it is possible to evaluate the degree of accuracy of the indices found in the literature for the assessment of mental fatigue during human-robot collaboration tasks.

[1] `https://niryo.com/product/ned-education-research-cobot/`
[2] `https://www.neuroelectrics.com/solutions/enobio/20`

Figure 3.3: Enobio20 device.

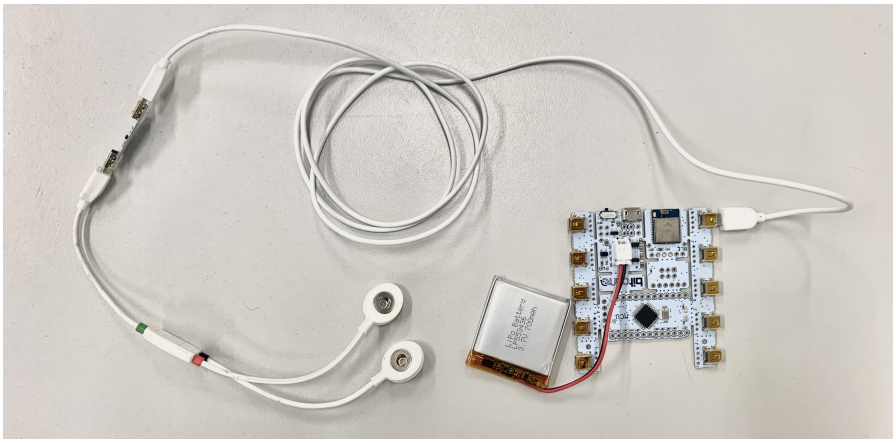


Figure 3.4: BITalino (R)Evolution Kit.

Materials and methods

A total of three male university students (age = 22 ± 1 years old) were analyzed. Participants provided written informed consent to participate in the study. The only exclusion criterion relates to the fact that users were not to be subjected to diseases or drugs affecting the nervous system.

Before starting the study, participants were instructed on the procedures and protocol requirements during the different trials. All participants underwent a period of familiarization with the equipment required for the test as they had no experience with robots. The experimental room was calm and the light and temperature were continuously regulated.

Participants conducted four different types of experiments: (A) listening to music, (B) watching a video, (C) performing human-robot collaborative assembly, and (D) solving sudoku game. In the first test, the user had to remain relaxed with his eyes closed listening to classical music with earphones, as shown in Fig. 3.5. In the second test,

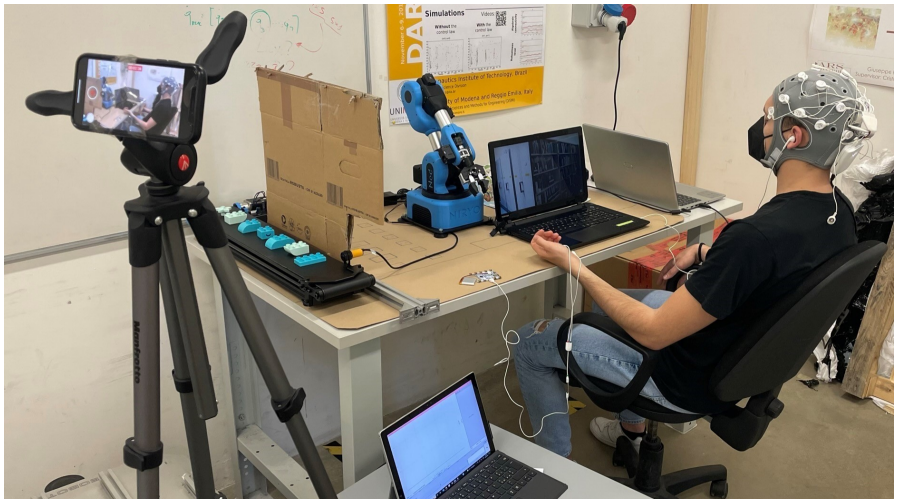


Figure 3.5: Experiment setup when subjects were listening to music.

the user had to remain relaxed while watching a video on a computer. The third test, which consisted of a collaborative robotics experiment, was organized in three tasks (C.1, C.2, C.3) of increasing difficulty. During these tests, the users had to assemble different figures by using "Duplo" bricks, supplied to them in a dedicated unloading box located inside the workstation through pick & place operations. Bricks were provided to the user in sequential or random order, depending on the task, by the robot and a conveyor equipped with a proximity sensor. In particular: regarding C.1, the user had to assemble one figure, the blocks of which were passed to them in order of construction and could see the photo of the figure at any time. About C.2, the user had to assemble two figures, whose blocks were passed to them in order of construction and in random order, moreover they could see the photo of the figures once only. Regarding C.3, the user had to assemble three figures, whose blocks were passed in order of construction and in random order and by means of a conveyor, moreover they could constantly see a drawing with different types of figures inside.

The setup for this test is shown in Fig. 3.6. In particular, the six constructions, shown in Fig. 3.7, were used during these tests, for a total of 44 pieces. In the fourth test, the user had to stay focused on playing a Sudoku puzzle of medium difficulty with a time limit. Between each experimental test (A, B, C, D) there was a pause of 3 minutes in order to prepare the experimental setup for the following test. This did not apply to the sub-tasks C.1, C.2, C.3 which were carried out in sequence.

Considering successive improvements due to pilot testing, the enrolled participants performed the following tests: Subject 1 did A, C, D; Subject 2 executed A, B, C; Subject 3 performed C.

During all the tests the participants were subjected to the acquisition of the EDA activity. At the end of the experiment, the participants had to fill in a questionnaire to collect their personal evaluation regarding their state of relaxation and concentration during the various tests.

The EEG data was evaluated using the Enobio20 device, a wireless electrode system, using a referential montage. Particularly, EEG was recorded from 19 scalp locations,

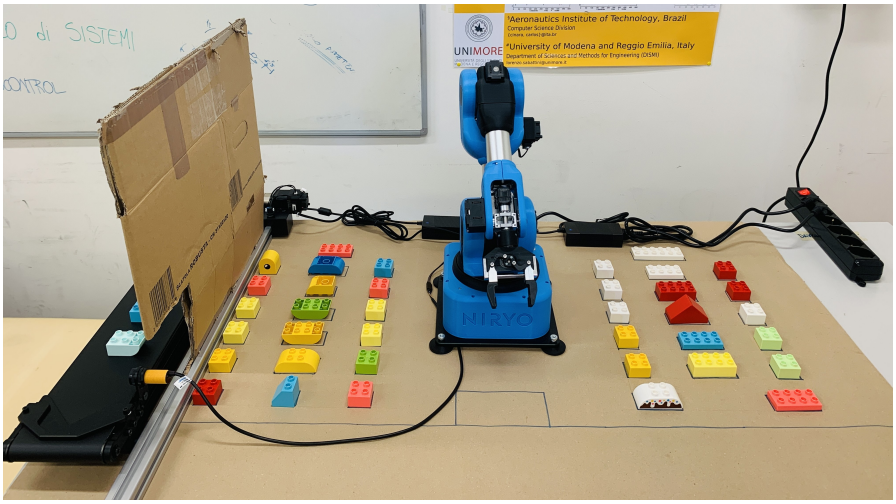


Figure 3.6: Setup for the human-robot collaboration test.

Figure 3.7: Figures to be assembled in the human-robot collaboration test.

according to the international 10-20 system: 17 EEG electrodes were Enobio NG Geltrode using electrode gel and 2 EOG electrodes were Enobio Stricktrodes. With reference to Fig. 3.8, the considered electrodes included frontal (Fz, F3, F4, F7, and F8), central (Cz, C3, and C4), temporal (T3, T4, T5, and T6), parietal (Pz, P3 and P4) and occipital (O1 and O2) locations, and two EOG electrodes (EOG1 and EOG2). Data were collected thorugh the software Neuroelectrics® Instrument Controller (NIC2)[3], which allows a computer to interact with Neuroelectrics devices. The electrodes placed in the mastoids served as a reference while recording on NIC2. Moreover, a sampling rate of 500 Hz was used.

MATLAB 2021.a with the EEGLAB 2021.1 toolbox were used for preprocessing and data analysis. Raw signals were filtered using a 50 Hz notch filter and a 0.1–30 Hz band-pass filter, while artifacts were corrected through ICA. The processed data were then grouped into three frequency bands: *Theta*, *Alpha*, and *Beta*.

EDA signals were processed following the methodology in [92], using the Weka software environment [93].

[3] `https://www.neuroelectrics.com/solution/software-integrations/nic2`

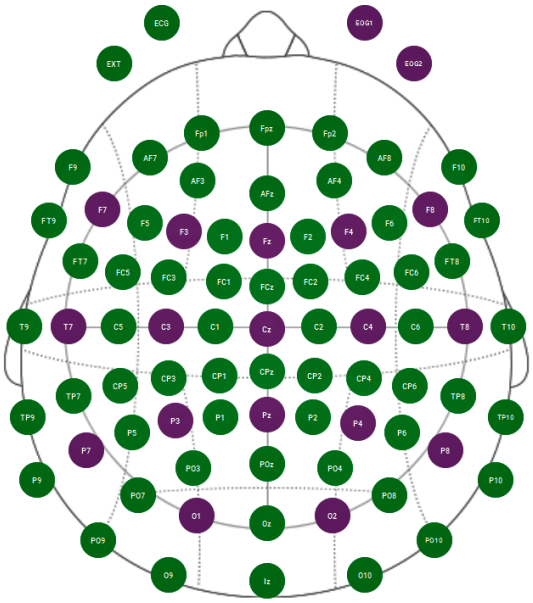


Figure 3.8: Channel locations according to the international 10-20 system.

Table 3.1: Spectral powers for the different frequency bands.

Rhythm Power	**Cortical Area**	**Channels**
Theta variation	Frontal e Temporal	F7, F3, F4, F8, Fz, T7, T8
Absolute theta variation	Frontal	Fz
Alpha variation	Parietal e Occipital	P7, P3, P4, P8, Pz, O1, O2
Absolute Alpha variation	Parietal	Pz
Beta variation	Frontal	F7, F3, F4, F8, Fz

Statistical analysis

After carrying out ICA, which allows to separate and identify independent sources in EEG data, the dataset was divided into epochs of 5 seconds, for a trade off between computational burden and reliability of results. In each epoch the spectral powers (µV/Hz) of the *Alpha*, *Beta*, *Theta* bands of the channels were calculated, considering the channels reported in Table 3.1.

Moreover, we computed the state-of-the-art indices reported in Table 3.2 [94, 95, 96, 97], founding that: i) Theta variation determines fatigue and mental workload; ii) Alpha variation decreases with arithmetic tasks and increases with creative thinking; iii) Beta variation refers to visual attention and short-term memory; iv) BAT determines involvement in tasks and mental effort; v) TA refers to increased mental load; vi) TA ASS provides the same measurement of TA, with the difference that it is obtained from the ratio of two absolute measurements that are obtained from certain channels on the scalp.

Table 3.2: Indices for comparison among frequency bands.

Indices	Definition
BAT	$\frac{Beta}{(Alpha+Theta)}$
TA	$\frac{Theta}{Alpha}$
TA ASS	$\frac{ThetaAss}{AlphaAss}$

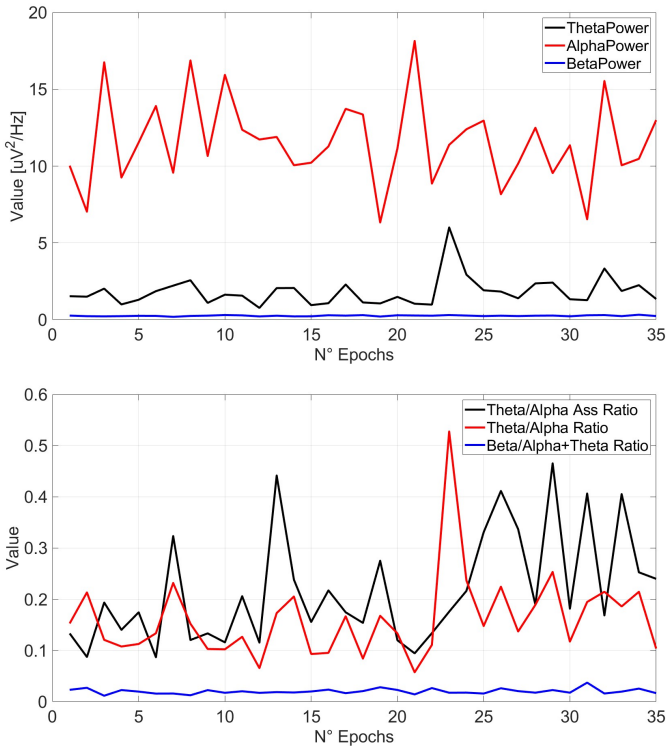


Figure 3.9: Spectral powers and synthetic indices for Subject 1 while listening to relaxing music (A).

After data processing, plots of each recording were created for each subject. Fig. 3.9, 3.10 and 3.11 show spectral powers and synthetic indices for Subject 1 in the different phases of the test (music, three collaborative tasks with the robot and Sudoku test). In addition, minimum, average and maximum values of each parameter of interest (see Tables 3.1 and 3.2) were computed and reported in Fig. 3.13.

Finally, as regards EDA activity, Fig. 3.12 plots recordings of Subject 1. It is noteworthy that an increase in the amplitude of the processed signal indicates an increase in stress and cognitive load on the user. Hence, Fig. 3.12 shows that Subject 1, according to the EDA activity, was more stressed during the tests at high workload (D, C.1, C.3) and less concentrated while listening to music (A).

Comparing the results obtained from the comparison of the two biosignals with the replies to questionnaire, it was possible to carry out an analysis of the indicators present in the literature for the analysis of EEG signals: the green tag in Fig. 3.13 has been attributed to the very good indices and the yellow tag to the sufficiently good indices. In particular, it was found that: (a) MAX THETA represents a very very good index for mental

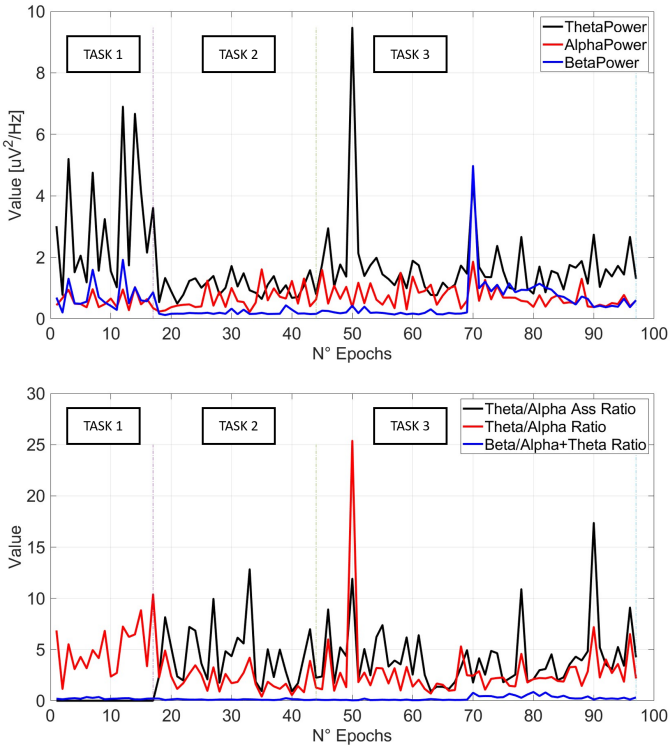


Figure 3.10: Spectral powers and synthetic indices for Subject 1 while performing three tasks of human-robot collaboration (C).

fatigue assessment. It is present when there is mental fatigue and high concentration and cognitive load. (b) MAX ALPHA represents a very good index. It is present with high values when subjects had closed eyes, otherwise high values were found during creative thinking due to mental effort. (c) MAX BETA is a sufficiently good index. It is related to visual awareness and is present when there is an increase in working memory. There were discrepancies in all participants with respect to the questionnaires. (d) MEAN THETA ASS is a very good index. It is related to the average concentration of the tests. (e) MEAN ALPHA ASS is a very good index. It is present with high values when subjects had closed eyes or during creative thinking due to mental effort. (f) MAX BAT is a sufficiently good index. It is related to mental effort. It has a discrepancy in Subject 2 as it should have a higher value in Task 2. (g) MAX TA is a very good index. It is related to the high cognitive load. (h) MIN TA is a very good index. It is related to low cognitive load. (i) MEAN TA ASS represents a sufficiently good index. It is related to the average stress during the tests, but did not prove reliable in our tests

On the basis of the considerations set out above, as regards Subject 1, it is possible to conclude that: (a) From the spectral power of the Theta rhythm, calculated in F7 - F3 - F4 - F8 - Fz - T7 - T8, it can be deduced that C.2 was by far the test that required the least mental effort; (b) From the spectral power of the Alpha rhythm, calculated in P7 - P3 - P4 - P8 - Pz - O1 - O2, it can be deduced that, during C.3 and D, an increase in creative thinking was found, probably due to to a mental effort; (c) The BAT index confirms a state of high concentration during D; (d) The TA index confirms a high workload during C.1, C.3, D and a reduced workload during A as the user is particularly relaxed. These results can be considered in agreement with those obtained

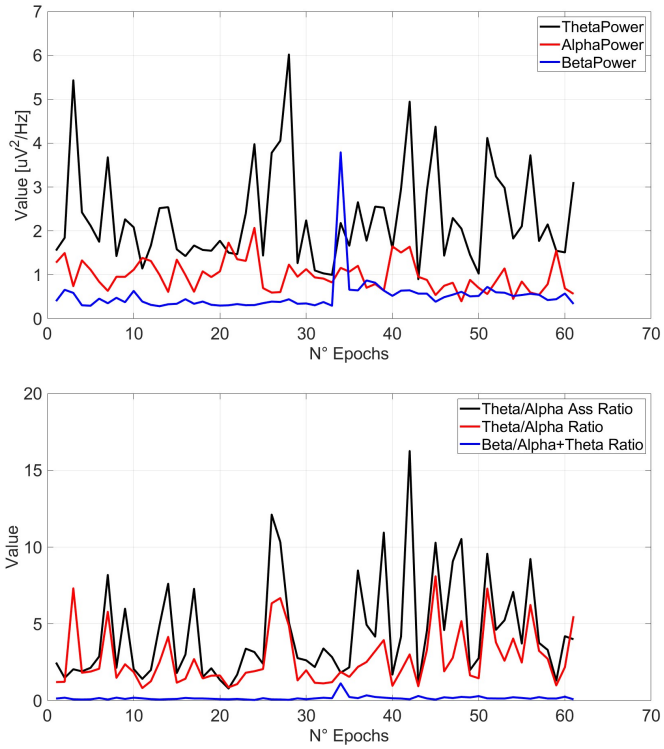


Figure 3.11: Spectral powers and synthetic indices for Subject 1 while performing a Sudoku test (D).

from the analysis of the electrodermal activity, where C.1, C.3, D were found to be the experiments with greater amplitude value than the C.2, A.

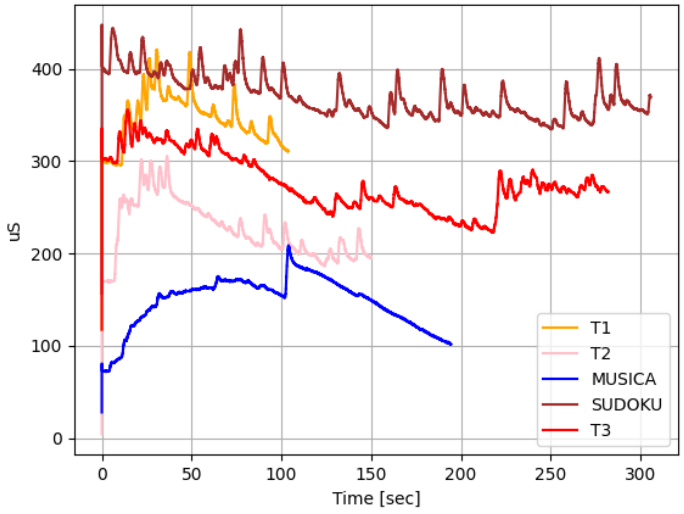


Figure 3.12: EDA activity for Subject 1.

	Mental fatigue and workload			More creative thinking			High concentration			Medium task difficulty (ABSOLUTE)		
	THETA			ALPHA			BETA			THETA ASS		
SUBJECT 1	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN
TASK1	6.90	3.00	0.78	1.03	0.59	0.28	1.92	0.75	0.20	0.00	0.00	0.00
TASK2	1.71	1.03	0.49	1.61	0.65	0.22	0.44	0.19	0.12	4.06	1.94	0.67
TASK3	9.47	1.71	0.76	1.86	0.74	0.30	4.97	0.58	0.14	4.83	2.19	0.66
TASK123	9.47	1.75	0.49	1.86	0.69	0.22	4.97	0.50	0.12	4.83	1.74	0.00
MUSIC	6.01	1.81	0.77	18.15	11.56	6.33	0.33	0.26	0.19	4.66	2.09	0.80
SUDOKU	6.02	2.31	0.90	2.07	0.99	0.40	3.79	0.52	0.29	14.66	3.21	1.00
SUBJECT 2	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN
TASK1	1.8242	1.131	0.522	1.3884	0.7634	0.386	0.9196	0.4807	0.206	2.9883	1.3327	0.587
TASK2	2.6496	1.433	0.712	2.5318	1.1048	0.623	0.7421	0.3409	0.205	3.0356	1.7228	0.793
TASK3	1.4935	0.913	0.439	2.55	1.1647	0.564	1.3896	0.3797	0.191	2.4004	1.3759	0.612
TASK123	2.6496	1.1	0.439	2.55	1.0737	0.386	1.3896	0.3874	0.191	3.0356	1.466	0.587
MUSIC	1.9564	1.096	0.723	28.7	11.595	0.263	1.004	0.5157	0.171	3.2526	1.892	1.059
VIDEO	1.4855	0.871	0.592	2.0472	0.962	0.422	0.6258	0.3213	0.186	2.8322	1.3636	0.826
SUBJECT 3	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN
TASK1	3.2023	1.348	0.71	3.3934	1.0153	0.592	0.704	0.3957	0.296	2.4398	1.3092	0.556
TASK2	0.9663	0.665	0.439	1.8075	0.6709	0.427	0.2845	0.1792	0.113	1.5418	0.8522	0.406
TASK3	12.1753	1.176	0.596	6.6732	1.1503	0.516	0.6696	0.34	0.164	11.518	1.4691	0.477
TASK123	12.1753	1.075	0.439	6.6732	1.0117	0.427	0.704	0.3084	0.113	11.518	1.2928	0.406

	Medium creative thinking (ABSOLUTE)			Mental effort			Higher Workload		Lower Workload		Brain Beat: Stress	
	ALPHA ASS			BAT			TA			TA ASS		
SUBJECT 1	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN
TASK1	1.19	0.57	0.22	0.36	0.22	0.13	10.39	5.15	1.15	0.00	0.00	0.00
TASK2	1.66	0.57	0.13	0.25	0.12	0.07	4.90	2.05	0.40	12.83	4.43	0.93
TASK3	1.55	0.62	0.19	0.86	0.25	0.04	25.38	2.90	0.70	17.36	4.21	0.94
TASK123	1.66	0.60	0.13	0.86	0.21	0.04	25.38	3.06	0.40	17.36	3.53	0.00
MUSIC	24.07	10.87	4.34	0.04	0.02	0.01	0.53	0.16	0.06	0.47	0.22	0.09
SUDOKU	2.57	0.93	0.27	1.13	0.17	0.05	8.10	2.72	0.83	16.25	4.47	0.79
SUBJECT 2	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN
TASK1	1.3064	0.6118	0.288	0.415	0.256	0.142	2.5937	1.585	0.6013	4.617	2.4597	0.751
TASK2	1.7578	0.8921	0.347	0.247	0.136	0.077	2.1635	1.348	0.4177	6.5246	2.2677	0.703
TASK3	1.4037	0.6733	0.184	0.765	0.192	0.082	1.7298	0.869	0.3501	11.318	2.4055	0.9
TASK123	1.7578	0.7238	0.184	0.765	0.188	0.077	2.5937	1.137	0.3501	11.318	2.3766	0.703
MUSIC	23.866	8.6462	0.147	0.188	0.067	0.019	4.523	0.491	0.031	18.314	1.5754	0.075
VIDEO	2.2531	0.7548	0.409	0.298	0.183	0.094	1.9023	1.032	0.3595	4.667	2.1183	0.371
SUBJECT 3	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN	MAX	MEAN	MIN
TASK1	0.8507	0.4449	0.225	0.259	0.181	0.107	2.7498	1.438	0.8976	4.5749	3.1417	1.025
TASK2	0.936	0.4787	0.265	0.223	0.138	0.078	1.5557	1.059	0.4713	3.3055	1.9576	0.758
TASK3	2.6105	0.6165	0.192	0.33	0.169	0.017	2.0393	1.029	0.4857	8.4294	2.6829	0.664
TASK123	2.6105	0.5572	0.192	0.33	0.163	0.017	2.7498	1.097	0.4713	8.4294	2.5706	0.664

Figure 3.13: Minimum, mean and maximum value of spectral powers and synthetic indices for each test participant, in each phase of the experimental procedure.

3.1.2 Multimodal stress evaluation

In the era of Industry 4.0, the study of HRC in advancing modern manufacturing and automation is paramount. An operator approaching a cobot may have feelings of distrust, and experience discomfort and stress, especially during the early stages of training. Human factors cannot be neglected: for efficient implementation, the complex psycho-physiological state and responses of the operator must be taken into consideration.

The objective of our work in [3] was to evaluate how learning cobot programming and engaging in the act of programming a cobot for the first time influence an individual's stress levels. For this purpose, this study used a learning platform defined in previous work [98], which provides a safe and interactive environment to learn about the capabilities and limitations of collaborative robots and helps users gain the skills and knowledge needed. To assess the users' stress while learning to program a robot, we recorded a wide set of psycho-physiological signals to monitor their involuntary responses induced by stress.

Materials and methods

The experiment consisted in training 21 participants (17 male and 4 female), between the ages of 22 and 29 (M = 22.43 years, SD = 2.09), novice to cobot programming to perform a pick-and-place task. The study comprises three distinct stages, outlined extensively as follows: introduction to the learning materials, baseline measurement task, and hands-on practice.

The educational framework has been meticulously designed to progressively furnish information essential for each phase of the tasks, facilitating an accessible learning experience for beginners in cobot programming. This incorporates informative slides, videos, and audio components. While they were exposed to the training program, their physiological activity was recorded and their subjective feedback was collected to assess the stress induced by HRC and robot learning. The recorded data are available in the publicly available dataset SenseCobot[4], described in [13].

To simulate a real-world situation of a novice operator approaching a cobot, the only inclusion criterion was the absence of any prior experience in cobot programming [13]. Participants, all management and mechatronic engineering students, were recruited voluntarily with online scheduling and were instructed to avoid substances like coffee, nicotine, and alcohol on the day of the experiment, as these could potentially affect brain activity and stress-related physiological signals. Each participant filled in a consent form in which they authorized the use of their data for scientific research purposes.

For this particular experiment, a cobot UR10e[5] equipped with a robotic gripper was used, but the procedure applies to other models and usage scenarios.

A learning platform was organized according to the previous work presented in [98],

[4] https://doi.org/10.5281/zenodo.8363762
[5] https://www.universal-robots.com/it/prodotti/ur10e/

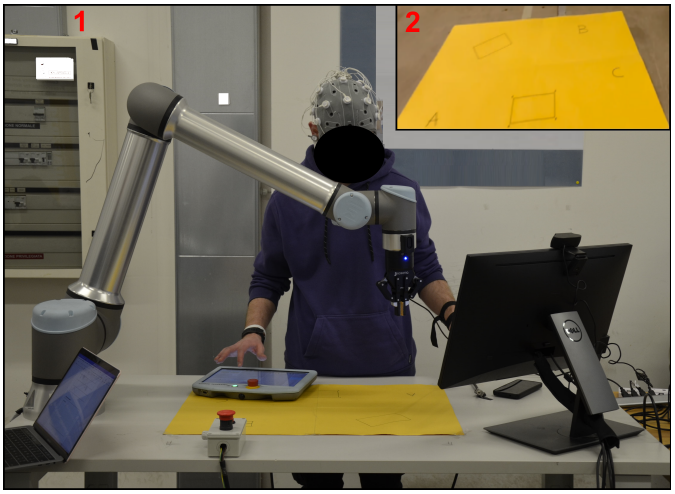


Figure 3.14: Experimental set-up. The participant (1), dressed with sensors in front of the workbench, has the cobot on their right and the monitor with learning materials on their left. On Top right (2), a detail of the workbench with the three points A, B, C and the rectangular shape where to place the box.

which aims at making cobot programming as easy as possible for novices. This previous work hypothesized that if unskilled workers were better informed about collaborative robots' functionalities and received systematic training, they could gain the ability to independently program cobots in an effective way. The platform provides information such as video, audio, and explanatory slides and guides learners through the different skills required for robot programming.

The workstation was arranged as depicted in Fig. 3.14. The attendee stood in front of a table, with a touchscreen monitor that had a webcam on the left side and the cobot on the right side. Three reference points (A, B, and C) and a box layout, necessary for the task, were marked on a cardboard on the table. The accessories necessary for task execution, together with a manual switch necessary for activating the free-drive mode, were placed within easy reach, to minimize operator's movements and distractions. The experiment was logically organized following a gradual progression of the complexity of the tasks.

Different wearable devices were used to collect several physiological signals. The iMotions software platform[6] was used, since it facilitates data synchronization and organization, thanks to the integration of different communication protocols.

A Shimmer3 ECG[7] with a sampling rate of 512 Hz and equipped with five Ag/AgCl electrodes was used to measure the ECG. Moreover, a Shimmer3 GSR[8] with a sampling rate of 512 Hz was used to detect real-time GSR. For brain activity monitoring, an Enobio 20 EEG Helmet with 20 channel electrodes positioned according to the 10:20 standard [14], organized in 17 EEG channels, 2 electrooculography (EOG) ocular electrodes, and a reference EXT channel, was used. The sampling rate of this device is 500 Hz. For recording facial expressions, the iMotions AFFDEX integrated toolkit was used [99]. This module is a convolutional and recurrent neural network trained for recording the 3D position

[6] https://imotions.com/
[7] https://shimmersensing.com/product/shimmer3-ecg-unit-2/
[8] https://shimmersensing.com/product/shimmer3-gsr-unit/

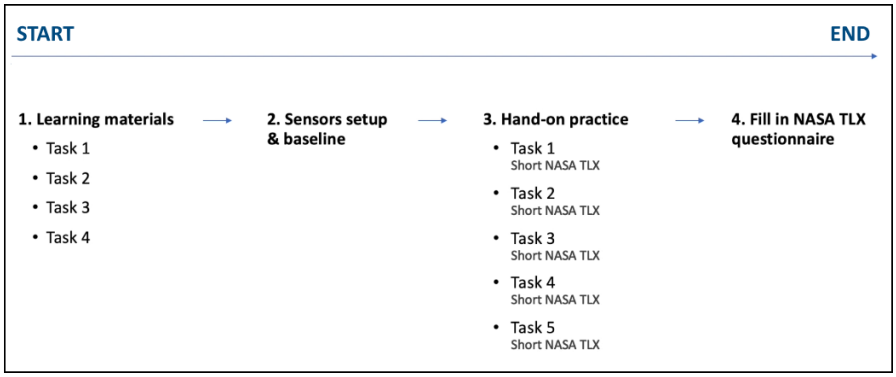


Figure 3.15: Experimental protocol. Temporal succession of the phases of the experiment. As can be seen in the figure, at the initial stage (learning materials), the participant performs the 4 tasks without constraints. Then, the participant is dressed with the sensors and performs the Baseline (sensors set-up and baseline). Subsequently, the 4 previously performed tasks plus a new one (Task_5) are repeated for the participant dressed with sensors, each interspersed with a questionnaire. The experiment ends with a final complete NASA-TLX.

of the user's head, detects the action units of the face, and recognizes the fundamental emotions (joy, surprise, fear, disgust, anger), plus neutral, sentimentality, attention, and confusion. AFFDEX measures accurately unfiltered and unbiased facial expressions of emotion, using just a standard webcam. In addition to physiological signals, other assessments were taken into consideration, such as time of task execution, number of errors, and programming mode used.

A detailed description of the experiment can be found in the previous work [13]. The experimental setup was structured to include the 4 macro-phases described hereafter, and summarized in Fig. 3.15. The tasks are performed in succession, with a short break between them necessary for the participant to complete the questions of the subjective questionnaire.

1. **Introduction to learning materials** - This phase aimed to introduce the users to cobot programming using dedicated learning material and assigning four basic tasks to complete [13]. Participants were not bound by time constraints and were allowed to refer to the instructions as needed, with no monitoring of errors. The four tasks performed are the following:
   - **Task 1** = moving the robotic gripper of the cobot arm from point A to point B;
   - **Task 2** = directing the cobot's arm to reach three specified points (A, B, and C);
   - **Task 3** = programming the cobot to reach an arbitrary height from point A(A') and then approach point A while maintaining its orientation perpendicular to the working surface. After a short pause, the cobot was to ascend back to point A';
   - **Task 4** = programming the cobot to perform a pick-and-place operation in which the user had to use the teach pendant to pick up a screw placed at point A and move it to point B by activating the robotic gripper (see rectangular shape in Fig. 3.14).

2. **Sensors setup and baseline** - The primary purpose of the baseline was to objectify and standardize the psycho-physical conditions for each participant, establishing a reference point for comparing signals obtained during other tasks. This phase involved recording the participant's signals while they were positioned in front of a stationary blue screen for three minutes, in the absence of sounds or other visual stimuli.

3. **Hands-on practice** - Participants were assigned to perform five tasks, with the first four being the same as those of the "Introduction to learning materials" phase, plus one additional new task: Task 5. For Task 5, participants were tasked with programming a cobot to manipulate a box initially positioned at a 45° angle relative to the surface, ensuring precise placement onto a predetermined location on the plane. Following this, participants were required to guide the gripper robot along a predefined path at a consistent speed, simulating a gluing operation. This task encompassed all the previously acquired skills and presented a more intricate challenge. These tasks were structured in a sequence of increasing complexity, as shown in Fig. 3.15 to simulate the challenges one might encounter in a real work environment. In all cases, users had the option of moving the cobot arm, via the teach pendant, using either jogging mode or free-drive mode. Furthermore, no constraints have been defined for the user, to observe the possible number of reactions, errors, and strategies without imposing bias. The effect of the user's choice was documented for the potential assessment of stress variation, errors, or variations in psycho-physiological signals.

4. **Questionnaire** - After each task, participants were asked to fill a reduced version of the NASA-TLX questionnaire, expressing their perceived level of stress concerning only two categories: physical exertion and the overall effort. It is worthwhile noting that the NASA-TLX questionnaire was formerly introduced to measure the overall task load. However, in the proposed analysis it was used for stress evaluation, as in previous works such as [91, 100, 101, 102]. The evaluation of the subjective stress and mental workload was expressed using a 7-point Likert scale (1: "very low"; 7: "very high"). If the sum of the scores of the two questions used was lower than 7, the task was labeled as NO STRESS; otherwise, it was labeled as STRESS [13].

Processing and analysis of recorded data

A signal processing pipeline was implemented in Python to extract the features and metrics that characterize stress. In particular, the Scipy library[9] was used for ECG signal processing, Neurokit 2 library[10] for GSR analysis, and MNE library[11] for processing EEG. Before the analysis of signals obtained and synchronized by iMotions, the timestamp was converted into a suitable format, and null values, gross recording errors, or files unusable for the study were eliminated. Due to recording errors, Participant 1's Baseline signal and Participant 10's entire ECG recording were discarded.

[9] `https://docs.scipy.org`
[10] `https://neurokit2.readthedocs.io/en/legacy_docs/`
[11] `https://mne.tools/stable/index.html`

The time windows chosen for metric extraction were set to a 10 s, an interval deemed sufficient to capture the slowest events (identified in this work as those from the GSR signal, according also with the literature [103]) and obtain meaningful HRV analysis [104].

Regarding the **ECG signal processing**, the time intervals between consecutive R peaks, known as RR intervals, and their derived measures are important parameters for stress detection. The steps performed to obtain this metric are described below:

1. **ECG data normalization:** The ECG signal was normalized by considering the 10^{th} and 90^{th} percentiles to mitigate subjective variations [105];
2. **ECG filtering:** To remove noise and artifacts corrupting the recorded signal, a band-pass filter with bandwidth from 10 Hz to 75 Hz was used [106];
3. **R peaks finding:** R peaks were detected using the "find peaks" function in the SciPy library. To discard erroneously detected peaks, the procedure in [107] was considered. Specifically, the fluctuations of the filtered ECG signal were smoothed by applying a Moving Average technique, with a window set to 1000 samples, and subsequently, a threshold, corresponding to one standard deviation above the moving average, was determined [107];
4. **RR intervals and HR calculation:** RR intervals between successive heartbeats were calculated by determining the differences between consecutive R-peaks. The instantaneous HR could then be derived, resulting in the number of beats per minute [108];
5. **HRV metrics extraction:** Some of the main metrics related to HRV, such as `PNN25`, `PNN50`, root mean square of successive differences (`RMSSD`), and RR intervals standard deviation (`SDNN`) were derived [109, 110, 111]). In particular, `SDNN` was calculated as standard deviation of RR intervals, `RMSSD` as mean of the squared differences between successive RR intervals in squared root. `PNN25` and `PNN50` were obtained as the percentage of consecutive RR intervals differences that are greater than 25 or 50 milliseconds respectively.

As a result, the ECG metrics used for subsequent analysis are `RR Intervals` expressed in ms and HRV metrics `PNN25`, `PNN50`, `RMSSD`, and `SDNN`.

Regarding **GSR signal processing**, the following steps were implemented, using the Neurokit 2 library:

1. **GSR filtering:** To enhance the signal-to-noise ratio of the recorded GSR signal, the `nk.eda_process` module was used. It consists in a two-step filtering process. Firstly, the high-frequency noise is removed. Secondly, since the sensor is mounted on the wrist during the execution of a dynamic task, motion artifacts are removed;
2. **Tonic and phasic components extraction:** tonic and phasic components of the GSR signal were extracted by a dedicated data processing, designed according to the model proposed by [112]. This was done by the Neurokit 2 `nk.eda_process` module;

3. **GSR features extraction:** Finally, parameter analysis and peak identification were executed with the `nk.eda_peaks` module.

The GSR metrics used for subsequent analysis are `Amplitude` (amplitude of the peaks in μS), `Rise Time` and `Recovery Time` (each of these in ms).

Regarding **EEG Signal Processing**, the MNE Python library was used, as it provides a wide range of modules for detailed EEG signal analysis. EEG processing was organized in the following manner:

1. **EEG signal pre-processing:** Recorded data were pre-processed by discarding the beginning and end of the recording signal (5 seconds) and downsampled from 500 Hz to 250 Hz. Channels related to EOG, necessary for detecting eye movement artifacts but useless for analysis, were then discarded;

2. **EEG filtering:** A band-pass filter with low cut-off frequency of 1 Hz and high cut-off frequency of 30 Hz was applied to remove noise and artifacts, according to previous works [113];

3. **Bad channels and artifacts removal:** Channels that exhibit statistically deviant behavior compared to the other channels were identified as bad channels and removed [114]. These channels exhibit very different behavior due to loss of electrode contact or other artifacts. In our work, the selection of bad channels was carried out with the kurtosis method of the MNE library. To carry out statistical comparisons, the following approach was considered: if the same channels were identified as bad for at least four of the five subsequent tasks, including the baseline task, they were removed. Subsequently, any movement-related artifacts, including ocular artifacts, were removed using the ICA method [115] from the remaining channels;

4. **Band power analysis:** The script used calculates band power by applying the Welch method to compute Power Spectral Density (PSD) of the frequency bands of interest (*Theta*, *Alpha*, and *Beta*) using the trapezoidal numerical integration method for each 5-second interval in the selected channels [116];

5. **EEG features extraction:** The ratio between *Beta* and *Alpha* components within the same channel was computed, together with the Asymmetry of Alpha between electrodes placed in symmetrical areas of the head. As for the findings reported in previous studies [117, 118, 119] regarding the *Theta* channel selection, the Power Spectrum for the specified frontal and temporal channels, namely F7, F3, F4, F8, Fz, T7, and T8, was averaged. Similarly, for the *Alpha* and *Beta* power spectra, the channels P7, P3, P4, P8, Pz, O1, O2, and F7, F3, F4, F8, Fz were chosen, respectively (see Table 3.1). The channels used in the evaluation of Alpha Asymmetry are two frontal channels F3 and F4. Alpha Asymmetry was calculated according to [80] as the subtraction of the alpha's power (α) natural logarithm of the left hemisphere channel (P3) from that of the the right hemisphere (P4), as provided by the following equation:

$$\text{Alpha Asymmetry} = \ln(\alpha_{P3}) - \ln(\alpha_{P4}) \tag{3.1}$$

Most research findings indicate that during times of stress, there tends to be increased alpha activity in the right frontal region compared to the left [120], resulting in a negative value of alpha asymmetry. As a result, the EEG metrics used for subsequent analysis are `Mean Alpha Power`, `Mean Beta Power`, `Mean Theta Power`, `Mean Beta/Alpha Power Ratio` and `Alpha Asymmetry`.

Lastly, we have performed the **emotion processing**. The raw AFFDEX module output of the iMotions software is a time series of the selected emotions: anger, disgust, fear, joy, sadness, and surprise. The algorithm implemented for analyzing emotions was structured in the following way:

1. **Data normalization:** Each column of the raw output was normalized by dividing the absolute value of each data point by the corresponding maximum value so that the signals vary between 0 and 1;
2. **Emotions features extraction:** For each frame of registration the dominant emotion recorded was selected, to have a time-dependent evolution of emotions.

Methodology for statistical analysis

The main objective of the statistical analysis was to evaluate whether a significant difference among the psycho-physiological parameters collected during task execution can be observed if compared with the baseline.

To this end, the following research questions were considered:

Q1: *Is the physiological response coherent with task complexity*?

Q2: *Is the physiological response strongly individual*?

Q3: *Are variations in the physiological response coherent with variations in the subjective perceptions*?

To answer the research questions, we considered the parametric t-test or non-parametric Mann Whitney test, parametric ANOVA or non-parametric Kruskal Wallis test, and parametric Tukey or non-parametric Dunn Post-hoc test, as illustrated in Fig. 3.16.

The tests used were all non-parametric, since the samples have different numbers of elements, and the condition of normality was not verified for all the metrics of interest. Moreover, the various tasks in the experimental protocol showed a progressively increasing level of complexity and the same sequence was maintained for all participants. As a result, statistical independence of the samples was not presumed for this experiment. For all statistical tests carried out for analysis, two significance values (α) of 0.05 and 0.01 were used.

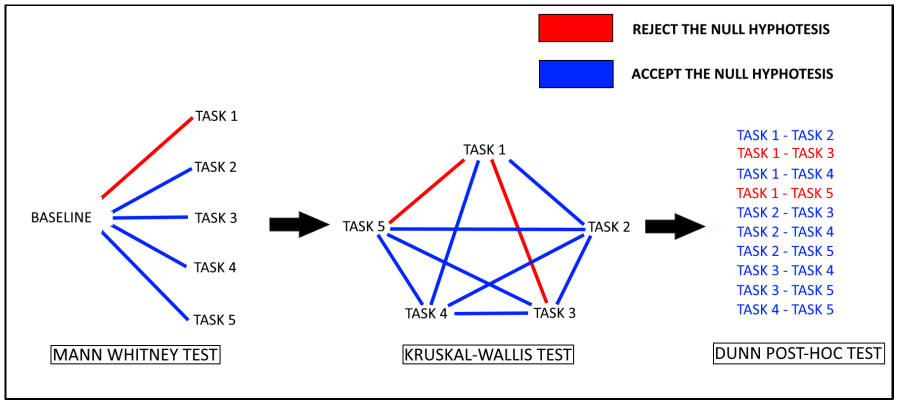


Figure 3.16: Succession of statistical tests performed for each metric extracted from each signal. In blue, the statistical test accepts the null hypothesis: samples confronted are similar. In red, the statistical test rejects the null hypothesis: samples confronted are different.

The statistical analysis is reported in detail below:

- **Normality test:** The data in each sample were tested using the D'Agostino and Pearson normality test with $\alpha = 0.05$, to determine if they follow a normal distribution [121]. This test was selected because of its robustness and precision even on modest-sized samples;
- **Comparison of physiological parameters between tasks and baseline:** During the execution of the five tasks, the signals of individual participants were statistically compared to their respective baseline (See Fig. 3.16). If the normality of the data was confirmed, a parametric t-test was conducted, otherwise, a non-parametric Mann-Whitney test with the same value was conducted instead [122]. The objective was to verify if the psycho-physiological parameters, as expected, were different compared to the resting condition;
- **Differences between tasks:** In order to reply to research question **Q1**, the study has to verify if the metrics extracted were different between tasks. If data were normally distributed, a parametric ANOVA test was used, otherwise, a non-parametric Kruskal-Wallis test (See Fig. 3.16); each of these tests can be applied to samples of a number greater than two [123];
- **Dunn Post Hoc test with Bonferroni Correction:** In order to reply to question **Q2**, the stress response among individuals has to be investigated, for each task. If the Kruskal-Wallis test yielded a rejection for $\alpha = 0.05$, that is, the compared samples are different, a subsequent parametric Tukey test or non-parametric Dunn test was conducted to assess variations among individual pairs of tasks, distinguishing those that exhibited differences from those that did not [124] (See Fig. 3.16);
- **Comparison with Subjective Evaluations:** In order to reply to research question **Q3**, the outcomes of the statistical analysis with the subjective assessments derived from the NASA-TLX questionnaire: this facilitated the harmonization of quantifiable physiological data with the participant's perspective. The objective was to evaluate how many times a variation detected in statistical analysis corresponds to a variation in subjective perception. This gives us information on the percentage of agreement between the observed stress-related physiological parameter and the subjective perception.

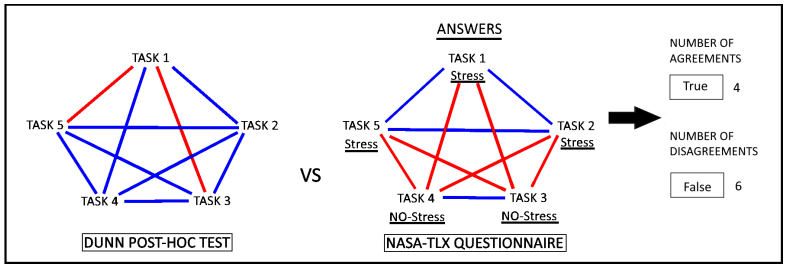


Figure 3.17: Schematic comparison between the statistical result obtained during the execution of the Dunn's Post Hoc test and the subjective responses from the NASA-TLX questionnaire. In blue, samples confronted with the Dunn Statistical test are similar, and answers from NASA-TLX are the same. In red, samples confronted with the Dunn Statistical test are different, as are the answers from NASA-TLX. On the right, in **NUMBER OF AGREEMENTS**, **TRUE**, represents the number of times in which a difference recorded by the statistical test corresponds to a difference observed in the questionnaire answers, **FALSE** otherwise.

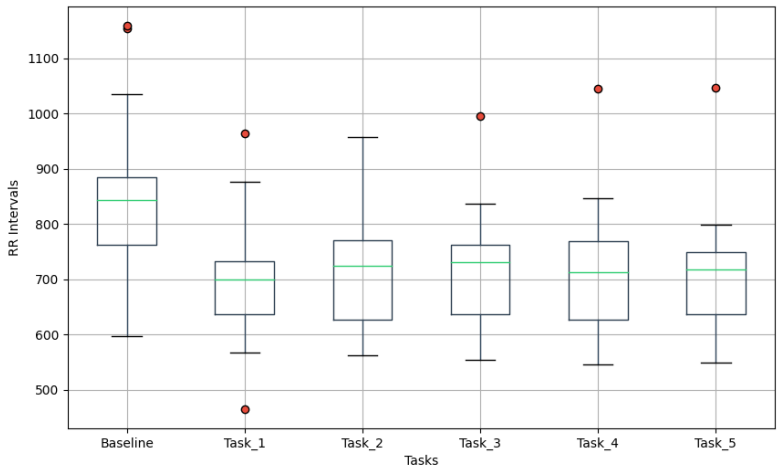


Figure 3.18: Box plot of ECG `RR Intervals` for baseline and the different robot programming tasks.

Statistical analysis was carried out using ad-hoc Python programs and Python standard libraries such as Scipy[12], Scikit-Learn[13] and Statsmodels[14].

Results

Table 3.3 reports mean and standard deviation values for RR interval, HRV, GSR, EEG, and emotions metrics. In particular, regarding **ECG analysis**, data from two subjects were discarded due to poor quality of recorded data [13], thus resulting in a total valid sample size of 19 subjects.

Figure 3.18 reports the box plot of `RR Intervals` recorded for all 19 valid samples. Moreover, Table 3.4 summarizes the number of subjects for which a statistically significant difference was observed for `RR Intervals` among baseline and the robot programming tasks, while Table 3.5 reports the same type of results for `PNN25`, `PNN50`, `RMSSD`, and `SDNN`. Specifically, the first set of rows reports the results of the Mann-Whitney test,

[12] `https://docs.scipy.org`
[13] `https://scikit-learn.org/stable/`
[14] `https://www.statsmodels.org/stable/index.html`

Tasks	ECG Metrics	Mean	Std	EEG Metrics	Mean	Std	GSR Metrics	Mean	Std
Baseline		947.50	240.33		11.90	7.72		271.55	17.40
Task 1		799.62	222.71		14.70	5.65		243.99	23.50
Task 2	RR	831.03	253.39	Alpha	16.49	7.33	Rise	246.50	37.37
Task 3	intervals	816.35	248.16	Power	17.07	8.02	Time	240.92	27.72
Task 4		826.06	269.43		15.75	8.24		268.64	27.70
Task 5		812.72	257.86		16.40	7.03		282.10	19.64
Baseline		83.56	10.13		21.89	11.34		323.34	38.92
Task 1		73.44	18.62		26.70	12.64		320.97	45.59
Task 2	PNN25	74.95	17.24	Beta Power	27.97	10.42	Recovery	312.92	46.27
Task 3		75.24	17.29		26.43	10.17	Time	309.73	47.50
Task 4		73.61	17.86		31.14	14.42		336.06	47.76
Task 5		72.91	18.64		33.18	11.98		378.75	38.20
Baseline		77.53	13.21		35.92	30.61		7.19	20.4
Task 1		65.63	20.26		50.20	31.77		40.3	48.8
Task 2	PNN50	67.22	18.67	Theta	55.70	35.68	Amplitude	85.1	159
Task 3		68.12	19.75	Power	56.10	35.48		94.4	157
Task 4		66.16	19.85		47.58	31.44		24.4	26.6
Task 5		65.38	20.80		45.49	24.92		15.2	20.9
Baseline		37.35	17.14		2.58	1.78			
Task 1		53.83	41.66		1.96	0.65			
Task 2	SDNN	55.14	43.36	Beta/Alpha	1.98	0.82			
Task 3		52.52	38.07	Ratio	1.84	0.83			
Task 4		43.65	31.77		2.19	0.79			
Task 5		44.91	28.41		2.31	0.76			
Baseline		38.80	23.39		-0.11	0.37			
Task 1		54.16	49.20		-0.15	0.38			
Task 2	RMSSD	54.15	49.26	Alpha	-0.16	0.40			
Task 3		52.35	44.28	Asymmetry	-0.16	0.37			
Task 4		42.23	37.22		-0.24	0.47			
Task 5		43.41	34.22		-0.04	0.34			

Table 3.3: Mean (**Mean**) and standard deviation (**Std**) of all the considered metrics for baseline and programming tasks. `RR Intervals`, `SDNN`, `RMSSD`, `Rise Time` and `Recovery Time` are in ms, `Alpha Power`, `Beta Power` and `Theta Power` are in $\frac{V^2}{Hz}$, `Amplitude` is in μS.

which was used to investigate differences in the pairs consisting of baseline and each robot programming task. As shown in Table 3.4, a statistically significant difference for `RR Intervals` is observed for all the considered 19 test participants, such as to reject the null hypothesis of similarity of sample distribution, while this number decreases in Table 3.5 for HRV metrics. In summary, a significant difference compared to baseline situation is observed for `RR Intervals` for all the considered subjects, while less average variability between tasks and baseline is found for HRV metrics. This is confirmed by Fig. 3.18 and Table 3.3. As shown in Fig. 3.18, net of some outliers, the baseline has the greatest median value of `RR Intervals`, while `RR Intervals` during the programming tasks are, on average, lower. This is indicative of an increase in the heart rate and greater activation of the subject. Furthermore, it is observed that the mean values of `RR Intervals` are lower in Task 1, Task 4, and Task 5; in particular, Task 5 seems to be the one with a smaller range of values. Moreover, when excluding the baseline, the most extensive range of values is observed in Task 1 and Task 2, suggesting a higher variability in the participant's responses, and, regarding the difference between the first and third quartile, the greater distribution of RR interval values is observed in Task 2 and Task 4.

To investigate further in detail the differences among the different tasks, we carried out the non-parametric Kruskal-Wallis test (second set of rows in Table 3.4 and Table 3.5). Indeed, this test is performed on more than two sample populations to determine if a significant difference in data distribution exists. As demonstrated in Table 3.4, for all 19

Test	Conditions	$\alpha = 0.05$	$\alpha = 0.01$
Mann-Whitney	Task 1 - Baseline	19	19
	Task 2 - Baseline	19	19
	Task 3 - Baseline	19	19
	Task 4 - Baseline	19	19
	Task 5 - Baseline	19	19
Kruskal-Wallis	All Tasks	19	19
Dunn Post-Hoc	Task 1 - Task 2	9	9
	Task 1 - Task 3	13	11
	Task 1 - Task 4	16	15
	Task 1 - Task 5	16	16
	Task 2 - Task 3	11	10
	Task 2 - Task 4	14	14
	Task 2 - Task 5	13	11
	Task 3 - Task 4	9	9
	Task 3 - Task 5	13	11
	Task 4 - Task 5	12	11

Table 3.4: Statistical tests for `RR intervals`. The columns on the right report the number of test participants for which a statistically significant difference was observed among the different conditions for $\alpha = 0.05$ and $\alpha = 0.01$. Rows in blue denote consecutive tests. Total sample size = 19.

subjects in our study, a statistically significant difference is observed among the distribution of `RR Intervals` collected during the five robot programming tasks, regardless of baseline values. This number decreases for HRV related metrics, as shown in Table 3.5. It appears that the distribution of raw `RR Intervals` shows a greater difference between tasks, while HRV metrics show less variability. However, this test does not provide insights about the differences within the tasks. To this end, we performed a Dunn Post-Hoc test, to assess which pair of conditions exhibited a statistically significant difference. The last set of rows (Dunn Post-Hoc) in Tables 3.4 and 3.5 reports, for all the possible pairs of robot programming tasks, the number of subjects for which statistically significant differences were found with this test. In the tables, we highlight in blue the pairs referred to conditions administered in sequence. The results show that `RR Intervals` in such pairs of consecutive tasks were more similar than when comparing tasks administered not in succession: indeed, the tasks were designed with increasing complexity. This observation is not always verified for HRV metrics (see Table 3.5).

Finally, we compared the distribution of RR intervals and HRV metrics with the results of NASA-TLX subjective questionnaires, to assess whether they agree or not. In particular, following Fig. 3.17, we computed how often the Dunn's Post Hoc test for each metrics and the replies to the NASA-TLX questionnaire detected a change in the stress level (from "STRESS" to "NO-STRESS" and vice versa) or the same stress level between any pair of tasks. Results are shown in Table 3.6, respectively, where we report the average level of agreement between statistical tests and subjective questionnaire. These results show how changes in the considered metrics were consistent with self-reported stress levels. In particular, the table shows that HRV metrics exhibit greater agreement with self-reported stress levels than `RR Intervals`.

The same procedure was carried out for **GSR analysis** of all the 21 test participants: Mann-Whitney test for pairwise comparison between baseline and each robot programming task, Kruskal-Wallis test among all the robot programming tasks and the Dunn Post-Hoc test for multiple pairwise comparisons between all pairs of robot programming

Test	Conditions	PNN25		PNN50		RMSSD		SDNN	
		$\alpha =$ 0.05	$\alpha =$ 0.01	$\alpha =$ 0.05	$\alpha =$ 0.01	$\alpha =$ 0.05	$\alpha =$ 0.01	$\alpha =$ 0.05	$\alpha =$ 0.01
Mann-Whitney	Task 1– Baseline	6	6	10	7	5	2	3	2
	Task 2– Baseline	8	5	7	5	6	5	3	1
	Task 3– Baseline	8	4	7	5	7	7	8	4
	Task 4– Baseline	11	7	10	8	7	7	7	5
	Task 5– Baseline	10	7	11	9	7	6	6	5
Kruskal-Wallis	All Tasks	5	5	5	5	4	4	6	6
Dunn Post-Hoc	Task 1– Task 2	2	2	1	1	0	0	2	2
	Task 1– Task 3	2	2	3	3	0	0	2	2
	Task 1– Task 4	2	2	1	1	0	0	2	2
	Task 1– Task 5	3	3	4	4	1	1	3	3
	Task 2– Task 3	1	1	3	3	2	2	3	3
	Task 2– Task 4	1	1	1	1	0	0	1	1
	Task 2– Task 5	1	1	0	0	0	0	2	2
	Task 3– Task 4	1	1	1	1	0	0	1	1
	Task 3– Task 5	1	1	2	2	0	0	3	3
	Task 4– Task 5	0	0	1	1	0	0	3	3

Table 3.5: Statistical tests for PNN25, PNN50, RMSSD, and SDNN. The columns on the right report the number of test participants for which a statistically significant difference was observed among the different conditions for $\alpha = 0.05$ and $\alpha = 0.01$. Rows in blue denote consecutive tests. Total sample size = 19.

Conditions	Agreement %				
	RR Intervals	PNN25	PNN50	RMSSD	SDNN
Task 1 - Task 2	47.37	78.95	84.21	84.21	89.47
Task 1 - Task 3	36.84	78.95	73.68	73.68	78.95
Task 1 - Task 4	31.58	63.16	57.89	57.89	63.16
Task 1 - Task 5	36.84	63.16	57.89	52.63	52.63
Task 2 - Task 3	52.63	84.21	73.68	68.42	73.68
Task 2 - Task 4	36.84	68.42	68.42	63.16	68.42
Task 2 - Task 5	47.37	52.63	57.89	52.63	68.42
Task 3 - Task 4	42.11	78.95	78.95	73.68	78.95
Task 3 - Task 5	57.89	63.16	57.89	63.16	63.16
Task 4 - Task 5	31.58	84.21	78.95	78.95	68.42
Average	**42.11**	**71.58**	**68.95**	**66.84**	**70.53**

Table 3.6: Agreement between RR Intervals, PNN25, PNN50, RMSSD, and SDNN with subjective responses from NASA-TLX, when comparing tasks. Values are in percentage over 19 subjects.

tasks. The results are reported in Table 3.7, which shows that Amplitude is the metrics with greater variability compared to the rest situation.

Finally, the percentage of agreement between the Dunn statistics tests performed on Rise Time, Recovery Time, and Amplitude and the respective subjective questionnaires are reported in Table 3.8. The Rise Time and Recovery Time metrics show the highest percentage of agreement with subjective perception (NASA-TLX), on average above 70%. Conversely, poor agreement is found for Amplitude, although this is the metrics exhibiting the greatest differences among the different experimental tasks and baseline.

Regarding **EEG analysis**, Table 3.9 summarizes the results of the statistical analyses reported previously, namely the Mann-Whitney, Kruskal-Wallis and Dunn Post-Hoc tests. The table shows that most differences were found between all the programming tasks and the baseline condition for all the metrics, while Alpha Asymmetry was the metrics that highlighted differences among the programming tasks for some subjects.

The comparisons of the brain parameters mean *Alpha*, *Beta*, and *Theta* Power spectrum, Alpha Asymmetry, and mean *Beta* to *Alpha* ratio of the power spectrum and the re-

Test	Conditions	Rise Time		Recovery Time		Amplitude	
		$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$
Mann-Whitney	Task 1 - Baseline	8	6	5	2	20	19
	Task 2 - Baseline	11	10	5	4	19	19
	Task 3 - Baseline	11	7	7	4	19	19
	Task 4 - Baseline	8	4	2	2	20	18
	Task 5 - Baseline	10	6	5	2	19	19
Kruskal-Wallis	All tasks	7	4	6	5	21	21
Dunn Post-Hoc	Task 1 - Task 2	7	3	1	1	12	12
	Task 1 - Task 3	3	2	1	1	17	17
	Task 1 - Task 4	5	4	1	1	16	14
	Task 1 - Task 5	5	1	1	1	17	13
	Task 2 - Task 3	1	1	2	1	18	17
	Task 2 - Task 4	5	4	2	2	16	16
	Task 2 - Task 5	5	4	3	2	18	17
	Task 3 - Task 4	4	4	1	0	16	14
	Task 3 - Task 5	3	3	1	0	16	16
	Task 4 - Task 5	1	1	1	1	12	10

Table 3.7: Statistical tests for GSR data. The columns on the right report the number of test participants for which a statistically significant difference was observed among the different conditions for $\alpha = 0.05$ and $\alpha = 0.01$. Rows in blue denote consecutive tests. Total sample size = 21.

Conditions	Agreement %		
	Rise Time	Recovery Time	Amplitude
Task 1 - Task 2	71.43	76.19	42.86
Task 1 - Task 3	71.43	66.67	42.86
Task 1 - Task 4	57.14	57.14	61.90
Task 1 - Task 5	57.14	66.67	47.62
Task 2 - Task 3	80.95	80.95	28.57
Task 2 - Task 4	76.19	80.95	33.33
Task 2 - Task 5	80.95	71.43	38.10
Task 3 - Task 4	71.43	76.19	33.33
Task 3 - Task 5	71.43	71.43	42.86
Task 4 - Task 5	76.19	80.95	66.67
Average	**71.43**	**72.86**	**43.81**

Table 3.8: Agreement between GSR metrics and subjective responses from NASA-TLX, when comparing tasks. Values are in percentage over 21 subjects.

sponses from the NASA-TLX questionnaires are reported in Table 3.10. The metrics derived from the EEG are among the most precise in describing and predicting the individual stress state, with accuracy percentages always above 50% and with `Theta Power Mean` close to 80%.

Lastly, statistical tests were performed to evaluate emotional responses during the execution of tasks. The results of the statistical **emotions analysis** are reported in Table 3.11. A multivariate analysis of variances (MANOVA) [125] was performed on the raw time series on the emotions recorded during the tasks. Before using this test, the requirements of normality of the data distribution and homomorphy of the covariance matrices were verified. The MANOVA returned that there is a noticeable distinction between the tasks for all 21 participants.

Following the use of the MANOVA test, for a more detailed statistical analysis of emotion signals, a Chi-square test for consistency [126] was conducted to compare the distribution of emotions for each participant in each task with their corresponding baseline. A Chi-square test for consistency [126] was conducted on the emotion distributions across the five tasks for each participant to determine whether there was a significant difference

		Mean Alpha Power		Mean Beta Power		Mean Theta Power	
Test	**Conditions**	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$
Mann-Whitney	Task 1 - Baseline	10	10	7	6	6	5
	Task 2 - Baseline	8	7	10	8	6	3
	Task 3 - Baseline	8	7	11	9	10	8
	Task 4 - Baseline	13	12	16	15	11	11
	Task 5 - Baseline	14	13	17	17	14	12
Kruskal-Wallis	All tasks	12	11	14	13	7	2
Dunn Post-Hoc	Task 1 - Task 2	1	0	2	1	0	0
	Task 1 - Task 3	0	0	3	1	0	0
	Task 1 - Task 4	4	2	3	2	1	1
	Task 1 - Task 5	4	2	4	2	1	1
	Task 2 - Task 3	1	0	2	2	0	0
	Task 2 - Task 4	2	1	3	2	0	0
	Task 2 - Task 5	5	3	4	2	1	1
	Task 3 - Task 4	0	0	3	2	0	0
	Task 3 - Task 5	5	3	7	6	1	1
	Task 4 - Task 5	4	2	4	4	1	0

		Mean Beta/Alpha Power Ratio		Alpha Asymmetry	
Test	**Conditions**	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$
Mann-Whitney	Task 1 - Baseline	8	6	8	7
	Task 2 - Baseline	8	8	6	6
	Task 3 - Baseline	8	8	6	5
	Task 4 - Baseline	15	12	11	10
	Task 5 - Baseline	13	12	8	8
Kruskal-Wallis	All tasks	10	10	12	12
Dunn Post-Hoc	Task 1 - Task 2	2	1	3	3
	Task 1 - Task 3	3	2	2	2
	Task 1 - Task 4	5	4	8	7
	Task 1 - Task 5	4	3	4	3
	Task 2 - Task 3	3	3	4	4
	Task 2 - Task 4	5	4	9	8
	Task 2 - Task 5	3	3	8	7
	Task 3 - Task 4	1	0	7	7
	Task 3 - Task 5	5	5	7	5
	Task 4 - Task 5	3	2	9	8

Table 3.9: Statistical tests for EEG data. For each metrics, the columns on the right report the number of test participants for which a statistically significant difference was observed among the different conditions for $\alpha = 0.05$ and $\alpha = 0.01$. Rows in blue denote consecutive tests. Total sample size = 21.

among them. Finally, a Bonferroni Post-Hoc test [127] was performed to evaluate those pairs of tasks that showed more differences. As a last result, the results of the comparisons with the NASA-TLX questionnaires are shown in Table 3.12.

Discussion

Figure 3.19 presents an overview of the metrics obtained from the analysis of psychophysiological signals and summarizes the agreement (in percentage) between subjective response and quantitative metrics derived from physiological signals and estimated emotions. The metrics were sorted from left to right following an increasing order of percentage of agreement between the metric and the subjective evaluation. So in the rightmost part, there are the metrics that demonstrate a higher correlation with subjective evaluations of stress from the NASA-TLX questionnaire.

Conditions	Agreement %				
	Mean Alpha Power	Mean Beta Power	Mean Theta Power	Mean Beta/Alpha Power Ratio	Alpha Asymmetry
Task 1 - Task 2	89.47	82.35	80.00	78.57	75.00
Task 1 - Task 3	73.68	58.82	73.30	64.29	58.33
Task 1 - Task 4	63.16	70.59	66.70	64.29	58.33
Task 1 - Task 5	57.89	58.82	60.00	64.29	83.33
Task 2 - Task 3	73.68	70.59	80.00	57.14	58.33
Task 2 - Task 4	78.95	76.47	80.00	57.14	50.00
Task 2 - Task 5	57.89	52.94	80.00	50.00	50.00
Task 3 - Task 4	78.95	82.35	86.70	78.57	41.67
Task 3 - Task 5	57.89	52.94	73.30	57.14	66.67
Task 4 - Task 5	73.68	70.59	86.70	71.43	33.33
Average	**70.53**	**67.65**	**76.67**	**64.29**	**57.50**

Table 3.10: Agreement between EEG metrics and subjective responses from NASA-TLX, when comparing tasks. Values are in percentage over 21 subjects.

Test	Conditions	$\alpha = 0.05$	$\alpha = 0.01$
MANOVA	All tasks	21	21
Chi square	Task 1 - Baseline	16	11
	Task 2 - Baseline	12	11
	Task 3 - Baseline	11	9
	Task 4 - Baseline	12	9
	Task 5 - Baseline	14	13
Chi square with consistency	All tasks	18	16
Bonferroni	Task 1 - Task 2	17	17
	Task 1 - Task 3	20	20
	Task 1 - Task 4	21	21
	Task 1 - Task 5	21	21
	Task 2 - Task 3	7	7
	Task 2 - Task 4	11	11
	Task 2 - Task 5	14	11
	Task 3 - Task 4	3	3
	Task 3 - Task 5	5	5
	Task 4 - Task 5	1	1

Table 3.11: Statistical tests for the analysis of emotional response. The columns on the right report the number of test participants for which a statistically significant difference was observed among the different conditions for $\alpha = 0.05$ and $\alpha = 0.01$. Rows in blue denote consecutive tests. Total sample size = 21.

The statistical examination of cardiac activity in terms of RR intervals and HRV metrics suggests some interesting findings. From a statistical perspective, the raw `RR Intervals` values sampled during task execution are notably and consistently lower than those recorded during rest conditions or baseline situations for all participants (see Table 3.3 and the results of the Mann-Whitney test in Table 3.4). This stress-related parameter decreases during task execution, suggesting a greater state of emotional arousal. Note that, as defined in the research question **Q2**, cardiac parameters, as well as individual perceptions, are highly variable between subjects (see Fig. 3.18). Furthermore, for all the samples analyzed, a statistically significant variation in the RR distribution is observed during different tasks (see Table 3.4), with the lowest values in Task 1, Task 4, and Task 5 (see Figure 3.18). The last two are the most complex tasks, while the first could represent a certain anxiety in approaching the cobot (Fig. 3.18). These results are in agreement with the research question **Q1**. In Table 3.4 the Dunn Post-Hoc test indicates lower values in the comparison between Task 1 vs Task 2, Task 2 vs Task 3, Task 3 vs Task 4, and Task 4 vs Task 5, all of which are consecutive tasks. This implies that the instances where a statistically significant difference is observed in the `RR Intervals` parameter are

Conditions	Agreement %
Task 1 - Task 2	38.10
Task 1 - Task 3	33.33
Task 1 - Task 4	38.10
Task 1 - Task 5	42.86
Task 2 - Task 3	57.14
Task 2 - Task 4	47.62
Task 2 - Task 5	57.14
Task 3 - Task 4	66.67
Task 3 - Task 5	52.38
Task 4 - Task 5	80.95
Average	**51.43**

Table 3.12: Agreement between emotional response and subjective responses from NASA-TLX, when comparing tasks. Values are in percentage over 21 subjects.

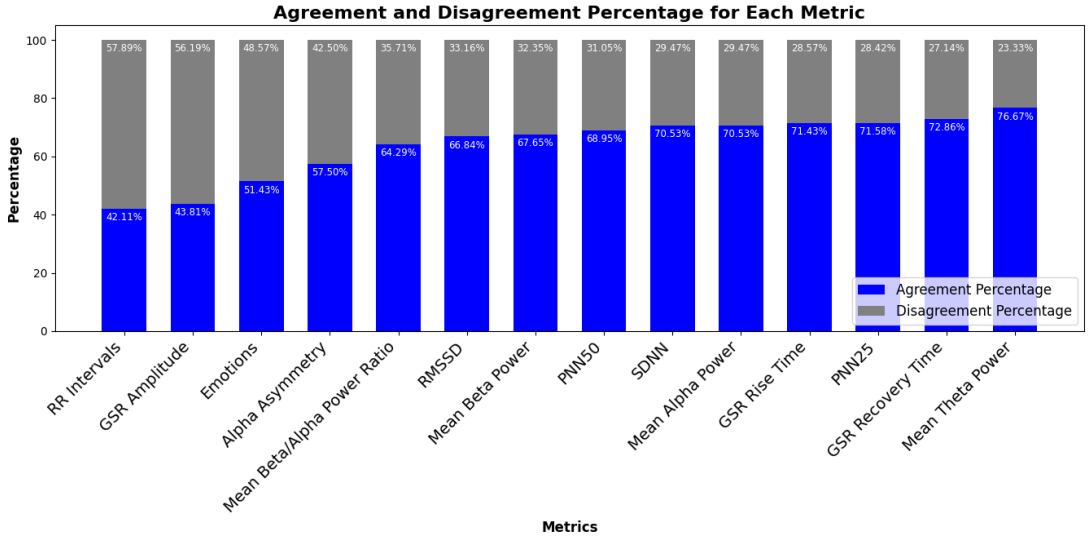


Figure 3.19: Percentage of agreement between psycho-physiological metrics and subjective questionnaire NASA-TLX.

fewer than those observed in non-consecutive tasks. The parameters appear to exhibit similarity, demonstrating temporal inertia, where preceding events seem to influence subsequent ones, irrespective of task complexity (Table 3.6).

In research, it is essential to assess the alignment between objective evaluations of stress-related parameters and subjective perceptions. The comparison of results from the Dunn test and NASA-TLX responses, specifically regarding `RR Intervals` distribution in the given experimental setup, reveals an agreement of 47.5%. The limited correlation observed between cardiac metrics and subjective perceptions appears to validate the constraints associated with relying solely on subjective assessments to describe stress [128]. Such assessments are heavily influenced by cultural and personal biases, as well as various objective and subjective variables.

HRV metrics, which are widely used in the literature to assess stress states, show lower differences than raw RR intervals between tasks and the baseline, as well as between different tasks (see Table 3.5). The HRV metrics decrease also during tasks compared with baseline, suggesting an higher stress level (see Table 3.3) . However, as illustrated in Fig. 3.19, `PNN25`, `PNN50`, `RMSSD`, and `SDNN` demonstrate strong precision, consistently aligning with subjective stress evaluations at a rate of 65% to 75%.

From the statistical analysis of GSR derived metrics in Table 3.7, it is clear that `Amplitude` is the parameter that shows more differences between the robot programming tasks and the rest condition, unlike `Rise Time` and `Recovery Time`, which appear to be less prone to variations compared when compared to a baseline scenario. Indeed, considering the Kruskal-Wallis outcomes with $\alpha = 0.01$, statistically significant differences were observed in 4 out of 21 samples for `Rise Time`, 5 out of 21 for `Recovery Time` and 21 out of 21 for `Amplitude`. Again with reference to Table 3.7, the outcomes of the Dunn's tests performed on samples with significant Kruskal-Wallis test results appear to validate the observations made in the ECG analysis: the metrics exhibit reduced variability throughout the execution of consecutive tasks, as indicated by the lower incidence of significant results in subsequent tests. Table 3.3 highlights that `Recovery Time`, a GSR metric directly related to stress, shows a positive increase during the evolution of tasks, with greater values on Task 4 and Task 5. `Amplitude`, shows greater values during task execution, `while Rise Time`, a metric inversely related to stress, has values that tend to be lower during activities compared to the baseline level.

Comparing the results obtained from the variation in parameters detected by the Dunn Post Hoc test and the NASA-TLX responses, we observe agreement percentages of 71.43% for `Rise Time`, 72.86% for `Recovery Time`, and 43.81% for `Amplitude`. As Fig. 3.19 shows, `Rise Time` and `Recovery Time` metrics stand out as excellent indicators for predicting and characterizing an individual's perception of stress.

The comprehensive statistical analysis conducted using metrics derived from brain signals proves to be some of the most accurate methods for evaluating perceived subjective stress within the scope of this cobot programming study, with an agreement percentage for `Theta Power Mean`, `Mean Alpha Power`, and `Mean Beta Power` respectively of 76.67%, 70.53%, and 67.65% (see Fig. 3.19). The trend and evolution of these metrics during the evolution of the tasks are reported in Table 3.3. The results of the Mann-Whitney test in Table 3.9 indicate that the derived EEG parameters show lower variability compared with the baseline, in contrast to the cardiac parameter. Furthermore, as discussed in research question **Q2**, there is variability in cerebral parameters across tasks, as evidenced by Kruskal-Wallis tests. This suggests distinct mental demands associated with tasks of varying complexity. The application of Dunn's Post Hoc test to brain parameters provides additional support for the hypothesis that there is a temporal dependence in the biological signal between tasks. This is evidenced by the lower significance values observed when comparing consecutive task.

Lastly, the capture of emotions in the dynamic setting of this project faced challenges due to occlusions and registration loss. Precision is likely higher in static scenarios, where the subject remains stationary in front of the camera, allowing for continuous recording. The level of concordance between recorded emotions and subjective assessments from NASA-TLX stands at 51.43% confirming research question **Q3**.

3.2 Adaptive interfaces and assistive technologies

While biosignals provide a powerful means of estimating users' cognitive and emotional states, a comprehensive approach to user understanding also requires robots to adapt their **physical interfaces** and **assistive functionalities** to heterogeneous user needs. In this perspective, adaptive interfaces are conceived not only to facilitate fluent collaboration in shared workspaces, but also to extend robotic assistance to individuals with specific impairments.

This section presents two representative examples: (i) **a low-cost 3D-printed electromagnetic gripper** specifically designed for collaborative tasks in Sec. 3.2.1, and (ii) **a wearable stereo vision-based obstacle detection system** developed for visually impaired users in Sec. 3.2.2. Both solutions contribute to enhancing social acceptance by reducing economic and technological barriers, while ensuring inclusivity, usability, and safety.

3.2.1 A low-cost 3D printed electromagnetic gripper for robotic arms

Robots and robotic manipulators were originally developed to supplement or replace humans in tasks involving dull, repetitive, dirty, dehumanizing, or dangerous work [129]. Generally, manipulators have different sorts of applications [130]. One of the common procedures performed by manipulators is **grasping**, i.e., manipulating and holding objects with the end-effector. In order to perform such operations, each manipulator needs a gripper mounted on its end-effector. Manipulation speed, object shape, weight and other characteristics are important factors in selecting the gripper type. However, some smart grippers are designed for general purposes and different object shapes. Secure grasping requires not only contacting the objects, but also preventing potential slip and damage while the objects are picked and placed. Robotic grippers face challenges in competing with human workers in terms of flexibility. With the evolution of automation in robot industries, grippers and grasping have become important topics in the robotic research community.

Several studies have focused on specific types of grippers, including **3D printed grippers** [131, 132, 133, 134]. A review on robotic grippers and a comparison of various types has been conducted in [135], revealing that, despite the extensive availability of grippers in the market, many share similar features, with limited stroke and force being the main limiting factors. The analysis suggests the need to develop new products with increased versatility and reduced weight to meet the emerging requirements of robotic systems. Although several **open-source** grippers have been proposed in the literature [136, 137], many struggle with gripping small or irregularly shaped objects or present challenges related to complexity and cost.

Conversely, some recent works have focused on **specialized grippers**. In [138], the authors introduced an innovative miniature magnetic gripper designed for minimally invasive surgery. This gripper, fabricated using 3D printing with photosensitive resin, combines magnetic and elastic materials to enable precise open-close motion under an

external magnetic field. Similarly, [139] presented a magnetic field-driven spherical gripper with adjustable stiffness, employing magnetorheological fluid and particle-jamming mechanisms for improved adaptability and efficiency. While these designs demonstrate significant innovation, they are not open source and involve intricate construction processes.

In [140], the authors described an electromagnetic gripper developed for industrial use in foundries, designed for gripping and handling large metal objects and semi-finished foundry products. It can generate extremely high attractive forces (150 N with a 0.1 mm air gap) and includes a driver that can control the supplied current based on the load. However, it is tailored for specific applications involving high currents (1800 mA), and its size and weight are not reasonably compatible with a typical robotic arm. In [141], the authors discussed a gripper using an electromagnet and a permanent magnet as an actuator to grasp objects through a water-filled membrane. It can grasp objects of any shape, compatible with the gripper's size. The design is easily scalable, with two different versions presented in the paper, and it does not require large currents or complex control electronics. However, despite these features, modulating the gripping force may be challenging, as it is proportional to the amount of water in the membrane.

Regarding cost-related aspects, several studies have proposed **low-cost grippers** for various applications. A 3D-printed multi-purpose gripper is presented in [137], which utilizes pneumatic actuators to control the device's fingers. This gripper is capable of grasping objects with varying shapes, sizes, stiffness, and surface areas, offering maximum flexibility compared to other designs. It is also an open-source project with a cost of 33$. Despite these advantages, the gripper features a complex mechanism composed of several parts that must be 3D printed with specific settings to be airtight and operational, and it is subject to wear and tear of its flexible components. In [142], the authors presented a low-cost, open-source three-finger robotic gripper designed for research and educational purposes. The 3D design model was developed and manufactured with a minimal number of 3D-printed components and a commercially available servo actuator, with a total cost of 400$. In [143], the authors proposed a cost-effective gripper intended for educational, research, and industrial applications requiring precise and flexible manipulation. This gripper is mounted on centimeter-scale, 3-degree-of-freedom (DOF) delta robots, made using commercially available linear actuators and 3D-printed soft materials, with a total cost of 300$. In [144], the authors proposed the DragonClaw gripper, a pneumatically actuated, 3D-printable three-fingered dexterous gripper with integrated magnetic tactile sensors. The gripper's 2-DOF thumb design enables a variety of precision and power grasps, providing excellent dexterity. Notably, this gripper includes seven magnetic-based sensors to detect finger deformation and react to external disturbances during tasks, with a cost of 500$.

Considering the existing literature, our work [2] proposes the design of a **light**, **open-source**, **easy to build** and **low-cost** gripper, able to operate without the need for high currents. As a result, the proposed approach involves the development of a **3D printed electromagnetic gripper**, to be used in pick and place scenarios involving small, irregularly shaped metal objects. This makes it a valuable tool with broad applicability

in any experimental setting, either industrial and academic.

To validate the proposed prototype, three tests were conducted: i) to determine the distance required for the electromagnetic gripper to successfully grasp a sample of ferromagnetic material with variable weight; ii) to assess the maximum weight the electromagnetic gripper can lift; iii) to assess the maximum weight the electromagnetic gripper can lift for pick and place operations, considering the sample in the most disadvantageous position. Subsequently, a thermal analysis of the gripper shell was performed to monitor the internal temperature and assess the reliability of the proposed design solution.

It is worthwhile noting that in the literature there is no agreed way to assess the performance of the above mention grippers. For instance, in [141], the authors presented the object diameter versus the water filling rate of the elastic membrane. In other works, such as [135], the authors compared stroke, force, weight, as well as a performance index referred to as the C-factor. Regarding the domain of electromagnetic grippers, in [140], the authors evaluated their gripper performance by showing various results, such as the force behavior as a function of the current. In our work [2], since a fixed current due to system design constraints is considered, the variation of the electromagnetic field over time as part of the thermal analysis is used.

Preliminaries

To design the electromagnet, its physical model is derived by considering the electromagnetic model of a linear coil and the principle of electromechanical energy conversion.

The magnetic circuit is considered as a linear magnetic structure, introducing some simplifications as in [145]:

- the magnetic reluctance of iron is assumed to be negligible. This assumption is justified by the significantly higher reluctance of the air gap. It simplifies the calculations and it is acceptable since the magnetic permeability of iron is much greater than that of the surrounding air;
- given their negligible influence, the effects of edges in magnetic structures were neglected, thus considering an idealized case.

These assumptions allow to derive the minimum number of turns required, given the geometrical characteristics of the electromagnet, the maximum current, the magnetic attraction force and the maximum distance at which to exert it.

The magnetic field of a linear coil is directly proportional to the supplied current. The electromagnet acts as a magnetomotive force generator, with series reluctances represented by the iron core and air gap, as shown in Fig. 3.20a. To this end, the reluctance of the magnetic path in the ferromagnetic core is denoted by $\mathcal{R}_{core}$ and the magnetic reluctance of the path in air is denoted by $\mathcal{R}_{air}$. It is possible to calculate the reluctance of the magnetic path in each material as follows:

$$\mathcal{R}_j = \frac{\ell_j}{\mu_o \mu_r \mathcal{A}},\ j \in \{core, air\} \tag{3.2}$$

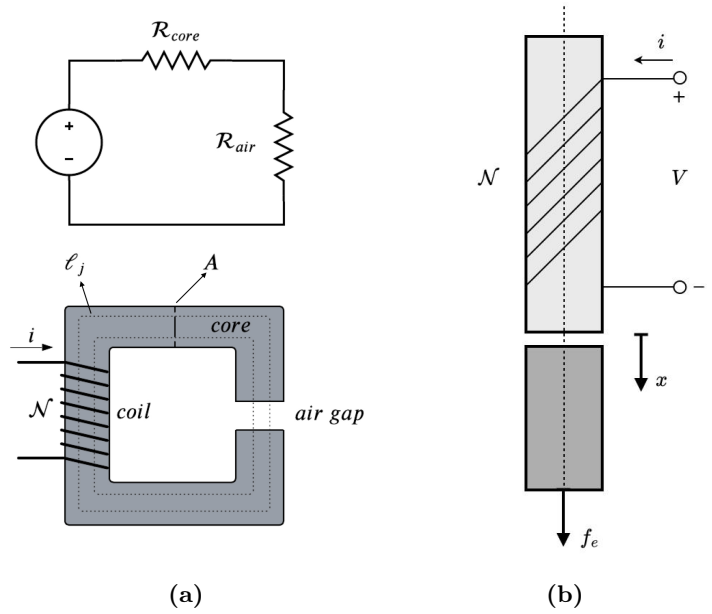


Figure 3.20: Scheme of magnetic reluctance: where (a) represents the magnetic and electric circuit analogy and (b) illustrates the electromagnet schematic.

where ℓ_j is the length of the average magnetic path through the j-th material, μ_o and μ_r are the magnetic permeability of air and core, respectively, and $\mathcal{A}$ is the cross section of the magnetic flux.

The work to move a mass corresponds to a change in the electromagnetic field's stored energy. The energy change equals the sum of the electrical circuit's work and the mechanical system's work [146]. In particular:

$$d\mathcal{W}_m = e\ i\ dt - f_e\ dx, \tag{3.3}$$

where $\mathcal{W}_m$ is the energy stored in the electromagnetic field, e is the induced electromotive force and f_e the external force acting on the object to be attracted, as shown in Fig. 3.20b.

The electromagnet design is based on the boundary condition where the object is at the maximum distance from the electromagnet pole, representing the highest air gap reluctance scenario, considering that reluctance is directly proportional to the average magnetic path length.

Design and prototyping

The proposed electromagnetic gripper, shown in Fig. 3.21 consists of three parts: the **electromagnetic core**, illustrated in Fig. 3.22a, a **conductor** wrapped around the electromagnet, shown in Fig. 3.22a, and a plastic **tool shell**, shown in Fig. 3.22b, that holds together the entire gripper. Specifically, an electromagnet characterized by a ferromagnetic core with a circular cross-section and a copper linear coil is considered.

The design of the tool shell has been conducted following some basic criteria, including:

1. the possibility of easily accessing the electronics, making any maintenance and

modification easier;

2. the use of easy to access fabrication technologies.

Additionally, the prototype is intended to be developed using 3D printing to reduce cost and weight, facilitating its creation. The availability of the 3D printed prototype allows easy adaptation to different types of robotic flanges.

To ensure the reproducibility of the gripper, the entire design is provided as open-source. 3D CAD files for constructing the shell, CAD drawings, assembly guide and video, and cost report for building the proposed electromagnetic gripper can be found here: https://doi.org/10.5281/zenodo.10822858.

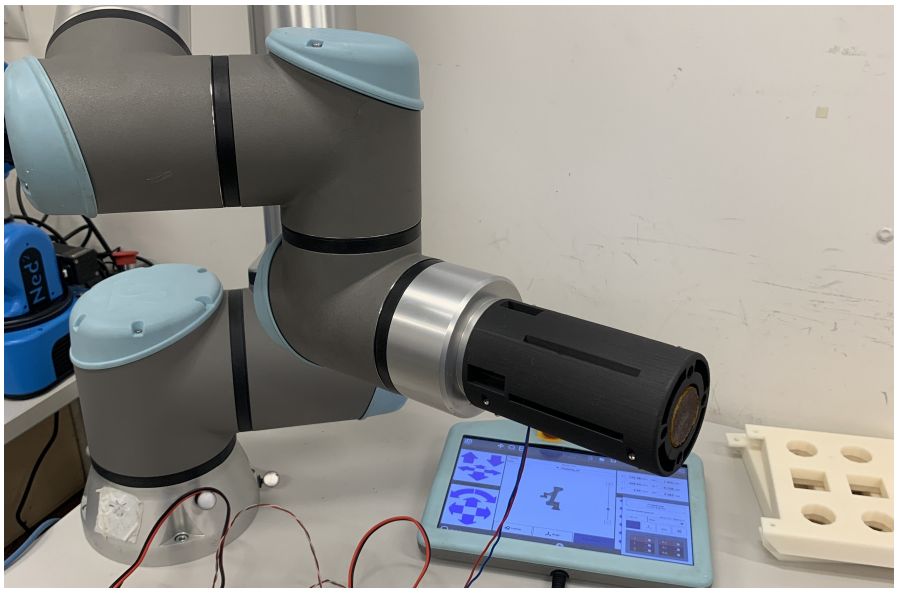


Figure 3.21: Proposed electromagnetic gripper.

(a) **(b)**

Figure 3.22: Electromagnetic gripper composed of: (a) a conductor wrapped around the electromagnetic core and (b) a plastic tool shell.

To design the electromagnet, considering an electromagnet characterized by a ferromagnetic core with a circular cross-section and a copper linear coil, it is necessary to choose the robot on which it will be mounted.

Any manipulator robot has different specifications that will result in different design constraints. In particular, the following design parameters need to be considered:

$$V = V^*, \tag{3.4a}$$

$$i \leq i^*, \tag{3.4b}$$

$$r < r^*, \tag{3.4c}$$

where i^* and V^* are current and voltage derived from the datasheet of the robot, and (3.4c) expresses the relation between the electromagnet radius, r, and the robot's flange radius, r^*.

Next, two design parameters need to be defined: the weight to be lifted, expressed with f^* and measured in N, and the target distance from the gripper at which the object must be successfully grasped, denoted as x^* and expressed in m. Thus, the following parameters are defined:

$$f = f^*, \tag{3.5a}$$

$$x = x^*, \tag{3.5b}$$

where f denotes the magnetic force to be generated; while x is the maximum distance at which such a force needs to be applied.

The area of the circular cross-section of the electromagnet, denoted as $\mathcal{A}$, is calculated as $\mathcal{A} = r^2\pi$, considering the electromagnet radius, r, as a design choice. The minimum number of turns, denoted by $\mathcal{N}$, required to generate such a force can be determined using Eq. (3.6), as described in [145]:

$$\mathcal{N} = \frac{1}{i^*} \cdot \sqrt{\frac{2(x^*)^2 \cdot f^*}{\mu_0 \cdot \mathcal{A}}}. \tag{3.6}$$

This determination is made with fixed parameters for the magnetic force f^* and the maximum distance x^* as outlined in Eq. (3.5). Additionally, the cross-section $\mathcal{A}$ and the magnet radius r are taken into account, along with i^* as defined in Eq. (3.4b).

Subsequently, it is necessary to calculate the size of the conductor to be wrapped around the electromagnet. Specifically, its minimum diameter, d_c, can be determined as reported in [147]:

$$d_c = \sqrt[3]{\frac{4\rho_{T_{max}} \cdot (i^*)^2 \cdot 10^3}{\pi^2 \cdot \lambda \cdot (T_{max} - T_a)}} \tag{3.7}$$

where $\rho_{T_{max}}$ is the copper resistivity at the temperature T_{max}; λ is air thermal conductivity coefficient; T_{max} and T_a are the maximum desired temperature and room temperature, set as design parameters.

In order to guarantee that the current i is lower than i^*, considering the voltage V^* in

Table 3.13: UR10e robot specifications

Parameter	Min	Typ	Max	Unit
Supply voltage in 24V mode	23.5	24	24.8	V
Supply voltage in 12V mode	11.5	12	24.8	V
Supply current (single pin)	-	1000	2000	mA
Flange diameter	0.035			m

Eq. (3.4), a limit for the target resistance R can be defined

$$R \geq \frac{V^*}{i^*}. \tag{3.8}$$

To check the value of the final current, the wire length ℓ needs to be calculated, considering the number $\mathcal{N}$ of turns obtained in Eq. (3.6) and the magnet radius r:

$$\ell = \mathcal{N} \cdot 2\pi r. \tag{3.9}$$

Hence, the wire resistance R_w becomes

$$R_w = \rho_s \cdot \ell, \tag{3.10}$$

where ρ_s is the copper resistivity per unit of length.

If the wire resistance is below the target resistance value R, defined in Eq. (3.8), during the manufacturing of the winding it will be possible to make more turns than the required minimum number of turns. By means of this procedure the following inequality can be satisfied:

$$i = \frac{V^*}{R_w} \leq i^*. \tag{3.11}$$

In the event of adjustments to various parameters, such as masses, distances, or the choice of a different robot or electromagnet radius, modifications to the design should be simple. For instance, altering masses or distances would require adjusting the design parameters f^* and x^* and repeating the sizing process. Conversely, opting for an electromagnet core with a larger radius would result in a reduction of the minimum number of turns $\mathcal{N}$. However, modifying the robot would be a bit more complex, as it would involve not only changes to design constraints such as voltage V^*, current i^*, and flange radius r^*, but also appropriate changes to the tool shell to align the holes in the tool shell with those in the robot flange.

In [2], we have chosen to design a electromagnetic gripper for the Universal Robots UR10e [148], as a **possible implementation**. This choice was motivated by the robot's reputation for precision and versatility. The design constraints introduced by the UR10e robot, in terms of electrical specifications and diameter of the robot's flange, are reported in Table 3.13. Considering these design constraints, the following design parameters are

obtained:

$$V^* = 24\ V, \tag{3.12a}$$

$$i^* = 1\ A, \tag{3.12b}$$

$$r^* = 0.035\ m, \tag{3.12c}$$

Next, the objective to lift approximately 200 g at a distance of 10±10% mm is considered. The following parameters can thus be defined:

$$f^* = 2\ N, \tag{3.13a}$$

$$x^* = 0.01\ m. \tag{3.13b}$$

The area of the circular cross-section of the electromagnet is calculated considering $r = 0.015\ m$ as a design choice, obtaining $\mathcal{A} = 7.068 \cdot 10^{-4}\ m^2$. The minimum number of turns required to generate such a force was determined using Eq. (3.6), considering $\mu_0 = 4\pi \cdot 10^{-7}\ H/m$. As a result, a number of turns $\mathcal{N}$ approximately equal to 671 is obtained, considering the requirements on payload express in Eq. (3.13) and the constraints for the UR10 robot in Eq. (3.12).

Subsequently, it is necessary to calculate the size of the conductor to be wrapped around the electromagnet, whose minimum diameter, d_c, can be determined as reported in Eq. (3.7) considering the following values: $\rho_{T_{max}} = 0.0197\ \frac{\Omega \cdot mm^2}{m}$ [149]; $\lambda = 15\ \frac{W}{^\circ C \cdot m^2}$ [146]; $T_{max} = 55^\circ C$ and $T_a = 20^\circ C$, set as design parameters.

In particular, $d_c = 0.248\ mm$ is obtained. Considering a safety factor to increase the minimum diameter by 20%, $d_c = 0.298\ mm$ is obtained. Therefore, the commercial diameter $d_c = 0.361\ mm$, corresponding to the commercial size 27 AWG, was chosen.

In order to guarantee that the current i is lower than $i^* = 1\ A$, considering the voltage V^* in Eq. (3.12), a limit for the target resistance R can be defined

$$R = \frac{V^*}{i^*} \geq 24\ \Omega. \tag{3.14}$$

To check the value of the final current, the wire length ℓ needs to be calculated, considering the number $\mathcal{N}$ equal to 671 and the magnet radius r:

$$\ell = \mathcal{N} \cdot 2\pi r = 63.24\ m. \tag{3.15}$$

Hence, the wire resistance R_w becomes

$$R_w = \rho_s \cdot \ell = 10.68\ \Omega, \tag{3.16}$$

where ρ_s is equal to 0.1689 Ω/m [150].

As a result, the wire resistance is significantly below the target resistance value R in Eq. (3.14). During the manufacturing of the winding, a number of turns $\mathcal{N}$ equal to 1650 was chosen, obtaining $\ell = 155.43$ m and $R_w = 26.25\ \Omega$,

in order to guarantee the use of a current slightly less than 1 A. Indeed:

$$i = \frac{V}{R_w} = 0.91 \ A. \tag{3.17}$$

Regarding the **circuit diagram**, the gripper was connected to a power supply by setting a voltage of 24 V and a current of 1 A, while a relay was chosen for the control function.

Specifically, according to the design parameters described in Eq. (3.12), it is necessary to choose a relay, driven at V^* DC, supporting an operating current greater than the nominal i^*. This device is connected to a digital output pin of the robot controller, through which it can be turned on and off.

The electrical configuration of the gripper thus results as in Fig. 3.23.

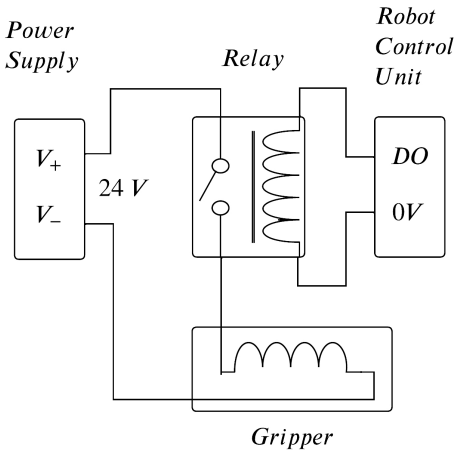


Figure 3.23: Circuit diagram. The relay's energising coil is connected to a digital output of the robot controller, indicated by DO pinout. The secondary circuit is connected to the external power supply and to the the gripper coil.

Regarding the **tool shell**, a shell of cylindrical shape, divided into two semi-cylindrical pieces, was chosen. This design, shown in Fig. 3.24, allows easy access to the electronics contained within. To connect the two half-shells, through-holes were provided on the side surface of the cylinder, allowing the shell to be closed with four M2x15 screws. The connection with the robot is made by a flange drilled opportunely to allow mounting of the gripper using four screws (M6 in the case of the UR10e robot).

The end opposite to the tool holder flange is left open, providing an edge where the core can be positioned. This approach minimizes material interference between the electromagnet and the object to be attracted, ensuring a firm grip while also facilitating airflow through designated openings as shown in Fig. 3.24.

A through-hole was then made near the base of the shell to allow for the housing of the connecting cable between the electronics of the gripper and the robot. Finally, several cuts were made in radial and axial directions on the cylindrical shell of the gripper in order to enhance removal of the heat generated by the passage of current in the coils of the winding.

The design described so far is also functional for the 3D printing process, since the two

half-shells can be arranged with the interface positioned on the printing bed. Notches were also made for inserting screws when mounting the gripper to facilitate installation.

Having the 3D CAD files of the gripper available, adapting the gripper to different types of robotic flanges is feasible, as it only requires matching the tool shell holes with the robot flange holes.

As the material for the prototype, onyx was chosen, a composite consisting of nylon with the addition of micro-carbon fibers, produced by Markforged. This material exhibits excellent mechanical properties, including high resistance and durability against chemicals and heat. Specifically, the material is characterized by a yield strength of $40MPa$, a strain at break of 25%, and a flexural strength of $71MPa$. A critical parameter for heat resistance is the bending softening temperature, which is 145 °C in this case.

The prototype was printed using a Markforged X7 printer with a printing time of 28 hours.

Figure 3.24: Render view of the gripper shell, with key features: (1) Assembling through-holes, (2) M2x15 assembling screws, (3) Mounting M6 screws and notches, (4) Shell flange, (5) Through-hole for cables, and (6) Radial and axial cuts for thermal dissipation.

Test and characterization

To assess the performance of the proposed design, three types of tests have been carried out through offline programming using the UR robot teach pendant: i) maximum grasping distance; ii) maximum grasping mass; iii) pick and place test. Moreover, to validate the capability of the electromagnetic gripper to handle objects with varying geometries, an additional test was performed using nine different objects. Notably, during this test, the objects were handed to the robot without any predefined orientation. A video demonstrating this test is available in the Zenodo repository.

Specifically, the teach pendant allows the robot to execute joint movements for free space movements using the *MoveJ* command, perform linear movements with *MoveL*, activate

and deactivate the gripper using the *Set* command and pause between operations with the *Wait* command, or execute *jogging* movements. Additionally, to perform these tests, it was necessary to use several tools such as an external power supply to provide 24 V voltage and 1 A current to the gripper, a relay for the control function, and a caliper to accurately measure the distance between the gripper and the sample to be lifted, used in the different tests.

- **Maximum grasping distance:** the purpose of this experiment was to determine the maximum distance at which the gripper could attract samples of different masses. This parameter is crucial as it indicates the distance within which the electromagnet can securely grasp objects of varying weights. Operationally: i) the tool was positioned above the mass to be attracted using small vertical *jogging* movements, with the electromagnet activated through the *Set* command; ii) Once the piece was grasped by the tool, the distance between the table and the piece was measured using a caliper.

  Considering the design parameters expressed in Eq. (3.13), this type of test allowed verifying the correct design of the electromagnetic gripper. In fact, it is possible to lift a weight of 200 g at a distance of $x = 9.18$ mm, as shown in the Table 3.14.
- **Maximum grasping mass:** the aim of this experiment was to determine the maximum mass that the robot can grasp. During the test, the robotic arm was programmed to: i) position the gripper to pick up the sample with a *MoveL* command; ii) activate the electromagnet to grasp it with the *Set* command; iii) lift it 100 mm above the table and hold the position for 5 seconds using the *MoveL* and the *Wait* commands; iv) lower and release it through the *MoveL* and *Set* operations. Specifically, for the execution of this test, a initial weight of 1000 g was used, and the sample weight was increased by 100 g increments until reaching a maximum weight limit of 1500 g at a distance of $x = 1.80$ mm, as shown in the Table 3.14. Each *MoveL* movement of the robot was performed at a speed of 250 mm/s.

  A video demonstrating the maximum weight test with a 1500 g specimen and the test failed with 1600 g is attached in the Zenodo repository.
- **Pick and place test:** in this test, the robotic arm was programmed to: i) position the gripper to pick up the sample with a *MoveL* command; ii) activate the electromagnet to grasp it with a *Set* command; ii) lift the object with a *MoveL* command; iv) move to the vertical of the predefined deposition point using a set of *MoveJ* and *Wait* commands; v) lower itself to an appropriate height to deactivate the gripper and deposit the picked object through *MoveL* and *Set* commands. In performing the set of *MoveJ* and *Wait* commands, the robotic arm was programmed to manipulate the sample so that it was parallel to the ground plane for 5 seconds, to verify that it was able to hold the mass in the most disadvantageous condition, as shown in Fig. 3.26. As reported in Table 3.14 the pick and place task was successfully accomplished for specimens of mass up to 300 g, while it failed for the 325 g specimen. During this test, a *MoveJ* speed of 60 $^\circ/s$ and a *MoveL* speed of 250 mm/s were considered.

  A video demonstrating the pick and place task with a 300 g specimen is attached in the Zenodo repository.

Figure 3.25: Test for the maximum grasping mass with sample of 1500 g.

Table 3.14: Experimental test results

Mass	Maximum attraction distance	Pick and place
200 g	9.18 mm	Yes
225 g	7.52 mm	Yes
250 g	5.96 mm	Yes
275 g	4.77 mm	Yes
300 g	4.25 mm	Yes
325 g	3.39 mm	No
1000 g	2.90 mm	No
1100 g	2.80 mm	No
1200 g	2.40 mm	No
1300 g	2.15 mm	No
1400 g	2.00 mm	No
1500 g	1.80 mm	No

Thermal analysis

While the device is in operation, the heat generated by conduction losses is transferred by conduction across the iron core, and by convection through the air gap between the coil and shell. A significant portion of the heat flowing through the air gap then reaches the shell, from where it is discharged towards the external environment, as shown in Fig. 3.27a. Such a process leads to an increase of the shell temperature, which could compromise mechanical properties and make it more challenging to handle the gripper with bare hands.

In order to assess the external surface temperature of the shell, two thermal analyses were carried out by means of infrared thermography: one in the case of the gripper under unloaded conditions (UC), and the other under loaded conditions (LC). A FLIR A-615

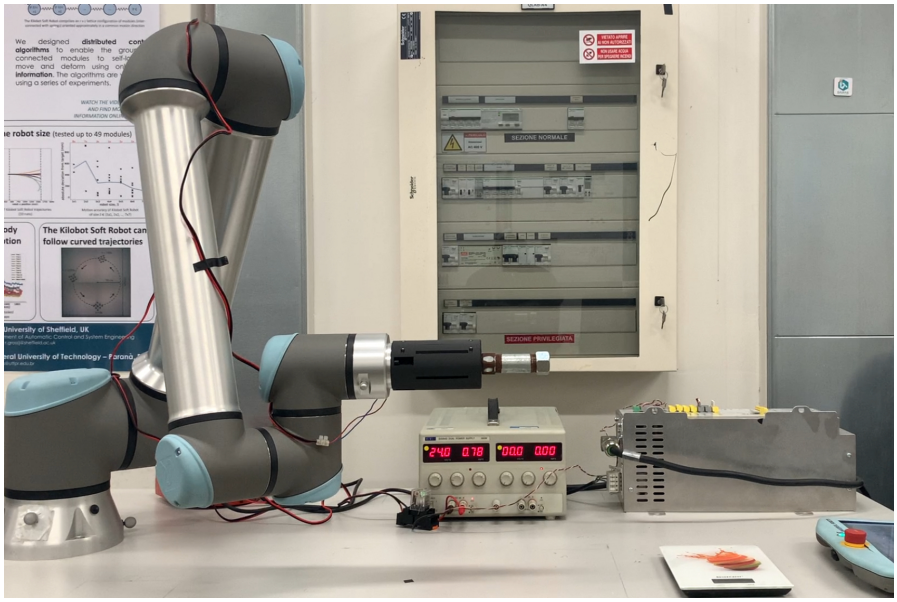


Figure 3.26: Pick and place test with sample weighing 300 g. The robotic arm is programmed to manipulate the sample parallel to the ground plane, ensuring a challenging condition for the gripper to hold the mass.

thermal camera was used, characterised by a thermal sensitivity $< 0.5°C$ at $30°C$, an accuracy of $\pm 2\%$, a resolution of 640×480 pixel, a spectral range of $7.5 - 14\ \mu m$ and a frame rate of 50 Hz.

Specifically, in the UC, measurements were conducted by holding the gripper at rest while powered on; the camera was placed in front of the gripper, at a distance of 37 cm, with an anti-reflective plate in the background to minimise noises from the nearby environment. In the LC, the measurement setup was the same as in the LC, except for the presence of a specimen, the measurement distance and the ambient temperature. The measurement distance was set to 70 cm to include the specimen held by the electro-magnetic gripper. The ambient temperature was $21.5°C$. The same specimen shown in Fig. 3.26, with a weight of 300 g, was used.

A schematic representation of the experimental setup is shown in Fig. 3.28.

The parameters for the thermographic analysis are listed in Table 3.15. The surface emissivity of the shell of the gripper is set to that of the material used in 3D printing (onyx). The solenoid emissivity should be the emissivity of the wire's external insulation. Due to the lack of information from the supplier, the insulation emissivity needs to be assumed. Generally, insulation materials have an emissivity between 0.8 and 0.95; a conservative estimate of 0.8 was assumed, since lower emissivity values imply higher surface temperatures. Furthermore, the analysis mainly focuses on the surface temperature of the shell of the gripper, and the emissivity of the solenoid is used only to have an indication of the ΔT with respect to the start of the experiment. The recording spanned 2 hours. In both LC and UC measurements, a snapshot was taken every 5 minutes, except for the first 30 minutes. During this period one snapshot per second (1fps) was taken, in order to better resolve the initial transient. Fig. 3.27b provides a schematic representation of the different areas where temperature is measured. The shell mean temperature is evaluated as the mean temperature of Box1. To properly assess only the shell temperature, it is

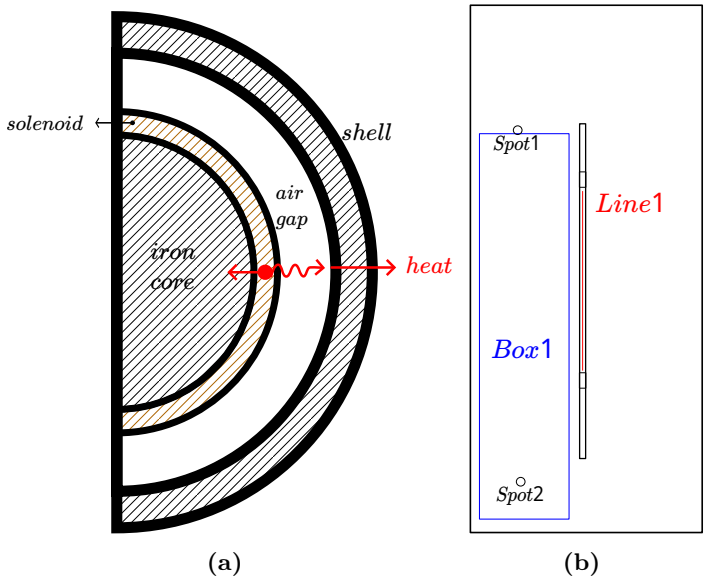


Figure 3.27: The (a) simplified scheme of the radial heat transfer inside the gripper and the (b) schematic representation of the Box1, Line1 and spots defined for data recording.

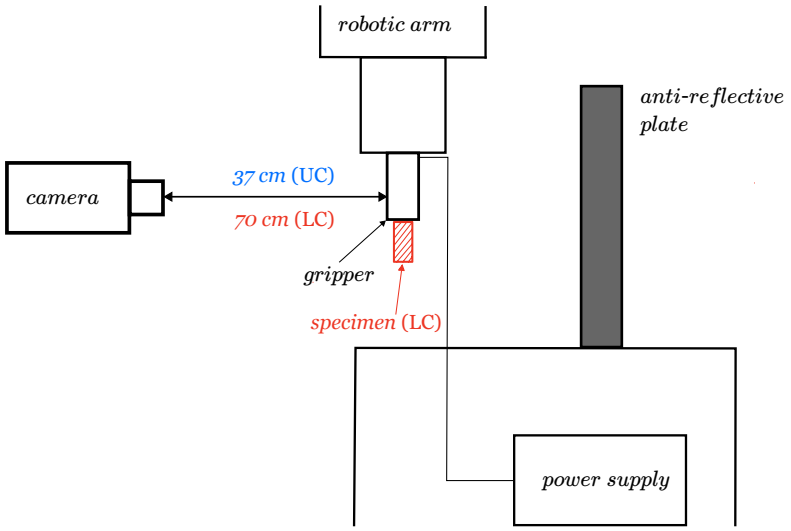


Figure 3.28: Experimental setup for the thermal analysis.

important that Box1 exclusively encompasses the shell material, without any interference from other materials (such as screws) or with the longitudinal cut. Therefore, the Box1 was defined as a sampling box spanning a periodic angular sector of the gripper.

In order to obtain a qualitative measurement of the solenoid mean temperature, another sampling line (Line1) is strategically placed within the central part of the slit. Finally, two further probes were defined to record the temperature difference between the upper and lower sides of the shell.

Fig. 3.29 presents a series of snapshots captured at different times in the UC measurements. In the first snapshot the temperature distribution at the initial time (i.e., when the power supply was switched on) is displayed. The second snapshot ($t = 5$ min) shows how heat mostly spreads from the cuts; this is because the cuts oppose a smaller thermal resistance with respect to the case. At $t = 20$ min, the temperature distribution is more uniform: the mean temperature in Box1 has increased by 10°C, while, according to the

Table 3.15: Thermographic analysis parameters

Parameter	UC Value	LC Value	Unit
Ambient temperature	23	21.5	$°C$
Solenoid emissivity	0.8	0.8	-
Shell emissivity	0.95	0.95	-
Supply voltage	24	24	V
Initial supply current	0.75	0.75	A
Measurement distance	0.37	0.7	m

average temperature in Line1, the mean winding temperature has increased by 48°C, reaching a temperature of $72°C$. Going ahead, it is evident how the temperature distribution always becomes more uniform in the central and upper side of the case, while the lower side always remains colder. This distribution may be attributed to natural convection effects transporting warmer air (both internal and external to the shell) upwards, and to the augmented heat transfer surface in the lower part of the gripper. After one hour, the mean temperature in Box1 reaches $41°C$, whereas the mean temperature on Line1 reaches $106°C$; this substantial difference is due to the convective heat transfer in the internal air gap and cuts, complemented by conduction heat transfer across the iron core. After two hours the temperature distribution looks qualitatively similar to that of the preceding images, the mean temperatures of Box1 reaches 45°C, i.e., an increase of 4°C in one hour. Such values indicate that it would be still possible to safely handle the gripper with bare hands. On the other hand, the estimated winding temperature reaches 114°C, for an increase of 8°C in one hour. At the end of the experiment, the gripper, and in particular the shell, does not show signs of failure; hence the material is suitable for long periods of continuous operation.

The evolution of temperature with time during the UC measurements, is shown in Fig. 3.31a, in which the brown curve is the winding mean temperature (mean temperature of Line1), the black curve is the shell mean temperature (mean temperature of Box1) and the green and blue curves represent respectively the temperature of Spot1 and Spot2. To make the graph more readable, the solid lines are obtained by performing a regression with the recorded data (represented by the markers). According to the plot, the thermal transient lasts about 60 minutes for the shell (black line), while for the solenoid it lasts nearly 2 hours (red line). As Fig. 3.29 shows, the lower side of the shell always exhibits a smaller temperature; indeed, in the plot, Spot2 (blue line) always remains below the mean temperature. After 120 minutes, Spot2 is 6°C colder, while Spot1 (green line), which is located in the upper side, is 6°C hotter with respect to the shell mean temperature. As already mentioned, the difference between the two points can be ascribed to the effects of natural convection, which determines a higher heat transfer rate in the lower part of the shell.

Interestingly, during the test, it was observed that, as the temperature increased, the current decreased compared to the initial supply current values shown in Tab. 3.15. This

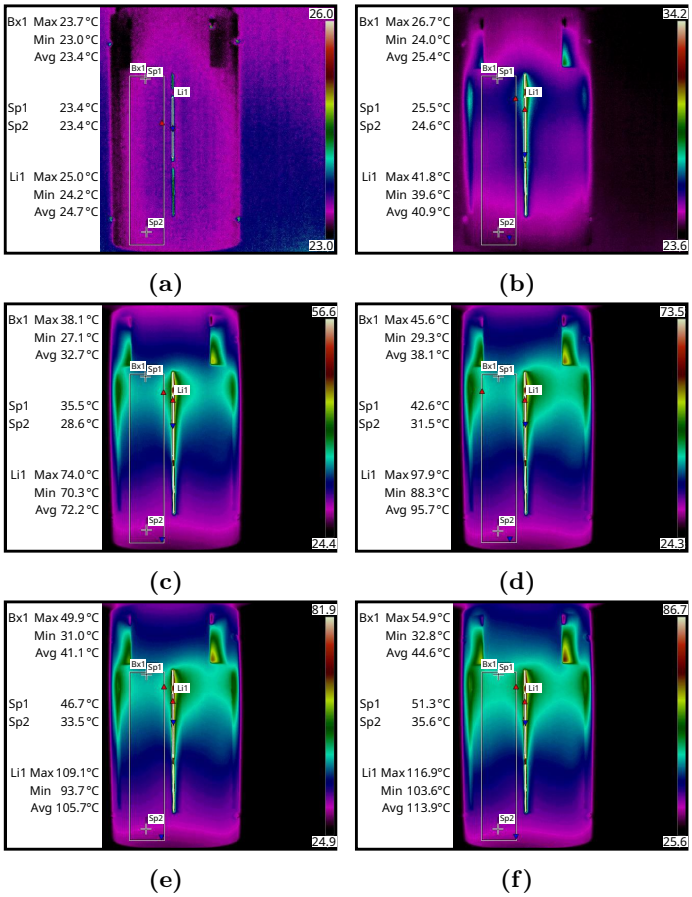


Figure 3.29: Thermal camera snapshots in the UC after: (a) 0 *min*; (b) 5 *min*; (c) 20 *min*; (d) 40 *min*; (e) 60 *min*; (f) 120 *min*.

behaviour can be attributed to the change in electrical resistivity with temperature

$$\rho = \rho_0[1 + \alpha(T - T_0)] \tag{3.18}$$

where ρ is the instantaneous electrical resistivity, ρ_0 is the reference electrical resistivity, α is the temperature coefficient, T represents the wire temperature and T_0 the temperature at which ρ_0 is evaluated.

For a linear coil, the magnetic field generated is directly proportional to the supplied current [151]:

$$\mathcal{B} = \frac{\mu \mathcal{N} i}{\ell_c}, \tag{3.19}$$

where $\mathcal{B}$ is the magnetic flux density, measured in Wb/m^2; μ is the magnetic permeability, expressed in H/m; and ℓ_c is the length of the iron core covered by the coil, expressed in m.

Considering that the only variable in Eq. (3.19) is the current intensity, the ratio between the instantaneous magnetic field and initial magnetic field can be derived as

$$\frac{B}{B_0} = \frac{i}{i_0} \tag{3.20}$$

where i_0 and B_0 are respectively the current and magnetic field intensity at time 0.

Using the following formulation for the wire resistance

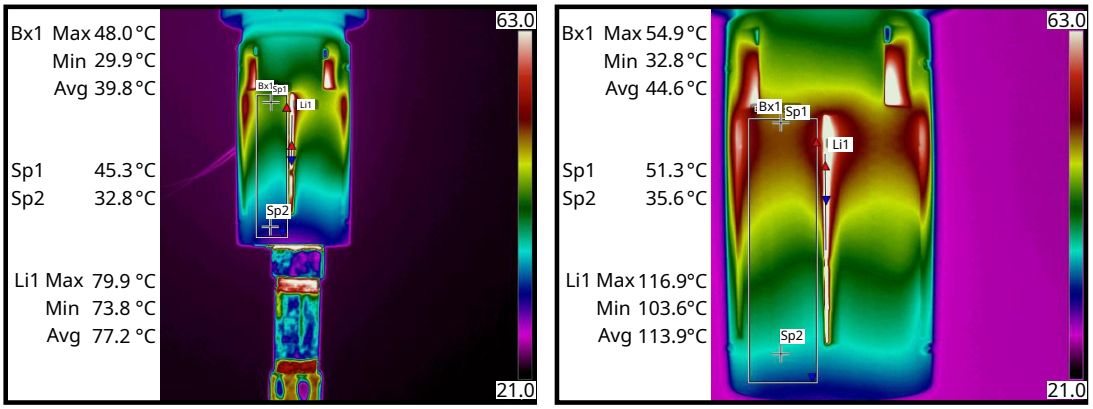
$$R = \rho \frac{\ell}{\mathcal{A}} \tag{3.21}$$

with $\mathcal{A}$ equal to the wire cross-section, then Eq. (3.20) can be rewritten as

$$\frac{B}{B_0} = \frac{V\mathcal{A}}{\rho\ell} \frac{1}{i_0}. \tag{3.22}$$

Figure 3.31b depicts the trend with time of the magnetic field intensity variation for $i_0 = 0.75\ A$, $V = 24\ V$, $\rho_0 = 1.71 \times 10^{-8}\ \Omega m$ and $T_0 = 24\ °C$ (the mean temperature of the wire at the beginning of the experiment). As for the temperature, the variation of the magnetic field is faster in the first minutes and then slows down as time progresses, until it stabilizes around its final value, that is around 72%. This implies that, in two hours, the magnetic field intensity decreases by about 28% with respect to the initial state; although this evaluation relies on a simplified set of equations, it can be an indication about the performance variation of the gripper with the increase of temperature.

As already explained above, the first measurements were performed in unloaded conditions, since any picked object would constitute an additional path for heat transfer, thus reducing the thermal load of the gripper. To confirm this statement, a second thermal analysis was carried out under LC.

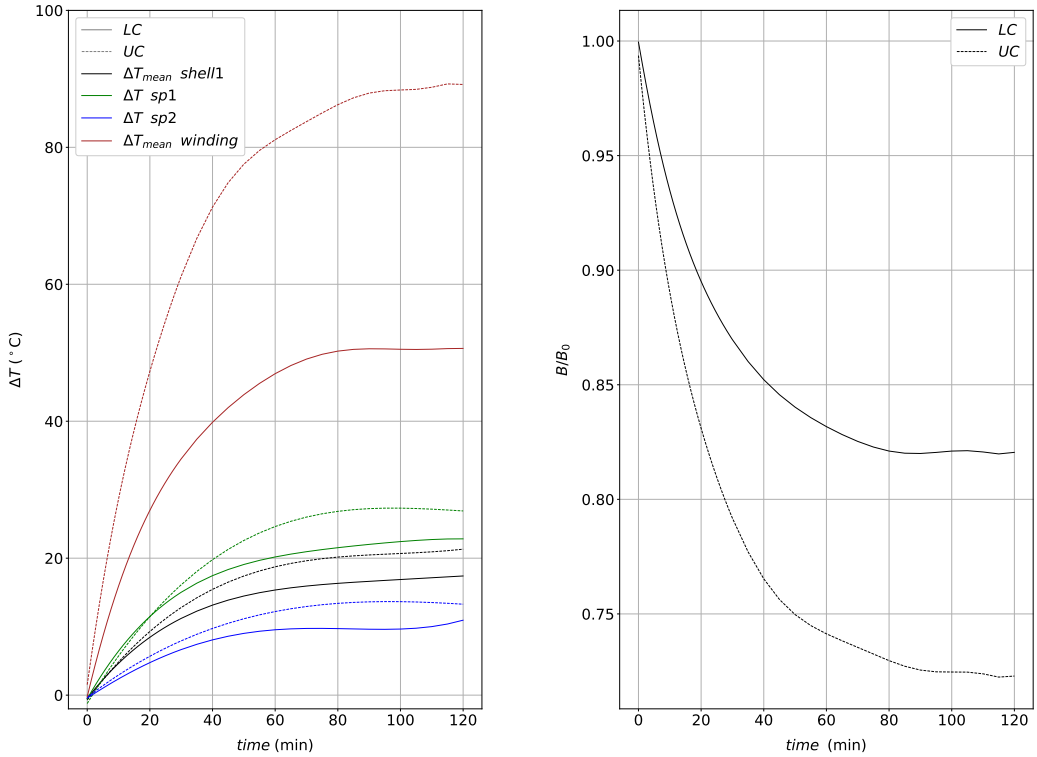


(a) **(b)**

Figure 3.30: Results of the thermal analysis comparison: (a) snapshot in LC case at $t = 120\ min$, (b) snapshot in the UC case at $t = 120\ min$.

Figure 3.30 presents two snapshots captured at the same time, $t = 120\ min$, in the LC and UC cases, respectively. The temperature distribution under steady-state conditions shows some differences, particularly in the lower part of the electromagnetic gripper, as shown in Fig. 3.30a, where a region with lower temperature is observed. This difference is most likely due to conductive heat transfer with the load itself. Moreover, in the LC, the temperature distribution near the upper edge of the ventilation slots appears more uniform. This could be attributed to a reduced heat exchange between the solenoid and the shell, in favor of conductive heat transfer between the solenoid and the load. It is worth mentioning that the temperature values of the load in Fig. 3.30a are not to be considered as true, since the image was processed with the emissivity of the shell, which is different from the emissivity of the load material (steel); in addition, since the load

cannot be approximated as a diffuse emitter, its recorded temperature values can change with the measurement direction.



(a) **(b)**

Figure 3.31: Results of the thermal analysis comparison with and without load (LC and UC, respectively): (a) temperature variation for Spot1, Spot2, Box1 and Line1, (b) magnetic field intensity variation.

The evolution of temperature over time is shown in Fig. 3.31a, where the comparison is based on the temperature variation rather than absolute temperature values, since the ambient temperatures in the two experiments differ as shown in Tab. 3.15. In particular, Fig. 3.31a shows the temperature variation from the initial time for Spot1, Spot2, Box1 (mean shell temperature) and Line1 (mean winding temperature) in both cases, highlighting that the load contributes to dissipating the heat generated in the solenoid, resulting in a lower temperature increase when the load is present (LC measurements). Although the absolute temperature values obtained from Line1 may not be highly reliable as the measurement is performed through one axial cut and the emissivity was assumed with a conservative value, the fact that the trend aligns with the other measured values further confirms that the load contributes to dissipating the generated heat. Figure 3.31b presents the comparison of the magnetic field intensity. As expected, since the winding temperature is lower in the LC, the resistivity variation is lower as deduced from (3.18). Thus, after 120 *min*, the magnetic field intensity drop is 18%, instead of 28% resulting in the UC. This finding highlights that the more the grasped object dissipates the heat generated in the winding, the less the magnetic field intensity decreases. Therefore, possible improvements can be made by decreasing the contact resistance with

the object and enhancing the thermal conductivity of the load and / or the heat transfer surface to the ambient. Finally, the relative temperature difference between the UC and LC measurements at the end of the experiment can be expressed as

$$\Delta\bar{T} = \frac{\Delta T_{UC} - \Delta T_{LC}}{\Delta T_{UC}}, \tag{3.23}$$

where ΔT_{UC} and ΔT_{LC} represent the difference between the first and last temperature values measured in the UC and LC measurements, respectively; $\Delta\bar{T}$ is 0.11 regarding the Box1, 0.12 for the Spot1 and Spot2, and 0.30 regarding the Line1. From this analysis, it can be concluded that the thermally most critical condition is when the gripper is not holding any object. Indeed, for all defined regions $\Delta\bar{T} > 0$ as the load dissipates part of the heat generated in the solenoid, resulting in reduced thermal stress on the shell.

3.2.2 A wearable stereo vision-based obstacle detection system for visually impaired individuals

In everyday life, human eyes are the primary means of acquiring information from the surrounding environment. According to the World Health Organization, 285 million people worldwide have visual impairments, including 39 million who are blind [152]. Blind and visually impaired individuals, therefore, face daily challenges in navigating autonomously in complex environments, such as urban or domestic settings. Traditionally, blind individuals rely on white canes or guide dogs for navigation. White canes help detect obstacles, while guide dogs assist the user in reaching their destination. However, both solutions provide limited environmental details, making navigation in complex environments a significant challenge. Indeed, tasks such as crossing streets, moving through crowds, and recognizing objects in unfamiliar places can be risky due to the inability to perceive depth, motion, and spatial relationships. These challenges affect personal safety, independence, and overall quality of life, emphasizing the need for more advanced assistive technologies.

Artificial intelligence (AI) and computer vision (CV) offer promising solutions by enabling real-time interpretation of the environment. By combining image processing algorithms with AI models, CV can accurately detect, recognize, and classify objects. These technologies have applications across various fields, such as healthcare [153], automotive [154] and industrial automation [155], enhancing accessibility and increasing efficiency.

In robotics, CV plays a crucial role in enabling robots to perceive and understand their surroundings, leading to advancements in tasks like autonomous navigation, object recognition, and robotic assistance [156]. By integrating CV, robots become more intuitive, responsive, and capable of executing complex tasks with enhanced precision. For example, in [12], we proposed an architecture for user identification and social navigation in mobile robotics. Their approach leverages CV algorithms for gesture and facial recognition to identify users and facilitate navigation in social contexts. Similarly, in [157], a YOLO-based model was introduced to enhance object detection for service robots, improving their efficiency in navigating indoor environments. Beyond navigation, CV is also useful in improving robotic manipulation and motion planning. In [158], the authors

proposed a layered architecture that incorporates visual foundation models to enhance task execution. Moreover, CV technologies can be integrated in HRC, as demonstrated in our work [3], where the *AFFDEX 2.0* module was employed to monitor real-time facial expressions during such interactions.

In the context of accessibility, CV offers a significant opportunity for the development of assistive systems for blind and visually impaired individuals, allowing the detection and classification of obstacles in their environment. Current solutions implement diverse approaches to enhance spatial perception. AIris [159] integrates high-resolution cameras and natural language processing (NLP) to provide real-time audio scene descriptions, achieving great performance in object recognition and text reading. SOMAVIP [160] employs depth cameras and IoT frameworks to deliver semantic feedback in urban environments. RealSense-based systems [161], using stereo depth sensors such as R200 and RS410, enable efficient obstacle detection and path generation. YOLO-based systems [162] focus on light RGB object detection, ensuring high speed at the cost of depth accuracy, while simpler configurations with ultrasonic sensors [163] prioritize cost efficiency.

Despite these innovations, assistive technologies still face persistent challenges. Many solutions rely on centralized processing or cloud infrastructures, introducing latency and limiting usability in offline environments or areas with poor connectivity. Others provide minimal or non-semantic feedback, reducing the user's ability to gain a detailed awareness of their surroundings. Another challenge is the integration of machine learning models into embedded systems due to the inherent limitations of these devices. Therefore, achieving a balance between advanced functionality, portability, cost efficiency, and adaptability to dynamic and unstructured scenarios remains an open challenge.

In our work [4], we propose a wearable computer vision system, schematized in Fig. 3.32, capable of detecting and classifying objects and obstacles while estimating and verbally indicating their distance in real time, thus enhancing users' autonomous mobility and safety. Specifically, the system employs a lightweight object detection model, *EfficientDet-Lite0*, in conjunction with the Intel RealSense D415 RGB-D sensor, which utilizes stereo vision technology. The system processes RGB and depth data in parallel on a Raspberry Pi 4, delivering real-time semantic feedback without relying on cloud infrastructure. In addition, portability and affordability enhance the system's usability for real-world deployment. By addressing challenges related to depth accuracy, real-time feedback, and user-friendly experience, our work contributes to the development of scalable, reliable, and accessible assistive technologies for individuals with visual impairments.

Hardware and software components

This section introduces the components used in the development of the proposed system. The hardware consists of the Intel RealSense D415 depth camera and the Raspberry Pi 4 single-board computer, which together provide real-time depth perception and computational capabilities.

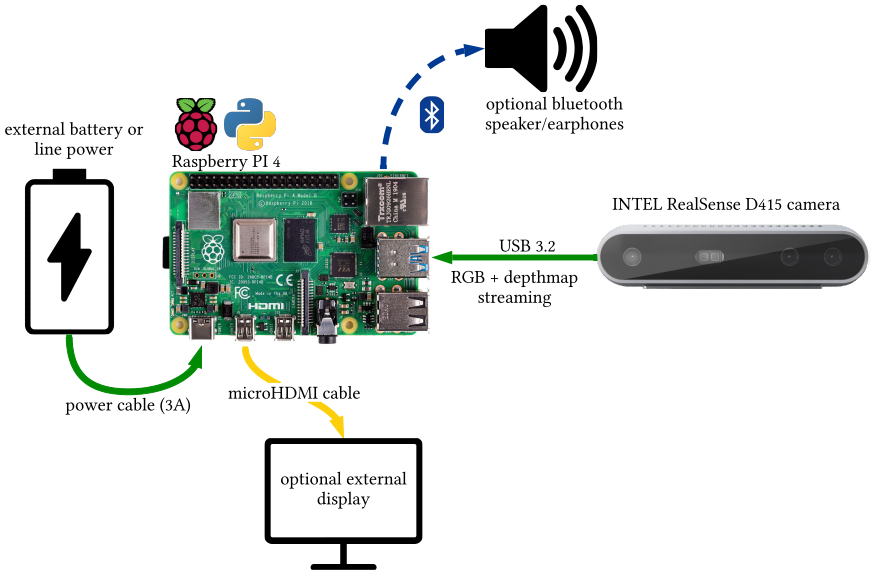


Figure 3.32: Hardware architecture layout of the proposed system.

The Intel RealSense D415 camera, shown in Fig. 3.32, is an advanced depth-sensing device designed for precise 3D perception. It employs an active stereo vision system that integrates two infrared sensors with an infrared projector, enabling the generation of high-resolution depth maps in real time. Its RGB sensor further allows the acquisition of color images synchronized with depth data, making it highly suitable for object recognition and 3D modeling applications. Complementing the camera, the Raspberry Pi 4, also depicted in Fig. 3.32, is a powerful single-board computer designed for embedded applications, featuring a compact size and a flexible design. It is well-suited for projects that require a balance between functionality and low power consumption.

The software methodologies implemented in the proposed system are the stereo vision technique for depth estimation and the *EfficientDet-Lite0* algorithm for object detection and classification. Particularly, **stereo vision** is the process of estimating the depth of scene points based on their positional variation between two image sensors placed at a known relative distance. In our work, stereo vision is employed to acquire depth data, which is then used for obstacle detection. This process relies on identifying correspondences between elements (objects or recognizable details) present in the input images. When an object appears in both images, its position differs slightly between the two; this difference, known as disparity, is used to compute the object's distance from the observation point. By iterating this operation for all identifiable points in both images, a disparity map is generated [164].

Specifically, the stereo vision process consists of four main stages:

- *Offline calibration*: to obtain the intrinsic parameters of the two cameras (such as focal length, image center, and lens distortion parameters) and the extrinsic parameters (i.e., the relative position of the two cameras, referred to as the baseline, and their relative rotation);
- *Rectification*: which removes lens distortions and reduces the correspondence search from a two-dimensional to a one-dimensional operation, thus accelerating disparity computation;

- *Stereo matching*: which aims at identifying homologous points in the stereo image pair;
- *Triangulation*: given the disparity map, baseline, and focal length, triangulation determines the 3D position of the corresponding points in space.

On the other hand, **object detection** is a fundamental task that involves identifying instances of objects within an image and classifying them into predefined categories. Numerous high-speed algorithms have been developed for object detection. Currently, object detection models can be divided into two main categories [165]: two-stage and one-stage detectors. The first consists of models that break down the task into multiple stages, following a *coarse-to-fine* approach. In contrast, the latter comprises models designed to complete the detection process in a single step using a single neural network.

Particularly, in our work [4], to ensure the real-time performance of the proposed system on a single-board computer, we adopt the *EfficientDet-Lite0* model [166]. *EfficientDet-Lite0* is a lightweight and optimized one-stage detector model designed for devices with limited resources and belongs to the EfficientDet family, which uses the EfficientNet convolutional neural network (CNN) as its backbone [167].

Proposed architecture

As introduced above, our work proposes a CV system equipped with an object detection model and stereo vision technology for depth estimation. The intended application scenario is assistive technology for visually impaired or blind individuals, providing them with audio feedback on obstacles in their surroundings.

The objective is to integrate these functionalities into a wearable platform and to leverage the combined information to detect and classify objects and obstacles in real time. Specifically, we consider the term *object* to an identified element in the scene that could potentially become an obstacle. In contrast, the term *obstacle* refers to any surface within the captured scene that, even if not associated with a specific object, poses a risk because its proximity falls below a critical distance threshold relative to the user.

The system was developed within a Python 3.7 virtual environment, utilizing key libraries such as NumPy [168], OpenCV [169], multiprocessing, TensorFlow [170], and PyRealSense2 [171], the latter being necessary for camera integration.

As schematized in Fig. 3.33, the system consists of two main processing pipelines that form its core:

- **Obstacle detection process**: by processing depth maps collected by the RGBD stereo camera, the system identifies areas in the scene that are at a critical distance, defined as below a certain threshold;
- **Object classification process**: utilizing the *EfficientDet-Lite0* model, the system detects objects in RGB frames and associates their position with that of the obstacles to provide a combined visualization and communication.

The detailed execution flow of the proposed architecture is illustrated in Fig. 3.34.

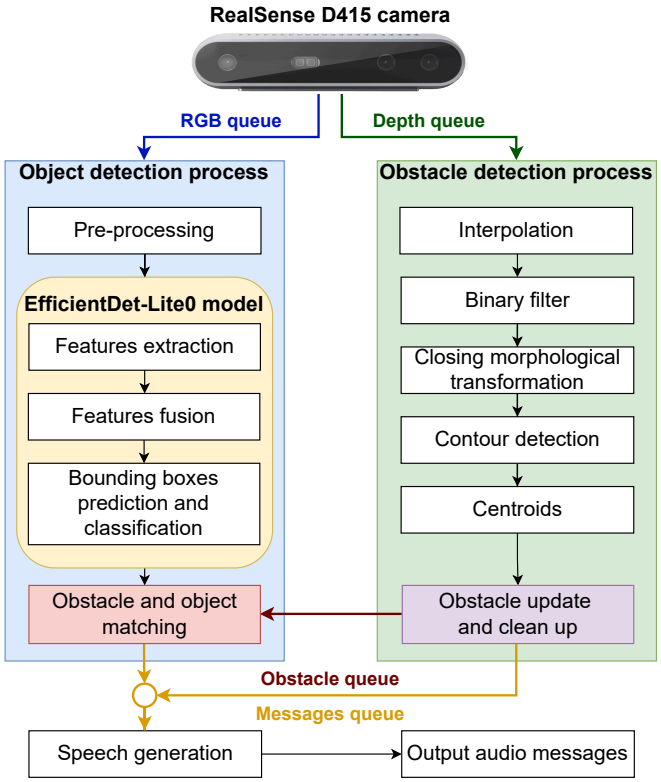


Figure 3.33: General overview of the image processing stages involved in the system, which consists of two main processing pipelines: the obstacle detection process (green), and the object detection process (blue).

Specifically, the process begins with a *main* execution thread, which launches three parallel processes to handle:

- data acquisition from the camera, defined as *stream reader process*;
- obstacle detection, defined as *obstacle detection process*;
- object detection, defined as *object detection and pairing process*.

Regarding the **Stream Reader process**, due to the computational demands of object and obstacle detection, which represent the main bottlenecks in the pipeline, a multiprocessing-based approach was adopted. This design choice, preferred over multithreading, enables parallel execution of independent processes, optimizing performance, particularly in embedded environments. The *stream reader process*, shown in yellow in Fig. 3.34, corresponds to Algorithm 1, which is highlighted in red within the yellow block. This process is responsible for acquiring depth and RGB frames from the RealSense camera. Specifically, it initializes a pipeline to capture frames, processes the incoming data, and inserts the frames into two separate queues: the *Depth queue* for depth data and the *RGB queue* for RGB data. The process runs continuously until a *stopEvent* is triggered, as detailed in Algorithm 1.

The second parallel process, referred to as the **obstacle detection process** and shown in green in Fig. 3.34, is responsible for real-time depth data processing and serves as a consumer of the *Depth queue*. It retrieves frames from the queue and applies *preprocessing and contour extraction*, followed by *obstacle contours sorting and filtering*. Through these operations, the process identifies a set of detected obstacles computing

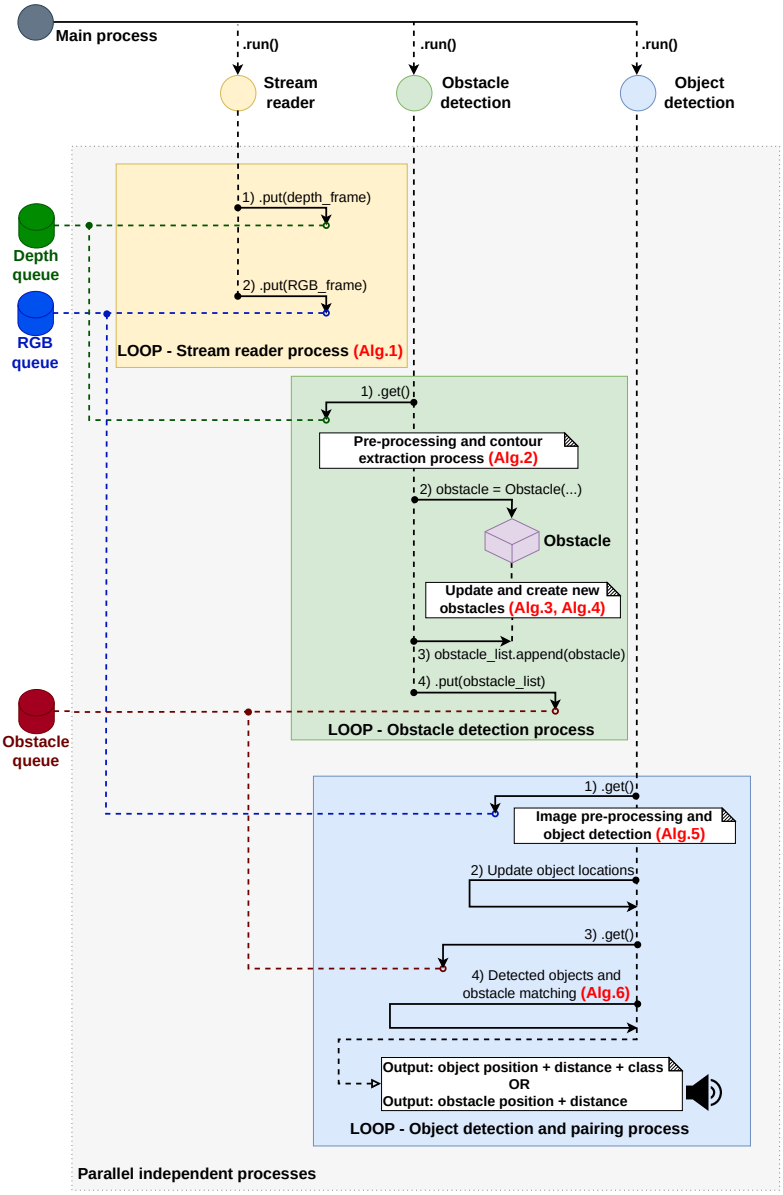


Figure 3.34: Architecture execution flow diagram consisting of three parallel processes for handling data acquisition from the camera (yellow), obstacle detection (green), and object detection (blue).

their corresponding distances.

Regarding *pre-processing and contour extraction*, given an RGB frame (Fig. 3.35a), a corresponding depth map can be derived, as illustrated in Fig. 3.35b. Considering a depth map as a two-dimensional matrix of numerical values representing the distance of projected points from the observation point, the *pre-processing and contour extraction process* aims to identify, sort, and filter the contours of obstacles in an image. As outlined in Algorithm 2, highlighted in red in Fig. 3.34 within the green block, the process begins with the generation of a binary mask by applying a depth threshold between 400 mm and 2500 mm (Fig. 3.35c). The lower bound corresponds to the camera's minimum detection range, while the upper bound is empirically determined to define the region where obstacles are considered potentially dangerous. Next, morphological operations are applied to the mask to reduce noise and enhance image quality. Finally, contour detection is performed, and only the n most significant obstacles (i.e., those with the

Algorithm 1: Stream reader process

```
Require: Depth queue, RGB queue, stopEvent
Ensure: Frames inserted into the queues
begin
    pipeline ← New pipeline;
    config ← New configuration;
    pipelineWrapper ← PipelineWrapper(pipeline);
    Start(pipeline, config);
    while ¬ IsSet(stopEvent) do
        frames ← WaitForFrames(pipeline);
        depthFrame, colorFrame ← GetFrames(frames);
        if depthFrame = NULL or colorFrame = NULL then
            continue;
        depthImage ← AsArray(GetData(depthFrame));
        colorImage ← AsArray(GetData(colorFrame));
        if IsFull(Depth queue) then
            Get(Depth queue);
        Put(Depth queue, depthImage);
        if IsFull(RGB queue) then
            Get(RGB queue);
        Put(RGB queue, colorImage);
    Stop(pipeline);
```

Algorithm 2: Pre-processing and contour extraction process

```
Require: Image frame, n
Ensure: n sorted and filtered contours by ContourArea
begin
    maskObstacle ← InRange(frame, 400, 2000);
    closedMask ← MorphologyEx(maskObstacle, MORPHCLOSE, (7, 7));
    contours ← FindContours(closedMask, ...);
    sortedContours ← Sort(contours, key ← ContourArea, reverse);
    filteredContours ← sortedContours[:n];
    return filteredContours;
```

largest contours) are retained. An example result is shown in Fig. 3.35d, where the extracted contour is highlighted in green.

Regarding *centroid computation*, after detecting obstacles, their centroids are computed using image moments to estimate their geometric centers. These centroids are then stored for tracking purposes. In particular, to ensure temporal consistency across frames, newly detected centroids are compared with previously identified obstacles following the procedure outlined in Algorithm 3, highlighted in red in Fig. 3.34 within the green block.

The comparison is based on the following criteria:

- *Proximity matching*: obstacles detected within a threshold of 50 pixels across consecutive frames are considered the same object, and their attributes (e.g., position and distance) are updated accordingly;
- *Merging overlapping detections*: if a newly detected centroid falls within an extended radius proportional to an existing obstacle's area, it is merged with that obstacle. Otherwise, it is registered as a new obstacle;
- *Filtering transient detections*: only obstacles corresponding to newly detected centroids are retained, while transient or noise-induced detections are discarded.

Specifically, the distance of each obstacle is calculated using the *compute_ distance* method of the Obstacle class, as described in Algorithm 4. This method identifies the closest point

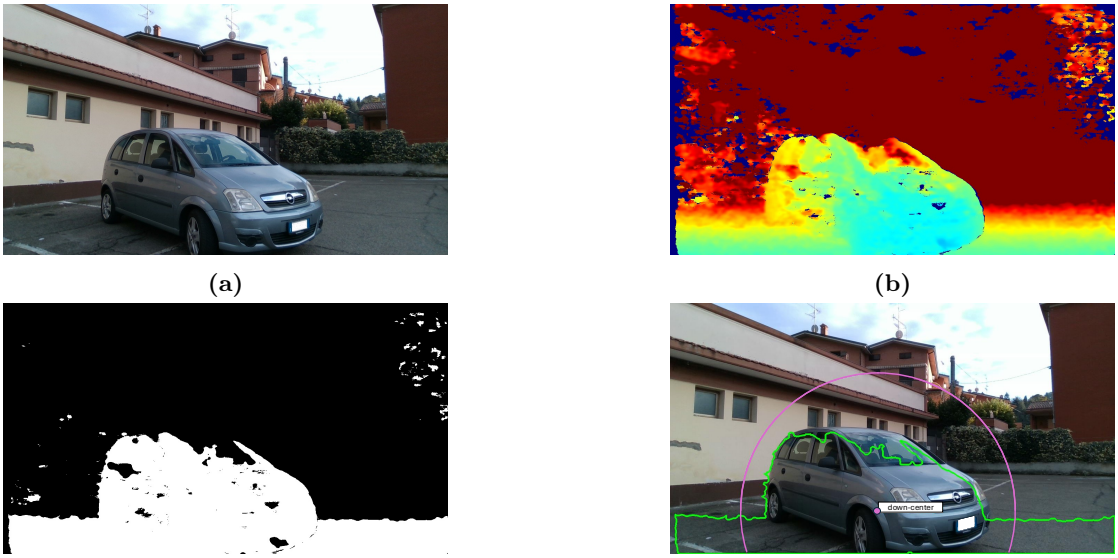


Figure 3.35: Output of the different stages of the *obstacle detection process* in an outdoor environment: (a) an RGB frame capturing a car; (b) the corresponding depth map; (c) the generated binary mask; and (d) the output of Algorithm 2 and Algorithm 3, where the detected contour is shown in green and the centroid in purple.

within the obstacle and stores the minimum distance as an internal attribute when the obstacle is created or updated.

Furthermore, based on the centroid's position, each obstacle is assigned a label indicating its approximate location within the frame. The frame is divided into six regions, defined by the combination of two vertical sections (*top* and *bottom*) and three horizontal sections (*left*, *center*, and *right*). An example is shown in Fig. 3.35d, where the computed centroid is highlighted in purple. This *centroid computation* approach enhances robustness by ensuring that only persistent obstacles are tracked over time, reducing the impact of spurious detections.

The last parallel process, referred to as the **object detection and pairing process** and shown in blue in Fig. 3.34, analyzes RGB frames to identify and classify relevant objects within a given context, such as cars, people, tables, or chairs. Subsequently, the process will be responsible for providing an audible verbal feedback to indicate the presence of the most dangerous obstacle for the user. In the proposed system, this process consists of four key steps:

1. **Model configuration:** the object detection model is set up with the following key parameters: i) *Number of threads*: optimized experimentally and set to 4 for efficient execution; ii) *Maximum objects per frame*: limited to 3 to ensure focused detection; iii) *Confidence threshold*: a minimum confidence score required for an object to be included in the results;

2. **Frame pre-processing and object detection:** before the RGB frame is processed by the model, the pre-processing steps described in Algorithm 5 are performed.

   The pre-processing includes horizontal flipping to achieve a selfie-view, color space

Algorithm 3: Update and create new obstacles

```
Require: List of previous obstacles: previousObstacles, detected center: detectedCenter,
         frame, contours extracted form the frame
Ensure: Updated list of obstacles: obstacles
begin
    found ← False;
    for o ∈ previousObstacles do
        if o.isNear(detectedCenter, 50) then
            o.distance ← o.calculateDistance(frame, contour);
            o.center ← detectedCenter;
            found ← True;
            break;
        if o.isNear(detectedCenter, o.radius) then
            o.area ← o.area + ContourArea(contour);
            o.updateRadius();
            o.distance ← o.calculateDistance(frame, contour);
            found ← True;
            break;
    if Not found then
        obstacles.Append(new Obstacle(ContourArea(contour), detectedCenter, frame,
          contour, useMinDistance=True));
```

Algorithm 4: Compute Distance from Depth Frame

```
Require: Depth frame depthFrame, contour contour
Ensure: Minimum valid distance or None
begin
    mask ← ZerosLike(depthFrame, uint8);
    DrawContours(mask, [contour], -1, 255, FILLED);
    distances ← depthFrame[mask == 255];
    validDistances ← distances[distances > 0];
    if length(validDistances) > 0 then
        return GetMin(validDistances);
    else
        return None;
```

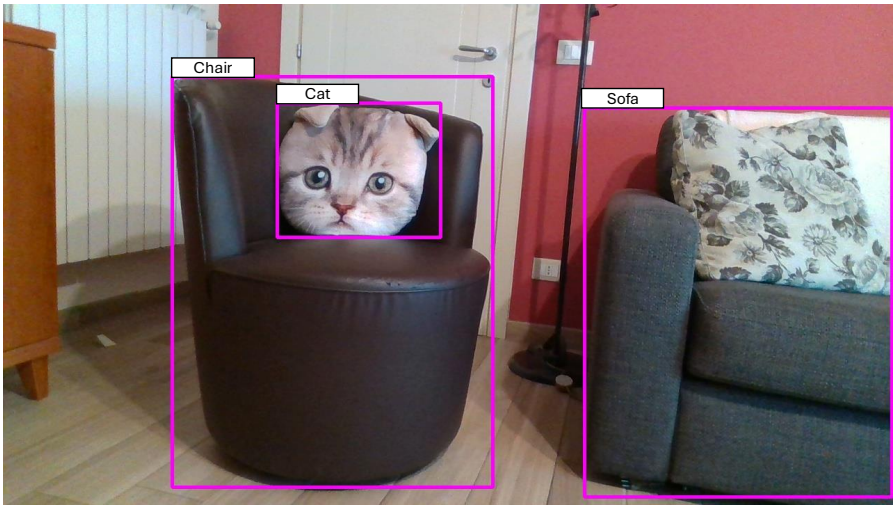


Figure 3.36: Output of the *image pre-processing and object detection algorithm* in indoor environment.

conversion from BGR to RGB, and resizing to dimensions suitable for the object detection model. Finally, the preprocessed image is converted into a tensor representation and passed to the object detector to obtain detection results. The output is a structured data object containing a list of detected classes within the frame, along with the dimensions and positions of the corresponding bounding boxes, as shown in Fig. 3.36.

Algorithm 5: Image pre-processing and object detection

```
Require: color image from RGB Queue: colorImage, object detector: detector
Ensure: Detection result: detectionResult
1 begin
2     image ← Flip(colorImage);
3     rgbImage ← ConvertColor(image, BGR2RGB);
4     resizedRgbImage ← Resize(rgbImage, (640, 480));
5     inputTensor ← CreateFromArray(resizedRgbImage);
6     detectionResult ← Detect(detector, inputTensor);
7     return detectionResult
```

3. **Distance estimation and pairing:** the final step in the *object detection and pairing process* associates detected objects with obstacles and determines their respective distances from the user. This step takes as input the list of obstacles received through the *Obstacle queue*, as illustrated in Fig. 3.34.

   Next, the association between detected objects and obstacles is determined using the procedure described in Algorithm 6. Specifically, an object is considered a match with an obstacle if their centers are within 100 pixels of each other. If a match is found, the object's bounding box, class, and distance are announced verbally and optionally displayed on the screen, as shown in Fig. 3.37. If no match is found, only the presence of the obstacle and its estimated distance are communicated.

   This real-time pairing of detected objects with obstacle distances enhances spatial awareness, ensuring accurate and meaningful feedback for the user.

4. **Verbal output feedback:** lastly, once the most dangerous obstacle for the user has been detected, the proposed system provides verbal acoustic feedback to signal

Figure 3.37: Output of the *detected objects and obstacle matching* algorithm in outdoor environment.

Algorithm 6: Detected objects and obstacle matching

```
Require: List of obstacles: obstaclesList, object center: center, frame
Ensure: Frame with distance information
1 begin
2     matched ← False;
3     for obstacle ∈ obstaclesList do
4         distance ← GetDistance(obstacle);
5         if distance ≠ None and obstacle.isNear(center, 100)
            DisplayBoundingBoxAndInfo(frame, obstacle);
6           AnnounceObjectInfo(obstacle);
7           matched ← True;
8           break;
9     if not matched  AnnounceObstaclePresence(distance);
```

its presence, as illustrated by the gray block within the blue block in Fig. 3.34. Specifically, the system announces the obstacle's position, corresponding to the centroid's label, along with its distance from the user. If the detected obstacle is a classified object, the system also specifies its category. Examples of output phrases are: "*Car, bottom-left, 2.56 meters*" or "*Obstacle, bottom-right, 1.2 meters*".

To avoid excessive audio messages that could be distracting in dynamic environments, obstacle notifications are issued at fixed, configurable intervals t rather than immediately after each detection. This approach filters out transient obstacles while ensuring timely alerts, as t is chosen to remain within safety limits given by the system's maximum obstacle detection time of 0.4498 s (Tab. 3.16).

Experimental validation

To experimentally validate the proposed architecture (Fig. 3.32), we conducted a performance test to evaluate the system's obstacle and object detection times, demonstrating that its performance meets the required specifications. This was followed by a pilot user study with subjects without visual impairments to assess the overall feasibility of the proposed system.

In the **performance test**, the system was tested for five consecutive minutes in an indoor

Statistics	Obstacle Detection [s]	Object Detection [s]
Mean	0.1363	2.4574
Variance	0.0024	0.6317
Standard deviation	0.0488	0.7948
Minimum	0.0495	0.5549
Maximum	0.4498	3.6320

Table 3.16: Performance of the system for obstacle and object detection.

environment, during which it detected various obstacles and objects. During execution, the processing times for each obstacle and object detection cycle were recorded in a log file and expressed in seconds. In total, data were collected from 844 cycles in which at least one obstacle was detected and 52 cycles in which at least one object was detected, yielding the results presented in Tab. 3.16.

From Table 3.16, the ratio of the average object detection time to the average obstacle detection time indicates that obstacle detection is, on average, approximately 18 times faster than object detection, achieving an execution rate of approximately 7 frames per second.

Considering the performance test results, the system meets the required specifications if the arbitrary audio feedback interval t is at least equal to the maximum obstacle detection time. In this case, the system ensures that obstacles are detected and reported in a timely manner. For instance, if $t = 10\ s$, the user receives auditory feedback every 10 s, and the values in Table 3.16 confirm that, even in the worst case, obstacles can be reported within a maximum delay of approximately 0.45 s. On the other hand, object detection, which includes classification, can take up to 3.63 s, but this remains within an acceptable range depending on the application's requirements.

Regarding the **pilot user study**, the study was conducted in a domestic environment filled with various objects. During testing, the camera was attached to the shirt worn by the user. The study showed that the system is suitable for real-world scenarios. Moreover, participants emphasized the need for improvements in the auditory cues provided by the system and suggested integrating haptic feedback via a smartphone to enhance user experience. Additionally, the user study revealed the need for stably attaching the camera to the shirt to prevent movement-related issues, which could cause errors and delays in obstacle detection. Ensuring a secure attachment would improve both the accuracy and reliability of the system, making it more effective in dynamic environments.

3.3 Outlook

The analyses presented in this chapter highlight how user understanding constitutes a fundamental building block of socially acceptable interaction. Among the different studies discussed, only the EEG-based experiment introduced in [14] must be regarded as a preliminary work. This investigation focused exclusively on brain activity and involved a limited number of participants, serving mainly as a methodological validation. Nevertheless, it demonstrated the feasibility of using EEG spectral features, particularly the *Theta*, *Alpha*, and *Beta* bands, as indicators of mental workload and engagement during collaborative assembly. Although preliminary, this study validated the experimental protocol, the signal-processing pipeline, and the indices that were later adopted in the extended multimodal framework.

Building upon this foundation, subsequent works refined the approach by integrating multiple physiological signals to obtain a more comprehensive and objective estimation of the operator's psychophysiological condition. In particular, the multimodal studies combined EEG, ECG, and GSR with subjective workload assessments (NASA-TLX) during first-time collaborative robot programming. This approach enabled a quantitative correlation between the physiological parameters and the perceived stress and effort, showing strong agreement levels between objective and subjective indicators.

At the same time, a complementary research effort focused on developing and validating a low-cost 3D-printed electromagnetic gripper. Conceived as an adaptive and accessible robotic interface, it was designed to facilitate safe and intuitive collaboration in shared workspaces. Its open-source nature and ease of fabrication make it suitable for educational, research, and assistive contexts, lowering both economic and technological barriers to human–robot interaction. Beyond its functional role in performing reliable pick-and-place operations with ferromagnetic parts, the gripper exemplifies how adaptive physical interfaces can enhance social acceptance by promoting inclusivity, usability, and safety.

In parallel, the work presented in [4] explored assistive perception for visually impaired users, proposing a wearable stereo vision–based system for obstacle detection and classification. The device interprets spatial information from the surrounding environment and delivers audio feedback to notify the user of obstacle type and position in real time. By translating visual data into intuitive auditory cues, the system enhances situational awareness and autonomy in everyday navigation tasks. This research exemplifies how adaptive sensing and multimodal feedback can extend robotic perception to assistive domains, fostering inclusivity and accessibility in human–technology interaction.

Together, these contributions demonstrate that user understanding should be treated as an integrated capability of the robot, rather than a separate sensing module. By combining physiological, perceptual, and behavioral information, the robot can adapt its actions in real time, modulating speed, trajectory, or autonomy according to the user's state. This ability forms the conceptual bridge to the next chapter, where user understanding is extended to social navigation, enabling mobile robots to regulate interpersonal distance and motion dynamics in a socially acceptable way.

Chapter 4

Social Navigation

In recent years, several service robots have been developed for various practical applications, defining a novel approach to navigation called **social navigation** [172]. A socially navigating robot acts and interacts with humans or other robots, achieving its navigation goals while modifying its behavior so the experience of agents around the robot is not degraded or is even enhanced.

Notably, reception and robotic guidance have emerged as particularly popular services, where robots are gradually replacing human personnel in assisting customers. In these scenarios, a mobile robot autonomously navigates the environment to guide a person to a specific location, facing the challenge of planning and completing a collision-free path through obstacles in the environment [173]. In particular, social navigation denotes the capability of a mobile robot to move in human-shared environments by not only avoiding physical collisions, but also complying with social norms and expectations. These conditions pose different challenges.

First, **safety** is a primary concern, since the robot must plan and execute a **collision-free path**, and **reduce its speed** in proximity of humans [10, 11, 173]. Another aspect to consider is the robot's behavior in **deadlock situations** [174]. In intersections or narrow doorways, the robot can either take the initiative to go first or proactively propose a solution by, for example, making a gesture, emitting a signal to allow others to pass first. Additionally, the robot may need to follow a person [175] or be followed, to guide them to a target destination [12]. In this case, the robot should keep the user within its **field of view**, reduce its speed if the human is slowing down or stop if the person is no longer following. Furthermore, there might be the need for the robot to **recognize the person** it is interacting with (or following or guiding) and adjust its behavior to accommodate individual preferences. Additionally, in social environments where robots and people coexist and robots move according to human needs, **space-time constraints** might be relevant to ensure that the robot can efficiently and safely manage its activities, especially in dynamic and crowded environments shared with people. Temporal constraints can take various forms, such as time limits to complete a specific task, time intervals to accomplish a sequence of actions, or priorities assigned to different activities based on their importance. These restrictions may be imposed by environmental conditions, user

preferences, or social requirements.

In contrast to classical navigation, social navigation explicitly targets human-centered requirements such as comfort, legibility, predictability, and respect of interpersonal spaces. Within this dissertation, social navigation is considered one of the key pillars of social acceptance: for a robotic system to be perceived as acceptable, navigation behavior must remain safe in the control-theoretic sense and socially appropriate in the HRI sense. This dual perspective calls for methods that can guarantee safety while accommodating temporal, contextual, and social constraints.

Early approaches to social navigation relied on **Artificial Potential Fields** (**APF**) [176] and the **Social Force Model** (**SFM**) [177]. These methods approximate human movement by simulating attractive and repulsive forces between agents and obstacles, allowing the robot to navigate while maintaining interpersonal distances. Although simple and widely adopted, APF-based approaches often require heuristic parameter tuning, which in some cases may lead to undesired robot behavior (e.g., jerky movement) [178]. Moreover, these methods can also suffer from local minima and robot stuck situations [179].

Other researchers explored **Deep Reinforcement Learning** (**DRL**) for social navigation [180, 181]. DRL approaches can generate adaptive behaviors by optimizing policies that minimize travel time and collisions with humans. However, their effectiveness is typically constrained by the complexity of human interactions, the reliance on accurately perceived pedestrian positions, and the difficulty of transferring trained models from simulation to real-world environments [182].

A third family of methods includes **search-based and sampling-based planners**. For instance, in [183], the authors presented an approach that utilizes a Transition-based RRT (T-RRT) [184], in which a Social Relationship Model (SRM) is used as a cost function to generate socially acceptable paths. Whilst their approach generates feasible and collision free paths, they do not compare against other approaches and their test environments only include static social agents.

More recently, in [185], the authors proposed an online social robot navigation framework that integrates a Social Heatmap to represent human density, a multilayered planning strategy for global and local navigation, and social cost functions to model human discomfort. Their system demonstrated improved performance and social compliance in large, crowded indoor environments using the Pepper robot, achieving higher success rates than previous DRL and APF-based methods.

Despite the progress achieved by the above methods, most approaches remain limited in their ability to formally guarantee safety and temporal consistency while adapting to human behavior. More recent works have therefore embraced *control-theoretic* and *logic-based* formulations to endow navigation with formal safety and timing properties. In particular, the studies presented in this thesis introduce a framework that combines **Control Barrier Functions** (**CBFs**) with **Signal Temporal Logic** (**STL**) to enforce spatio-temporal specifications, including obstacle avoidance, nonlinear velocity limits, and angular constraints for user visibility, through a quadratic program solved at control rate [10, 11]. The methodology is further validated and deployed on a real omnidi-

rectional platform, demonstrating that the same guarantees can be preserved beyond simulation [1], and extended with robustness to measurement noise via an **Unscented Kalman Filter** in [6].

While CBF-STL ensures rigorous safety and timing, **social intelligent navigation** also requires high-level reasoning about context and user needs (e.g., when to *reach*, *follow*, or *accompany*; how fast to move; which social distance to maintain). To this end, the chapter complements the control layer with a **Large Language Model**-based reasoning layer that interprets environmental and user-related cues and translates them into navigation constraints and parameters, which are then enforced by the low-level CBF controller (see Section 4.2).

The remainder of this chapter is organized as follows:

- **Section 4.1** introduces the proposed CBF-based STL social navigation framework. It provides a macro description of the algorithmic pipeline and recalls the theoretical components (CBFs for safety sets, STL for temporal specifications, and the resulting quadratic problem (QP)), together with its uncertainty-aware extension and notes on deployment.
- **Section 4.2** presents the *Intelligent Social Navigation* framework: a two-level architecture in which an LLM-based high-level module infers navigation mode, speed bounds, and social distances from contextual and user cues, while the low-level CBF controller guarantees safe execution.
- **Section 4.3** discusses limitations and future directions.

4.1 CBF-based STL social navigation

Among the various facets of social acceptance, **social navigation** represents a key factor for mobile robots operating in dynamic and human-populated environments. The problem addressed in this section is to ensure that a robot can reach a destination position while satisfying both safety and time requirements, even in the presence of nonlinear speed constraints and avoiding static and dynamic obstacles. To this end, the proposed framework combines CBFs and STL in a unified control formulation that guarantees the satisfaction of spatio-temporal constraints through a real-time optimization problem.

Moreover, this section first describes the robot control system considered in Sec. 4.1.1, as well as the concepts of CBF and STL discussed in Sec. 4.1.2 and Sec. 4.1.3, respectively. Then, it introduces the proposed framework in Sec. 4.1.4, followed by the experimental validation in Sec. 4.1.5, the real-world deployment in Sec. 4.1.6, and the robust extension in Sec. 4.1.7.

In the following, we denote $\mathbb{R}$ as the set of real numbers, while $\mathbb{R}^n$ is the n-dimensional real vector space. Non-negative and positive real numbers are $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$, respectively. A class $\mathcal{K}$ function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a continuous and strictly increasing function with $\alpha(0) = 0$.

4.1.1 Control system

Consider $x \in \mathcal{D} \subseteq \mathbb{R}^n$ and $u \in \mathcal{U} \subseteq \mathbb{R}^m$ be the state and input of a nonlinear input-affine control system:

$$\dot{x} = f(x) + g(x)u, \tag{4.1}$$

The functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are assumed to be locally Lipschitz continuous [10]. Given a control signal $u : [t_0, t_1] \to \mathcal{U}$, the signal $x : [t_0, t_1] \to \mathbb{R}^n$ is a solution to (4.1) if x is absolutely continuous and $x(t)$ satisfies (4.1) for all $t \in [t_0, t_1]$.

4.1.2 Control barrier functions

CBFs provide a formal framework for enforcing safety constraints through forward invariance of a **safe set**. Following CBF theory [186, 187, 188], we desire a given safe set $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$ to be **forward invariant** for the system, that is, $\forall x(0) \in \mathcal{C},\ x(t) \in \mathcal{C}\ \forall t \geq 0$. This property ensures that the specified set remains forward invariant with respect to the system's dynamics. Consequently, if the system starts within this set, it will always stay within it [186]. The forward invariance of the safe set can be achieved by means of the definition of the CBF.

Let $\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}$ be the set of configurations that satisfy the safety requirements for the system, also known as the safe set $\mathcal{C}$, which is defined as the super level set of a smooth function $h : \mathbb{R}^n \to \mathbb{R}$ with $\frac{\partial h}{\partial x}(x) \neq 0\ \forall x \in \partial\mathcal{C}$, where $\partial\mathcal{C} = \{x \in \mathbb{R}^n : h(x) = 0\}$. CBFs are defined by the following relation, considering an extended

class $\mathcal{K}$ function α such that for the system (4.1) and for all $x \in \mathcal{C}$:

$$\exists\, u \; s.t. \; \dot{h}(x,u) \geq -\alpha(h(x)) \Leftrightarrow \mathcal{C} \text{ is forward invariant.}$$

With this condition it is possible to set up an optimization-based control problem that modifies the desired control input $u_{des} \in \mathbb{R}^m$ and the actuated one so that the latter satisfies the CBF constraint, solving the following optimization problem:

$$u(x) = \underset{u \in \mathcal{U}}{\operatorname{argmin}} \; \frac{1}{2}||u - u_{des}||^2 \tag{4.2a}$$

$$\text{s.t. } \dot{h}(x,u) \geq -\alpha(h(x)). \tag{4.2b}$$

When dealing with multiple barrier functions, it is possible to utilize a **smooth approximation** of the min operator [6, 189] in order to approximate the set of all barriers into a single one, thus obtaining the most constraining barrier function. For k barrier functions $h_l : \mathbb{R}^n \to \mathbb{R}$ where $l \in \{1, ..., k\}$, let

$$h^*(x) := -\frac{1}{\eta} \ln \left(\sum_{l=1}^{k} \exp(-\eta h_l(x)) \right) \quad \text{with } \eta > 0. \tag{4.3}$$

Note that $\min_{l \in \{1,...,k\}} h_l(x) \approx h^*(x)$ where the precision of this approximation improves as η increases. This is beneficial because, if $h^*(x) \geq 0$, it ensures that $h_l(x) \geq 0$ for each $l \in \{1, ..., k\}$.

4.1.3 Signal temporal logic

We utilize STL as a temporal logic formalism due to its capability to express both qualitative and quantitative requirements of systems in continuous domains [190, 191]. It offers a natural and compact approach to analyze a robot's motion in a continuously evolving space-time environment. Let $s : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ be a continuous-time signal. STL involves logical predicates, denoted by μ, whose truth values are evaluated over continuous signals $s(t)$. The predicates [192] are obtained after evaluation of a predicate function $h : \mathbb{R}^n \to \mathbb{R}$ as:

$$\mu \coloneqq \begin{cases} \text{True} & \text{if } h(s(t)) \geq 0 \\ \text{False} & \text{if } h(s(t)) < 0. \end{cases} \tag{4.4}$$

In this thesis, the continuous signal is the system's state trajectory at time t, namely $x(t)$. The STL syntax [193] of an STL formula ϕ can be associated with one of the various expressions defined by the grammar in (4.5):

$$\phi ::= \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \, U_{[a,b]} \, \phi_2. \tag{4.5}$$

In particular, ϕ can be associated with the Boolean True ($\top$) signifying that the formula is always true; with the predicate μ indicating that ϕ holds true when the predicate is satisfied as shown in (4.4); with the "negation" operator meaning that ϕ is true when the negated formula $\neg\phi$ is false; with the "conjunction" operator of two STL formulas $\phi_1 \wedge \phi_2$ where ϕ is true when both ϕ_1 and ϕ_2 are simultaneously true; or with the "until" temporal

operator $\phi_1 U_{[a,b]} \phi_2$ where $a, b \in \mathbb{R}_{\geq 0}$ represent the bounds of the interval defined by the temporal operator, with $a \leq b$. In this context, ϕ is true when ϕ_1 becomes true and remains so within the specified time interval before ϕ_2 becomes true. To further enhance the expressiveness of STL, two additional operators are introduced: the "eventually" temporal operator, defined as $F_{[a,b]}\phi := \top U_{[a,b]} \phi$, and the "always" temporal operator, defined as $G_{[a,b]}\phi := \neg F_{[a,b]} \neg \phi$, where $G_{[a,b]}\phi$ is satisfied if ϕ is not violated during the interval $[a, b]$. The satisfaction relation $(x, t) \vDash \phi$ indicates that the signal $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, e.g., a solution of (4.1), satisfies ϕ at time t.

For a signal $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, the STL semantics [189] are defined recursively as follows:

$$
\begin{aligned}
(x,t) &\models \top && \Leftrightarrow \text{holds by definition},\\
(x,t) &\models \mu && \Leftrightarrow h(x(t)) \geq 0,\\
(x,t) &\models \neg\phi && \Leftrightarrow \neg((x,t) \models \phi),\\
(x,t) &\models \phi_1 \wedge \phi_2 && \Leftrightarrow (x,t) \models \phi_1 \wedge (x,t) \models \phi_2,\\
(x,t) &\models \phi_1 U_{[a,b]} \phi_2 && \Leftrightarrow \exists t_1 \in [t+a, t+b] \text{ s.t. } (x,t_1) \models \phi_2,\\
& && \quad \wedge \forall t_2 \in [t, t_1], (x,t_2) \models \phi_1,\\
(x,t) &\models F_{[a,b]}\phi && \Leftrightarrow \exists t_1 \in [t+a, t+b] \text{ s.t. } (x,t_1) \models \phi,\\
(x,t) &\models G_{[a,b]}\phi && \Leftrightarrow \forall t_1 \in [t+a, t+b], (x,t_1) \models \phi.
\end{aligned}
$$

All STL temporal operators have bounded time intervals in continuous time. The horizon of an STL formula is the minimum time needed to decide its satisfaction. For an STL formula that has no nested operators, its horizon is determined by the largest upper bound of the time intervals of all operators [194].

Finally, it is possible to discuss the quality of satisfaction by defining the quantitative semantics $\rho^{\phi}(x, t) \in \mathbb{R}$, which indicates how robustly a signal x satisfies ϕ at time t [195], thus obtaining a robustness value ρ instead of a Boolean value. Furthermore, it holds that $(x, t) \vDash \phi$ if $\rho^{\phi}(x, t) > 0$ and $(x, t) \vDash \phi$ implies $\rho^{\phi}(x, t) \geq 0$.

CBFs encoding STL formulation

In [196], the authors have established a connection between a function $\mathfrak{b} : \mathbb{R}^n \times [t_0, t_1] \to \mathbb{R}$, later shown to be a valid Control Barrier Function (vCBF), and the STL semantics of ϕ. In particular, if this function is in accordance with the conditions expressed in [189] (i.e., Steps A, B and C in [189]), then, for a given signal $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ with $\mathfrak{b}(x(t), t) \geq 0$ for all $t \geq 0$, it holds that $(x, 0) \vDash \phi$. Hence, $x(t) \in \mathcal{C}(t)$ for all $t \geq 0$ implies $(x, 0) \vDash \phi$.

Similar to [189], a switching mechanism can be used introducing $\mathfrak{o}_l : \mathbb{R}_{\geq 0} \to \{0, 1\}$ into the smooth approximation of the min operator (4.3), obtaining (4.6):

$$
\mathfrak{b}(x,t) := -\frac{1}{\eta} \ln \left(\sum_{l=1}^{k} \mathfrak{o}_l(t) \exp(-\eta \mathfrak{b}_l(x,t)) \right) \quad \text{with } \eta > 0. \tag{4.6}
$$

k is again the total number of functions $\mathfrak{b}_l(x, t)$ obtained as in [189] (Steps A, B and

C) and each $\mathfrak{b}_l(x,t)$ corresponds to either an "always", "eventually", or "until" operator with a corresponding time interval $\mathbb{I} = [a_l, b_l]$. If this mechanism is employed, it is necessary to remove the single functions $\mathfrak{b}_l(x,t)$ from $\mathfrak{b}(x,t)$ when the corresponding "always", "eventually", or "until" operator is satisfied. With these conditions it is possible to synthesize a QP that renders $\mathcal{C}(t)$ forward invariant when $\mathfrak{b}(x,t)$ is a vCBF. Therefore, we can consider:

$$\min_{u \in \mathcal{U}} u^T Q u \tag{4.7a}$$

$$\text{s.t. } \frac{\partial \mathfrak{b}(x,t)}{\partial x}(f(x) + g(x)u) + \frac{\partial \mathfrak{b}(x,t)}{\partial t} \geq -\alpha(\mathfrak{b}(x,t)) \tag{4.7b}$$

where $Q \in \mathbb{R}^{m \times m}$ is a positive semi-definite matrix. This convex optimization problem is feasible if $\mathfrak{b}(x,t)$ is a vCBF. The role of the $\alpha(\cdot)$ function [186] is to provide to the designer a way to modulate the action of the CBF, depending on whether a more conservative or aggressive behaviour is desired.

Construction of $\mathfrak{b}(x,t)$

The construction of the function $\mathfrak{b}(x,t)$ follows the procedure originally proposed in [196], which establishes the link between STL specifications and CBFs. For completeness, the method is briefly summarized below.

- Consider $\phi := F_{[t_a,t_b]}\mu$ or $\phi := G_{[t_a,t_b]}\mu$ and let

$$t^* := \begin{cases} t_b & \text{if } F_{[t_a,t_b]}\mu \ , \ t_b > 0 \\ t_a & \text{if } G_{[t_a,t_b]}\mu \ , \ t_a \geq 0. \end{cases} \tag{4.8}$$

- Let

$$h^{opt} := \sup_{x \in \mathbb{R}^n} h(x). \tag{4.9}$$

- Since we aim at satisfying ϕ with robustness threshold $r \in \mathbb{R}_{\geq 0}$, i.e., $\rho^{\phi}(x,0) \geq r$, then choose

$$r \in \begin{cases} (0, h^{opt}) & \text{if } t^* > 0 \\ (0, h(x(0))) & \text{if } t^* \geq 0. \end{cases} \tag{4.10}$$

- Consider $\mathfrak{b}(x,t) := -\gamma(t) + h(x)$, where $\gamma(t)$ is a non-decreasing, piecewise linear function defined as:

$$\gamma(t) := \begin{cases} \frac{\gamma_\infty - \gamma_0}{t^*} t + \gamma_0 & \text{if } t < t^* \\ \gamma_\infty & \text{otherwise.} \end{cases} \tag{4.11}$$

- Next, let

$$\gamma_0 \in (-\infty, h(x(0)), \tag{4.12a}$$

$$\gamma_\infty \in (\max(r, \gamma_0), h^{opt}). \tag{4.12b}$$

By following this procedure, it is possible to compute $\mathfrak{b}(x,t)$, and consequently its partial

derivatives $\frac{\partial \mathfrak{b}(x,t)}{\partial x}$ and $\frac{\partial \mathfrak{b}(x,t)}{\partial t}$, which are used in the quadratic program formulation (4.7b).

4.1.4 Proposed framework

In [1, 6, 10, 11], we propose a motion planning approach based on CBF and STL for completing a task at any time within a time interval in a dynamic system subject to nonlinear velocity constraints, angular constraint and collision avoidance. To this end, we consider the STL fragment [196] reported in (4.13). In particular, we divide STL formulas into two categories: ψ, defining an elementary class, and ϕ, defining a composite class that includes temporal operators and conjunctions of multiple elementary STL formulas:

$$\psi ::= \top \mid \mu \mid \neg\mu \mid \psi_1 \wedge \psi_2 \tag{4.13a}$$

$$\phi ::= G_{[a,b]}\psi \mid F_{[a,b]}\psi \mid \psi_1 U_{[a,b]}\psi_2 \mid \phi_1 \wedge \phi_2 \tag{4.13b}$$

where ψ_1, ψ_2 are formulas of the class ψ given in (4.13a), whereas ϕ_1 and ϕ_2 are formulas of the class ϕ given in (4.13b). Compared to [189], we make similar assumptions:

Assumption 1. *For an STL formula ϕ defined according to* (4.13b)*, there exists a constant $C \geq 0$ such that $(x, 0) \vDash \phi \implies ||x(t)|| \leq C \; \forall t \geq 0$.*

This guarantees that trajectories $x(t)$ are bounded.

Assumption 2. *The vector function $g(x)$ in* (4.1) *is such that $g(x)g(x)^T$ is positive definite for all $x \in \mathbb{R}^n$.*

In the following, we discuss the formulation of the specific constraints considered in this framework, namely collision avoidance, nonlinear velocity, and angular constraints.

Collision avoidance

Let's assume that the robot is located at the current position $x_0 \in \mathbb{R}^2$, and that a list of detected obstacle positions $x_{obs} \in \mathbb{R}^2$ is available within a certain sensing range. For a given obstacle i, approximated as a circle with radius $R_i \in \mathbb{R}_{\geq 0}$, the collision avoidance constraint can be satisfied by defining the barrier function:

$$h_{obs}(x) = ||x_0 - x_{obs_i}||^2 - R_i^2.$$

The function $h_{obs}(x)$ is a CBF if the control input u ensures that

$$\dot{h}_{obs}(x, u) = 2(x_0 - x_{obs_i})u \geq \alpha(h_{obs}(x)).$$

Thus, the most constraining barrier function $h^*_{obs}(x)$ can be defined using a smooth approximation of the min operator (4.3), leading to the constraint formulation in (4.17c).

Nonlinear velocity constraints

Let us consider a robot at the current position $x_0 = [x_{0_x}, x_{0_y}]^T$, and suppose it needs to reach the goal $x_G = [x_{G_x}, x_{G_y}]^T$ within a time $t^* = t_b$ seconds, with a certain tolerance distance ε from x_G. This problem can be expressed in terms of the following STL formula:

$$\phi : F_{[t_a,t_b]}(||x_0 - x_G|| < \varepsilon).$$

To construct $\mathfrak{b}(x,t)$, in the case of the "eventually" operator, according to Sec. 4.1.4, it is necessary to consider t^* as the time t_b belonging to the interval $\mathbb{I} = [t_a; t_b]$. This process does not allow the desired task to be completed at any time within the interval $\mathbb{I}$, but it will be completed at t_b, unless the vCBF is constructed with a different $t^* < t_b \in \mathbb{I}$. Furthermore, considering a system that must provide safety-critical guarantees, it is necessary to take into account the presence of velocity constraints that may limit the system dynamics during task execution. In [10], we propose a two-step construction procedure that allows addressing these constraints.

1. The **first step** involves determining the average velocity, $v_{average}$, that the robot would have along the path to be crossed. Assuming there are no obstacles, the robot will follow the minimum Euclidean distance path, thus obtaining that $v_{average}$, set by the STL formula ϕ, is given by the ratio between the variation of the distance traveled Δs_{tot}, and the time t^* imposed by ϕ, as defined in (4.14):

$$\Delta s_{tot} = \sqrt{(x_{0_x} - x_{G_x})^2 + (x_{0_y} - x_{G_y})^2}, \tag{4.14a}$$

$$v_{average} = \frac{\Delta s_{tot}}{t^*}. \tag{4.14b}$$

   In the event that one or more obstacles are present along the robot's path, the distance it will need to traverse will be greater, leading to an increase in velocity to ensure the satisfaction of the STL specification ϕ (i.e., to ensure that the task is completed by t^*). Introducing the velocity constraint v_{max}, two sub-cases can be considered: the first, $v_{max} \geq v_{average}$ in which the specification ϕ continues to be satisfied; the second, $v_{max} < v_{average}$, in which the STL specification ϕ cannot be satisfied. Consider, for example, the case shown in Fig. 4.1: in this situation, the STL specification requires the robot to reach the destination x_G from the initial position x_0, covering a distance of $\Delta s_{tot} = 17\ m$, in $t^* = 10\ s$. As a result, $v_{average} = 1.7\ m/s$, which does not allow the satisfaction of the specification in the last tract of $1.5\ m$, shown in grey, since $v_{max} < v_{average}$. A similar scenario that considers the presence of an obstacle is depicted in Fig. 4.2. In this case as well, the robot should have an average velocity $v_{average} = 1.7\ m/s$ to ensure the satisfaction of the specification within $t^* = 10\ s$. This cannot be achieved as it conflicts with the velocity constraint of $v_{max} = 1\ m/s$ when the robot enters in the area shown in yellow near the obstacle, since $v_{max} < v_{average}$.

2. This **second step** provides a solution that allows resolving the issue introduced by the velocity constraints and simultaneously addresses the problem associated with

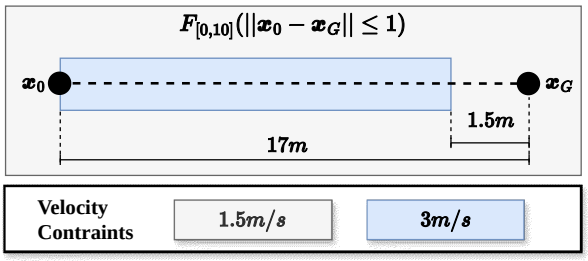


Figure 4.1: Example 1: The robot must reach the final state x_G within 10 seconds while being subject to two maximum velocity constraints defined by different colored areas along its path.

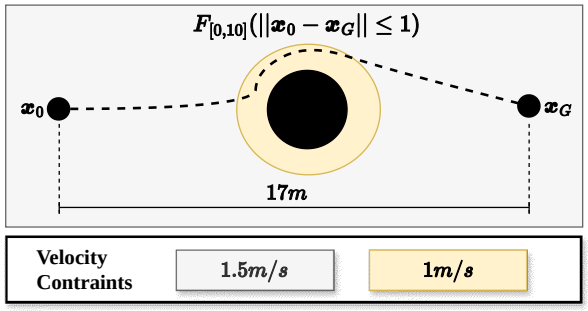


Figure 4.2: Example 2: The robot must reach the final state x_G within 10 seconds while being subject to two maximum velocity constraints defined by different colored areas along its path in the case of obstacle avoidance.

the inability to conclude a task at any time within the interval $\mathbb{I}$, considering the "eventually" operator.

In particular, we propose the computation of a dynamically defined bound t^*_{new}, given by

$$t^*_{new} = \frac{\Delta s}{(P_i - P_r P_c)\, v_{max}(t)} \tag{4.15}$$

where:

- Δs is the remaining path space;
- $v_{max}(t)$ is the maximum velocity;
- P_i represents the *initial percentage*, taking a user-defined value $0.5 \leq P_i < 1$ which allows the $v_{average}$ to be increased so as to complete the specification within $\mathbb{I}$.
- P_r is the *reduction percentage*, taking an arbitrary value $0 < P_i < 0.2$ which allows to decrement *initial percentage* in case the solver does not converge to solution.
- P_c is the *percentage counter*, with an initial value of zero, and it is incremented in case the QP fails to find a solution.

Specifically, the solution consists in instantaneously checking the maximum velocity $v_{max}(t)$ and dynamically computing a t^*_{new} based on the remaining distance Δs and the weighted maximum allowable velocity $(P_i - P_r P_c)\, v_{max}(t)$. This computation needs to be implemented whenever a change in v_{max} occurs or when the quadratic problem does not converge to a feasible solution due to the exceeding of speed constraints. Hence, the quantity t^*_{new} will be used to compute a new barrier $\mathfrak{b}(x,t)$,

as explained in Sec. 4.1.4. This procedure will allow the robot to travel the route at a speed greater than $v_{average}$, thus allowing the specification to be satisfied in the interval $\mathbb{I}$, and to overcome the problem depicted in Figs. 4.1 and 4.2.

Regarding nonlinear velocity constraints, considering the velocity vector $v_{real} = [v_x, v_y, w]^T$, we introduce an additional constraint within the QP problem by imposing the squared norm of the nonlinear velocity to be $\|v_x + v_y\| \leq v_{max}$, leading to the constraint formulation in (4.17d).

Angular constraint

Exploiting the methodology presented in [11, 188], the FOV constraint in (4.17e) can be defined. Specifically, considering $\mathcal{W}$ as the world reference frame, $\mathcal{R}$ as the local robot reference frame, and $\mathcal{H}$ as the local human reference frame, the FOV constraint is formulated based on the relative position of the detected user, represented by the components ${}^{\mathcal{W}}x_H$ and ${}^{\mathcal{W}}y_H$, with respect to the robot state $x = [{}^{\mathcal{W}}x, {}^{\mathcal{W}}y, {}^{\mathcal{W}}\theta]$. This relationship is expressed as:

$$ {}^{\mathcal{R}}p_{\mathcal{H}} = [{}^{\mathcal{W}}x_H - {}^{\mathcal{W}}x, {}^{\mathcal{W}}y_H - {}^{\mathcal{W}}y]^T. $$

Next, the constraint is split into two components that are treated as different constraints entered into an optimization solver. To this end, we introduce the following $h_{fov}(\cdot)$ for the robot:

$$ h_{fov}({}^{\mathcal{R}}p_{\mathcal{H}}) = \begin{bmatrix} h_1({}^{\mathcal{R}}p_{\mathcal{H}}) \\ h_2({}^{\mathcal{R}}p_{\mathcal{H}}) \end{bmatrix} = \lambda \begin{bmatrix} tan(\frac{\beta}{2}) & 1 \\ tan(\frac{\beta}{2}) & -1 \end{bmatrix} {}^{\mathcal{R}}p_{\mathcal{H}}. \tag{4.16} $$

Imposing the condition $h_{fov}(\cdot) \geq [0, 0]^T$ is equivalent to constraining the robot's orientation such that the user remains within the angular FOV β. The parameter λ takes a value of -1 if the constraint is applied behind the robot in accompany mode, or 1 during the reach and follow modes. The time derivative of (4.16) is then expressed as:

$$ \dot{h}_{fov}({}^{\mathcal{R}}p_{\mathcal{H}}, u) = \frac{\partial h_{fov}({}^{\mathcal{R}}p_{\mathcal{H}})}{\partial {}^{\mathcal{R}}p_{\mathcal{H}}} {}^{\mathcal{R}}\dot{p}_{\mathcal{H}} = \lambda \begin{bmatrix} \tan\left(\frac{\beta}{2}\right) & 1 \\ \tan\left(\frac{\beta}{2}\right) & -1 \end{bmatrix} {}^{\mathcal{R}}\dot{p}_{\mathcal{H}}, $$

where the velocity of human with respect to the robot, ${}^{\mathcal{R}}\dot{p}_{\mathcal{H}}$, is then expressed through kinematic computations:

$$ {}^{\mathcal{R}}\dot{p}_{\mathcal{H}} = {}^{\mathcal{R}}R_{\mathcal{W}} {}^{\mathcal{W}}\dot{p}_{\mathcal{H}} - \begin{bmatrix} {}^{\mathcal{R}}v_x \\ {}^{\mathcal{R}}v_y \end{bmatrix} + \omega \begin{bmatrix} {}^{\mathcal{W}}y_H - {}^{\mathcal{W}}y \\ -({}^{\mathcal{W}}x_H - {}^{\mathcal{W}}x) \end{bmatrix}, $$

considering the rotation matrix ${}^{\mathcal{R}}R_{\mathcal{W}}$:

$$ {}^{\mathcal{R}}R_{\mathcal{W}} = \begin{bmatrix} cos({}^{\mathcal{W}}\theta) & sin({}^{\mathcal{W}}\theta) \\ -sin({}^{\mathcal{W}}\theta) & cos({}^{\mathcal{W}}\theta) \end{bmatrix}. $$

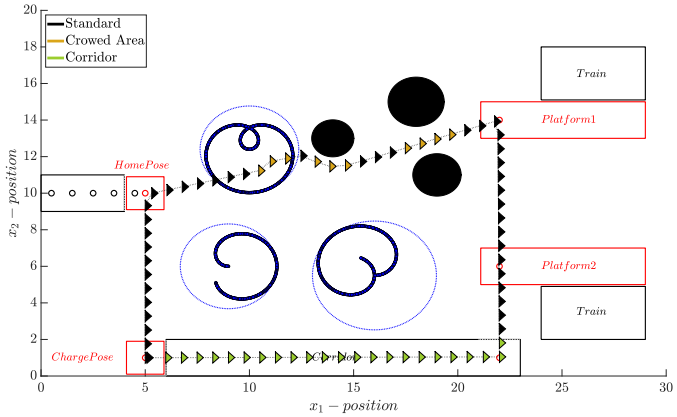


Figure 4.3: Robot trajectory in simulated environment.

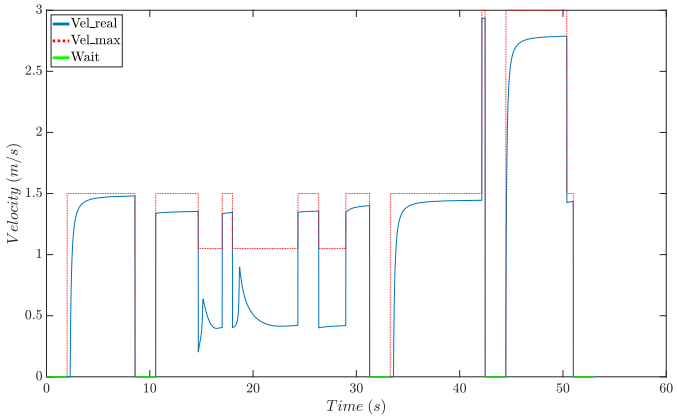


Figure 4.4: Time evolution of real speed in reference to maximum speed.

In conclusion, the following quadratic optimization problem can be formulated:

$$\min_{u \in \mathcal{U}} u^T Q u \tag{4.17a}$$

$$\text{s.t.} \quad \frac{\partial \mathfrak{b}(x,t)}{\partial x}(\dot{x}) + \frac{\partial \mathfrak{b}(x,t)}{\partial t} \geq -\alpha(\mathfrak{b}(x,t)), \tag{4.17b}$$

$$\dot{h}^*_{obs}(x,u) \geq -\alpha(h^*_{obs}(x)), \tag{4.17c}$$

$$\|v_x + v_y\| \leq v_{max}, \tag{4.17d}$$

$$\dot{h}_{fov}({}^{\mathcal{R}}p_{\mathcal{H}}, u) \geq -\alpha(h_{fov}({}^{\mathcal{R}}p_{\mathcal{H}})). \tag{4.17e}$$

4.1.5 Experimental validation

In order to test the proposed framework, we have conducted two different experimental validation in Matlab. Specifically, in [10], we simulated a social navigation application, shown in Fig. 4.3 and in the accompanying video[1].

Let us consider a scenario in which a robot is located in a train station and needs to guide a person to a platform. Depending on the robot's position, the velocity constraint v_{max} can vary, creating different operational modes:

- *Standard*: the default mode, with $v_{max} = 1.5\ m/s$;
- *Crowded Area*: when the robot is close to dynamic or static obstacles, with $v_{max} =$

[1]DOI: https://doi.org/10.5281/zenodo.10075373

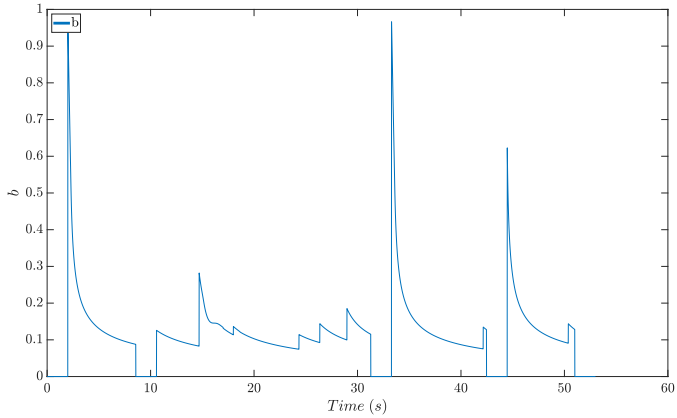


Figure 4.5: The behavior of the control barrier function $\mathfrak{b}(x,t)$ has a value always greater than zero, indicating the satisfaction of the STL formula ϕ.

1.05 m/s;

- *Corridor*: when the robot is within a corridor, where human access is not allowed and we assume there are not obstacles. Here, we set $v_{max} = 3\ m/s$.

For the simulation, we employed a three-wheeled omnidirectional robot model as implemented in [189]. The simulation environment incorporates six obstacles, comprising three static and three dynamic obstacles, randomly distributed in the environment. To create a more realistic simulation of dynamic obstacle behavior, distinct rhodonea curve trajectories are assigned to each obstacle. To successfully guide the user to their destination, the robot starts from the *charge_pose* and proceeds to the *home_pose*. Subsequently, it must safely guide the user to the platform while avoiding collisions. Upon completing the task, the robot will enter the *corridor* and ultimately return to the initial position. The robot is expected to execute this sequence of operations within various time intervals, for a maximum total of 60 seconds. The temporal constraints are expressed through the following STL formula: $\phi = \phi' \wedge \phi'' \wedge \phi''' \wedge \phi''''$ with a certain tolerance $\varepsilon = 0.2\ m$ and velocity constrains, where:

$$\begin{aligned}
\phi' &:= F_{[0,10]}(||x - x_{HOME_POSE}|| \leq \varepsilon) \\
\phi'' &:= F_{[10,40]}(||x - x_{PLATFORM_1}|| \leq \varepsilon) \\
\phi''' &:= F_{[40,50]}(||x - x_{CORRIDOR}|| \leq \varepsilon) \\
\phi'''' &:= F_{[50,60]}(||x - x_{CHARGE_POSE}|| \leq \varepsilon).
\end{aligned}$$

Figure 4.4 shows the real velocity profile along the path that satisfies the nonlinear velocity constraints, shown in red. For simplicity, we considered the robot internal dynamics $f(x)$ equal to zero. As a result, its velocity is given by $v_{real} = g(x)u$, from (4.1). During the simulation, pauses were introduced at the end of each STL specification, as indicated by the green segments in Fig. 4.4, where the robot is not moving.

In addition, as reported in Table 4.1, it can be observed that each specification is satisfied within the time interval defined by its own temporal operator. As a result, by employing this approach, it was possible to decrease the execution time of the STL formula ϕ to approximately 47 s instead of 60 s. The performance of the function $\mathfrak{b}(x,t)$, shown in Fig. 4.5, demonstrates the satisfaction of the formula ϕ throughout the entire simulation,

Table 4.1: Time comparison between constraint in the STL formula and actual path duration. Time in [s].

	ϕ'	ϕ''	ϕ'''	ϕ''''	t_{total}
STL constraint	10	30	10	10	60
Actual path duration	6.51	25.06	9.01	6.50	47.08

as it ensures that its value is always greater than zero. Using the proposed approach, it is possible to observe the results of motion planning, which has allowed for the identification of a valid path for the robot and the satisfaction of STL specifications subject to nonlinear velocity constraints, ensuring the compliance with safety guarantees.

Regarding the angular constraint, in [11], we simulated a scenario, shown in Fig. 4.6 and in the accompanying video[2], using a similar simulation proposed in [10]. The input control u are the angular velocities of the robot, considering the three-wheeled omnidirectional robot model described in [197]. We assumed maintaining the user at an arbitrary distance with a certain level of noise from the robot.

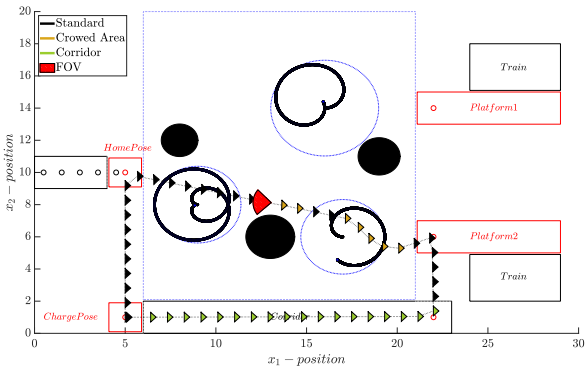


Figure 4.6: The robot starts from *ChargePose* and, upon detecting the user, moves to *HomePose*. The robot guides the user to the *Platform2* while ensuring velocity constraints, angular constraints, and collision avoidance. Subsequently, the robot returns to the home pose through an obstacle-free corridor.

Using the proposed approach, it is possible to observe the results of motion planning, which allows for the identification of a valid path for the robot and the satisfaction of STL specifications subject to nonlinear velocity constraints, angular constraint, ensuring the compliance with safety guarantees.

4.1.6 Real-world deployment

In [1], we provided a practical implementation of the CBF–based STL social navigation framework in real-world conditions with omnidirectional robots, extending our previous studies [10, 11]. Additionally, we release the associated ROS2 Humble code, providing instruction on how to scale the considered framework to any robotic platform. To validate the proposed system, we conducted different tests in a real-world scenario, as shown

[2]DOI: https://doi.org/10.5281/zenodo.10255241

Figure 4.7: Real-world environment with omnidirectional mobile robot and two unknown obstacles.

in Fig. 4.7, demonstrating that the architecture successfully satisfies the constraints considered.

To implement the proposed architecture with ROS2 Humble, an omnidirectional mobile robot equipped with laser scanners, a rear-facing camera and an onboard speaker is employed. The implementation code is available on GitHub[3]. The proposed architecture, organized into several functional blocks, enables a mobile robot to navigate towards a destination in a socially acceptable manner while satisfying multiple constraints and interacting vocally with the user.

As illustrated in Fig. 4.8, the architecture is organized into four main functional blocks. The *INPUT* block (blue) receives the user request, which specifies the navigation goals and temporal constraints. The *camera node* (green) estimates the human–robot distance using the rear camera, while the *Robot platform* (orange) provides localization information and laser scanner data. Finally, the *CBF–STL block* (red) integrates these inputs to compute the robot's velocity commands, as well as service calls to adjust the speaker volume and trigger vocal interactions. In particular:

- **INPUT request:** the *INPUT* request is a message of type `my_interfaces/msg/Input` published on the `/input` topic. This message can be generated either manually from the terminal or automatically from an external node. When received, it triggers the robot's navigation toward one or more destinations under temporal constraints. Specifically, *INPUT* request includes the following parameters: a `temp_operator`, corresponding to the STL operators eventually (F), always (G), or until (U); `goal` and `goal2`; and the temporal bounds `ta`, `tb`, and `tab`.
  An example of *INPUT* request is: `ros2 topic pub /input my_interfaces/msg/Input "temp_operator: 'F', goal: [9.0, 8.0], goal2: [0.0, 0.0], ta: 1, tb: 5, tab: 0" -once`.
- **Camera node:** this node is used to compute the distance between the camera and the user during robot navigation. The resulting human–robot distance is exploited by the *CBF–STL block* to enable vocal interaction, allowing the robot to notify the user to either approach or move away depending on a predefined threshold. In particular, the *camera node* publishes the distance value as a `std_msgs/Float32` message on the `/human_distance` topic, as shown in Fig. 4.8.

[3] `https://github.com/andrearuo/omnibase_ws`

- **ROBOT platform:** after performing environment mapping, the robot platform employs Adaptive Monte Carlo Localization (AMCL), which estimates the position and orientation of the robot within the environment. The localization data are published on the `/amcl_pose` topic and provided as input to the *CBF–STL block*, together with the laser scanner data published on the `/scan` topic, which are used to detect obstacles in the robot's surroundings in real-time. Moreover, the robot receives velocity commands through the `/cmd_vel` topic, processes volume adjustments via the `/volume` service, and handles text-to-speech (TTS) output through the `/say` service.
- **CBF-STL block:** this block generates the robot's velocity commands to drive it toward the destination while satisfying multiple constraints. Specifically, upon receiving an *INPUT* request, the *manager cbf-stl node* first sets the robot's volume via the `/volume` service by providing the value of the `VOLUME_MAX` variable. Next, a request is sent to the *cbf-stl node* through the `/input_request` service to initiate the navigation.

  During task execution, the human-robot distance measured by the *camera node* is continuously monitored to provide acoustic notifications to the user. In particular, an acoustic alert is triggered to inform the user to move away if their distance falls below the threshold `MIN_DISTANCE_NOTIFICATION` value. In contrast, a different notification instructs the user to approach if their distance exceeds the threshold `MAX_DISTANCE_NOTIFICATION` value.

  The *cbf-stl node*, developed on the basis of our previous work [10, 11], solves a quadratic optimization problem, through the ALGLIB library [198], formulated as in (4.17). The output of this optimization problem, u, is the velocity command published on the `/cmd_vel` topic, while simultaneously enforcing several key constraints, such as: the spatio-temporal constraints, corresponding to (4.17b); the collision avoidance, through (4.17c); the velocity constraints in (4.17d) and angular constraints as shown in (4.17e). In particular, spatio-temporal constraints are satisfied by encoding STL into CBFs using parameters defined in the *INPUT* request with the procedure exposed in [10, 189].

  Regarding collision avoidance, the system is able to avoid unknown obstacles by processing messages of type `my_interfaces/msg/Obstacles` on the `/obstacles` topic, which include the following parameters: the number of detected obstacles (`o_number`), their radii (`o_radius`), and their positions (`ox_pose` and `oy_pose`). These parameters are provided by the *check obstacle node*, which combines the robot's localization from the `/amcl_pose` topic with laser scanner data from the `/scan` topic in order to cluster obstacles as walls or columns. Specifically, clustering is performed using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) library [199], excluding laser scanner measurements beyond the arbitrary threshold value defined by the variable `CHECK_OBSTACLE_RADIUS`. Subsequently, the *check obstacle node* assigns a `RADIUS_WALL` value when an obstacle is classified as a column, and the `RADIUS_COLUMN` value when it is classified as a wall. In terms of velocity constraints, the maximum velocity, v_{max}, is set by the parameter `MAX_VEL_ROBOT`, which is dynamically reduced to `OBS_VEL_ROBOT` if an obstacle is detected within the `OBS_DIST_ROBOT` value of the robot.

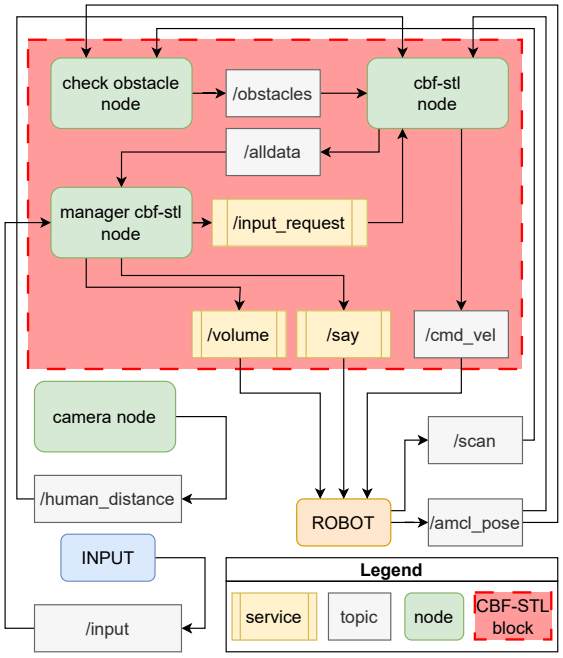


Figure 4.8: Overall architecture of the proposed system.

About angular constraints, the system ensures the goal remains within the robot's FOV by using the procedure described in [11]. It leverages the goal position from the *INPUT* request, the robot's pose from the `/robot_pose` topic, and an arbitrary FOV angle variable `FOV_GOAL_RADIANS`.
These CBFs can be modulated by leveraging the $\alpha(\cdot)$ functions, as reported in [186], depending on whether a more conservative or a more aggressive behavior is desired. Specifically, within the *manager cbf-stl node* it is possible to adjust the parameters `ESP_STL` and `K_STL`, while within the *cbf-stl node* the parameters `ESP_OBS`, `ESP_FOV_GOAL`, `K_OBS`, and `K_FOV_GOAL` can be tuned. Moreover, every 0.02 s (i.e., with frequency 50 Hz), the *cbf-stl node* publishes navigation data on the `/alldata` topic, which are then saved by the *manager cbf-stl node*, and the command velocity to the robot.

Experimental setup

The proposed framework was validated using the PAL Robotics TIAGo Pro robot [200] connected to an external PC equipped with an Intel Ultra 7 CPU, 16 GB RAM, and an NVIDIA RTX 4050 GPU, running Ubuntu 22.04 and ROS2 Humble. In particular, the robot employed in this work is composed of an omnidirectional base with four wheels, computer with CPU Intel i5, RAM 8 GB, SSD 250 GB and 802.11ax, Dual Band, 2x2 Wi-Fi 6 + Bluetooth 5.2, two laser 360° LIDAR FOV, 1 x 6 DoF IMU and two speakers. The communication between the robot and the PC was established via a dedicated SSH Ethernet connection.

A ZED2 AI Stereo Camera [201] is used, mounted on the rear expansion panel of the robot through a custom 3D-printed support and connected to the PC via a USB 3.1 cable. In order to obtain the human–robot distance, the *camera node* (provided in the shared code with the name *zed2 camera node*) interfaces with the *zed-ros2-wrapper* [202],

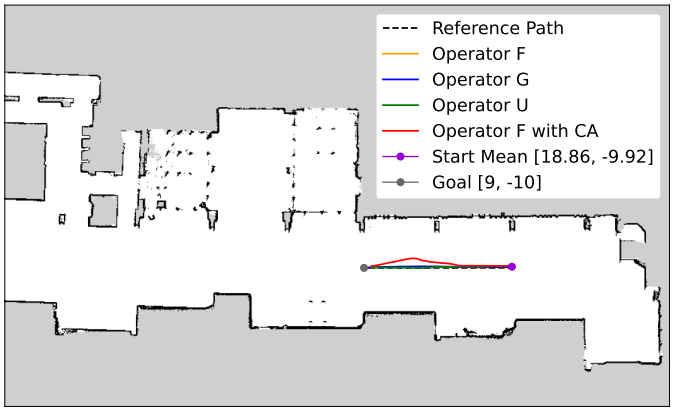


Figure 4.9: 2D map containing the average path taken by each of the four cases, relative to the reference path.

which bridges the camera with ROS2 through its ZED SDK 5.0 [203]. By enabling the `bt_enabled` parameter and setting the `body_format` to `BODY_18` [204] within the *common_stereo.yaml* file in the *zed_wrapper*, the *camera node* can select a reference landmark (landmark ID 1 in our case, corresponding to the neck). The entire architecture was executed on the external PC, with the exception of AMCL localization, which was performed onboard the robot.

The experiments were conducted in a public convention center, as illustrated in Fig. 4.7, whose 2D map is reconstructed in Fig. 4.9. The environment includes the presence of unknown static obstacles and people who can move freely within the area. Moreover, the arbitrary parameters are selected as in Table 4.2.

Table 4.2: Parameters used in the experimental validation

Variable	Value
`MIN_DISTANCE_NOTIFICATION`	$0.8\ m$
`MIN_DISTANCE_NOTIFICATION`	$3.5\ m$
`CHECK_OBSTACLE_RADIUS`	$3.0\ m$
`RADIUS_WALL`	$0.4\ m$
`RADIUS_COLUMN`	$0.6\ m$
`MAX_VEL_ROBOT`	$0.7\ m/s$
`OBS_VEL_ROBOT`	$0.5\ m/s$
`OBS_DIST_ROBOT`	$1.5\ m$
`FOV_GOAL_RADIANS`	$\frac{5\pi}{180}\ radians$

Experimental validation

To validate our proposed architecture, we conducted a total of 35 tests in four different cases. Specifically, the tests involved reaching a predefined goal position using three different temporal operators in combination with collision avoidance, velocity limit, and angular constraint. In detail, 10 tests were performed using the temporal operator eventually (F), 10 using the operator always (G), 10 using the operator until (U), and 5 using the operator F in the presence of two random and unknown static obstacles, as shown in

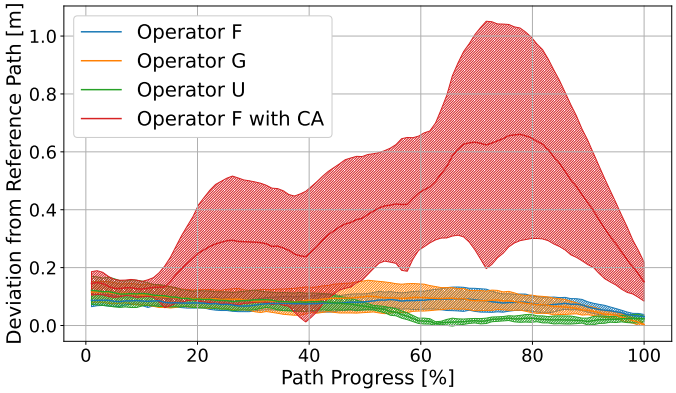


Figure 4.10: Deviation in meters (mean ± standard deviation) of the four cases relative to the reference path.

the attached video *CA test example.mp4*[4].

Additionally, during the task, the robot was able to measure the user's position relative to itself using the rear camera. Based on these data, the robot can communicate with the user, instructing them to approach or move away if the predefined distance limits are exceeded.

Following the tests, we considered the starting midpoint of the 35 tests and the predefined destination at (9, -10), thus obtaining the reference path shown in black in Fig. 4.9. This reference path was used to calculate the robot's deviation from it, as shown in Fig. 4.10. The results demonstrate that the robot's trajectory closely followed the reference path with minimal deviation for each temporal operator, unlike the last case involving random obstacles along the path.

Finally, the execution times of each simulation were analyzed, and the results are presented in Fig. 4.11. These results highlight that in all cases the simulations satisfied the time constraints.

Moreover, a final test was carried out, as shown in the attached video *DA test example.mp4*[4], in which the robot guides a user to an office in the convention center, demonstrating the system's responsiveness even in the presence of dynamic obstacles.

Scalability to other hardware platforms

The shared code refers to the implementation shown in Fig. 4.8 with the hardware setup described in Section 4.1.6. To adapt the proposed architecture to different hardware platforms, the following nodes must be modified:

- *check obstacle node*: in the case of a different arrangement of laser scanners, or when only a single laser scanner is available, the node must be modified to process the incoming data accordingly. Specifically, the data may arrive from a topic different from `/scan`, and the transformations of the laser scanners with respect to the robot's local reference frame must be updated. The *check obstacle node* must still publish the previously defined parameters on the `/obstacles` topic;

[4] `https://doi.org/10.5281/zenodo.17980308`

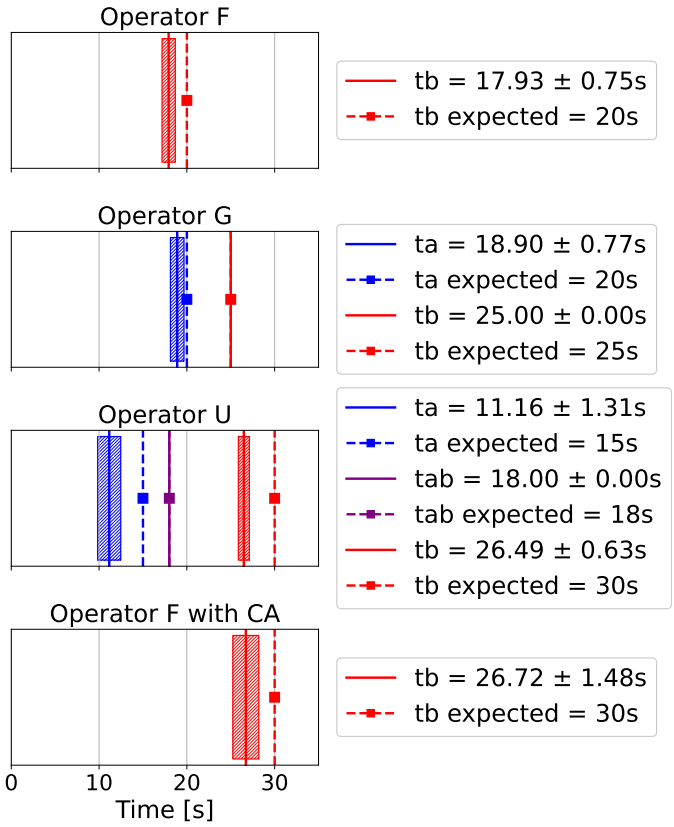


Figure 4.11: Average execution times with their respective variances for each of the four cases.

- *manager cbf-stl node*: when using a robot other than those provided by PAL Robotics, the `SayActionClient` class must be adapted to comply with the TTS protocol of the target platform. In particular, the *manager cbf-stl node* relies on the function `SayActionClient.send_speech_goal(const std::string &text)` to transmit voice messages to the robot, which must therefore be re-implemented to ensure compatibility;
- *camera node*: if a different camera is used in place of the ZED2 adopted in this work, the corresponding SDK, wrappers, and interfaces must be installed. Furthermore, the *camera node* must be modified to ensure compatibility with the new device. Importantly, the new camera node must still publish the measured human–robot distance on the `/human_distance` topic using the `std_msgs/Float32` message type, so as to ensure interoperability with the rest of the architecture;
- *map configuration*: users must create their own YAML map file to reflect the operating environment of the robot. The map should be generated through standard SLAM or mapping procedures available in ROS2 and stored in the workspace. The YAML file must then be loaded onto the robot and referenced in the navigation stack to enable proper localization and path planning.

4.1.7 Uncertainty

The growing complexity of modern robotic systems and the environments in which they operate necessitates careful consideration of safety, increasing the challenges associated with social navigation. Moreover, localization measurements are often affected by random

noise, which can lead to unsafe behavior if not addressed in the control design. Therefore, it is crucial to ensure robustness against **measurement noise**.

In [205], the authors proposed a framework for safety-critical control of systems with erroneous state estimates. This framework leverages CBFs and unifies the method of backup sets for synthesizing control invariant sets with robustness requirements. The outcome is the synthesis of measurement-robust CBFs, which provide guarantees for safe behavior in the presence of imperfect measurements. The works in [206, 207] considered robust CBF formulations with worst-case disturbance bounds to ensure safety. Specifically, in [206], the authors introduced the concept of robust-CBF, which, when combined with input-to-state stable control Lyapunov functions, yields controllers for constrained nonlinear systems under known, partially known, and unknown disturbances. In [207], the authors proposed a robust control Lyapunov function and a CBF-based QP controller with an **Unscented Kalman filter** (**UKF**) which has the capability to mitigate the effects of state disturbance and measurement errors. Safety guarantees in the presence of measurement noise were addressed for stochastic systems in [208]. The authors presented a framework for CBFs in stochastic systems subject to Gaussian process and measurement noise. They developed CBFs for environments with incomplete state information, with states estimated via sensors corrupted by Gaussian noise. It has been proven that the proposed CBFs ensure safety when the state estimate is within a specified bound of the true state, a condition achieved using an extended Kalman filter (EKF) when the system is linear or when the process and measurement noise are sufficiently small.

Building on our previous works [10, 11], in [6] we proposed a robust CBF-based STL motion planning methodology combined with an UKF to mitigate the effects of noise on state measurement. Hence, our methodology ensures task completion at any time within a specified time interval for dynamic systems subject to nonlinear velocity constraints, collision avoidance, and erroneous state measurements.

Unscented Kalman filter

The UKF describes a filtering algorithm for estimating the state of a nonlinear system, affected by **random noises**, based on a sequence of observations and control information [207]. Specifically, we consider the following dynamic nonlinear time-invariant discrete-time system with additive noise:

$$\begin{aligned} x_{k+1} &= F(x_k, u_k) + v_k, \\ y_k &= H(x_k) + n_k, \end{aligned} \tag{4.18}$$

where $x_k \in \mathbb{R}^n$ denotes the unobserved state of the system at time k, $y_k \in \mathbb{R}^p$ is the observed measurement signal at time k, v_k is a zero-mean Gaussian noise process with covariance matrix $Q \in \mathbb{R}^{n\times n}$ at time k, n_k is a zero-mean Gaussian noise observation with covariance matrix $R \in \mathbb{R}^{p\times p}$ at time k, and $u_k \in \mathbb{R}^n$ is the control input at time k. Without loss of generality, we assume here that v and n are uncorrelated. F and H are the state transition function and the measurement function respectively [209]. In general, F and H are not linear. In order to estimate the probability distribution of the

system's state using efficient Kalman filtering, one has to linearize the functions F and H. While the EKF performs this linearization using Taylor series expansion, which can introduce errors in the posterior mean and covariance, the UKF uses a more accurate stochastic approximation, known as the **unscented transformation**. This method provides a better estimate of the posterior distribution without requiring linearization, while maintaining the same computational complexity as the EKF [210].

It is possible to estimate the state using the UKF algorithm given in [210], considering that the input is the mean and covariance of the estimate at time k, along with the control input, u_k, and the observation, y_k.

Proposed approach

In the presence of noise affecting the robot's position, the direct measurement becomes unreliable, necessitating **estimation of the state**. In particular, the estimated state, in contrast to noisy measurements, is utilized to compute a vCBF.

UKF requires a discrete-time state transition function, F in (4.18), but the robot's model considered in (4.1) is continuous-time. To compute the discrete-time approximation of the continuous-time model, we rely on the Euler discretization method [211]. Considering the robot's dynamical system in (4.1), the resulting discrete-time state transition function, defined in (4.18) is:

$$F(x_k, u_k) = x_k + (f(x_k) + g(x_k)u_k)T_s, \tag{4.19}$$

where the accuracy of this approximation depends on the sample time T_s. Moreover, we assume that the measurement function, H in (4.18), is an identity matrix since the robot's output is its position, which coincides with the state.

Following the formulation of UKF [210], the augmented state is defined as the concatenation of the original state x and noise variables v and n:

$$x^a = [x^T, v^T, n^T]^T \in \mathbb{R}^L, \text{ with } L = 2n + p. \tag{4.20}$$

The UKF is initialized with:

$$\hat{x}_0 = \mathbb{E}[x_0], \tag{4.21}$$

$$P_0 = \mathbb{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T], \tag{4.22}$$

$$\hat{x}_0^a = \mathbb{E}[x_0^a] = [\hat{x}_0^T, 0^T, 0^T], \tag{4.23}$$

$$P_0^a = \mathbb{E}[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}, \tag{4.24}$$

where $\mathbb{E}$ is the expected value, $\hat{x}_0$ is the initial state estimation, $\hat{x}_0^a$ is the initial augmented state estimation, while $P_0 \in \mathbb{R}^{n \times n}$ and $P_0^a \in \mathbb{R}^{L \times L}$ are the covariance matrix of $\hat{x}_0$ and $\hat{x}_0^a$ respectively. Since v and n are white Gaussian noise, their mean over time is always zero. At time step k, the state x_k^a can be estimated according to the algorithm in [210].

The correct setting of the Q and R matrices helps make the filters more robust. Usually, the process noise covariance matrix Q is determined using an innovation-based method [212], which relies on the difference between the actual measurement and the predicted position value from the system model. Conversely, the measurement noise covariance matrix R is often determined empirically, accounting for uncertainties in the tracking data. When empirical determination is not feasible, it is possible to assume a diagonal structure for R and estimate it from the knowledge of y and x as $R = I \cdot ||\varepsilon||^2$, with $||\varepsilon||^2 = ||y - Hx||^2$, as discussed in [209].

The **robust CBF-STL motion planning algorithm** can be expressed from the QP in (4.7), that ensures $\mathcal{C}(t)$ remains forward invariant, when $\mathfrak{b}(\hat{x}, t)$ is a vCBF, considering the estimated state $\hat{x}$ as follows:

$$\min_{u \in \mathcal{U}} \; u^T Q u \tag{4.25a}$$

$$\text{s.t.} \;\; \frac{\partial \mathfrak{b}(\hat{x}, t)}{\partial \hat{x}}(f(\hat{x}) + g(\hat{x})u) + \frac{\partial \mathfrak{b}(\hat{x}, t)}{\partial t} \geq -\alpha(\mathfrak{b}(\hat{x}, t)), \tag{4.25b}$$

$$\|v_x + v_y\| \leq v_{max}.$$

In particular, we aim to minimize the control input u while satisfying the following constraints: i) the CBF-STL constraint ensures task completion within a specified time interval. The function $\mathfrak{b}(x, t)$, leveraging the smooth approximation of the min operator in (4.3), incorporates both static and dynamic obstacle avoidance barriers, thereby enabling collision avoidance. In this case, the constraint includes the state estimate derived from UKF, as opposed to the approach used in [10, 11]; ii) the nonlinear velocity constraint, as we are interested in constraining the velocity norm, which requires the quadratic norm of the omnidirectional robot's velocities to be less than or equal to v_{max}.

Simulation results

To validate the proposed approach, we consider a scenario in which an omnidirectional robot is in a social environment and needs to guide a person to various points of interest (e.g., $x_{pose\_1} = (8; 2)$ and $x_{pose\_2} = (5; 13)$). The simulation environment includes two static and two dynamic obstacles. Hence, the robot must plan and execute a collision-free path with the ability to reprogram its trajectory, reduce its speed in proximity of obstacles, or accelerate in obstacle-free areas. Moreover, we assume that depending on the robot's position, the velocity constraint v_{max} can vary, creating different operational modes: i) *Standard mode*: the default mode; ii) *Reduction mode*: when the robot is close to dynamic or static obstacles; iii) *Speed mode*: when the robot is going on a path free of obstacles. Arbitrarily, we set $v_{max} = \{1.5;\ 1.05;\ 3\}\ m/s$ for the different modes, respectively.

Additionally, we assume the initial position of the robot is known, and we consider that the robot must satisfy the following STL specification:

$$\phi = \phi_1 U_{[15,45]} \phi_2, \tag{4.26}$$

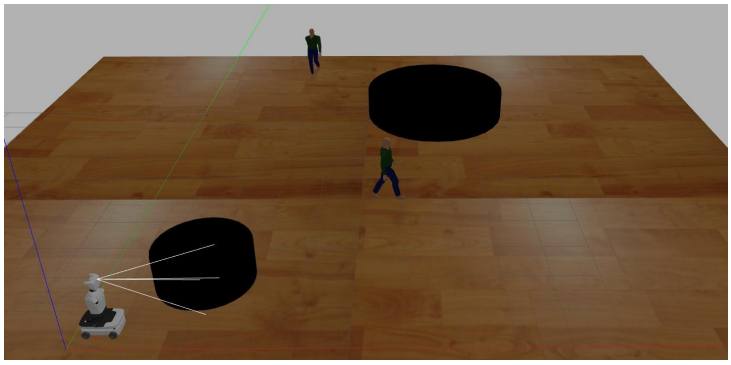


Figure 4.12: Gazebo simulation of the considered environment, where we placed two static and two dynamic obstacles (black cylinders and humans).

where:

$$\begin{aligned}\phi_1 &= G_{[15,18]}(||x - x_{pose\_1}|| \leq \varepsilon),\\ \phi_2 &= F_{[18,45]}(||x - x_{pose\_2}|| \leq \varepsilon).\end{aligned} \tag{4.27}$$

In particular, (4.26) defines the conjunction of the two specifications, ϕ_1 and ϕ_2, in (4.27). The first specification, ϕ_1, requires the robot to reach pose $x_{pose\_1} = (8; 2)$ with a tolerance $\varepsilon = 0.2\ m$, within 15 s, and remain in that position until 18 s. Then, ϕ_2, requires the robot to reach pose $x_{pose\_2} = (5; 13)$ with tolerance $\varepsilon = 0.2\ m$ within 45 s.

To evaluate the proposed framework, we simulated a social navigation application in Gazebo using ROS2 Humble across three different use cases, as demonstrated in the accompanying video[5]. In the first scenario, the ideal case without disturbances is considered, following the methodology outlined in [10]. In the second scenario, disturbances are introduced, resulting in localization uncertainties for the robot, again employing the methodology from [10]. In the third scenario, we build on the second case by applying the robust CBF-STL motion planning algorithm proposed in this work. The layout of the Gazebo simulation is depicted in Fig. 4.12, while Fig. 4.13 shows the 2D layout, with the colored areas representing the robot's three different operational modes.

Assuming an ideal scenario with no disturbances in the state measurement, and applying the framework described in [10], we obtain the results presented in Fig. 4.14. In particular, the satisfaction of the ϕ specification is indicated by the fact that the robot reaches the first destination in about 11 s, and leaves for the second destination at the time of 18 s, arriving within 45 s. In addition, the adherence of the velocity constraints and the positivity of the barrier are ensured.

In the real scenario, we consider the use of a robot location sensor that introduces uncertainty into the state measurements. Specifically, we assume that the sensor provides the robot's position with a uniform error distribution between -0.05 and $0.05\ m$. The results presented in Fig. 4.15 were obtained by applying the methodology described in [10]. These results demonstrate the satisfaction of the specification in (4.26) and conformity to speed constraints; however, the constant positivity of the barrier is not guaranteed. In particular, the top image in Fig. 4.15 shows that the robot has continuous speed spikes,

[5]DOI: https://doi.org/10.5281/zenodo.13768594

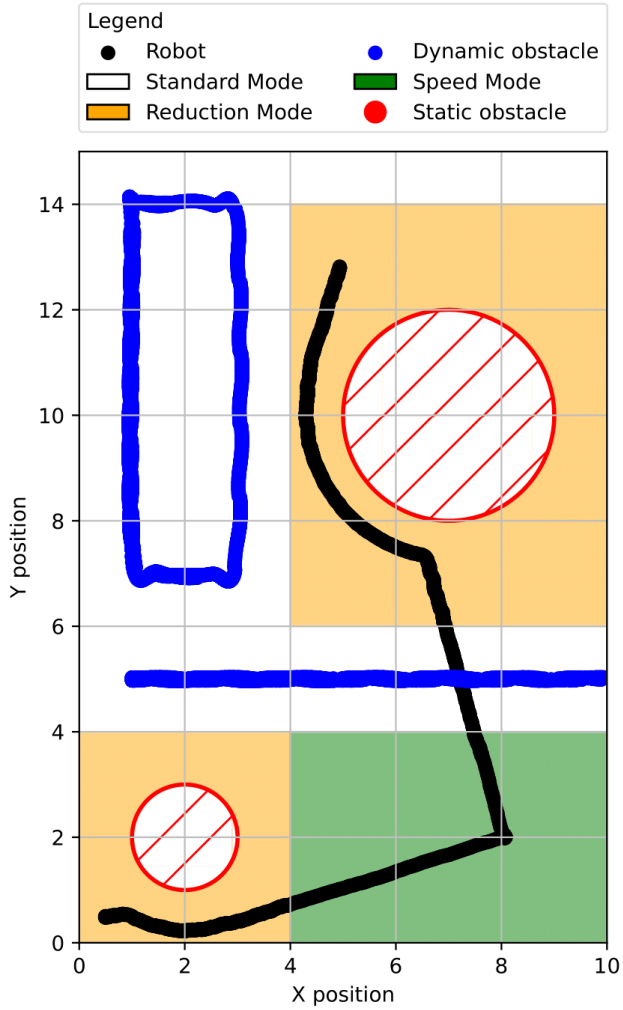


Figure 4.13: Robot's path is shown in black, the paths of the dynamic obstacles are shown in blue, the static obstacle positions are depicted in red, while the orange rectangles represent the areas where the robot must operate in reduction mode, the green areas represent where the robot must be in speed mode, and the white areas indicate the default mode.

caused by disturbances, resulting in problems with energy consumption and negative user experience. An important point to note is that the small disturbance in the algorithm proposed in [10] does not ensure a safe trajectory execution.

This issue can be addressed using the method proposed in this work, with the results illustrated in Fig. 4.16. In the UKF, T_s has been chosen as the reciprocal of the QP simulation execution frequency equal to 20 ms. Q and R are computed as follows: Euler's method was used to predict the robot's future position in the first case, with no measurement error assumed. The resulting position deviation was 0.01 m. To reflect this deviation in the process noise covariance matrix Q, the variance was calculated as the square of the deviation value. Therefore, a variance of 0.0001 was assigned to the diagonal elements of the Q matrix to accurately represent the predicted uncertainty in the robot's position. Assuming the uniformly distributed noise, as described in the second case, the variance terms of the R matrix, which are equal to the square of the disturbance value, were determined to be 0.0025.

In particular, Fig. 4.16 shows the satisfaction of the specification ϕ as well as adherence to speed constraints and barrier positivity in the case of uncertainties in measurements

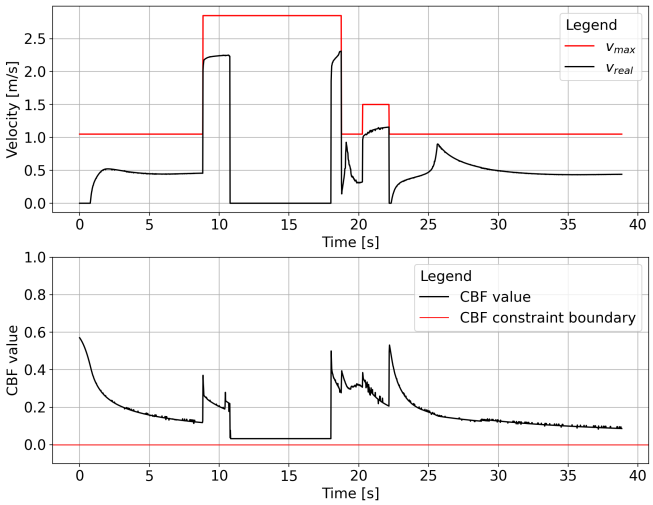


Figure 4.14: Considering the ideal case, the top plot shows the robot's speed in black, while the maximum allowed speed is shown in red. The second plot depicts the barrier value in black, which does not fall below the constraint boundary shown in red.

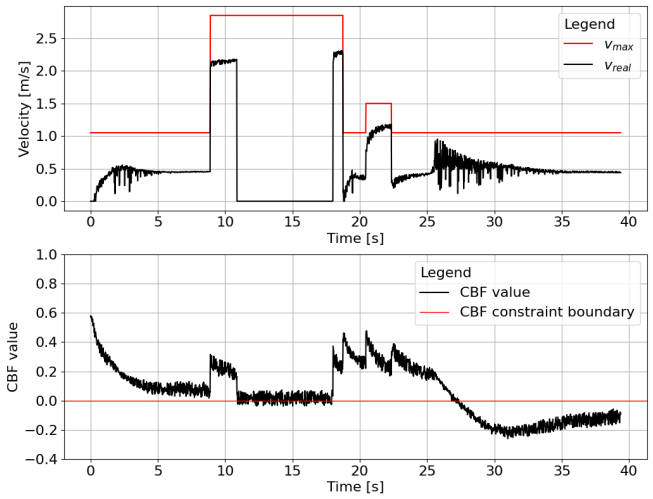


Figure 4.15: Considering the real case with a uniform error distribution between -0.05 and $0.05\ m$, in the top image, the robot's speed is shown in black, while the maximum allowed speed is shown in red. In the second image, the barrier value is depicted in black while the constraint barrier boundary is shown in red.

of the state. Moreover, the mean distance between the corrupted observed state and the true position obtained from Gazebo, x_{real}, in the second case, is equal to 0.038, while the mean distance between the estimated state $\hat{x}$ and x_{real}, in the third case, is equal to 0.018. This difference highlights the significant reduction in errors achieved by the proposed method compared to the method presented in [10] when the state is subject to measurement noise.

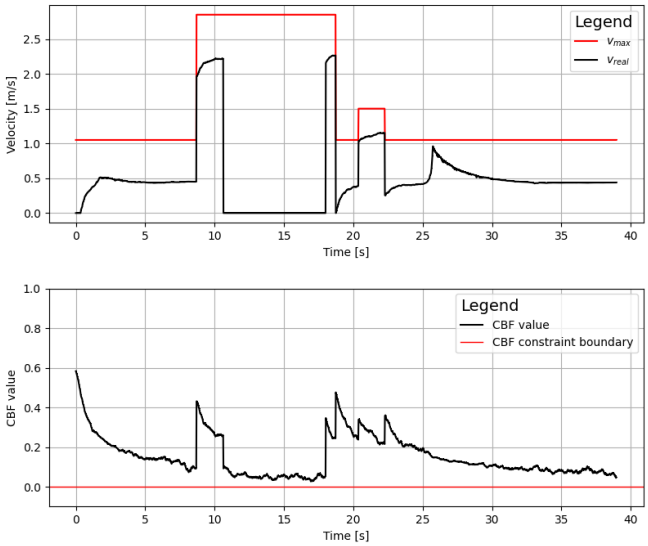


Figure 4.16: Considering the real case with the proposed approach, in the top image, the robot's speed is shown in black, while the maximum allowed speed is shown in red. In the second image, the barrier value is depicted in black while the constraint barrier boundary is shown in red.

4.2 Intelligent social navigation

While the previous section focused on low-level control approaches for **social navigation** based on CBFs and STL, this section explores the integration of high-level reasoning mechanisms into the navigation process. The concept of **intelligent social navigation** emerges from the need to enhance robot adaptability and contextual awareness in dynamic human-shared environments.

One of the key challenges in this field is enabling robots to assist people beyond the limits of traditional robotics expertise and to effectively understand and respond to human and social cues, a process known as **social perception** [213, 214]. For successful interaction with humans and an appropriate response to their requests and needs, service robots should recognize and comprehend their interlocutor. This requires the development of intelligent systems capable of engaging with users in a fluid and context-aware manner, responding to **social cues** such as gestures, body language, verbal and non-verbal stimuli, and spatial positioning.

As many other fields, the rise of artificial intelligence (AI) also impacted the HRI domain. In particular, the integration of **LLMs** with robotic systems has introduced a transformative paradigm, providing unprecedented capabilities in natural language understanding and task execution [215]. Beyond perception, LLMs can be used to score potential next actions and even generate action sequences directly from natural language instructions, without requiring additional domain-specific knowledge. Several research efforts have leveraged these capabilities to improve robotic perception, planning, and decision-making in dynamic environments, exploring different methodologies to improve interaction and adaptation. For instance, the LLM-GROP approach, presented in [216], focuses on extracting commonsense knowledge from LLMs to infer semantically valid object configurations and integrate them with task and motion planning to generalize to varying scene geometry. In [217], the authors proposed a method to personalize household cleanup by learning user preferences for object placement. Their approach uses language-based planning and few-shot learning with LLMs to infer generalized preferences from limited interactions. This enables robots to adapt quickly to different users and apply learned preferences to future tasks.

More related to the social navigation domain, in [218], the authors proposed a novel hybrid approach called Socially-Aware navigation Large Language Model framework (SALM), which integrates LLM and Deep Reinforcement Learning to navigate through human-filled public spaces and provide multiple social services. SALM infers global planning from human-in-the-loop commands in real-time, and encodes social information into a LLM-based large navigation model for low-level motion execution.

Despite the growing use of LLMs across various sectors, enabling service robots to perceive and interpret contextual and environmental cues remains an open research challenge. This capability would allow robots to act as socially intelligent agents, adapting their actions to the social context (Fig. 4.17). For example, a robot could accompany a user, follow them to a destination, or approach a group interested in interacting. Un-

Figure 4.17: Social robot operating into a human-shared scenario.

derstanding the operational context would also enable robots to consider social and psychological factors, such as maintaining appropriate interpersonal distance based on emotional state or adjusting speed to match the user's age and mobility.

In [5], we propose a two-layer architecture that modulates navigation constraints for a social robot based on its operating environment:

- the **high-level reasoning layer** uses LLMs to perceive contextual and environmental cues from the social context. Based on this information, the LLMs generate low-level constraints that guide the robot's navigation, ensuring its movements are natural and socially acceptable. For example, social factors such as the required distance the robot must maintain from the user and its maximum allowable speed influence navigation decisions. Additionally, the robot's navigation mode, categorized as reach, follow, or accompany, is dynamically adjusted according to the surrounding context;
- the **low-level layer** handles navigation using CBFs as a reactive method. Using a reactive approach, the service robot can dynamically adjust its control, ensuring smooth and socially acceptable interactions with users. Leveraging the high-level reasoning layer, the low-level navigation system incorporates social constraints, enhancing user safety and acceptance of the robot's behavior.

4.2.1 Large language model

LLMs, widely used in robotics, automation, and interactive systems, generate and interpret human-like text and support structured decision-making. A key feature is **Function Calling** (**FC**), which enhances their utility by enabling structured outputs that external systems can execute. This ensures responses adhere to predefined formats, improving reliability—an essential aspect for tasks like robotic navigation.

In this context, LLMs can be represented as a function approximation:

$$f_\phi : \mathcal{P} \times \mathcal{S} \to \mathcal{T},$$

where f_ϕ is the LLM function parameterized by ϕ (the actual domain), $\mathcal{P}$ is the space of

natural language prompts, $\mathcal{S}$ is the set of the functions schema, $\mathcal{T}$ is the space of possible structured text outputs. Given a function schema s (e.g., the robot function to call) and a natural language prompt p, the LLM function generates a mapping $f_\phi(p, s) \rightarrow t^*$, where $t^* \in \mathcal{T}_\mathcal{S}$, the subset of outputs conforming to $\mathcal{S}$. The structured output t^* is parsed into a function call: $q : \mathcal{T}_\mathcal{S} \rightarrow \mathcal{E}$ where $\mathcal{E}$ is the set of executable robot operations, which finally maps the selected function to call to a specific robot behavior.

4.2.2 Social zones

Along with navigation constraints, another important element regards the **social zones**. Social norms, often implicit and unspoken, are challenging to define and implement in robotics. When robots move to avoid humans, it is important to ensure not only physical safety, which prevents collisions, but also psychological safety, which avoids causing disturbance or discomfort to people [219]. **Proximity** plays a key role in avoidance behaviors, with research on proxemics examining personal space, a fundamental aspect of social convention that, when invaded, can cause discomfort.

Using the procedure exposed in [220] and considering a and b as generic Cartesian coordinates, the **human social interaction space** is modeled as a two-dimensional asymmetric Gaussian function, $f(a, b) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_{>0}$, centered at the user's position (x_H, y_H), which is determined as follows:

$$f(a,b) = e^{-\left(\frac{(d\cos(\gamma))^2}{2\sigma_a^2} + \frac{(d\sin(\gamma))^2}{2\sigma_b^2}\right)}, \tag{4.28}$$

where:

$$\begin{aligned} d &= \sqrt{(a - x_H)^2 + (b - y_H)^2}, \\ \theta &= atan2((b - y_H), (a - x_H)), \\ \theta^{po} &= \text{rotation of the function } f \text{ minus } \frac{\pi}{2}, \\ \gamma &= \theta - \theta^{po}, \text{ converted in } (-\pi, \pi]. \end{aligned}$$

The values of σ_a and σ_b in (4.28) vary among σ_l, σ_h, σ_s and σ_r according to γ in the four different quadrants, as shown in Fig. 4.18.

Thus, by arbitrarily choosing l_r, the length of the square representing the intimate zone of a human, and the values of d_i, according to [220], the four variances σ_l, σ_h, σ_s and σ_r for the function $f(a, b)$ are calculated as follows:

$$\sigma_i = \frac{1}{2}\sqrt{\frac{(d_i + 0.5l_r)^2}{ln10}}, \text{with } i = \{l, h, s, r\}. \tag{4.29}$$

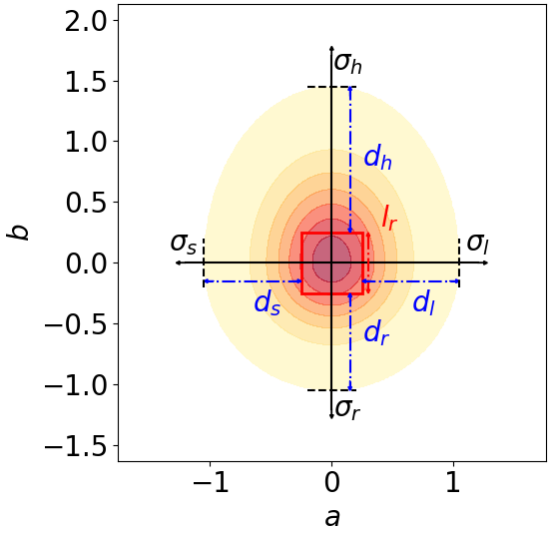


Figure 4.18: Human social interaction space.

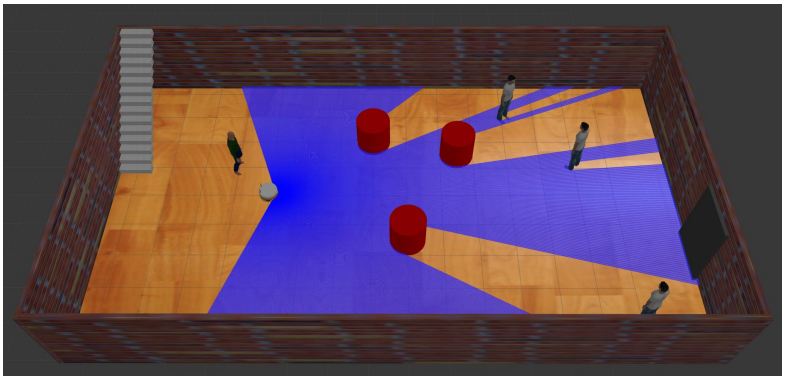


Figure 4.19: Simulated scenario in Gazebo: the robot performs navigation tasks while adhering to social navigation constraints. The environment also includes three static obstacles (red) and a door (black).

4.2.3 Proposed framework

Consider a scenario where a robot navigates a social environment with static obstacles (e.g., objects, stationary individuals) and dynamic ones (e.g., moving people). The robot has access to contextual and environmental data, including user IDs, preferences, positions, and map data. As illustrated in Fig. 4.20, the proposed architecture enables the high-level reasoning layer to interpret social cues and adapt navigation accordingly. If certain information is unavailable, the robot retrieves it from a dedicated database (as shown in [214]). The high-level layer also determines the appropriate navigation mode by adjusting low-level navigation constraints. In particular, we consider the different scenarios where a user might need to reach a target position, be accompanied by the robot or be followed by the robot. To this end, we introduce three navigation modes, which we name reach, accompany, or follow. To ensure socially acceptable behavior, the robot should maintain greater distance from users who appear uncomfortable and dynamically adjusts its speed based on user characteristics, moving faster with younger individuals and slower with elderly or impaired users.

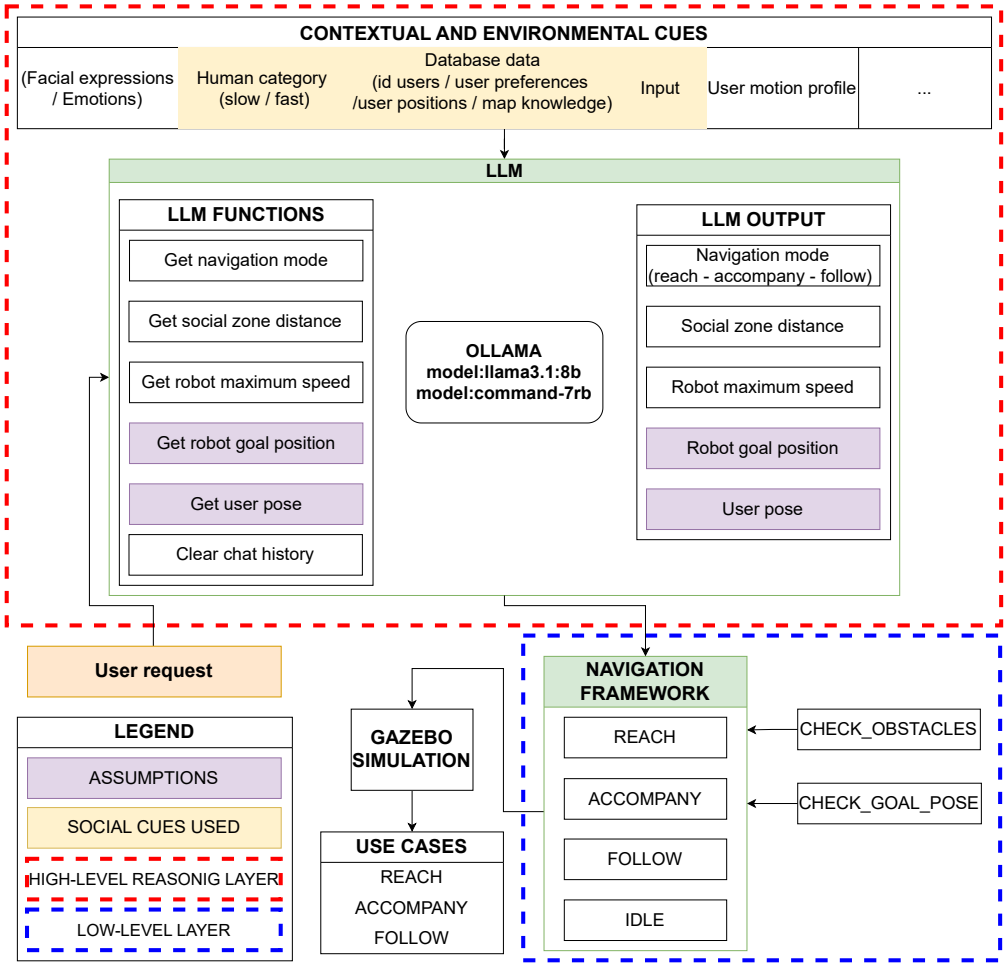


Figure 4.20: Proposed system architecture: high-level reasoning layer (red) and low-level layer (blue).

System Architecture Referring to the architecture diagram shown in Fig. 4.20, after the user request for robot assistance, depicted in orange, the high-level reasoning layer, highlighted in red, processes the request along with contextual and environmental cues extracted from the robot's operational area. This layer then generates specific outputs via FC, which are subsequently passed to the low-level layer, marked in blue. The low-level layer employs CBFs to ensure smooth and socially acceptable behaviors while adhering to social navigation constraints, such as collision avoidance, velocity limits, and angular constraints. Furthermore, this layer continuously detects obstacles around the robot using the *check_obstacles* block and determines the optimal goal pose when the robot is instructed to reach an individual or a group using the *check_goal_pose* block.

High-level reasoning Although the architecture conceptually allows for the integration of various types of contextual and environmental cues, our implementation specifically employs those highlighted in yellow in Fig. 4.20. Following the user's input request and the considered social cues, the high-level reasoning layer performs an initial FC request to determine whether the user intends to initiate a conversation with the robot, be accompanied, be followed, or be reached at a specific location. Such FC request is implemented through the *request context* mentioned in Tab. 4.3, which contains sample prompts specifying the robot context used in the FC processes.

Following a human request, the high-level layer performs different queries to assess first the contextual information and later the task to accomplish. In the first step, the robot relies on the LLM engine to update some specific parameters of the interacting user. In particular, based on the robot database and possible sensor input, the LLM engine of the architecture retrieves fundamental information for the social intelligent navigation. In the specific case of our work in [5], we considered the following parameters: the **user id** or their name; the user's **category**, classified as either slow (e.g., elderly individuals or people with impairments) or fast (e.g., young individuals); the **confidence score**, a numeric value indicating the user comfort with robotic systems; the **user's current position**, **speed** and **target destination**. Each field can be set to an unknown value if the corresponding information is unavailable.

Next, by exploiting the information retrieved about the user and contextual environment, and the explicit received request, the LLM executes a series of FC requests, using the functions illustrated in Fig. 4.20. The FC outputs are mapped to specific robot behaviors and then passed to the low-level layer of the architecture. These outputs include:

- the **navigation mode**, choosing between reach, accompany and follow;
- the **robot maximum speed**, related to the context included in Tab. 4.3, by providing the LLM system with the *robot maximum speed context*. This value will be applied in the reach and accompany modes, while, in the follow mode, the robot will use the human speed parameter as its maximum speed;
- the **social zone distance**, by providing the LLM with the *social zone distance context* (see Tab. 4.3), based on Hall's social zone and proximity distance definitions [220]. This distance varies depending on the detected confidence score of the user. It is used in the reach mode to determine the robot's goal pose and in the accompany and follow modes to set the distance the robot should maintain from the user during the task. Additionally, this parameter is used as d_h to compute σ_h in (4.29), which determines how far the robot should position itself in front of the user. Meanwhile d_l, d_s and d_r are considered fixed, as a mobile robot positioned behind or to either side of a human is generally less dangerous than one in front at the same distance, as reported in [220].

The high-level reasoning layer is designed to retain user input requests relevant to the same context within the LLM's **chat history**. This prevents information loss, enhancing the robustness of the architecture. For instance, if a user initially requests to be followed into a specific location (e.g., the kitchen) but later the robot perceives discomfort or confusion by the user, the high-level layer adjusts the navigation parameters accordingly by leveraging the historical context of the interaction. However, when the context, and thus the interacting user, changes, the layer resets the chat history.

Navigation framework The low-level layer is used as a navigation framework to perform reach, follow, and accompany tasks through the use of CBFs. Receiving input parameters from the high-level reasoning layer enables the low-level layer to execute smooth robot movements with socially acceptable behaviors while adhering to social navigation constraints. In particular, the navigation framework is defined by the optimization prob-

Table 4.3: LLM system prompts list

request context
You are a robot navigation assistant. You know information about the most engaged user parameters: $\{json.dumps(most\_engaged\_user.model\_dump())\}$ and the actual LLM output parameters: $\{json.dumps(llm\_output.model\_dump())\}$ Based on the context of the user's request, you are able, by utilizing the available tools $\{tools\}$, to provide various outputs such as: • The parameters needed to accompany a specific user in json format. • The parameters needed to reach a specific user in json format. • The parameters needed to follow a specific user in json format. • Reset the context when the interaction with the user is finished. • Provide the chat history. • Otherwise, anwers to generic questions.
robot maximum speed context
You are a robot navigation assistant with the following strict adherence rule: - Reply only in the following format: • robot maximum speed: < value > $[m/s]$ • short explanation of the reasoning You have access to information about the user's category parameter: $\{json.dumps(most\_engaged\_user.model\_dump())\}$. Your task is to determine the robot maximum speed following these rules: • The robot maximum speed for the robot is a float value between $0.1[m/s]$ and $0.8[m/s]$. • If the user is Fast: the robot can move at a faster speed, prioritizing efficiency and responsiveness. • If the user is Slow: the robot should move at a moderate speed to ensure safety and comfort. • If the user is Impaired: the robot should prioritize safety and move at a slow speed. • If the user's state is unclear, unspecified, or unknown, use the default robot maximum speed of $0.5[m/s]$.
social zone distance context
You are a robot navigation assistant with the following strict adherence rule: - Reply only in the following format: • social zone distance: < value > $[m]$ • short explanation of the reasoning You have access to information about the user's confidence parameter: $\{json.dumps(most\_engaged\_user.model\_dump())\}$. • A high confidence value indicates that the user is likely to prefer closer interaction with the robot. • A low confidence value indicates the user may prefer more distance. The social zone distance is a float value between $1.0[m]$ and $1.4[m]$. Your task is to determine the appropriate social zone distance based solely on the user's confidence state, following these guidelines: • Choose a smaller distance (closer to $1.0[m]$) if the user feels comfortable (e.g., confident, relaxed, open, calm, secure, reassured). • Choose a larger distance (closer to $1.4[m]$) if the user feels uncomfortable (e.g., anxious, nervous, afraid, suspicious, restless, tense, worried, hesitant, defensive). • If the user's state is unclear, unspecified, or unknown, default the social zone distance to $1.2[m]$.

lem in (4.30), where collision avoidance, velocity constraints, and angular constraints are considered as social navigation constraints, as expressed in (4.30b).

The control input $u \in \mathbb{R}^2$, representing the velocity components $v_x \in \mathbb{R}$ and $v_y \in \mathbb{R}$, is computed with reference to the optimization problem in (4.2) as:

$$u(x), \delta = \underset{u \in \mathcal{U}, \delta \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{2}||u - u_{des}||^2 + m\delta^2 \tag{4.30a}$$

$$\begin{aligned} \text{s.t. } & \dot{h}^*_{obs}(x, u) \geq -\alpha(h^*_{obs}(x)), \\ & \|v_x + v_y\| \leq v_{max}, \\ & \dot{h}_{fov}({}^{\mathcal{R}}p_{\mathcal{H}}, u) \geq -\alpha(h_{fov}({}^{\mathcal{R}}p_{\mathcal{H}})) - \delta. \end{aligned} \tag{4.30b}$$

Here, the desired control input $u_{des} \in \mathbb{R}^2$ is designed to drive the system toward the target position $x_{tgt} \in \mathbb{R}^2$ relative to the current robot pose $x_{act} \in \mathbb{R}^2$, which corresponds to the first two components of the state $x \in \mathbb{R}^3$. Thus, u_{des} is computed as: $u_{des} = K(x_{tgt} - x_{act})$, where $K \in \mathbb{R}_{\geq 0}$ is a proportional gain. Moreover, we consider a

slack variable $\delta \in \mathbb{R}$ with a slack variable weight $m \in \mathbb{R}_{\geq 0}$ in order to relax the angular constraint in cases where the constraint cannot be strictly satisfied due to system limitations (e.g., the use of a differential-drive robot).

Within the navigation framework, the *check_ obstacles* module is responsible for continuously elaborating the laser scan data of the robot, generating a list of detected obstacle positions, $x_{obs} \in \mathbb{R}^2$, at a certain distance from it. Thus, the most constraining barrier function $h^*_{obs}(x)$ can be defined using a smooth approximation of the min operator (4.3), leading to the constraint formulation in (4.30b). Regarding the non-linear velocity constraint, given the robot maximum speed parameter, $v_{max} \in \mathbb{R}_{\geq 0}$, provided by the high-level reasoning layer, the constraint is imposed by ensuring that the squared norm of the velocity satisfies $\|v_x + v_y\| \leq v_{max}$ as expressed in (4.30b). Exploiting the methodology presented in [11, 188], the FOV constraint in (4.30b) can be defined, as shown in Sec. 4.1.4.

Within the navigation framework, the social zone distance parameter regulates the robot's interaction with humans across different navigation modes. In the reach mode, it determines the optimal position for the robot to approach the interacting user based on their location and orientation, following the procedure described in [221]. In the accompany mode, this parameter ensures that the robot does not advance toward the target destination if the distance between the robot and the user exceeds the predefined social zone distance. Finally, in the follow mode, it prevents the robot from approaching the user beyond the social zone distance.

4.2.4 Experimental validation

To validate the proposed architecture, we conducted Gazebo simulations with a differential-drive robot operating at 50 Hz in a social environment, populated with different users and unknown static obstacles, as shown in Fig. 4.19 and in the accompanying video[6]. As LLM engine, we considered the *llama3.1:8b* and *command-7rb* models.

In the considered scenario, the goal of the robot is to assist the users to satisfy their requests. To demonstrate the effectiveness of the proposed approach, we defined two different use cases:

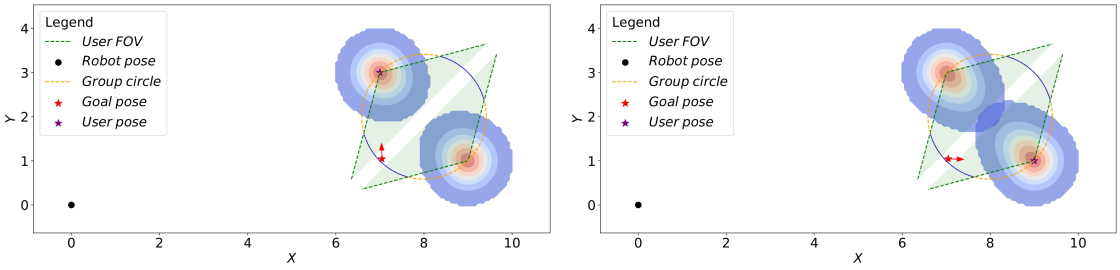
- **reach**: based on the environmental and social cues, the regulation in the reaching behavior of the robot towards a specific user is assessed;
- **follow-accompany**: the capacity of the robot to infer the correct navigation mode, modulating it based on sequential human requests is demonstrated.

Finally, to further evaluate the performance of the high-level reasoning layer, we assessed the system's accuracy in determining the correct navigation parameters. It is important to note that the robot's maximum speed and the social zone distance are empirically constrained, as shown in Tab. 4.3. These constraints help prevent potential hallucinations from the LLM engine, ensuring user safety.

[6]DOI: https://doi.org/10.5281/zenodo.15075165

Table 4.4: Accuracy of the LLM output parameters in 100 different requests

Parameter	Navigation Mode	Max Speed	Social Distance
Accuracy	87%	76%	85%



(a) **(b)**

Figure 4.21: Goal pose estimation for the two cases in the reach scenario assuming $d_l, d_s, d_r = 0.8\ m$ and $l_r = 0.5\ m$. In (a), for the fast user case, the social zone distance is set to $1.0\ m$. In (b), for the slow user case, the social zone distance is set to $1.4\ m$. The purple star represents the user pose, while the red star indicates the robot goal pose, with the required robot orientation shown by the red arrow.

Reach scenario In this scenario, two different user inputs are considered: "*Hello I am User1. I am a young man and I am happy to see you. Please, reach me*" and "*Hello I am User2. I am 86 years old and I have not confidence with robotics. Please, reach me*". Following these requests, for the first user the LLM provides the following parameters: {*navigation mode* : *reach*, *robot maximum speed* : 0.7, *social zone distance* : 1.0, *goal pose* : $[0.0, 0.0]$, *human pose* : $[7.0, 3.0]$}. The justification for the robot maximum speed value was: "*Based on your category as a young user, the robot can move at a faster speed, prioritizing efficiency and responsiveness*". For the social zone distance, the LLM explained: "*User1 is confident, relaxed, open, calm, secure, reassured*". Regarding the human pose, it is assumed to be known and it is used within the *check_goal_pose* block in order to get the best goal pose. Initially, the LLM sets this goal pose to a null value to reach the user. Figure 4.21a illustrates this process, showing the robot's current position, the users in the environment along with their social zones, and the selected goal of the robot. In particular, using the procedure in [221], the robot goal pose (red star) is determined as a position in the space that falls within the FOV of the users (dashed green lines) and is outside their human social interaction space (blue area).

Similarly, for the *User2*, the LLM provided the following parameters: {*navigation mode* : *reach*, *robot maximum speed* : 0.5, *social zone distance* : 1.4, *goal pose* : $[0.0, 0.0]$, *human pose* : $[9.0, 1.0]$}. Figs. 4.22 present the velocity plots for the two scenarios, while Figs. 4.21 show the outputs of *check_goal_pose*, where the social zone distance values are used to determine the goal poses.

Follow-accompany scenario In this test case, the user requests the robot to follow them to a specific location (i.e., the black door in Fig. 4.19). The user is *User3* and request specifically the robot to follow them slowly.

In this context, the resulting parameters are: {*navigation mode* : *follow*, *robot*

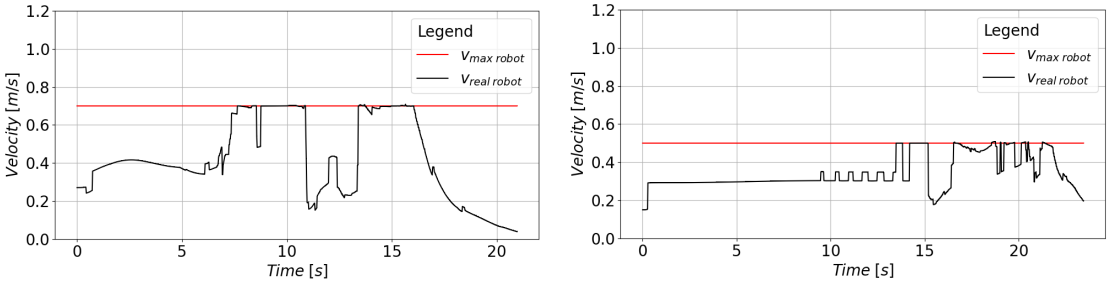


Figure 4.22: Velocity plots in the reach scenario. The robot's speed is shown in black, while its maximum speed is shown in red. In (a), for the fast user case, the robot maximum speed is set to 0.7 m/s. In (b), for the slow user case, the robot maximum speed is set to 0.5 m/s.

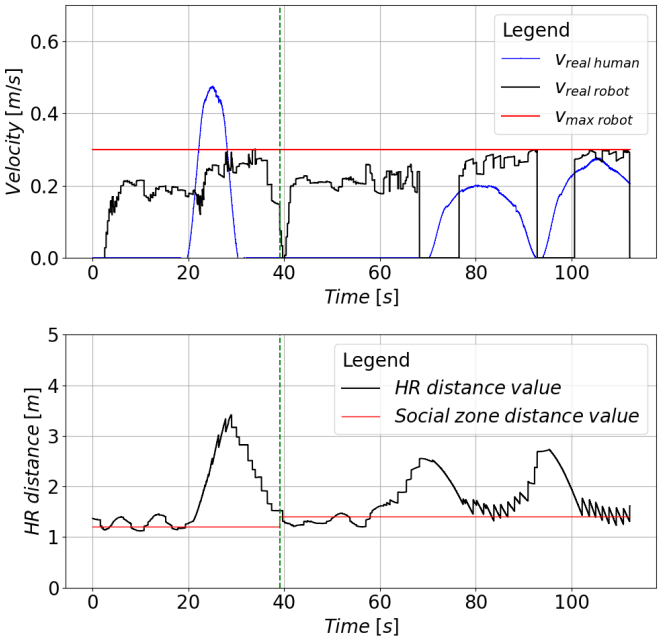


Figure 4.23: In the follow-accompany scenario, the top plot shows the robot's speed in black, and the robot maximum speed in red. The second plot depicts the distance between the user and the robot in black, and the social zone distance in red. The dashed vertical green line separates the follow navigation mode (left) from the accompany mode (right).

maximum speed : 0.3, *social zone distance* : 1.2, *goal pose* : $[10.0, -2.0]$, *human pose* : $[0.0, 0.0]\}$. The justification for the social zone distance value was: "*User's state is unclear, unspecified or unknown*". The human pose in the follow scenario is assumed to be known.

Following an event, the user experiences discomfort by the presence of the robot behind them and prefers to be accompanied by it that will drive the navigation. At this point, the resulting parameters become: $\{$*navigation mode* : *accompany*, *robot maximum speed* : 0.3, *social zone distance* : 1.4, *goal pose* : $[10.0, -2.0]$, *human pose* : $[0.0, 0.0]\}$. The navigation mode is updated and the social zone distance is modified with the following justification: "*The user feels uncomfortable, so a larger distance is appropriate*", while the values of goal pose and robot maximum speed are retained from the previous context. Finally, Fig. 4.23 presents the velocity and the distance between the most engaged user and the robot in the follow-accompany scenario.

Performance evaluation To validate the proper functioning of the high-level reasoning layer, different queries not involving the low-level layer of the architecture have been performed. In particular, 100 different requests were sent, combining the three types of navigation, specifying the user emotion and categories. Examples of these requests are variation of the following *"Hi, I am Mia. I am using a wheelchair. Please adjust accordingly your speed and accompany me to the kitchen"*, *"Hey, I am Dylan. Stay far to me while following me to the workshop"* and similar. With the execution of these queries we evaluated the number of correct parameters generation compatible with the provided context. In particular, results have been manually checked to assess their correct feasibility. Results of this test are reported in Tab. 4.4, where the accuracy (i.e., total correct responses over all the requests) is reported for the three parameters, obtaining a mean of the accuracy of 82%, demonstrating a satisfactory response from the LLM engine.

4.3 Outlook

The studies presented in this chapter have addressed the problem of enabling socially acceptable navigation in human-shared environments from both a control-theoretic and a reasoning perspective.

The first part introduced a rigorous framework based on the combination of CBFs and STL, which provides formal safety and spatio-temporal guarantees for mobile robots. Through this formulation, navigation tasks can be completed while respecting velocity constraints, ensuring obstacle avoidance, and maintaining compliance with temporal specifications. The proposed controller proved effective in simulation and real-world experiments, demonstrating that safety-critical guarantees can be achieved even under noisy localization and dynamic environments. However, while this approach ensures mathematically verifiable behavior, it remains dependent on predefined constraints and lacks contextual awareness beyond the encoded specifications.

Building upon this foundation, the second part of the chapter introduced the concept of intelligent social navigation, where contextual and affective information are integrated into the navigation loop through LLMs. In this framework, high-level reasoning is used to interpret social cues, user characteristics, and environmental conditions, which are then translated into low-level constraints enforced by the CBF-based controller. This architecture enables the robot to adapt its behavior dynamically according to the perceived social context.

As future work, a key direction will be to close the loop within the LLM-based reasoning process itself. At present, the reasoning module is activated only upon explicit input from the user. In future implementations, this loop could become autonomous, reactivating the LLM after a fixed time interval or when a significant change occurs in the environment or in the user's internal state, such as a variation in emotional condition or engagement level detected through multimodal sensing. This would enable continuous monitoring and adaptation, allowing the robot to proactively update its understanding of the social context and adjust its navigation parameters accordingly, ultimately moving toward a closed-loop socially intelligent navigation system.

Chapter 5

Main Findings and Future Directions

In this thesis, the main focus was to investigate how the concept of **social acceptance in robotics** can be realized through two interconnected pillars: *user understanding* and *social navigation*. By combining human-centered design, physiological sensing, and control theory, this work addressed social acceptance as a **systemic property**, one that emerges from the interplay between perception, cognition, and action in human-robot interaction.

By enabling robots to understand users through classification and biosignal analysis, and to navigate socially through CBF-based STL motion planning and LLM reasoning, this thesis laid the foundations for robotic systems that are not only effective, but also trusted, accepted, and embraced by people in real-world contexts.

5.1 Main Findings

Throughout this thesis, several key contributions have been achieved, spanning theoretical, methodological, and experimental perspectives across the two main research pillars: *user understanding*, and *social navigation*.

In Chapter 2, the concept of **social acceptance in robotics** was examined from a systemic perspective, emphasizing its nature as a multi-layer property that combines functional, behavioral, and socio-ethical dimensions of design. A systematic analysis of requirements [7] revealed that social acceptance depends on several complementary factors, each influencing how users perceive trust, safety, and transparency in robotic systems. Building upon this analysis, an **architecture for socially acceptable systems** was proposed in [9]. This modular architecture integrates detection, reasoning, and interaction modules designed to guarantee ethical, transparent, and trustworthy behavior. It supports distributed implementation of social requirements across hardware and software layers, ensuring compliance with human-centered regulations and promoting scalability for different application domains.

Chapter 3 focused on **user understanding** as a key enabling mechanism for social acceptance. The presented studies investigated how robots can infer the user's mental and emotional state by analyzing multimodal physiological signals. Through the integration of EEG, ECG, GSR, and facial expression analysis [3, 13, 14], this work demonstrated that psychophysiological indices can reliably estimate mental workload, stress, and engagement during collaborative robot programming tasks. The findings highlighted that variations in *Alpha*, *Beta*, and *Theta* EEG bands, together with heart rate variability and electrodermal features, serve as strong predictors of cognitive effort. These results have demonstrated that user understanding is not merely an auxiliary function, as it can enable robots to adapt their behavior based on real-time perception of the user's cognitive and affective state.

Lastly, in Chapter 4, the focus shifted from perception to control, addressing **social navigation** as the ability of robots to move safely in human-shared environments. To this end, a motion planning framework based on CBFs and STL [1, 6, 10, 11] was developed to ensure the satisfaction of spatial and temporal constraints under safety-critical conditions. This framework enables robots to reach the target destination in respect of collision avoidance, velocity regulation, and field-of-view constraint, even in uncertain or dynamic scenarios. Further extensions introduced **LLMs** into the control architecture [5]. The resulting system combined high-level reasoning with low-level control, representing a significant step toward the realization of **socially intelligent navigation**, where robots are capable of both understanding and adhering to human social norms while maintaining technical reliability and safety.

5.2 Future Directions

While the methodologies proposed in this thesis have established a framework connecting perception, cognition, and control for socially acceptable robotics, several research directions remain open for future exploration.

First, the biosignals analyzed in controlled experiments could be incorporated into experiments involving physical robots operating in social contexts. Embedding wearable or contactless physiological sensing into the interaction loop would allow the robot to continuously monitor the user's cognitive and emotional state, allowing dynamic adaptation of speed, distance, or dialogue strategies.

A second direction concerns the combination of biosignal processing with LLMs. By embedding physiological features into the input space of an LLM, it will be possible to create reasoning systems that interpret both linguistic and physiological cues. This multimodal representation could enable the robot to infer not only the semantic content of the user's verbal expression, but also their affective and cognitive state, thereby facilitating empathetic and context-sensitive responses.

Lastly, another possible direction is to close the loop between perception, reasoning, and control. The information derived from the physiological and behavioral data of the user will serve as feedback to the LLM reasoning layer, which in turn will update

the low-level control policy. This closed-loop interaction between user understanding, language-based reasoning, and robot control will represent a major step toward truly adaptive and socially acceptable robotic systems.

Bibliography

[1] A. Ruo, L. Sabattini, and V. Villani, "Deployment of a CBF-based STL social navigation framework on omnidirectional mobile robots," in *Submitted to: Robotics and Automation Practice (RA-P)*, 2025.

[2] A. Ruo, L. Bernardi, L. Campanelli, M. Grespan, D. Trane, R. Sedoni, D. Angeli, L. Sabattini, and V. Villani, "A low-cost 3D printed electromagnetic gripper for robotic arms," *Mechatronics*, vol. 110, p. 103374, 2025.

[3] S. Borghi, A. Ruo, L. Sabattini, M. Peruzzini, and V. Villani, "Assessing operator stress in collaborative robotics: A multimodal approach," *Applied Ergonomics*, vol. 123, p. 104418, 2025.

[4] A. Ruo, D. Napolitano, and V. Villani, "A wearable stereo vision-based obstacle detection system for visually impaired individuals," in *Workshop on Agents and Robots for reliable Engineered Autonomy*. Springer, 2025, pp. 35–50.

[5] A. Ruo, J. Cacace, M. Dalmau-Moreno, L. Sabattini, and V. Villani, "An LLM-based architecture for socially intelligent robot navigation based on social cues," in *2025 34th IEEE international conference on robot and human interactive communication (RO-MAN)*. IEEE, 2025.

[6] A. Ruo, L. Sabattini, and V. Villani, "Robust CBF-based STL motion planning for socially responsible robot navigation in the presence of measurement noise," in *2025 American Control Conference (ACC)*, 2025.

[7] A. Ruo, S. Arreghini, L. Capra, R. De Chiara, V. Di Pasquale, A. Giusti, C. Iani, A. Paolillo, D. Petrak, A. Plaum, M. Quamara, L. Sabattini, V. Schmuck, P. Servillo, F. Zurolo, and V. Villani, "Systematic analysis of requirements for socially acceptable service robots," in *17th International Workshop on Human-Friendly Robotics*, 2024.

[8] A. Coppari, S. Proia, A. Ruo, F. Favali, L. Sabattini, C. Secchi, V. Villani, M. Piazzola, and L. Capra, "A large language model-based motion planning for human-robot interaction: An experimental case study," in *17th International Workshop on Human-Friendly Robotics*, 2024.

[9] M. Quamara, L. Capra, V. Villani, C. Carlevaro, O. Celiktutan, A. Peretti, M. Piazzola, A. Ruo, L. Sabattini, V. Schmuck, *et al.*, "Towards a modular architecture for extended reality systems," in *8th International Conference on Artificial Intelligence and Virtual Reality (2024)*. Springer, 2024.

[10] A. Ruo, L. Sabattini, and V. Villani, "CBF-based motion planning for socially responsible robot navigation guaranteeing STL specification," in *2024 European Control Conference (ECC)*, 2024, pp. 122–127.

[11] ——, "CBF-based STL motion planning for social navigation in crowded environment," in *European Robotic Forum*, 2024.

[12] ——, "Follow me: an architecture for user identification and social navigation with a mobile robot." in *European Robotic Forum*, 2024.

[13] S. Borghi, F. Zucchi, E. Prati, A. Ruo, V. Villani, L. Sabattini, and M. Peruzzini, "Unlocking human-robot dynamics: Introducing sensecobot, a novel multimodal dataset on industry 4.0," in *2024 19th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2024, pp. 880–884.

[14] A. Ruo, V. Villani, and L. Sabattini, "Use of EEG signals for mental workload assessment in human-robot collaboration," in *Human-Friendly Robotics 2022*, P. Borja, C. Della Santina, L. Peternel, and E. Torta, Eds. Cham: Springer International Publishing, 2023, pp. 233–247.

[15] SERMAS Consortium, "Socially-Acceptable Extended Reality Models and Systems (SERMAS)," https://sermasproject.eu/, 2025, Project website, Accessed: October 2025.

[16] N. Savela, T. Turja, and A. Oksanen, "Social acceptance of robots in different occupational fields: a systematic literature review," *International Journal of Social Robotics*, vol. 10, no. 4, pp. 493–502, 2018.

[17] J. Holland, L. Kingston, C. McCarthy, E. Armstrong, P. O'Dwyer, F. Merz, and M. McConnell, "Service robots in the healthcare sector," *Robotics*, vol. 10, no. 1, p. 47, 2021.

[18] B. G. Draghici, A. E. Dobre, M. Misaros, and O. P. Stan, "Development of a human service robot application using pepper robot as a museum guide," in *2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*. IEEE, 2022, pp. 1–5.

[19] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Field and Service Robotics: Results of the 10th International Conference*. Springer, 2016.

[20] R. Stricker, S. Müller, E. Einhorn, C. Schröter, M. Volkhardt, K. Debes, and H. Gross, "Interactive mobile robots guiding visitors in a university building," in *IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012.

[21] M. E. Foster, B. Craenen, A. Deshmukh, O. Lemon, E. Bastianelli, C. Dondrup, I. Papaioannou, A. Vanzo, J.-M. Odobez, O. Canévet, *et al.*, "Mummer: Socially intelligent human-robot interaction in public spaces," *arXiv preprint arXiv:1909.06749*, 2019.

[22] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[23] A. V. Savkin, A. S. Matveev, M. Hoy, and C. Wang, *Safe robot navigation among moving and steady obstacles*. Butterworth-Heinemann, 2015.

[24] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and autonomous systems*, vol. 42, no. 3-4, pp. 143–166, 2003.

[25] S. Arreghini, G. Abbate, A. Giusti, and A. Paolillo, "A service robot in the wild: Analysis of users intentions, robot behaviors, and their impact on the interaction," in *IEEE/RSJ Int. Conf. Intelligent Robots Sys*, 2024.

[26] G. Abbate, A. Giusti, V. Schmuck, O. Celiktutan, and A. Paolillo, "Self-supervised prediction of the intention to interact with a service robot," *Robotics and Autonomous Systems*, vol. 171, p. 104568, 2024.

[27] S. Arreghini, G. Abbate, A. Giusti, and A. Paolillo, "A long-range mutual gaze detector for hri," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 2024, pp. 870–874.

[28] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS quarterly*, pp. 319–340, 1989.

[29] J. Wirtz, P. G. Patterson, W. H. Kunz, T. Gruber, V. N. Lu, S. Paluch, and A. Martins, "Brave new world: service robots in the frontline," *Journal of Service Management*, vol. 29, no. 5, pp. 907–931, 2018.

[30] A.-H. Chiang and S. Trimi, "Impacts of service robots on service quality," *Service Business*, vol. 14, no. 3, pp. 439–459, 2020.

[31] Y. Liu, X. Wang, Y. Du, and S. Wang, "Service robots vs. human staff: The effect of service agents and service exclusion on unethical consumer behavior," *Journal of Hospitality and Tourism Management*, vol. 55, pp. 401–415, 2023.

[32] A. Çöltekin, I. Lochhead, M. Madden, S. Christophe, A. Devaux, C. Pettit, O. Lock, S. Shukla, L. Herman, Z. Stachoň, *et al.*, "Extended reality in spatial sciences: A review of research challenges and future directions," *ISPRS International Journal of Geo-Information*, vol. 9, no. 7, p. 439, 2020.

[33] C. Andrews, M. K. Southworth, J. N. Silva, and J. R. Silva, "Extended reality in medical practice," *Current treatment options in cardiovascular medicine*, vol. 21, pp. 1–12, 2019.

[34] L. A. Cardenas-Robledo, Ó. Hernández-Uribe, C. Reta, and J. A. Cantoral-Ceballos, "Extended reality applications in industry 4.0.-a systematic literature review," *Telematics and Informatics*, p. 101863, 2022.

[35] T. Andrade and D. Bastos, "Extended reality in iot scenarios: Concepts, applications and future trends," in *2019 5th Experiment International Conference (exp. at'19)*. IEEE, 2019, pp. 107–112.

[36] S. Doolani, C. Wessels, V. Kanal, C. Sevastopoulos, A. Jaiswal, H. Nambiappan, and F. Makedon, "A review of extended reality (xr) technologies for manufacturing training," *Technologies*, vol. 8, no. 4, p. 77, 2020.

[37] T. M. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and M. Vilar-Montesinos, "A fog computing and cloudlet based augmented reality system for the industry 4.0 shipyard," *Sensors*, vol. 18, no. 6, p. 1798, 2018.

[38] F. G. Hamza-Lup, C. Hughes, and J. P. Rolland, "Sensors in distributed mixed reality environments," *arXiv preprint arXiv:1811.11955*, 2018.

[39] S. Lee, G. Lee, G. Choi, B.-h. Roh, and J. Kang, "Integration of onem2m-based iot service platform and mixed reality device," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–4.

[40] M. Serras, L. García-Sardiña, B. Simões, H. Álvarez, and J. Arambarri, "Dialogue enhanced extended reality: Interactive system for the operator 4.0," *Applied Sciences*, vol. 10, no. 11, p. 3960, 2020.

[41] J. M. Beer, A. Prakash, T. L. Mitzner, and W. A. Rogers, "Understanding robot acceptance," *Georgia Institute of Technology*, pp. 1–45, 2011.

[42] I. Gaudiello, E. Zibetti, S. Lefort, M. Chetouani, and S. Ivaldi, "Trust as indicator of robot functional and social acceptance. an experimental study on user conformation to icub answers," *Computers in Human Behavior*, vol. 61, pp. 633–655, 2016.

[43] M. Heerink, B. Kröse, V. Evers, and B. Wielinga, "Assessing acceptance of assistive social agent technology by older adults: the almere model," 2010.

[44] T. Ogino, Y. Matsuda, D. Kawabata, K. Yamazaki, A. Kimura, F. Shibata, *et al.*, "A distributed framework for creating mobile mixed reality systems," in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*. IEEE, 2014, pp. 327–331.

[45] N. Pereira, A. Rowe, M. W. Farb, I. Liang, E. Lu, and E. Riebling, "Arena: The augmented reality edge networking architecture," in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2021, pp. 479–488.

[46] R. T. Fielding, "Rest: architectural styles and the design of network-based software architectures," *Doctoral dissertation, University of California*, 2000.

[47] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM computing surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003.

[48] Swagger Community, "Swagger," https://swagger.io/, 2011, accessed: October 2025.

[49] Openapis, "Openapis.org," https://www.openapis.org, accessed: October 2025.

[50] Asyncapi, "Asyncapi.com," https://www.asyncapi.com, accessed: October 2025.

[51] M. Jones, J. Bradley, and N. Sakimura, "Json web token (jwt)," Internet Engineering Task Force (IETF), Tech. Rep., 2015.

[52] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture (no. nist special publication (sp) 800–207)," *National Institute of Standards and Technology*, 2020.

[53] Justadudewhohacks, "Faceapi," https://github.com/justadudewhohacks/face-api.js, 2014, accessed: October 2025.

[54] S. Team, "Speechbrain," https://speechbrain.github.io/, 2022, accessed: October 2025.

[55] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[56] Ros2, "Ros - robot operating system," https://www.ros.org/, 2023, accessed: October 2025.

[57] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *Electrical Engineering and Systems Science*, 2022.

[58] Google, "Speech-to-text," https://cloud.google.com/speech-to-text, 2023, accessed: October 2025.

[59] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, L. Zhao, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, and B. Ge, "Summary of chatgpt-related research and perspective towards the future of large language models," *Meta-Radiology (2023)100017*, 2023.

[60] Google, "Model card for flan-t5 xxl," https://huggingface.co/google/flan-t5-xxl, 2023, accessed: October 2025.

[61] M. I. Ahmad, O. Mubin, and J. Orlando, "A systematic review of adaptivity in human-robot interaction," *Multimodal Technologies and Interaction*, vol. 1, no. 3, p. 14, 2017.

[62] G. Giannakakis, D. Grigoriadis, K. Giannakaki, O. Simantiraki, A. Roniotis, and M. Tsiknakis, "Review on psychological stress detection using biosignals," *IEEE transactions on affective computing*, vol. 13, no. 1, pp. 440–460, 2019.

[63] R. A. Bryant, M. J. Friedman, D. Spiegel, R. Ursano, and J. Strain, "A review of acute stress disorder in dsm-5," *Focus*, vol. 9, no. 3, pp. 335–350, 2011.

[64] G. Giannakakis, D. Grigoriadis, K. Giannakaki, O. Simantiraki, A. Roniotis, and M. Tsiknakis, "Review on psychological stress detection using biosignals," *IEEE Transactions on Affective Computing*, vol. PP, pp. 1–1, 07 2019.

[65] D. Andersson, "Real-time ecg for objective stress level measurement," 2017.

[66] A. B. De Luna, *Basic electrocardiography: normal and abnormal ECG patterns*. John Wiley & Sons, 2008.

[67] T. F. o. t. E. S. o. C. t. N. A. S. o. P. Electrophysiology, "Heart rate variability: standards of measurement, physiological interpretation, and clinical use," *Circulation*, vol. 93, no. 5, pp. 1043–1065, 1996.

[68] W. Boucsein, *Electrodermal activity*. Springer Science & Business Media, 2012.

[69] Z. Zou and S. Ergan, "Evaluating the effectiveness of biometric sensors and their signal features for classifying human experience in virtual environments," *Advanced Engineering Informatics*, vol. 49, p. 101358, 2021.

[70] R. Hoehn-Saric, D. R. McLeod, and W. D. Zimmerli, "Somatic manifestations in women with generalized anxiety disorder: Psychophysiological responses to psychological stress," *Archives of general psychiatry*, vol. 46, no. 12, pp. 1113–1119, 1989.

[71] G. Acerbi, E. Rovini, S. Betti, A. Tirri, J. F. Rónai, A. Sirianni, J. Agrimi, L. Eusebi, and F. Cavallo, "A wearable system for stress detection through physiological data analysis," in *Ambient Assisted Living: Italian Forum 2016 7*. Springer, 2017, pp. 31–50.

[72] N. A. Zulkurnaini, R. S. S. A. Kadir, Z. H. Murat, and R. M. Isa, "The comparison between listening to al-quran and listening to classical music on the brainwave signal for the alpha band," in *2012 Third International Conference on Intelligent Systems Modelling and Simulation*. IEEE, 2012, pp. 181–186.

[73] G. Buzsaki, *Rhythms of the Brain*. Oxford university press, 2006.

[74] S. Chandra, A. K. Jaiswal, R. Singh, D. Jha, and A. P. Mittal, "Mental stress: Neurophysiology and its regulation by sudarshan kriya yoga," *International journal of yoga*, vol. 10, no. 2, p. 67, 2017.

[75] G. Xavier, A. Su Ting, and N. Fauzan, "Exploratory study of brain waves and corresponding brain regions of fatigue on-call doctors using quantitative electroencephalogram," *Journal of occupational health*, vol. 62, no. 1, p. e12121, 2020.

[76] G. Teo, G. Matthews, L. Reinerman-Jones, and D. Barber, "Adaptive aiding with an individualized workload model based on psychophysiological measures," *Human-Intelligent Systems Integration*, vol. 2, pp. 1–15, 2020.

[77] T. Hayashi, E. Okamoto, H. Nishimura, Y. Mizuno-Matsumoto, R. Ishii, and S. Ukai, "Beta activities in eeg associated with emotional stress," *International Journal of Intelligent Computing in Medical Sciences & Image Processing*, vol. 3, no. 1, pp. 57–68, 2009.

[78] L. M. Gatzke-Kopp, M. K. Jetha, and S. J. Segalowitz, "The role of resting frontal eeg asymmetry in psychopathology: Afferent or efferent filter?" *Developmental psychobiology*, vol. 56, no. 1, pp. 73–85, 2014.

[79] G. Berretz, J. Packheiser, O. T. Wolf, and S. Ocklenburg, "Acute stress increases left hemispheric activity measured via changes in frontal alpha asymmetries," *Iscience*, vol. 25, no. 2, 2022.

[80] G. Giannakakis, M. Pediaditis, D. Manousos, E. Kazantzaki, F. Chiarugi, P. G. Simos, K. Marias, and M. Tsiknakis, "Stress and anxiety detection using facial cues from videos," *Biomedical Signal Processing and Control*, vol. 31, pp. 89–101, 2017.

[81] S. G. Hart, "Nasa task load index (tlx)," *Human Performance Research Group NASA Ames Research Center Moffett Field, California*, 1986.

[82] P. S. Tsang and V. L. Velazquez, "Diagnosticity and multidimensional subjective workload ratings," *Ergonomics*, vol. 39, no. 3, pp. 358–381, 1996.

[83] M. A. Vidulich and P. S. Tsang, "Mental workload and situation awareness," *Handbook of human factors and ergonomics*, pp. 243–273, 2012.

[84] J. R. Crawford and J. D. Henry, "The positive and negative affect schedule (panas): Construct validity, measurement properties and normative data in a large non-clinical sample," *British journal of clinical psychology*, vol. 43, no. 3, pp. 245–265, 2004.

[85] S. Cohen, T. Kamarck, R. Mermelstein, *et al.*, "Perceived stress scale," *Measuring stress: A guide for health and social scientists*, vol. 10, no. 2, pp. 1–2, 1994.

[86] T. H. Holmes and R. H. Rahe, "The social readjustment rating scale." *Journal of psychosomatic research*, 1967.

[87] M. A. Mariscal, S. Ortiz Barcina, S. García Herrero, and E. M. López Perea, "Working with collaborative robots and its influence on levels of working stress," *International Journal of Computer Integrated Manufacturing*, pp. 1–20, 2023.

[88] Z. Zakeri, A. Arif, A. Omurtag, P. Breedon, and A. Khalid, "Multimodal assessment of cognitive workload using neural, subjective and behavioural measures in smart factory settings," *Sensors*, vol. 23, no. 21, p. 8926, 2023.

[89] C. Carissoli, L. Negri, M. Bassi, F. A. Storm, and A. Delle Fave, "Mental workload and human-robot interaction in collaborative tasks: A scoping review," *International Journal of Human-Computer Interaction*, pp. 1–20, 2023.

[90] A. Pollak, M. Paliga, M. M. Pulopulos, B. Kozusznik, and M. W. Kozusznik, "Stress in manual and autonomous modes of collaboration with a cobot," *Computers in Human Behavior*, vol. 112, p. 106469, 2020.

[91] A. Bussolan, S. Baraldo, L. M. Gambardella, and A. Valente, "Assessing the impact of human-robot collaboration on stress levels and cognitive load in industrial assembly tasks," in *ISR Europe 2023; 56th International Symposium on Robotics*. VDE, 2023, pp. 78–85.

[92] R. Martinez, A. Salazar-Ramirez, A. Arruti, E. Irigoyen, J. I. Martin, and J. Muguerza, "A self-paced relaxation response detection system based on galvanic skin response analysis," *IEEE Access*, vol. 7, pp. 43 730–43 741, 2019.

[93] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[94] R. Fernandez Rojas, E. Debie, J. Fidock, M. Barlow, K. Kasmarik, S. Anavatti, M. Garratt, and H. Abbass, "Electroencephalographic workload indicators during teleoperation of an unmanned aerial vehicle shepherding a swarm of unmanned ground vehicles in contested environments," *Frontiers in neuroscience*, vol. 14, p. 40, 2020.

[95] A. Holm, K. Lukander, J. Korpela, M. Sallinen, and K. M. Müller, "Estimating brain load from the eeg," *The Scientific World Journal*, vol. 9, no. 1, pp. 639–651, 2009.

[96] J. P. Fuentes-García, S. Villafaina, D. Collado-Mateo, R. Cano-Plasencia, and N. Gusi, "Chess players increase the theta power spectrum when the difficulty of the opponent increases: an eeg study," *International journal of environmental research and public health*, vol. 17, no. 1, p. 46, 2020.

[97] B. Raufi and L. Longo, "An evaluation of the eeg alpha-to-theta and theta-to-alpha band ratios as indexes of mental workload," *Frontiers in Neuroinformatics*, vol. 16, p. 861967, 2022.

[98] A. Hansen, V. Villani, A. Pupa, and A. H. Lassen, "Introducing novice operators to collaborative robots: a hands-on approach for learning and training," *Authorea Preprints*, 2023.

[99] M. Bishay, K. Preston, M. Strafuss, G. Page, J. Turcot, and M. Mavadati, "Affdex 2.0: A real-time facial expression analysis toolkit," *arXiv preprint arXiv:2202.12059*, 2022.

[100] J. Favre-Félix, M. Dziadzko, C. Bauer, A. Duclos, J.-J. Lehot, T. Rimmelé, and M. Lilot, "High-fidelity simulation to assess task load index and performance: a prospective observational study," *Turkish Journal of Anaesthesiology and Reanimation*, vol. 50, no. 4, p. 282, 2022.

[101] F. K. Ghanavati, A. Choobineh, S. Keshavarzi, A. A. Nasihatkon, and A. S. J. Roodbandi, "Assessment of mental workload and its association with work ability in control room operators," *La Medicina del lavoro*, vol. 110, no. 5, p. 389, 2019.

[102] S. Said, M. Gozdzik, T. R. Roche, J. Braun, J. Rössler, A. Kaserer, D. R. Spahn, C. B. Nöthiger, and D. W. Tscholl, "Validation of the raw national aeronautics and space administration task load index (nasa-tlx) questionnaire to assess perceived workload in patient monitoring tasks: pooled analysis study using mixed models," *Journal of medical Internet research*, vol. 22, no. 9, p. e19472, 2020.

[103] M. Memar and A. Mokaribolhassan, "Stress level classification using statistical analysis of skin conductance signal while driving," *SN Applied Sciences*, vol. 3, no. 1, p. 64, 2021.

[104] H. M. Melo, T. C. Martins, L. M. Nascimento, A. A. Hoeller, R. Walz, and E. Takase, "Ultra-short heart rate variability recording reliability: The effect of controlled paced breathing," *Annals of noninvasive electrocardiology*, vol. 23, no. 5, p. e12565, 2018.

[105] S. Hong, W. Zhang, C. Sun, Y. Zhou, and H. Li, "Practical lessons on 12-lead ecg classification: meta-analysis of methods from physionet/computing in cardiology challenge 2020," *Frontiers in Physiology*, vol. 12, p. 2505, 2022.

[106] W. J. Tompkins, "Biomedical digital signal processing," *Editorial Prentice Hall*, 1993.

[107] M. Kassinopoulos, R. M. Harper, M. Guye, L. Lemieux, and B. Diehl, "Altered relationship between heart rate variability and fmri-based functional connectivity in people with epilepsy," *Frontiers in Neurology*, vol. 12, p. 671890, 2021.

[108] P. Mayapur, "A review on detection and performance analysis on rr interval methods for ecg," *Int. J. Innov. Res. Sci. Eng. Technol*, vol. 7, pp. 11 019–11 025, 2018.

[109] R. E. Kleiger, P. K. Stein, and J. T. Bigger Jr, "Heart rate variability: measurement and clinical utility," *Annals of Noninvasive Electrocardiology*, vol. 10, no. 1, pp. 88–101, 2005.

[110] F. Shaffer and J. P. Ginsberg, "An overview of heart rate variability metrics and norms," *Frontiers in public health*, vol. 5, p. 258, 2017.

[111] V. Villani, M. Righi, L. Sabattini, and C. Secchi, "Wearable devices for the assessment of cognitive effort for human–robot interaction," *IEEE Sensors Journal*, vol. 20, no. 21, pp. 13 047–13 056, 2020.

[112] S. A. Hossein Aqajari, E. K. Naeini, M. A. Mehrabadi, S. Labbaf, N. Dutt, and A. M. Rahmani, "pyeda: An open-source python toolkit for pre-processing and feature extraction of electrodermal activity," *Procedia Computer Science*, vol. 184, pp. 99–106, 2021, the 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[113] I. Karpiel, Z. Kurasz, R. Kurasz, and K. Duch, "The influence of filters on eeg-erp testing: Analysis of motor cortex in healthy subjects," *Sensors*, vol. 21, no. 22, p. 7711, 2021.

[114] T. Alotaiby, F. E. A. El-Samie, S. A. Alshebeili, and I. Ahmad, "A review of channel selection algorithms for eeg signal processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, pp. 1–21, 2015.

[115] A. Hyvärinen, "Independent component analysis: recent advances," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20110534, 2013.

[116] S. Huberty, V. Carter Leno, S. J. van Noordt, R. Bedford, A. Pickles, J. A. Desjardins, S. J. Webb, B. Team, and M. Elsabbagh, "Association between spectral electroencephalography power and autism risk and diagnosis in early development," *Autism Research*, vol. 14, no. 7, pp. 1390–1403, 2021.

[117] B. Raufi and L. Longo, "An evaluation of the eeg alpha-to-theta and theta-to-alpha band ratios as indexes of mental workload," *Frontiers in Neuroinformatics*, vol. 16, p. 44, 2022.

[118] J. P. Fuentes-García, S. Villafaina, D. Collado-Mateo, R. Cano-Plasencia, and N. Gusi, "Chess players increase the theta power spectrum when the difficulty of the opponent increases: an eeg study," *International journal of environmental research and public health*, vol. 17, no. 1, p. 46, 2020.

[119] A. Holm, K. Lukander, J. Korpela, M. Sallinen, and K. M. Müller, "Estimating brain load from the eeg," *TheScientificWorldJOURNAL*, vol. 9, pp. 639–651, 2009.

[120] U. R. Acharya, S. V. Sree, P. C. A. Ang, R. Yanti, and J. S. Suri, "Application of non-linear and wavelet based features for the automated identification of epileptic eeg signals," *International journal of neural systems*, vol. 22, no. 02, p. 1250002, 2012.

[121] A. Ghasemi and S. Zahediasl, "Normality tests for statistical analysis: a guide for non-statisticians," *International journal of endocrinology and metabolism*, vol. 10, no. 2, p. 486, 2012.

[122] M. P. Fay and M. A. Proschan, "Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules," *Statistics surveys*, vol. 4, p. 1, 2010.

[123] G. W. Corder and D. I. Foreman, "Nonparametric statistics for non-statisticians," 2011.

[124] A. Dinno, "Nonparametric pairwise multiple comparisons in independent groups using dunn's test," *The Stata Journal*, vol. 15, no. 1, pp. 292–300, 2015.

[125] R. Warne, "A primer on multivariate analysis of variance (manova) for behavioral scientists," *Practical Assessment, Research, and Evaluation*, vol. 19, no. 1, 2019.

[126] M. Aslam and F. Smarandache, *Chi-square test for imprecise data in consistency table*. Infinite Study, 2023.

[127] M. Aslam and M. Albassam, "Presenting post hoc multiple comparison tests under neutrosophic statistics," *Journal of King Saud University-Science*, vol. 32, no. 6, pp. 2728–2732, 2020.

[128] T. Föhr, A. Tolvanen, T. Myllymäki, E. Järvelä-Reijonen, S. Rantala, R. Korpela, K. Peuhkuri, M. Kolehmainen, S. Puttonen, R. Lappalainen, *et al.*, "Subjective stress, objective heart rate variability-based stress, and recovery on workdays among overweight and psychologically distressed individuals: a cross-sectional study," *Journal of Occupational Medicine and Toxicology*, vol. 10, pp. 1–9, 2015.

[129] B. Zhang, Y. Xie, J. Zhou, K. Wang, and Z. Zhang, "State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review," *Computers and Electronics in Agriculture*, vol. 177, p. 105694, 2020.

[130] Z. Samadikhoshkho, K. Zareinia, and F. Janabi-Sharifi, "A brief review on robotic grippers classifications," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, 2019, pp. 1–4.

[131] Z. Wang, D. S. Chathuranga, and S. Hirai, "3D printed soft gripper for automatic lunch box packing," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016, pp. 503–508.

[132] G. L. Goh, G. D. Goh, V. P. Nguyen, W. Toh, S. Lee, X. Li, B. D. Sunil, J. Y. Lim, Z. Li, A. K. Sinha, *et al.*, "A 3D printing-enabled artificially innervated smart soft gripper with variable joint stiffness," *Advanced Materials Technologies*, vol. 8, no. 24, p. 2301426, 2023.

[133] C. Tawk, Y. Gao, R. Mutlu, and G. Alici, "Fully 3D printed monolithic soft gripper with high conformal grasping capability," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2019, pp. 1139–1144.

[134] M. Dragusanu, S. Marullo, M. Malvezzi, G. M. Achilli, M. C. Valigi, D. Prattichizzo, and G. Salvietti, "The DressGripper: A collaborative gripper with electromagnetic fingertips for dressing assistance," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7479–7486, 2022.

[135] L. Birglen and T. Schlicht, "A statistical review of industrial robotic grippers," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 88–97, 2018.

[136] Y. Tlegenov, K. Telegenov, and A. Shintemirov, "An open-source 3D printed underactuated robotic gripper," in *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2014, pp. 1–6.

[137] C. Tawk, A. Gillett, M. in het Panhuis, G. M. Spinks, and G. Alici, "A 3D-printed omni-purpose soft gripper," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1268–1275, 2019.

[138] M. Li, F. Zhao, X. Li, M. Li, S. Liu, and M. Li, "Design and fabrication of a novel miniature magnetic gripper," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1964–1970.

[139] R. Li, W. Qin, G. Li, M. Shou, X. Wang, X. Huang, X. Gong, C.-H. Lee, Y. Chen, and P.-a. Yang, "Design and optimization of the magnetic field-driven spherical gripper with adjustable stiffness," *Materials & Design*, vol. 235, p. 112391, 2023.

[140] A. Fort, E. Landi, M. Barbieri, V. Vignoli, and M. Mugnaini, "An electromagnetic gripper for multiple picking of iron laser-cut parts," in *2022 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 2022, pp. 1–6.

[141] Y. Tsugami and T. Nishida, "Simple structured gripper using electromagnet and permanent magnet," in *International Conference on ICT Robotics*, vol. 2017, 2017, pp. 25–26.

[142] K. Telegenov, Y. Tlegenov, and A. Shintemirov, "A low-cost open-source 3-d-printed three-finger gripper platform for research and educational purposes," *IEEE Access*, vol. 3, pp. 638–647, 2015.

[143] P. Mannam, A. Rudich, K. L. Zhang, M. Veloso, O. Kroemer, and Z. Temel, "A low-cost compliant gripper using cooperative mini-delta robots for dexterous manipulation," in *Robotics science and systems*, 2021.

[144] V. H. Sundaram, R. Bhirangi, M. E. Rentschler, A. Gupta, and T. Hellebrekers, "Dragonclaw: A low-cost pneumatic gripper with integrated magnetic sensing," in *2023 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2023, pp. 1–8.

[145] G. Rizzoni, A. Andreotti, L. Verolino, *et al.*, *Elettrotecnica - Principi e applicazioni*. McGraw-Hill, 2004.

[146] G. Ortolani and E. Venturi, *Manuale di elettrotecnica elettronica e automazione*. Hoepli, 2017.

[147] U. S. A. Department of Energy, *Thermodynamics, Heat Transfer, and Fluid Flow*, 1992.

[148] Universal Robots, "Universal robots e-series user manual," https://www.universal-robots.com/download/manuals-e-seriesur20ur30/user/ur10e/59/user-manual-ur10e-e-series-sw-59-english-international-en/, 2021, accessed: October 2025.

[149] The Engineering ToolBox, "Overall heat transfer coefficient: Definitions, values & applications," https://www.engineeringtoolbox.com/overall-heat-transfer-coefficient-d_434.html, 2003, accessed: October 2025.

[150] ASTM: Advancing Standards Transforming Markets, *ASTM B258-18: Standard Specification for Standard Nominal Diameters and Cross-Sectional Areas of AWG Sizes of Solid Round Wires Used as Electrical Conductors*. ASTM International West Conshohocken, PA, 2018.

[151] A. E. Fitzgerald, C. Kingsley, and S. D. Umans, *Electric machinery*. McGraw-Hill, 2003.

[152] D. Ni, A. Song, L. Tian, X. Xu, and D. Chen, "A walking assistant robotic system for the visually impaired based on computer vision and tactile perception," *Int. Journal of Social Robot.*, 2015.

[153] T. Saba, "Computer vision for microscopic skin cancer diagnosis using handcrafted and non-handcrafted features," *Microscopy Res. and Tech.*, 2021.

[154] V. Nguyen, H. Kim, S. Jun, and K. Boo, "A study on real-time detection method of lane and vehicle for lane change assistant system using vision system on highway," *Eng. science and technol., an Int. journal*, 2018.

[155] G. Granlund and H. Knutsson, *Signal processing for computer vision*. Springer Science & Business Media, 2013.

[156] C. Che, H. Zheng, Z. Huang, W. Jiang, and B. Liu, "Intelligent robotic control system based on computer vision technology," *arXiv preprint arXiv:2404.01116*, 2024.

[157] K. J. Singh, D. S. Kapoor, K. Thakur, A. Sharma, and X. Gao, "Computer-vision based object detection and recognition for service robot in indoor environment." *Comput., Mater. & Continua*, 2022.

[158] C. Danu, R. Chauhan, C. Bhatt, S. Devliyal, and S. Gupta, "Structered deep visual models for robot manipulation," in *Int. Conf. on Intell. and Innovative Technol. in Comput., Elect. and Electron.* IEEE, 2024.

[159] D. D. Brilli, E. Georgaras, S. Tsilivaki, N. Melanitis, and K. Nikita, "Airis: An ai-powered wearable assistive device for the visually impaired," *arXiv preprint arXiv:2405.07606*, 2024.

[160] W. Khan, A. Hussain, B. N. Khan, and K. Crockett, "Outdoor mobility aid for people with visual impairment: Obstacle detection and responsive framework for the scene perception during the outdoor mobility of people with visual impairment," *Expert Syst. with Appl.*, 2023.

[161] K. Yang, K. Wang, S. Lin, J. Bai, L. Bergasa, and R. Arroyo, "Long-range traversability awareness and low-lying obstacle negotiation with realsense for the visually impaired," in *1st Int. Conf. on Inf. Science and Syst.*, 2018.

[162] A. Datta, A. Sarkar, A. Bhattashali, A. Meruva, and V. Manikandan, "Enhanced environmental perception for visually impaired: A real-time object detection and distance estimation approach," in *IEEE Int. Conf. on Interdisciplinary Approaches in Technol. and Management for Social Innovation*. IEEE, 2024.

[163] K. Shahira, S. Tripathy, and A. Lijiya, "Obstacle detection, depth estimation and warning system for visually impaired people," in *TENCON IEEE Region 10 Conf.* IEEE, 2019.

[164] K. Y. Kok and P. Rajendran, "A review on stereo vision algorithm: Challenges and solutions," *ECTI Transactions on Comput. and Inf. Technol.*, 2019.

[165] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

[166] M. Tan, R. Pang, and Q. Le, "Efficientdet: Scalable and efficient object detection," in *IEEE/CVF Conf. on comput. vision and pattern recog.*, 2020.

[167] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Int. Conf. on Mach. Learning*. PMLR, 2019.

[168] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sept. 2020.

[169] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[170] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[171] Intel RealSense SDK Developers, "pyrealsense2 python wrapper for intel realsense sdk 2.0," https://pypi.org/project/pyrealsense2/, 2025, accessed: October 2025.

[172] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra, *et al.*, "Principles and guidelines for evaluating social robot navigation algorithms," *ACM Transactions on Human-Robot Interaction*, vol. 14, no. 2, pp. 1–65, 2025.

[173] K. Song, Y. Chiu, L. Kang, S. Song, C. Yang, P. Lu, and S. Ou, "Navigation control design of a mobile robot by integrating obstacle avoidance and lidar slam," in *Int. Conf. on Syst., Man, and Cybern.* IEEE, 2018.

[174] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra, *et al.*, "Principles and guidelines for evaluating social robot navigation algorithms," *arXiv preprint arXiv:2306.16740*, 2023.

[175] F. Rollo, A. Zunino, G. Raiola, F. Amadio, A. Ajoudani, and N. Tsagarakis, "Followme: a robust person following framework based on visual re-identification and gestures," in *Int. Conf. on Adv. Robot. and Its Social Impacts (ARSO)*. IEEE, 2023.

[176] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 500–505.

[177] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[178] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.

[179] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2347–2354.

[180] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[181] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[182] S. Guillen-Ruiz, J. P. Bandera, A. Hidalgo-Paniagua, and A. Bandera, "Evolution of socially-aware robot navigation," *Electronics*, vol. 12, no. 7, p. 1570, 2023.

[183] P. Patompak, S. Jeong, N. Y. Chong, and I. Nilkhamhang, "Mobile robot navigation for human-robot social interaction," in *2016 16th international conference on control, automation and systems (ICCAS)*. IEEE, 2016, pp. 1298–1303.

[184] L. Jaillet, J. Cortés, and T. Siméon, "Transition-based rrt for path planning in continuous cost spaces," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2145–2150.

[185] S. Silva, N. Verdezoto, D. Paillacho, S. Millan-Norman, and J. D. Hernández, "Online social robot navigation in indoor, large and crowded environments," in *Int. Conf. on Robot. and Autom.* IEEE, 2023.

[186] F. Ferraguti, C. T. Landi, A. Singletary, H. Lin, A. Ames, C. Secchi, and M. Bonfè, "Safety and efficiency in robotics: The control barrier functions approach," *IEEE Robot. & Autom. Magazine*, 2022.

[187] A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Eur. Control Conf.* IEEE, 2019.

[188] F. Bertoncelli, V. Radhakrishnan, M. Catellani, G. Loianno, and L. Sabattini, "Directed graph topology preservation in multi-robot systems with limited field of view using control barrier functions," *IEEE Access*, 2024.

[189] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *Control Syst. letters*, 2018.

[190] A. Zehfroosh and H. G. Tanner, "Non-smooth control barrier navigation functions for stl motion planning," *Frontiers in Robot. and AI*, 2022.

[191] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, and S. Sankaranarayanan, "Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications," *Lectures on Runtime Verification: Introductory and Advanced Topics*, 2018.

[192] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Int. Symp. on Formal Techniques in Real-Time and Fault-Tolerant Syst.* Springer, 2004.

[193] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *Control Syst. letters*, 2019.

[194] G. Yang, C. Belta, and R. Tron, "Continuous-time signal temporal logic planning with control barrier functions," in *Am. Control Conf.* IEEE, 2020.

[195] L. Lindemann and D. V. Dimarogonas, "Decentralized control barrier functions for coupled multi-agent systems under signal temporal logic tasks," in *Eur. Control Conf.* IEEE, 2019.

[196] ——, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *Transactions on Control of Netw. Syst.*, 2020.

[197] Y. Liu, J. J. Zhu, R. Williams II, and J. Wu, "Omni-directional mobile robot controller based on trajectory linearization," *Robot. and Auton. Syst.*, 2008.

[198] S. Bochkanov, "Alglib," http://www.alglib.net, accessed: October 2025.

[199] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[200] PAL Robotics, "Tiago pro mobile manipulator," https://pal-robotics.com/robot/tiago-pro/, 2025, accessed: October 2025.

[201] Stereolabs, "Zed 2 stereo camera," https://www.stereolabs.com/zed-2/, accessed: October 2025.

[202] ——, "Zed ros2 wrapper," https://github.com/stereolabs/zed-ros2-wrapper, 2025, accessed: October 2025.

[203] ——, "Zed sdk – developers release," https://www.stereolabs.com/en-it/developers/release, 2025, accessed: October 2025.

[204] Stereolabs, "Body tracking overview," https://www.stereolabs.com/docs/body-tracking, 2025, accessed: October 2025.

[205] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: Certainty in safety with uncertainty in state," in *Int. Conf. on Intell. Robots and Syst.* IEEE, 2021.

[206] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, 2018.

[207] R. Takano and M. Yamakita, "Robust constrained stabilization control using control lyapunov and control barrier function in the presence of measurement noises," in *Conf. on Control Technol. and Appl.* IEEE, 2018.

[208] A. Clark, "Control barrier functions for stochastic systems," *Automatica*, 2021.

[209] F. Bianchi, S. Formentin, and L. Piroddi, "Process noise covariance estimation via stochastic approximation," *Int. Journal of Adaptive Control and Signal Process.*, 2020.

[210] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter," *Kalman filtering and neural networks*, 2001.

[211] J. C. Butcher, *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[212] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation," in *power & energy society general meeting*. IEEE, 2017.

[213] L. Morillo-Mendez, M. Schrooten, A. Loutfi, and O. M. Mozos, "Age-related differences in the perception of robotic referential gaze in human-robot interaction," *Int. Journal of Social Robot*, 2024.

[214] J. Bravo, J. Cacace, D. Dal Moro, D. Serrano, and M. Dalmau-Moreno, "Improving robot social perception in human-robot-interaction using multi-modal cues," in *Iberian Robot Conf.* IEEE, 2024.

[215] C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, "Large language models for human–robot interaction: A review," *Biomimetic Intell. and Robot*, 2023.

[216] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," in *Int. Conf. on Intell. Robots and Syst.* IEEE, 2023.

[217] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Auton. Robots*, 2023.

[218] W. Wang, I. Obi, A. Bera, and B. Min, "Unifying large language model and deep reinforcement learning for human-in-loop interactive socially-aware navigation," 2025.

[219] M. Neggers, R. Cuijpers, P. Ruijten, and W. IJsselsteijn, "Determining shape and size of personal space of a human when passed by a robot," *Int. Journal of Social Robot*, 2022.

[220] W. Chen, T. Zhang, and Y. Zou, "Mobile robot path planning based on social interaction space in social environment," *Int. Journal of Adv. Robotic Syst.*, 2018.

[221] X. Truong and T. Ngo, "Dynamic social zone based mobile robot navigation for human comfortable safety in social environments," *Int. Journal of Social Robot*, 2016.