

University of Cassino and Southern Lazio

Department of Electrical and Information Engineering

Doctoral Course in Methods, Models and Technologies for Engineering
Cycle XXXVII

Shared Control Techniques for Assistive Robotics Applications and Human-Robot Interaction Scenarios

---

SSD: ING-INF/04

Supervisor:
Prof. Filippo Arrichiello

Author:
Francesca Patriarca

Coordinator:
Prof. Fabrizio Marignetti

UNIVERSITÀ DEGLI STUDI DI CASSINO E DEL LAZIO MERIDIONALE
CORSO DI DOTTORATO IN METODI, MODELLI E TECNOLOGIE PER L'INGEGNERIA

Author: Francesca Patriarca
Title: Shared control techniques for assistive robotics applications and Human-Robot Interaction scenarios
Department: Department of Electrical and Information Engineering
Degree: PHILOSOPHIAE DOCTOR

Permission is herewith granted to the university to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author:

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION. THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

"*Science is but a perversion of itself unless it has as its ultimate goal the betterment of humanity.*"

"*La scienza non è altro che una perversione di se stessa a meno che non abbia come obiettivo finale il miglioramento dell'umanità.*"

— *Nikola Tesla*

Abstract

This study presents a comprehensive study of shared control techniques in assistive robotics and human-robot interaction scenarios, with a particular emphasis on the integration of human-machine interfaces based on bioelectric signals. The aim is to address the growing need for assistive systems to improve the quality of life for individuals with motor disabilities, while simultaneously advancing collaborative robotic applications in several fields, including healthcare, rehabilitation, and industrial manufacturing.

This thesis at first presents a review of the fundamental principles of assistive robotics, focusing on the importance of autonomy in daily activities for individuals with disabilities. The field of assistive robotics, including prosthetics, exoskeletons, and robotic wheelchairs, is receiving increasing attention also due to the possibility of integrating human-machine interfaces based on physiological signals, such as electroencephalography and electromyography. The integration of these bioelectrical signals enables robots to interpret human intentions, thereby facilitating the performance of tasks that would otherwise require the constant assistance of an assistant, particularly in the context of assistive robotics applications. This work presents the implementation of brain-computer interfaces in shared control architectures, with a particular emphasis on electroencephalographic signals, which capture brain activity and facilitate user control of robotic systems through cognitive paradigms. The effectiveness of electroencephalography-based shared control systems has been demonstrated in teleoperation experiments involving mobile robots. These experiments have shown that such systems can enhance user autonomy and provide intelligent robotic assistance in tasks such as navigation and obstacle avoidance. They can also be used to control a robotic arm in manipulating objects in teleoperation scenarios.

Furthermore, this thesis addresses the applications of shared control in human-robot collaboration, with a particular focus on scenarios that require physical interaction and co-manipulation tasks. In such contexts, shared control frameworks that are guided by electromyography play a pivotal role, translating muscle activity into control signals for robotic

manipulators. This research presents the development and implementation of control systems based on myoelectric signals for robotic arms in co-manipulation tasks, wherein the human and robot work together to accomplish complex tasks necessitating precision and flexibility. The objective of shared control strategies is to adapt to the operator's intentions and adjust the robot's behavior in real time, thereby ensuring safe and efficient collaboration. Techniques such as admittance control, which adjusts the robot's stiffness and compliance based on human input, are employed to guarantee smooth and natural interactions.

Experimental validation represents a significant component of the research process. A number of experiments have been conducted with the objective of evaluating the performance of proposed shared control systems in both assistive and collaborative robotic applications. For example, a comprehensive investigation of teleoperating a mobile robot via a brain-computer interface is presented, wherein users navigate challenging environments while the robot proactively assists in tasks such as path planning and obstacle avoidance. Furthermore, a scenario in which the user employs electroencephalographic signals to manipulate objects via the control of a robotic arm is also presented. Similarly, experiments involving electromyography-based control of robotic manipulators demonstrate the feasibility of using bioelectrical signals to perform co-management tasks, thereby providing insights into how shared control systems can enhance productivity and reduce user workload. The experimental results demonstrate that shared control techniques not only enhance the autonomy and efficiency of assistive robotic systems, but also improve the safety and usability of human-robot interactions in collaborative scenarios. The results indicate that the incorporation of bioelectric signals into shared control frameworks provides notable benefits in terms of adaptability and responsiveness to user requirements, thereby facilitating the customization of robotic assistance across a diverse range of applications.

In conclusion, this thesis makes a significant contribution to the field of robotics by proposing innovative shared control architectures that bridge the gap between human capabilities and robotic assistance. The research enables more seamless and intuitive human-robot collaboration through the use of bioelectric signals, thereby opening up new avenues for assistive and collaborative robotic applications in medical and industrial settings.

Contents

Abstract **vii**

1 Introduction **1**
1.1 Motivation 1
1.2 Advanced Robotics 3
1.2.1 Assistive Robotics 4
1.2.2 Collaborative Robotics 7
1.2.3 Human-Robot Interaction 10
1.3 Shared Control 12
1.3.1 Shared control techniques for assistive robotics . . . 12
1.3.2 Shared control techniques for human-robot collaboration and interaction 14
1.4 Thesis outline and contribution 18

2 Bioelectrical Signals **21**
2.1 Electroencephalography (EEG) 22
2.1.1 Nervous System 23
2.1.2 EEG signals acquisition 24
2.1.3 Brain Computer Interface (BCI) 26
2.2 Electromyography (EMG) 27
2.2.1 Electromyographic signals 29
2.2.2 EMG Sensors 31

3 Background **33**
3.1 Kinematics 33
3.1.1 Rigid Body Position and Orientation 33
3.2 Differential Kinematics for a Robotic Manipulator 35
3.3 Differential Inverse Kinematics 36
3.4 Kinematic Singularities 38
3.5 Set-based Task Priority Inverse Kinematics 39

3.6 Differential Kinematics for a Non Holonomic Mobile Robot 44
3.7 Force Control in Robotic Manipulation 46
3.7.1 Impedance Control 47
3.7.2 Admittance Control 48

4 Shared Control Applications for Assistive Robotics 51
4.1 Introduction and Related works 51
4.2 Null-space-based shared control of a mobile robot using motor imagery based brain-computer interface 54
4.2.1 Shared control architecture 55
BCI based Motor Imagery 57
4.2.2 Null-space-based shared control strategy 60
Differential-drive robots kinematics and low-level controller 60
Elementary behaviors 62
Velocities fusion 64
4.2.3 Experiments 65
Implemented functional architecture 65
Experimental validation 66
4.3 Shared control architecture for teleoperation of a robotic manipulator through BCI 70
4.3.1 Shared Control Architecture 71
4.3.2 BCI based Motor Imagery 72
4.3.3 Robot end-effector control strategy 73
4.3.4 Joints controller 76
4.3.5 Experimental Results 77
System Architecture 77
Experimental Validation 79

5 Shared Control Application for Human-Robot Interaction 83
5.1 EMG-based shared control framework for human-robot co-manipulation tasks 85
5.1.1 Problem description 85
5.1.2 Shared control architecture 87
5.1.3 Low-level layer 88
Inverse kinematics and joints controller 89
Admittance control 90

5.1.4 High-level layer 91
EMG Signal Analysis 92
Finite State Machine 93
5.1.5 Experimental validation 94
Experimental Setup 94
Training and test of the classifier 95
Experimental results 96
5.2 EMG-driven shared control architecture for human-robot co-manipulation tasks 101
5.2.1 Problem Description 101
5.2.2 Shared control architecture 103
5.2.3 Low-level layer 105
Inverse kinematics and joints controller 105
Admittance control 105
5.2.4 High-level layer 106
EMG signal processing and classification 106
Finite State Machine 107
5.2.5 Experimental validation 108
Experimental Setup 108
Training and test of the classifier 108
Experimental results 109

6 Conclusions and Future Works **119**

List of Figures

1.1 An example of assistive robotics applications. 5
1.2 An example of exoskeletons for rehabilitation therapy. 6
1.3 An example of collaborative robotics applications. 8
1.4 An illustrative example of a human-robot physical interaction. . 11

2.1 Block diagram of the general processing that raw bioelectric signals must undergo in order to be converted into commands that can be sent to external devices. 22
2.2 Cerebral cortex lobes. 23
2.3 The 10-20 System. It is an internationally recognized method to describe and apply the location of scalp electrodes in the context of an EEG exam, polysomnograph sleep study, or voluntary lab research. 24
2.4 Graphical representation of the Motor Unit. 29
2.5 The recruitment of four motor units. 30

3.1 Example of robot manipulator with 3 links. 35
3.2 This figure presents the evolution of a generic set-based task, with particular attention to the activation thresholds ($\sigma_{a,u}$ and $\sigma_{a,l}$) and the safety thresholds ($\sigma_{s,u}$ and $\sigma_{s,l}$.) 40
3.3 Solution tree example. 42
3.4 Workflow of the algorithm. 44
3.5 Genaral block diagram of impedance control. 48
3.6 Genaral block diagram of admittance control. 49

4.1 On the left: evolution of the free force F_{free} with $\psi = 0.4$, $\omega = 0.3$. On the right: BCI force F_{BCI} with $\alpha_1 = 6.4$ and $\alpha_2 = 0.4$. 59
4.2 Activation function with $\delta = 0.05$ and $\varepsilon = 0.02$. 61
4.3 Example of velocity related to a user command given for 2.5s. . 63
4.4 Implemented functional architecture. 65

4.5 Snapshot of the built scenario for the experimental validation. The user is required to teleoperate the Turtlebot across three distinct sections: an obstacle-free area (depicted in blue), an area with a simulated narrow passage (depicted in green), and an area with three obstacles (depicted in red). 66
4.6 The path performed during one of the experiments highlighted according to the active control modes: in blue the path executed in *free* mode, in green in the *autonomous* mode, and in red in the *guided* mode. 68
4.7 Linear and angular velocities over time during one of the runs. The background colour is indicative of the active control mode. The *free* mode is represented by the blue line, the *autonomous* mode by the green line, and the *guided* mode by the red line. . 69
4.8 Snapshot of the built scenario for the experimental validation. The user is required to teleoperate the Jaco^2 robotic arm using BCI to manipulate objects in the space. 71
4.9 Main components of the proposed architecture. 78
4.10 Screenshots of the video taken during an experiment. 79
4.11 Input-output of the filter over time during the experiment. . . 80
4.12 Path followed by the end-effector in one of the experiment runs. In the plot are highlighted the Start and Stop points, the two ROAs and the bottle in positions of interest. 81

5.1 The scenario shows the experimental setup in which a human operator performs a co-manipulation task wearing four EMG sensors and using a Jaco^2 manipulator. 87
5.2 Proposed shared control architecture. The High-Level layer is responsible for the EMG signal acquisition and classification. Then, the recognized class is used into a finite-state machine algorithm that outputs a set of admittance gains. The Low-Level layer is responsible for the motion control of the manipulator, and it employs a variable admittance controller, an inverse kinematics algorithm and a joints controller. 88
5.3 Finite State Machine. 94
5.4 The confusion matrix obtained after training the classifier. . . . 97

5.5 The path followed by the manipulator's end-effector in one of the experiments performed. The path is highlighted with two colors based on the two operational modes: *Low-Damping* mode is marked in green and *High-Damping* mode is highlighted in blue. 98
5.6 Experimental results: a) EMG Classifier Output; b) Force measured by the F/T sensor; c) End-Effector Linear Velocity. The different colored areas on all graphs correspond to the two different control modes: green and blue represent the *Low-Damping* and the *High-Damping* modes, respectively. 100
5.7 The scenario shows the experimental setup in which a human operator performs a co-manipulation task using a Jaco^2 manipulator. On the dominant arm of the human operator, four Delsys Trigno Avanti EMG sensors are placed on the biceps and triceps brachii, flexor carpi radialis and extensor carpi ulnaris muscles. 103
5.8 Proposed shared control architecture. The High-Level layer is responsible for the EMG signal acquisition and classification. Then, the recognized class is used into a finite-state machine algorithm that outputs a set of admittance gains. The Low-Level layer is responsible for the motion control of the manipulator, and employs a variable admittance controller, an inverse kinematics algorithm and a joints controller. 104
5.9 Finite State Machine. Starting from the *no-contact* mode, the FSM switches to a different control mode based on the class triggered by the human operator and the time it is maintained. 107
5.10 The series of images displayed above illustrates one of the experiments conducted by Subject 1. 112
5.11 The confusion matrices obtained following the training of the classifier for all participants: a) the confusion matrix obtained for the Subject 1; b) the confusion matrix obtained for the Subject 2; c) the confusion matrix obtained for the Subject 3; d) the confusion matrix obtained for the Subject 4. 113
5.12 The path followed by the manipulator's end-effector in one of the experiments performed by the Subject 1. The path is highlighted with three colors based on the three operational modes: *no-contact* mode is marked in green, *contact* mode is highlighted in blue and *hold* mode is colored in red. 115

5.13 Results of one of the experiments of the Subject 1: a) EMG Classifier Output, i.e. the statistical mode of the last 1.5s of predicted classes; b) Force measured by the F/T sensor; c) End-Effector Linear Velocity. The different colored areas on all graphs correspond to the three different control modes: green, blue and red represent the *no-contact*, *contact*, and *hold* modes, respectively. 116

List of Tables

2.1 The five types of brainwaves commonly recognized. 25

5.1 Experimental parameters used in the proposed architecture. 96
5.2 Table containing performance evaluation metrics. 98
5.3 Experimental parameters used in the proposed architecture. 111
5.4 Table containing performance evaluation metrics 114

Chapter 1

Introduction

1.1 Motivation

In recent decades, there has been a growing focus on enhancing the quality of life for individuals with severe motor disabilities to increase their independence by reducing the presence of caregivers. Indeed, for people with motor impairments, daily-life activities, such as drinking or dressing, require continuous assistance. The use of devices that can provide an independent mobility can therefore have a significant impact on quality of life from both a practical and psychological perspective. As a result, there has been a growing interest in the field of assistive robotics and the scientific community is investing significant resources in the development of robotic solutions that can replace the need for caregivers. This is also made possible by the advent of new technologies and devices that enable users to interact with and control robotic systems. In this context, Human-Machine Interfaces (HMIs) have assumed an increasingly important role. Both the robotic systems used to assist individuals with disabilities and the human-machine interface employed to facilitate user command generation for the system are of paramount importance in assistive robotics. These interfaces may be based on a variety of physiological signals, including ElectroOculoGraphic (EOG), ElectroEncephaloGraphic (EEG), and ElectroMyoGraphic (EMG) signals.

In addition to assistive robotics, however, the use of HMIs also plays an important role in collaborative robotics and human-robot interaction scenarios, as they represent a contact point between the user and robotic systems. The objective of scientific research in these fields is to enhance

the efficiency, safety, and ease of use of these interfaces, thereby optimizing cooperation between humans and machines. In industrial manufacturing, healthcare, and rescue environments, the capability to interact intuitively and securely with robots is essential for reducing errors, increasing productivity, and ensuring operator safety. The advancement of sophisticated HMIs can also facilitate enhanced adaptability and personalization of interactions, enabling progressively natural integration of robotics into everyday life and human work. From the user's perspective, the operational mode of robotic systems may depend on the level of autonomy provided/required of the system itself; while from the device's perspective, this corresponds to different control modes of robotic devices, usually closely related to the HMI used to generate and communicate commands.
In this context, shared control modes, which allow both human and robot to contribute significantly to the task execution, have received particular attention. Furthermore, it is very important to maintain a balance between the contributions of the user and those of the robot. The level of robot control can vary depending on the user's requirements, ranging from complete user control to more autonomous modes in which the robot handles a multitude of aspects of the task to be performed. This allows robots to interpret user commands intelligently and autonomously handle complex or safety-critical tasks, thereby facilitating more intuitive and reliable interaction. Thus, in shared control mode the user is involved in the control loop as an active part, continuously generating high-frequency motion commands, that are then translated by the control software into low-level functions for the robot, following all security policies.

To illustrate, in assistive robotics, EMG sensors can be employed to detect the muscle activity of residual limb movement, thereby enabling the control of a prosthetic arm. Similarly, regarding brain-computer interfaces (BCIs), individuals with severe motor disabilities can utilize EEG signals to operate wheelchairs or robotic exoskeletons. In the context of rehabilitation, shared control with HMI integration enables patients to actively engage in therapy using robotic exoskeletons, thereby promoting motor recovery while ensuring the safe execution of movements. The integration of bioelectrical signals into shared control systems enables robots to interpret physiological inputs and intelligently share control with users. This creates an adaptive system where users provide high-level intentions,

like movement commands, while the robot autonomously handles tasks such as obstacle avoidance and precision control. This reduces cognitive load and enhances safety and usability, especially in complex or physically demanding tasks. These systems integrate user input with autonomous decision-making, enhancing adaptability and real-time responsiveness to user needs and environmental changes.

The objective of this research is to implement a more fluid and adaptive human-robot collaboration, utilizing the capabilities of human-machine interfaces to bridge the gap between human capabilities and robotic assistance in both medical and industrial settings. The goal is to examine and expand the shared control methodologies, offering insights into how the integration of HMIs can result in more responsive, adaptable, and effective assistive robotic systems. Furthermore, the project seeks to examine how the integration of HMI technologies into shared control frameworks can revolutionize collaborative robotics applications, resulting in more intuitive, responsive, and effective systems that enable users to entrust tasks, that are strenuous for humans, to the robot while allowing them to continue performing high-level tasks such as management and decision-making that robotic systems are unable to fulfill. In this manner, the capabilities of both operators can be integrated: the precision and dexterity of robots and the problem-solving abilities of humans can be combined to achieve tasks in a more effective and efficient manner.

1.2 Advanced Robotics

The field of advanced robotics encompasses a wide range of cutting-edge technologies and applications that are transforming various fields, including healthcare, research, and industrial automation, among others. Significant advances in perception, deep learning, and artificial intelligence technologies have enabled the proliferation of robotic systems capable of autonomous decision-making [1], thereby facilitating more effective implementation of artificial intelligence-based solutions and enhanced robot-human collaboration [2]. Furthermore, the advent of soft robotics, driven by innovations in soft material actuation and sensing, has provided new possibilities in domains such as surgical procedures and endoscopic applications [3]. In contrast, in the field of industrial robotics there is a growing emphasis on miniaturization, energy efficiency, and actuator precision,

with applications encompassing industrial, medical, service, and military robots [4]. The integration of advanced technologies such as artificial intelligence and robotics has the potential to transform healthcare: this is evidenced by the increasing use of robots to support clinical practices such as robot-assisted surgeries and needle biopsies [5, 6]. These robotic systems are equipped with high-definition vision systems, advanced control algorithms, and innovative technologies such as the EndoWrist, which facilitate greater accuracy and efficiency in medical procedures [6, 7]. Research on human-robot interactions is advancing rapidly, with the goal of creating more efficient robots that can communicate effectively with people [8], leading to the integration of robots into shared spaces and collaborative tasks with humans, indicating a shift toward more interactive and cooperative robotic systems. Advances in soft and flexible robots have paved the way for their application in the medical field, where bionics, flexible actuation, sensing technologies, and intelligent control algorithms are improving their capabilities for medical procedures [9]. The use of soft robotics is also increasing in smart manufacturing is also on the rise, offering increased productivity, flexibility, automation, and safety in industrial environments [10]. In addition, the integration of robotics and artificial intelligence in healthcare is facilitating the development of innovative solutions that enhance the quality and accessibility of healthcare services [11]. Robots are increasingly being used in healthcare to assist in tasks ranging from substation inspection to complex clinical research, showing the diverse applications of robotics in healthcare [12, 13].

1.2.1 Assistive Robotics

Advanced robotics and assistive robotics are interrelated that aim to enhance the quality of life for humans through technological innovation. The growth of advanced robotics has enabled the creation of assistive solutions that are tailored to the specific needs of individuals with disabilities, the elderly, and those who are undergoing rehabilitation [14]. These fields have a shared focus on providing practical assistance, companionship, and increasing the independence of users, particularly those who have physical or cognitive limitations. In healthcare, assistive robotics (Figure 1.1) plays a pivotal role, encompassing a wide range of practices aimed at improving patient outcomes, enhancing surgical precision, and facilitating rehabilitation efforts [15]. For example, robot-assisted surgery



Figure 1.1: An example of assistive robotics applications.

enables surgeons to perform operations with enhanced dexterity and precision, reducing invasiveness and postoperative complications [16, 17, 18]. Moreover, wearable robots have demonstrated efficacy in medical rehabilitation, particularly in military contexts, employing motorized mechanisms and sensor technology to enhance users' mobility and functionality [19]. The integration of robotics into rehabilitation (Figure 1.2) programs has demonstrated the ability to enhance outcomes and facilitate more efficient regaining of independence. For example, the development of assistive robots for specific medical procedures, such as deep brain stimulation, highlights the precision and accuracy that robotic assistance can offer in complex interventions [20]. In rehabilitation therapy, advanced and assistive robotics facilitate the delivery of high-dose, high-intensity training, which is a crucial element in the successful recovery from motor disorders [21]. Research has shown that robot-assisted therapy facilitates consistent training, reliable performance assessment, and greater independence during exercises, which is particularly beneficial for stroke patients [22]. By enabling the repetition and intensity of practice, these systems enhance the recovery process, offering significant advantages over traditional rehabilitation methods.



Figure 1.2: An example of exoskeletons for rehabilitation therapy.

The incorporation of artificial intelligence (AI) into assistive robotics has led to significant advancements in the field, allowing for the development of intelligent, adaptive systems that can respond to the specific needs of users.. For example, the integration of AI-based perception capabilities into assistive platforms has been demonstrated to enhance their functionality and responsiveness, thereby improving their effectiveness in a range of contexts [23]. Such adaptability is of particular importance in rehabilitation settings, where patients' needs can vary significantly over time.

The design and functionality of assistive robots are evolving to address the unique challenges of their intended applications. Autonomous assistive robots include advanced capabilities such as object detection and recognition, which allow them to adapt quickly to dynamic environments [24]. Unlike traditional industrial robots, assistive robots emphasize user interaction, safety, and ease of use, making them more suitable for personal assistance tasks [25].

In addition to medical applications, assistive robotics plays a role in home

care, performing tasks such as medication reminders and physical assistance. These systems alleviate the burden on caregivers while improving users' autonomy. Recent advancements in soft robotics have introduced new possibilities for assistive devices that provide smoother and safer interactions, thereby reducing the risk of injury in sensitive contexts such as rehabilitation [26].

Collaborative robots (cobots) exemplify the role of advanced robotics in care applications, designed to work alongside humans in various settings. They relieve healthcare professionals of monotonous tasks, thereby allowing them to focus on more complex responsibilities [27]. The evolution of human-robot interaction has enabled these cobots to provide personalized support in both industrial and personal settings, fostering a sense of partnership that enhances motivation and engagement during rehabilitation [28].

1.2.2 Collaborative Robotics

Collaborative robotics represent a significant evolution in advanced robotics, with safety and human interaction becoming central to their design and functionality [29]. Unlike traditional industrial robots, which typically operate in isolation or behind safety barriers, collaborative robots or cobots are designed to share workspaces with humans (Figure 1.3), enabling interaction in dynamic environments without the need for physical separation [30, 31]. This transition has been driven by the desire to improve productivity, enhance safety, and reduce costs in various applications [32], making cobots particularly valuable in industrial settings such as manufacturing and agriculture. The evolution of robotics has led to advanced robots capable of performing complex tasks autonomously, while cobots are specifically designed to work alongside human workers, combining the precision and strength of robotics with human cognitive capabilities. The increasing demand for collaborative robots in industrial contexts highlights their potential to transform traditional workflows by combining human cognitive capabilities with the precision and strength of robotics [33]. This collaboration aims to improve efficiency in various fields, from manufacturing to healthcare [34]. Technological advances such as intention-aware motion planning have significantly influenced the capabilities of collaborative robots, allowing them



Figure 1.3: An example of collaborative robotics applications.

to anticipate human movements and generate collision-free trajectories, thereby increasing safety and efficiency [35]. These advances also allow humans to delegate repetitive or physically demanding tasks to robots, reducing the risk of injury and increasing productivity [36]. Such capabilities are critical to cobots' ability to adapt to changing task layouts and optimize production processes [31].

One of the distinguishing characteristics of collaborative robots is their ability to perform joint manipulation tasks with human operators, achieving seamless human-machine cooperation through physical interactions [37, 38]. These interactions go beyond simple task sharing to include cognitive capabilities that enable cobots to predict human behavior and adapt to human needs during collaborative processes [39]. As robotics technology advances, the introduction of cobots has changed the dynamics of human-robot interaction, requiring new skills and training for human workers to ensure effective collaboration [40]. Safety is a critical aspect of integrating collaborative robots into human environments. Ensuring safe human-robot collaboration requires comprehensive risk assessments and the implementation of rigorous safety protocols [41]. These measures, along with advanced technologies such as robot skin and collaborative perception systems, help maintain safe interactions in dynamic

environments [42, 43]. Risk management and virtual reality simulations further contribute to the development of safe collaborative systems by allowing thorough testing without direct human-robot interaction [44, 45]. This ensures that cobots can operate effectively while maintaining high safety standards in various industrial applications, such as automotive manufacturing [46].

The potential of collaborative robotics extends beyond industrial environments. Its versatility is evident in applications such as cloud-based 3D mapping, coordinated search operations, and user-centred manipulation tasks, which improve the working conditions of operators in various production activities [47, 48]. Advances in AI and machine learning have further enhanced the capabilities of cobots, enabling them to better understand tasks, interact naturally with human operators, and adapt to unstructured environments [49, 50]. As the field of collaborative robotics continues to evolve, the role of cobots is expanding, fostering greater flexibility and adaptability in modern manufacturing systems [51]. The integration of cobots into Industry 4.0 has been facilitated by digital tools that simulate interactions, offering new opportunities for increased productivity and efficiency [52]. The synergy between AI and robotics plays a crucial role in optimising collaborative processes, improving operational efficiency and reshaping business models through automation [53]. This transformation highlights the importance of developing intuitive interfaces and lightweight robots equipped with sensors to ensure safe and efficient human-robot collaboration [54, 55]. The development of bio-inspired soft robots adds another dimension to the field, as their flexibility makes them suitable for delicate tasks and complex environments, promoting trust and cooperation in human-robot interactions [56]. In the future, research needs to address multi-agent scenarios to understand the dynamics of group interactions, going beyond individual robot-human collaboration to develop effective collaborative strategies for more complex environments [57]. The future of collaborative robotics lies in further enhancing these interactions through AI algorithms and improved communication systems, enabling robots to better understand and respond to human needs and behaviours [58, 59].

1.2.3 Human-Robot Interaction

As mentioned above, the concept of collaborative robotics is closely related to the idea that robots and humans can work together in a shared workspace, leveraging each other's strengths. While collaborative robots are designed for tasks that require high precision and repeatability, humans contribute cognitive and problem-solving skills to handle complex and unpredictable scenarios [60, 61]. This synergy is essential to maximize productivity in modern industrial environments, especially as industries transition to the Industry 4.0 era, where automation and human collaboration are critical [62, 63]. The fields of collaborative robotics and human-robot interaction (HRI) focus on integrating robots into human environments to achieve common goals. Effective HRI involves designing robots that can operate safely in close proximity to human workers, thereby facilitating productive interaction. This relationship is particularly relevant in industrial applications, where cobots perform repetitive tasks while humans handle more complex decision-making processes [64, 61]. In healthcare, collaborative robots provide patient assistance, support medical staff, and ensure patient safety [63, 65]. A key aspect of HRI is ensuring the safety of these interactions. Effective collaboration requires robots to adapt their behavior based on human actions to ensure safe and efficient teamwork. This includes equipping collaborative robots with sophisticated safety features and following standards such as ISO 10218 and ISO/TS 15066 to guide the safe design and operation of collaborative robots [63, 66, 67, 68]. In addition, bio-inspired design principles such as flexible robot skins and compliant control systems enable robots to interact with humans in a more natural way, enhancing safety and improving the overall collaborative experience [66, 69].

The evolution of HRI research reflects a broader trend towards understanding and improving human-robot collaboration. It has evolved from early studies of teleoperation to more sophisticated approaches focusing on shared attention and mutual understanding in social robotics [70, 71]. Advances in machine learning and AI allow robots to learn from human demonstrations, which is crucial for adapting to human actions and fostering seamless interaction [72, 73]. These capabilities are essential for developing robots that can operate autonomously while responding effectively to human input. Another key concept of HRI is the physical



Figure 1.4: An illustrative example of a human-robot physical interaction.

human-robot interaction (Figure 1.4), that involves direct physical contact between humans and robots, requiring precise coordination of forces and movements. In this case, it is crucial to understand the dynamics of physical interaction between humans and robots, as cobots must operate in spaces shared with humans, thus requiring a focus on safety and effective interaction [74, 75]. Effective pHRI is particularly important in unstructured environments such as surgery, where the ability of robots to understand and respond to human actions enhances their utility [76]. In this context, control strategies such as admittance control are used to ensure smooth physical interactions [77]. Safety remains a key concern, with adaptive control methods proposed to optimize interaction safety and comply with safety standards such as ISO10218 [78, 79].

As collaborative robotics evolves, ongoing research continues to address challenges such as task responsibility ambiguity, effective communication strategies, and the development of robust safety protocols [57, 80]. By deepening our understanding of these dynamics, the field can make progress towards creating enhanced collaborative systems that improve both productivity and human well-being.

1.3 Shared Control

The concept of shared control entails a collaborative approach in which both humans and intelligent machines contribute to decision-making and task execution [81]. This concept is particularly relevant in scenarios where humans and robots work together to achieve a common goal, with the objective of enhancing the effectiveness of their interactions by aligning the robot's actions with the human partner's intentions [82]. Indeed, in shared control schemes, the user is integrated into the control loop as an active part, generating high-frequency motion commands that are then converted by the control software into low-level references for the actuators in compliance with all robot and human constraints, especially constraints for human safety. It combines the advantages of humans with those derived from the autonomy of robotic systems, and distributes control between the human and robotic partners, thereby enhancing performance and safety [83]. The distribution of authority between human and machine can vary, with the possibility of gradual transitions in control responsibility, particularly in haptic shared control scenarios [84]. This dynamic allocation of control is crucial for achieving optimal performance, as demonstrated in studies showing benefits such as improved vehicle control, reduced control effort and decreased demand for visual attention.

1.3.1 Shared control techniques for assistive robotics

Shared control techniques have been employed to address particular challenges in the field of assistive robotics. For instance, Gaussian processes have been utilized to facilitate shared control of semi-autonomous mobile robots while maintaining reliable connectivity [85]. Furthermore, vector field-based approaches for guiding assistive mobile robots improve user experience and control precision by integrating user input with autonomous behaviors, allowing smooth navigation across different settings. This shared control enhances natural interaction, usability, and acceptance of assistive robotic systems [86]. Moreover, shared control techniques have played a pivotal role in the development of shared haptic control methods for task such as robotic cutting under non-holonomic constraints. These methods provide users with haptic feedback and guidance during complex tasks, enhancing user comprehension and control

precision and improving performance and user satisfaction [87]. Shared control has also been employed in the development of robotic wheelchairs, demonstrating its potential to enhance user control and safety in mobility assistance devices [88].

Instead, in the assistive teleoperation, shared autonomy methods have been employed to enhance the safety and precision of robotic control by combining user input with the robot's autonomous controls [89]. These approaches ensure users receive essential support for effective robot handling, especially in challenging environments. Through shared autonomy, assistive robots adapt to user input and environmental changes, enhancing system robustness and performance. Shared autonomy also plays a pivotal role in assistive robotic manipulation to augment human control signals with robotic autonomy, thereby facilitating collaborative and synergistic operations between users and robots. Furthermore, it demonstrates the efficacy of shared autonomy in improving the operation of robotic arms through a combination of shared autonomy and body-machine interfaces [90] to enable individuals with limited mobility to perform tasks that would otherwise be challenging or impossible, expanding the scope of tasks that can be accomplished [91] and facilitating the effective control and navigation of assistive robots such as wheelchairs, walkers, or robotic arms [92].

A significant application of shared control techniques in assistive robotics is the use of brain-computer interfaces, which allow users to control assistive robots through neural signals in a direct and intuitive way, for example to enable users to navigate mobile robots with greater accuracy and efficiency, thereby enhancing their overall autonomy and independence [93] or to control robots through paradigms such as motor imagery. This leads to specific modulations in brain EEG signals, including sensorimotor rhythms (SMRs), which can be used to voluntarily control external devices such as robots or exoskeletons [94]. Furthermore, EEG signals have been employed to rectify robot errors in real time, thereby illustrating the potential of EEG-based feedback methodologies to increase the responsiveness and adaptability of assistive robots, enhancing user experience and system performance [95]. In addition, the development of noninvasive BCIs has enabled the control of robotic manipulators based on motor-imagery paradigms, allowing users to transmit their mental intentions to the robot through EEG data [96] by decoding EEG signals

associated with specific mental states or motor intentions.

Recent advances have also highlighted the potential for integrating functional near-infrared spectroscopy (fNIRS) with electroencephalography in hybrid brain-computer interface systems. This combination exploits the temporal resolution of EEG and the spatial resolution of fNIRS, thereby facilitating more robust control and feedback mechanisms for assistive devices. For example, EEG-fNIRS systems have demonstrated enhanced classification accuracy in motor imagery tasks by integrating insights from neurovascular coupling with electrophysiological signals [97, 98]. These hybrid systems enhance the control and functionality of assistive robotics by improving the responsiveness of robotic arms and mobility devices. Furthermore, novel methods such as fNIRS-guided attention networks have been developed to integrate features from both modalities, resulting in a notable improvement in performance in mental arithmetic and motor imagery tasks [98]. Potential applications include motor rehabilitation, where these hybrid systems provide more comprehensive feedback and adaptive control, further optimizing user interaction with assistive robots [99].

1.3.2 Shared control techniques for human-robot collaboration and interaction

In collaborative human-robot control for moving and transferring objects, shared autonomy systems play a pivotal role in enabling autonomous robots to perform tasks in collaboration with humans [100], as for the development of autonomous collaborative robots in the field of physical human-robot interaction [101]. In these systems, both the human and the robot are responsible for controlling the movement. However, the robot's control increases in accordance with its growing understanding of the human's intentions [102]. Some of the proposed shared control strategies include the use of surface EMG signals to control mobile robots, ensuring obstacle avoidance and safe navigation in complex environments [103]. These signals play a crucial role in collaborative robotics and human-robot interaction, as they provide a direct interface between human muscle activity and robotic control systems. Model-based classifiers for remote control of robots [104], the integration of muscle activity

into movement patterns for physical interactions based on haptic communication [105], the use of surface electromyographic signals to improve teleoperation [106] and the development of biofeedback mechanisms for personalized robotic rehabilitation [107] are representative examples of the use of EMG signals in collaborative scenarios. Recent developments in shared control frameworks also emphasize the combination of electromyography and force myography (FMG) signals to enhance human-robot collaboration. The integration of both EMG and FMG sensors allows the respective strengths of each modality to be exploited. EMG is able to capture muscle activation, while FMG is able to detect pressure changes, providing a more comprehensive data set for decoding human intent. For example, hybrid measurement systems using both EMG and FMG sensors have demonstrated high accuracy in identifying specific gestures, achieving classification accuracies of up to 99%. This increases their utility in precise robotic control and rehabilitation tasks [108]. In addition, the incorporation of EMG and FMG into multimodal interfaces has significantly improved the precision and robustness of gesture recognition in dynamic contexts, facilitating intuitive control of prosthetics and other assistive devices [109]. In addition, the combination of EMG and FMG signals has been used to improve active rehabilitation systems. For example, innovative hydrogel-based sensor systems collect these multimodal signals to decode human movement intentions with high precision, achieving gesture recognition accuracies above 90% and illustrating the potential for robust control in active rehabilitation scenarios [110]. These results highlight the potential of integrating EMG and FMG to improve shared control frameworks by enabling more intuitive and precise human-robot interactions. This dual-sensor approach addresses the limitations of EMG-only systems, which are susceptible to noise, while providing complementary data for improved control and adaptability in a variety of applications.

The implementation of advanced control algorithms such as simultaneous localization and mapping (SLAM) is also enabled by the use of EMG signals, allowing mobile robots to navigate unknown environments based on electromyographic input [111]. In addition, studies have demonstrated the feasibility of lifting human-robot teams based on muscle activity measured by EMG signals [112], suggesting the potential for shared workload

scenarios between humans and robots. These signals can also be incorporated into collaborative robotic systems for teleoperation purposes [113], for instance to control robotic arms [114, 113], track upper limb movements [115], and improve cooperative manipulation tasks by employing data-driven modeling of muscle activities [116]. Advances in EMG-based control strategies have led to innovative approaches in robotics, such as the design of control maps for EMG-guided multichannel prostheses and supernumerary limbs [117], the identification of the EMG-human arm strength relationship for rehabilitation purposes [118], estimation of human hand stiffness to improve variable stiffness manipulation in robotic systems [119], and implementation of teleimpedance control using EMG data to control robot stiffness [120].

The ability to facilitate physical human-robot interaction during collaborative tasks is crucial for shared control, as it enhances both adaptability and safety in these interactions. Impedance/admittance control strategies are a commonly employed method for managing interaction dynamics, whereby robots are controlled as virtual dynamic systems with adjustable impedance parameters [121, 122]. This has led to the design of adaptive robotic control mechanisms to enable shared control in scenarios where humans dominate the interaction, using impedance control techniques to achieve a balance between reference tracking and human assistance [123]. Rather than relying only on their intrinsic dynamics, robots can adjust their compliance and motion synchronization capabilities to enhance collaboration with humans and interactions with the environment [124]. For non-backdrivable manipulators, this adaptability allows the replication of dynamic behavior to improve its physical interaction capabilities [125]. A significant advancement in improving adaptability and security has been the development of variable admittance control methodologies. In contrast to fixed admittance control, which is defined by static parameters, variable admittance control allows robots to adapt their behaviors in response to environmental conditions, acceleration, velocity, inertia, and human intention [126, 127]. For example, the integration of impedance controllers and safety-aware compliance mechanisms allows collaborative robots to navigate shared workspaces while considering energy efficiency and ensuring user safety [128]. By adapting parameters such as stiffness or damping in real time, robots can enhance their versatility and meet task-specific requirements, providing appropriate assistance or compliance

as needed [129]. In physical human-robot interaction, this real-time adjustment fosters intuitive interactions and aligns robot movements more closely with human intentions [130, 131]. This is of particular importance in scenarios such as rehabilitation training, where responsive control is essential for ensuring safety and efficacy effectiveness [132].

In pHRI, the strategic selection and adaptation of admittance control parameters are crucial for the establishment of stable and effective interactions between robots, the surrounding environment, and human operators. The dynamic adjustment of damping or stiffness parameters is particularly advantageous in applications that necessitate precision and stability, such as robotic surgery or micromanipulation tasks [133]. Notable research findings, as evidenced by the authors in [134], emphasize the significance of parameter adaptation in ensuring the stability and quality of interactions. In collaborative assembly tasks, adaptive control mechanisms that incorporate human cognitive strategies, such as virtual boundary walls, have demonstrated enhanced safety and efficiency [135]. Furthermore, techniques that integrate variable admittance control with human motion prediction or EMG-based muscle co-activation have enabled a more seamless alignment with human actions, thereby facilitating more effective human-robot interactions [131, 136]. Recent studies have investigated the potential of machine learning, including reinforcement learning and neural networks, to enhance variable admittance control. This approach allows robots to learn and adjust admittance parameters based on past experiences or human feedback [137, 138]. This intelligent adaptation is beneficial in dynamic environments, particularly in situations where continuous performance improvement is necessary, such as in complex manipulation or interactive surgical procedures [125]. The incorporation of fuzzy inference systems into admittance control serves to illustrate the advancement of decision-making processes that emulate those of humans, thereby enhancing the adaptability of robots in cooperative scenarios [139].

1.4 Thesis outline and contribution

This thesis makes a significant contribution to the field of assistive robotics and human-robot interaction through the development and validation of shared control architectures. The research promotes the integration of bioelectrical signals, particularly electroencephalographic and electromyographic signals, into robotic systems, enabling intuitive and adaptive human-robot collaboration. By leveraging these signals, this thesis proposes new shared control frameworks that integrate human cognitive or physical intent and robotic execution to ensure the usability and safety of robotic systems.

The work introduces EEG-based shared control strategies that facilitate intuitive navigation and interaction with robotic assistive systems. The use of brain-computer interfaces enables people with severe motor disabilities to control robots and mobile manipulators, thereby achieving a level of autonomy in tasks such as navigation, obstacle avoidance, and object manipulation. These systems are designed to adapt dynamically to the user's intentions, thereby reducing the cognitive load and maintaining precise control. In addition to EEG-guided approaches, this thesis presents EMG-based frameworks for co-manipulation tasks. These frameworks are designed to interpret muscle activity to guide robots in human-robot interaction scenarios. They enable real-time adaptation to user actions and ensure smooth, safe, and effective collaboration. By employing sophisticated signal processing techniques, the systems can effectively classify user input, effectively translating physiological signals into actions that seamlessly integrate with robotic behavior.

Another remarkable aspect of this work is the introduction of adaptive control methodologies, such as variable admittance control, which allow robots to modulate their behavior in response to environmental and user-specific factors. These techniques guarantee that robots remain responsive and compliant during interactions, particularly in unstructured environment or physically demanding tasks. The effectiveness of these systems has been validated through a series of experiments that have demonstrated improvements in user autonomy, task efficiency, and safety in a wide range of applications, including collaborative industrial tasks. Through meticulous experimental verification and an emphasis on practical applicability, this thesis illustrates the potential of shared control

systems to improve human-robot collaboration in significant ways. In addition to experimentally validating current shared control methodologies, the contributions aim to elucidate the transformative role of bioelectric signals in the realization of intuitive robotic systems. These advances have the potential to enable more continuous, personalized robotic assistance and efficient human-robot interaction in industrial settings.

The chapters that follow present the thesis in the following way:

- **Chapter 2 - Bioelectrical Signals:** this chapter introduces the concept of bioelectrical signals, with a particular focus on two primary types: electroencephalography and electromyography. It explains how these signals can be processed to extract characteristic features, which are then utilized by classifiers to distinguish among classes that are subsequently employed to control external devices.
- **Chapter 3 - Background:** the background chapter establishes the theoretical foundation for the thesis, encompassing fundamental robotics concepts such as kinematics and dynamics. It introduces the concepts of rigid body position, inverse kinematics for robotic manipulators, and differential kinematics for mobile robots. These concepts are fundamental to an understanding of how robots can interpret human intentions through bioelectrical signals and execute physical tasks.
- **Chapter 4 - Shared Control Applications for Assistive Robotics:** this chapter examines the implementation of shared control in assistive robotics, with a particular emphasis on mobile robots and robotic manipulators. It presents a shared control architecture based on motor imagination through brain-computer interfaces, which enables users to control robots using their electroencephalogram (EEG) signals. The effectiveness of this system is validated through experiments with mobile robot navigating obstacle environments and robotic arm for object manipulation.
- **Chapter 5 - Shared Control Application for Human-Robot Interaction:** the emphasis in this section is on the topic of human-robot interaction, particularly in the context of tasks that require physical collaboration. The chapter provides a detailed account of

the use of EMG signals in the handle of collaborative manipulation tasks between humans and robots. A shared control architecture enables the robot to adapt its behaviour in real time to the user's muscle activity, thereby facilitating a smooth and safe collaboration. The experimental setups serve to validate the approach, demonstrating its applicability in real-world scenarios.

- **Chapter 6 - Conclusions and Future Works:** the final chapter provides a synthesis of the thesis's findings and offers insights into future avenues of inquiry for shared control systems. It underscores the potential for enhancing autonomy in assistive robotics through the continued enhancement of bioelectric signal processing and the investigation of novel applications of shared control in more complex environments. A comprehensive overview of the shared control methodologies explored and developed throughout the thesis is provided, with a particular emphasis on assistive applications and human-robot collaboration.

The research activities described in this thesis have resulted in the publication and submission of the following papers:

- F. Patriarca, G. Gillini, P. D. Lillo, and F. Arrichiello, "Null-space based shared control of a mobile robot using motor imagery based brain-computer interface," in 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2022, pp. 2263–2268.
- F. Patriarca, P. D. Lillo, and F. Arrichiello, "Emg-based shared control framework for human-robot co-manipulation tasks," *in Proceedings of* ICINCO 2024 – 21st International Conference on Informatics in Control, Automation and Robotics, 2024.
- F. Patriarca, P. D. Lillo, and F. Arrichiello, "Emg-driven shared control architecture for human-robot co-manipulation tasks," *Submitted to* IEEE Transactions on Human-Machine Systems, 2024.

Chapter 2

Bioelectrical Signals

Bioelectric signals are defined as the electrical impulses generated by living cells, particularly neurons and muscle fibers. These signals play a crucial role in the functioning of biological systems and could be used in various physiological processes, including movement, sensation, and cognition. The study and application of these technologies have become increasingly important for the development of advanced human-machine interaction systems. The ability to capture, analyze, and interpret these signals has led to significant advances in medical diagnostics, neurotechnology, and robotics, paving the way for more intuitive and responsive control systems that can interact with the human body in a natural and efficient manner. In the field of robotics, bioelectric signals refer to the electrical activities generated by biological systems, particularly in the context of human-machine interaction. Signals such as electromyography and electroencephalography, which represent the electrical activities of the brain and muscles, respectively, are crucial for enabling robots to interpret human intentions and facilitate rehabilitation processes. In the field of human-machine interfaces, there is a growing trend of utilizing bioelectric signals to develop systems that enhance the responsiveness and adaptability of control mechanisms, due to the fact that these signals facilitate the translation of human intentions into machine actions. This capability is of particular significance in applications that necessitate precise, real-time control, including assistive technologies, rehabilitation devices, and advanced robotics. By leveraging bioelectric signals, researchers can create systems that closely mimic natural human movements and responses, thereby enhancing the usability and effectiveness of these technologies [1].

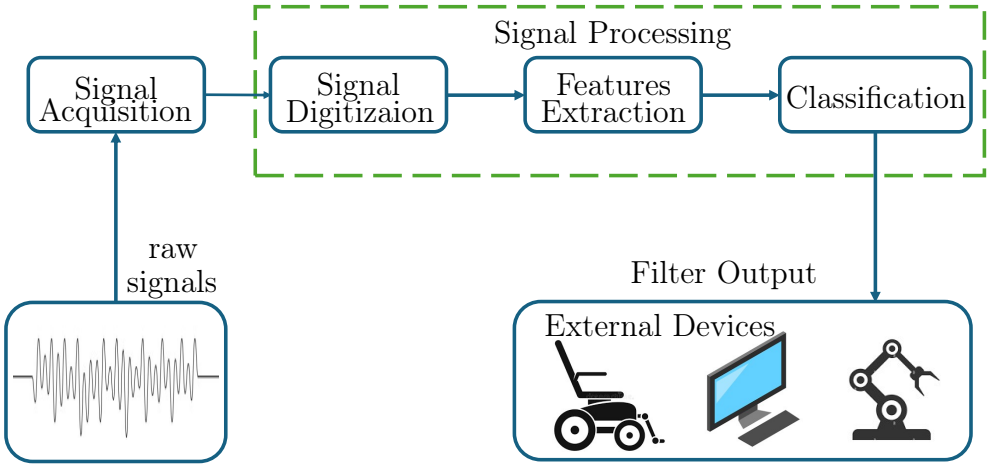


Figure 2.1: Block diagram of the general processing that raw bioelectric signals must undergo in order to be converted into commands that can be sent to external devices.

In any case, it is essential that all bioelectric signals are collected and processed in an appropriate manner. In general, following an initial acquisition phase, during which the signals are collected, they are digitized and processed to extract some characteristic elements of the signals, called *features*. Subsequently, the extracted features are sent to a classifier, which is employed to distinguish between different classes of the signals to translate into motion commands for external devices. A graphical representation of the general processing scheme of bioelectric signals is provided in Figure 2.1.

2.1 Electroencephalography (EEG)

Electroencephalography (EEG) is a technique that records the electrical activity of the brain by measuring voltage fluctuations resulting from ionic currents within the brain's neurons [140], providing valuable insights into the functioning of the nervous system and in diagnosing functional disturbances within the central nervous system.

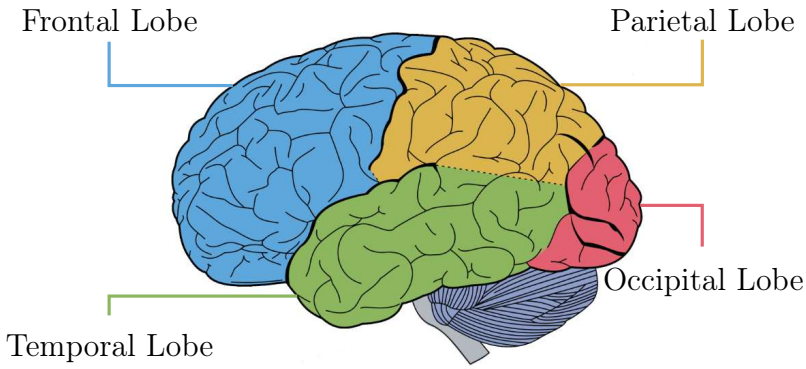


Figure 2.2: Cerebral cortex lobes.

2.1.1 Nervous System

The nervous system is composed of a variety of organs, including the brain, spinal cord, and sense organs such as the eye and ear [141]. These organs process information from external and internal stimuli and transmit signals to initiate responses. The functional units of the nervous system are neurons, which are specialized for intercellular communication. The nervous system is divided into two parts: the Central Nervous System (CNS), which includes the brain and spinal cord and integrates, processes, and coordinates sensory information and motor commands; and Peripheral Nervous System (PNS), which consists of nerves external to the CNS and includes the somatic and autonomic nervous systems that provide sensory information to the CNS and transmit motor commands to peripheral tissues and systems.

The CNS includes the brain, composed of two hemispheres connected by the corpus callosum, and the thalamic nuclei, which send sensory information to the basal nuclei and cerebral cortex. The cerebral cortex is divided into different lobes, as illustrated in Figure 2.2:

- Frontal Lobe, that contains the primary motor cortex, responsible for voluntary movements, attention, short-term memory tasks, planning, and motivation;
- Parietal Lobe, that integrates sensory information, spatial sense, navigation, and touch;

- Temporal Lobe, that processes sensory input, visual memories, language comprehension, and emotion.
- Occipital Lobe, that integrates visual information for image recognition and elaboration.

2.1.2 EEG signals acquisition

EEG signals are generated by the synchronous firing of cortical neurons and are fundamental to the understanding of brain function, the diagnosis of neurological disorders, and the development of brain-computer interfaces [142]. Clinically, EEG refers to the recording of the spontaneous electrical activity of the brain over a period of time, recorded by multiple electrodes placed on the scalp. An internationally accepted method for describing and applying the position of electrodes on the scalp is the 10-20 system, shown in Figure 2.3. This system is based on the fact that the effective distances between adjacent electrodes are 10% or 20% of the total front-to-back or right-to-left distance of the skull.

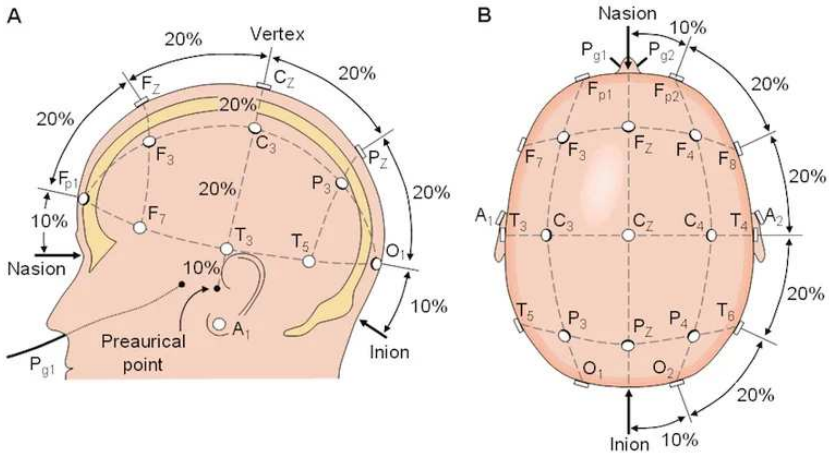


Figure 2.3: The 10-20 System. It is an internationally recognized method to describe and apply the location of scalp electrodes in the context of an EEG exam, polysomnograph sleep study, or voluntary lab research.

EEG signals can be classified into different frequency bands, each of which is associated with different mental states and cognitive processes [142,

143]. There are five widely recognized brain waves [144], referred to as delta, theta, alpha, beta, and gamma waves, and the main frequencies of human EEG waves are listed in Table 2.1 along with their characteristics. In order to integrate these signals into robot control loops, it is necessary to acquire, process and classify processes to convert them into commands for robotic systems.

EEG Waves		
Frequency band	**Frequency**	**Brain states**
Gamma γ	> 35 Hz	Concentration
Beta β	$12 - 35$ Hz	Anxiety dominant, active attention
Alpha α	$8 - 12$ Hz	Very relaxed, passive attention
Theta θ	$4 - 8$ Hz	Deep relaxed
Delta δ	$0.5 - 4$ Hz	Sleep

Table 2.1: The five types of brainwaves commonly recognized.

There are multiple paradigms that can be employed to acquire signals from the human brain. One may utilize event-related potentials (ERPs), which are very small voltages generated in brain structures in response to specific events or stimuli [145]. In humans, these fall into two categories: sensory ERPs, which are the initial waves or components that peak approximately in the first 100 milliseconds after the stimulus; and cognitive ERPs, which are generated in the later parts and reflect how the subject evaluates the stimulus. Among cognitive ERPs, the P300 wave is an elicited component in decision-making processes [146] that manifests as a positive voltage deflection with a latency (delay between stimulus and response) of approximately 250-500 ms [147]. In contrast, the Steady-State Visually Evoked Potential (SSVEP) represents a response to periodic visual stimuli. The spectrum of this type of potential is composed of constant peaks corresponding to the frequency of the stimulus and its harmonics. The most common SSVEP stimulation pattern is the On/Off stimulus, where the user focuses the attention on a blinking object at a constant frequency [148]. An alternative paradigm is provided by **motor imagery**, which will be referenced in following chapters. Motor imagery refers to the cognitive process of simulating a movement [147] without

actual physical execution, which generates distinct patterns of electrical activity in the brain that can be detected by electroencephalography (EEG). Specifically, it is a cognitive process in which a subject imagines performing a movement without actually executing it, or, in other words, the specific motor action is activated without any motor output. Accordingly, this paradigm necessitates the activation of brain regions implicated in the planning and execution of the movement. Motor imagery is associated with the specific activation of the neural circuits involved in the early stage of motor control, which includes the supplementary motor area, primary motor cortex, inferior parietal cortex, basal ganglia, and cerebellum. This mental trial engages the same neural pathways that would be activated during actual movement, thereby facilitating motor learning and enhanced performance. The research demonstrates that motor imagery engages brain regions that are analogous to those implicated in actual motor execution. For example, studies employing functional imaging techniques have demonstrated that the primary motor cortex undergoes alterations in excitability during motor imagery, analogous to those observed during actual movement [149, 150]. This indicates that this paradigm not only engages cognitive processes but also induces physiological responses that mirror those of physical actions, thereby reinforcing the concept of functional equivalence between imagined and performed movements [151].

2.1.3 Brain Computer Interface (BCI)

Brain-computer interfaces are sophisticated systems that enable direct communication between the brain and external devices by interpreting brain signals to control actions without the need for muscle activity. These interfaces are particularly beneficial for individuals with disabilities, as they facilitate the use of computers, control of robotic limbs, and communication through speech synthesis, all through the act of thinking about the desired action. In addition to assistive technologies, BCIs are also being explored for the purpose of enhancing human capabilities in areas such as gaming, virtual reality, and teleoperation, where they can provide more immersive and responsive experiences. The development of BCIs is a multidisciplinary effort involving advances in neuroscience, signal processing, machine learning, and robotics with the goal of improving the speed, accuracy, and usability of these systems. The recent

advancements in adaptive algorithms and machine learning have significantly improved the performance of BCIs, making them more practical for everyday use and expanding their potential applications.

Signals from the brain are extracted and processed in accordance with the adopted paradigm, which classifies them into operations to be sent to an external device, such as a computer to display the result or a robot to control its movement. Based on the placement of electrodes, three types of BCI can be distinguished:

- Invasive BCIs, in which electrodes are implanted directly under the scalp, offering the advantage of providing more accurate readings of brain signals;
- Partially invasive BCIs, in which electrodes are implanted inside the skull but are located outside the brain rather than inside the gray matter;
- Non-invasive BCIs, in which electrodes are placed on the scalp, and are the ones considered in this paper.

To classify signals recorded with a BCI, the signal must undergo a series of processing steps, as illustrated in the block diagram in Figure 2.1. The process begins with the *acquisition* of the signals, that are then amplified and digitized. This is followed by *signal processing*, which occurs in accordance with the paradigm being considered, and then the *translation algorithm*, where commands for *output devices* are classified according to the extracted features after processing EEG signals. The output device then executes the previously classified desired command and, finally, an *operating protocol* decodes the operations to be performed according to the desired command.

2.2 Electromyography (EMG)

Electromyography is a diagnostic procedure utilized to evaluate and record the electrical activity generated by skeletal muscles. It belongs to the field of neurophysiology and allows for the analysis of muscle activity at rest and during voluntary muscle activation through the use of needles and electrodes. This technique allows for the study of the peripheral nervous system, described in Section 2.1.1, from a functional

perspective, and it is of considerable importance in the diagnosis of neurological diseases and as an aid in muscle rehabilitation. Electromyographic activity can be used to decode a movement or the intention to perform a movement, and is used to gather information about biological signals, which returns a measure of the sum of muscle activation potentials within human muscles.

The analysis of the electrical impulse from muscles is conducted using an electromyograph, an electronic device capable of translating muscle activity into graphic form. Two distinct types of electromyography can be identified:

- Surface electromyography (sEMG), which permits the examination of the muscle's electrical signal through the placement of electrodes on the surface of the skin for the purpose of detecting muscle activity in a non-invasive manner [152];
- Needle electromyography (ENG), which enables the analysis of muscle electrical activity via subcutaneous needle electrodes [153].

Electrical signals recorded during electromyography reflect the depolarization and repolarization of muscle fibers, thereby providing insights into muscle activation patterns during rest and movement [154].

Electromyography technology is widely used in the development of human-robot interfaces, where it plays a pivotal role in enabling robots to assist in tasks that require fine motor control. In the context of rehabilitation, for instance, EMG-based exoskeletons are employed to facilitate the recuperation of motor function by enabling patients to respond to the user's muscle activation patterns [155, 156]. Furthermore, the integration of EMG signals into robotic systems can enhance the precision of movement control, thereby improving the overall efficacy of rehabilitation therapies [157]. Additionally, EMG is employed in sports science and ergonomics to optimize performance and reduce the risk of injury by providing detailed information about muscle function during various activities [158].

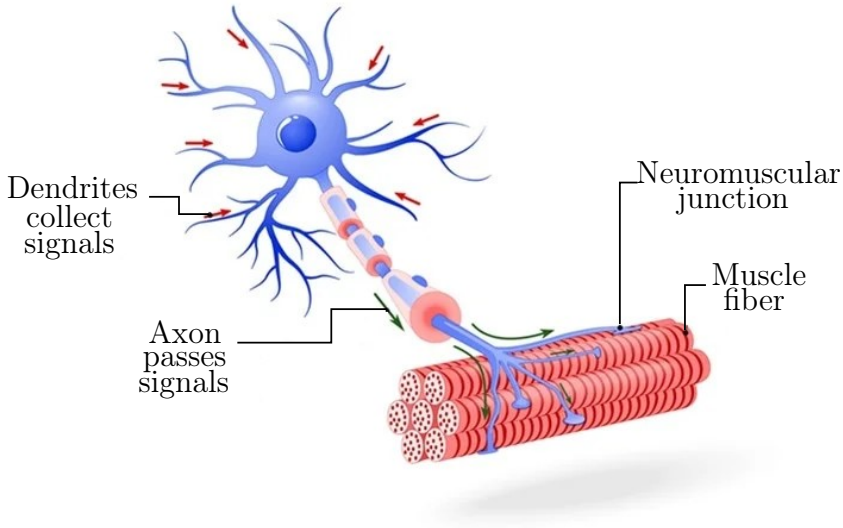


Figure 2.4: Graphical representation of the Motor Unit.

2.2.1 Electromyographic signals

EMG signals are obtained from the sum of the action potentials emanating from each of the muscle fibers that make up the motor unit, also called MUAP (Motor Unit Action Potential). The Motor Unit (MU), on the other hand, is composed of a collection of motoneurons[1] and the muscle fibers they innervate, as shown in the Figure 2.4.

The generation of EMG signals is mainly due to the depolarization of motor units: when a muscle contracts, action potentials are generated that lead to the release of neurotransmitters at the neuromuscular junction, which in turn causes muscle fibers to contract. This electrical activity can be recorded and analyzed to assess muscle performance and fatigue. For example, studies have shown that the mean frequency of the EMG signal decreases with muscle fatigue, indicating a change in muscle fiber recruitment patterns during sustained contractions [159]. As shown in the Figure 2.4, muscle fibers, or fibrocells, are cylindrical muscle cells

[1] A motoneuron is a type of neuron whose cell body is located in the motor cortex, brainstem, or spinal cord. Its axon (fiber) projects to the spinal cord or extends beyond it to directly or indirectly control effector organs, primarily muscles and glands.

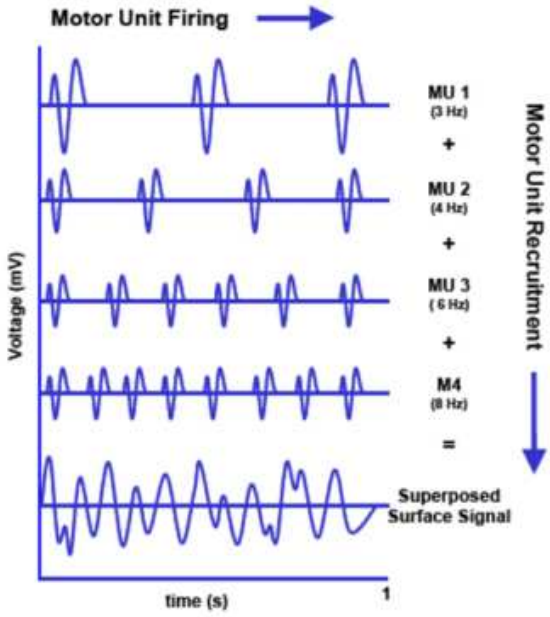


Figure 2.5: The recruitment of four motor units.

of small diameter (10 to 100 μm) and variable length that are interconnected by connective tissue to form the entire muscle. An EMG signal is represented as an oscillating wave on a time axis, exhibiting characteristics such as amplitude, duration, and frequency. The electromyographic signal, obtained through the use of needle or surface electrodes, is then amplified and filtered. Following this, it undergoes analog-to-digital conversion, resulting in the appearance of a sine wave of a random nature on an interface, as illustrated in the scheme in the Figure 2.1.

To achieve the most accurate detection possible, surface electrodes must be placed in an appropriate manner on the muscle of interest because the amplitude and frequency content of the measured signals change depending on the positioning of the electrode relative to the muscle fiber of the motor unit. The force exerted during voluntary contraction of a muscle is dependent on two parameters: the *firing* rate or frequency, that is the frequency of muscle activation, and the number of motor units engaged in the motor activity, also referred to as UM *recruitment*. The greater the number of motor units involved, the greater the force exerted. Figure 2.5 illustrates an example of a signal in which four motor units are recruited.

The firing rate is dependent upon the specific task, the duration of the contraction, and the magnitude of the muscle contraction. Firing represents a stimulus transmitted from the brain to the spinal cord, which propagates it along the axon connected to the muscle fibers. The muscle fibers respond to firing by generating action potentials at various points in the muscle. The sum of these potentials is ultimately detected by the electrode and the resulting muscle contraction occurs irregularly in both duration and proportion, rendering this parameter a random variable. The resulting muscle contraction is irregular in duration and magnitude, and for this reason this parameter can be considered a random variable. These parameters vary depending on the muscle involved, the speed of execution of the movement, and muscle fatigue. Each EMG signal is unique: it is impossible to obtain the same signal even when the same force, speed, and motor task characteristics are employed. This phenomenon can be attributed to the intrinsic randomness of the EMG signal, which is influenced by several factors. These include the irregular firing patterns of motor units, the stochastic nature of muscle fiber stress responses, and the diverse characteristics of muscle fiber recruitment. Other factors that could affect the EMG signal include the conductivity characteristics of skin tissue, physiological conditions and body temperature, cross talk from muscles adjacent to electrode placement [160], and external noise due to other external electromagnetic sources that could affect the EMG signal. Techniques such as wavelet filtering have been developed to enhance the quality of EMG signals by reducing noise and artifacts, thereby improving the fidelity of the collected data [161]. Accurate interpretation of EMG signals is crucial for both clinical diagnostics and the development of advanced technologies that rely on muscle activity for control and feedback.

2.2.2 EMG Sensors

Electromyography (EMG) sensors are utilized to detect and record the electrical activity of muscles during both voluntary and involuntary muscle activity, with the aim of analyzing muscle function and movement. The main function of EMG sensors is to provide data in real time on muscle activity, which can be used for both clinical and research purposes. For example, surface electromyography (sEMG) is a widely utilized method for the assessment of muscle activation times, particularly in

dynamic movements [162]. These sensors allow for the continuous monitoring of muscle activity in a wide range of contexts, including athletic training and rehabilitation therapy. They are also an essential component in the design of prosthetic limbs, exoskeletons, and rehabilitation devices, enabling users to control these devices based on their muscle activity. Recent advances in sensor technology have resulted in the development of more sophisticated and user-friendly EMG sensors, including wireless and stretchable devices that enhance comfort and usability for patients [163, 164].

In accordance with the specific electromyography procedure being conducted, the utilization of needle electrodes or surface electrodes may be required. The latter can be subdivided into two categories: active and passive electrodes. The distinction between these categories is based on the presence of a pre-amplification circuit, which serves to reduce input noise. The use of active sensors obviates the necessity for signal amplification and filtering procedures, due to the presence of a pre-amplification circuit. Furthermore, the selected positioning surface determines the categorization of electrodes into three distinct groups:

- Monopolar, in which two electrodes are used, one of which detects the signal and another that serves as a reference. However, the use of these electrodes results in the detection of all electrical signals within the affected region, including those that are not directly associated with muscle activity and, therefore, constitute a source of interference or noise.
- Bipolar, that include three electrodes: two that detect electrical impulses and a third that serves as a reference in an area external to the detection zone. This configuration allows for a significant reduction in the noise that is superimposed on the signal.
- Multipolar, which incorporates multiple electrodes for the detection, facilitating a more precise acquisition process and reducing noise as much as possible.

Moreover, the integration of EMG sensors with other technologies, such as inertial measurement units (IMUs) and machine learning algorithms, has led to significant enhancements in the precision and utility of muscle activity assessments.

Chapter 3

Background

This chapter offers an overview of the concepts of kinematics and dynamics as they pertain to robotic systems. In particular, it addresses the kinematics and dynamics of robotic manipulators, differential kinematics, and the prioritization of multiple tasks to be performed.

3.1 Kinematics

The field of *kinematics* is concerned with the geometrical aspects of motion. Before describing the kinematic model of a robotic system, it is important to consider that it can be schematized as a kinematics chain of rigid bodies. In this context, the overall motion can be understood as the composition of the elementary motions of all the rigid bodies that comprise it.

3.1.1 Rigid Body Position and Orientation

A rigid body, which is an ideal representation of a body that does not deform or change shape, can be described in space as a function of its position and orientation, also referred to as *pose*, with respect to a reference frame. Accordingly, the pose of a rigid body with respect to an inertial frame $\boldsymbol{\mathcal{F}_I}$ is defined as follows:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} & \boldsymbol{\varphi} \end{bmatrix}^T \in \mathbb{R}^6 \tag{3.1}$$

where $\boldsymbol{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T \in \mathbb{R}^3$ and $\boldsymbol{\varphi} \in \mathbb{R}^3$ are the position and orientation vectors, respectively.

With regard to orientation, there are numerous methods for its representation. One of the most widely recognized is that of Euler's angles, which employs the following notation for orientation:

$$\boldsymbol{\varphi} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \in \mathbb{R}^3 \tag{3.2}$$

where the angles ϕ, θ and ψ correspond to *roll*, *pitch* and *yaw* angles, respectively. This is a minimal representation of orientation, defined by three independent parameters.
Given a base frame $\boldsymbol{\mathcal{F}_B}$, it is possible to obtain the corresponding rotation matrix using the ZYX-convention in the current frame as follows:

$$\boldsymbol{R}_B^I = \begin{bmatrix} c_\psi\, c_\theta & -s_\psi\, c_\phi + c_\psi\, s_\theta\, s_\phi & s_\psi\, s_\phi + c_\psi\, c_\phi\, s_\theta \\ s_\psi\, c_\theta & c_\psi\, c_\phi + s_\phi\, s_\theta\, s_\psi & -c_\psi\, s_\phi + s_\theta\, s_\psi\, c_\phi \\ -s_\theta & c_\theta\, s_\phi & c_\theta\, c_\phi \end{bmatrix} \tag{3.3}$$

where $c_* = cos(*)$ and $s_* = sin(*)$. However, this representation is affected by a problem known as the *singularity representation*, which results in the conversion from the rotation matrix to the Euler angle representation being undefined when θ assumes values such as $\pm\frac{\pi}{2}$ rad. Consequently, in some cases, this representation is not suitable and it may be necessary to employ a non-minimal representation, such as the *Unit Quaternion*.
The Unit Quaternion is a non-minimal set of four parameters for representing the orientation of a rigid body. Let consider the orientation between two frames with the same origin, which is defined as follows:

$$\boldsymbol{R}_e(\alpha) = cos\alpha \boldsymbol{I}_{3\times 3} + (1 - cos\alpha)\boldsymbol{e}\boldsymbol{e}^T - sin\alpha \boldsymbol{S}(\boldsymbol{e}) \tag{3.4}$$

where $\boldsymbol{e}$ is the unit vector that identifies the axis in which a rotation of the α angle is required to align the two frames, $\boldsymbol{I}_{3\times 3}$ is the identity matrix of order 3 and $\boldsymbol{S}$ is the skew matrix, that performs the cross product between two (3×1) vectors. The unit quaternion is defined as

$$\mathcal{Q}(\eta, \boldsymbol{\varepsilon}) \tag{3.5}$$

where

$$\eta = cos\frac{\alpha}{2}, \quad \boldsymbol{\varepsilon} = \boldsymbol{e}sin\frac{\alpha}{2} \tag{3.6}$$

are the scalar and vector part respectively, satisfying the following condition:

$$\eta^2 + \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = 1. \tag{3.7}$$

Furthermore, since the two quaternions $\mathcal{Q}(\eta, \boldsymbol{\varepsilon})$ and $\mathcal{Q}(-\eta, -\boldsymbol{\varepsilon})$ represent the same orientation, in order to guarantee the representation's uniqueness, in Eq. (3.6) it is assumed that $\eta \geq 0$ corresponds to $\alpha \in [-\pi; \pi]$.

3.2 Differential Kinematics for a Robotic Manipulator

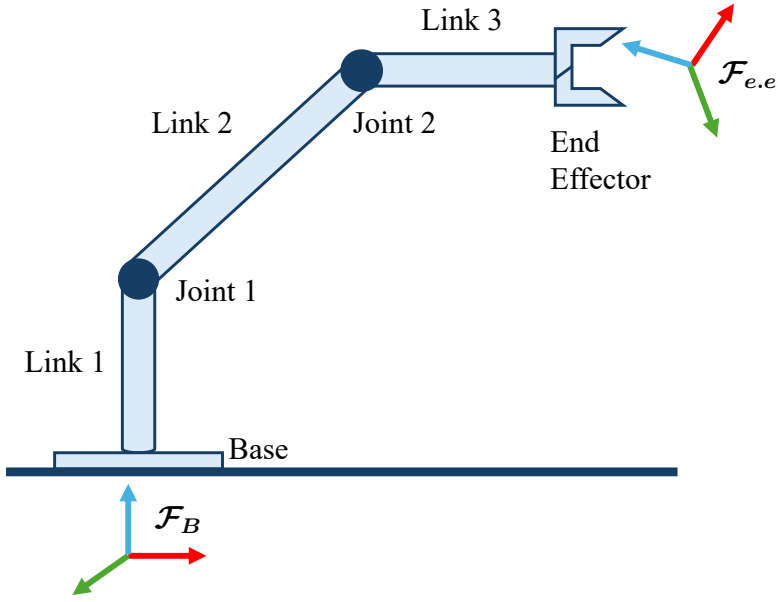


Figure 3.1: Example of robot manipulator with 3 links.

A manipulator can be considered as a kinematic chain consisting of a series of rigid bodies (links) connected by mechanical pairs or joints (as shown in Figure 3.1). One end of the manipulator is a constrained base, while the other end is an end-effector (e.e.), which enables the manipulation of objects in space. The robots structure is characterized by a number

of degrees of freedom (DOFs) which determine its configuration. Each degree of freedom is associated with a joint, with a variable linked to it.

The direct kinematics model describes the relationship between the joints variables $\boldsymbol{q}$, which is expressed in the *joint space*, and the end-effector pose $\boldsymbol{x}_{ee}$ in the *operational space*. Thus, the direct kinematics is expressed as follows:

$$\boldsymbol{x}_{ee} = \boldsymbol{k}(\boldsymbol{q}), \tag{3.8}$$

where the vector $\boldsymbol{x}_{ee}$ is of dimension $m \times 1$, the vector $\boldsymbol{q}$ is of dimension $n \times 1$, and the non-linear function $\boldsymbol{k}$ allows the operational space variables to be calculated on the basis of the joint space variables.

Since the relationship expressed in Eq. (3.8) is non-linear, the calculation of the e.e. configuration $\boldsymbol{x}_{ee}$ is not trivial. For relatively simple robotic structures, several geometric approaches can be employed. However, when the number of degrees of freedom (DoFs) is large, the number of potential solutions can be infinite. To address this issue, differential kinematics is used to identify the relationship between the joint velocities $\dot{\boldsymbol{q}}$ and the linear velocities $\boldsymbol{v}_{ee}$ and angular velocities $\boldsymbol{\omega}_{ee}$ of the end-effector. This relationship can be expressed as follows:

$$\begin{bmatrix} \boldsymbol{v}_{ee} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.9}$$

with $\boldsymbol{J}(\boldsymbol{q})$ the Jacobian matrix that maps the n-dimensional joints velocities into m-dimensional end-effector velocities.

3.3 Differential Inverse Kinematics

In the case of a general robotic system with n degrees of freedom, the state is described by the joint values $\boldsymbol{q} = \begin{bmatrix} q_1, & q_2, & \ldots, & q_n \end{bmatrix}^T \in \mathbb{R}^n$. A task may be defined as a generic m-dimensional control objective as a function of the system state $\boldsymbol{\sigma}(\boldsymbol{q}) \in \mathbb{R}^m$. The inverse kinematics problem may therefore be defined as the problem of identifying the $\boldsymbol{q}$ vector that brings $\boldsymbol{\sigma}(\boldsymbol{q})$ to a desired value $\boldsymbol{\sigma}_d$:

$$\boldsymbol{\sigma}(\boldsymbol{q}) = \boldsymbol{\sigma}_d. \tag{3.10}$$

As the aforementioned equation is non-linear, it is not feasible to determine the solution by simply inverting the relation. By differentiating equation (3.10) with respect to time, the following can be derived:

$$\dot{\boldsymbol{\sigma}}(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{3.11}$$

where $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\sigma}(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{m \times n}$ represents the task Jacobian matrix, whereas $\dot{\boldsymbol{q}}$ represents the system velocity vector. This enables the establishment of a linear mapping between the task-space velocity and the system velocity.

In order to compute the system velocity corresponding to the specific task time derivative, it is necessary to invert the linear mapping presented in equation (3.11). In the case where the rank of $\boldsymbol{J}$ is full, it is possible to invert the relationship directly, obtaining

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{\sigma}}(\boldsymbol{q}). \tag{3.12}$$

The digital implementation of Eq. (3.12), which relies on a numerical integration procedure to obtain joint positions, is susceptible to numerical drift. This issue can be addressed through the use of the Closed-Loop Inverse Kinematics (CLIK) algorithm [165], which establishes a correlation between the system velocity calculation and the error associated with the task:

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}, \tag{3.13}$$

where $\boldsymbol{\sigma}_d$ is the desired task value. The equation (3.12) can be rewritten as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{K}\tilde{\boldsymbol{\sigma}}) \tag{3.14}$$

where $\boldsymbol{K}$ is a positive-definite matrix of gains and $\dot{\boldsymbol{\sigma}}_d$ is the desired task velocity.

Indeed, when a robot manipulator is redundant, the matrix $\boldsymbol{J}$ is not square, and thus inversion is not possible. A robotic system is defined as redundant if the number of degrees of freedom exceeds the minimum required to accomplish a specific task. In this case, the Jacobian matrix is no longer invertible and equation (3.9) admits infinite solutions. Therefore, kinematic redundancy can be exploited to perform other tasks.

In this case, a constrained minimization problem is imposed in order to solve the system, in which the cost function is given by the following quadratic function [166]:

$$g(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}}. \tag{3.15}$$

The above function identifies the solution that minimizes the joint velocity norm among all possible options and the equation (3.12) becomes:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{\sigma}}(\boldsymbol{q}) \tag{3.16}$$

where $\boldsymbol{J}^{\dagger}(\boldsymbol{q}) = \boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))^{-1}$ is the Moore-Penrose pseudoinverse of the Jacobian matrix. Also in this case, the digital implementation of the equation results in a numerical drift, thus equation (3.14) can be written as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{K}\tilde{\boldsymbol{\sigma}}). \tag{3.17}$$

3.4 Kinematic Singularities

The manipulator joint configurations at which the matrix $\boldsymbol{J}$ is rank deficient, i.e., $rank(\boldsymbol{J}) < m$, are referred to as *kinematic singularities*. When the system is in proximity to these specific configurations, a small increment in the task value can result in high joint velocities [167]. Furthermore, the structure mobility is reduced.

Kinematic singularities can be classified into two categories: those occurring at the boundary and those occurring within the workspace. In the latter case, three singularity types can be identified: shoulder, elbow, and wrist, respectively.

The kinematic singularity problem can be resolved by employing the Damped Least-Square (DLS) pseudoinverse, which is defined as follows:

$$\boldsymbol{J}^{\dagger\lambda}_{DLS}(\boldsymbol{q}) = \boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}) + \lambda^2\boldsymbol{I}_m)^{-1} \tag{3.18}$$

where $\lambda \in \mathbb{R}$ is a damping factor and $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ is the identity matrix.

3.5 Set-based Task Priority Inverse Kinematics

The system velocity obtained via (3.16) lies in the subspace $\boldsymbol{\mathcal{R}(J)}$. However, in the case of redundant systems, it is possible to compute system velocity that fulfills more control objectives simultaneously by exploiting the orthogonal complement $\boldsymbol{\mathcal{N}(J)} \neq \emptyset$ of $\boldsymbol{\mathcal{R}(J)}$. In this case, the goal of the control algorithm is to identify the system velocity that optimally fulfills all tasks, given the specified priority order. To combine a hierarchy of h tasks, it is essential to assign h levels of priority i, ranging from the primary one ($i = 1$) to the lowest priority one ($i = h$), and to sort them accordingly. Based on this assumption, the reference system velocity can be computed recursively, as illustrated in [168]:

$$\dot{\boldsymbol{q}}_h = \sum_{i=1}^{h} \left(\boldsymbol{J}_i(\boldsymbol{q})\boldsymbol{N}_{i-1}^{A}\right)^{\dagger} \left(\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{K}_i\tilde{\boldsymbol{\sigma}}_i - \boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}}_{i-1}\right), \tag{3.19}$$

where $\boldsymbol{N}_i^A$ represents the *null space* projector of the augmented Jacobian matrix, which is obtained by stacking all the task Jacobian matrices from task 1 to i in the following manner:

$$\boldsymbol{J}_i^A(\boldsymbol{q}) = \begin{bmatrix}\boldsymbol{J}_1^T(\boldsymbol{q}) & \boldsymbol{J}_2^T(\boldsymbol{q}) & \dots & \boldsymbol{J}_i^T(\boldsymbol{q})\end{bmatrix}^T, \tag{3.20}$$

and is defined as:

$$\boldsymbol{N}_i^A = \boldsymbol{I} - (\boldsymbol{J}_i^A)^{\dagger}\boldsymbol{J}_i^A. \tag{3.21}$$

$\boldsymbol{N}_0^A = \boldsymbol{I}$, $\dot{\boldsymbol{q}}_0 = 0$ and $\dot{\boldsymbol{q}}_i$ is the solution at i-th iteration, i.e., the reference system velocity fulfilling all the hierarchy tasks.

The tasks can be classified into two main categories: *equality-based* tasks and *set-based* tasks. Equality-based tasks have an exact desired value as the control objective. Examples of such tasks include laying the final effector. Set-based tasks, which are usually known as inequality constraints, have a set of possible values. Obstacle avoidance is an example of a set-based task. The latter are addressed by expressing the inverse kinematics problem in a sequence of quadratic programming problems and solving them through iterative algorithms [169, 170]. However, this approach is computationally intensive. In [171], set-based tasks are incorporated into

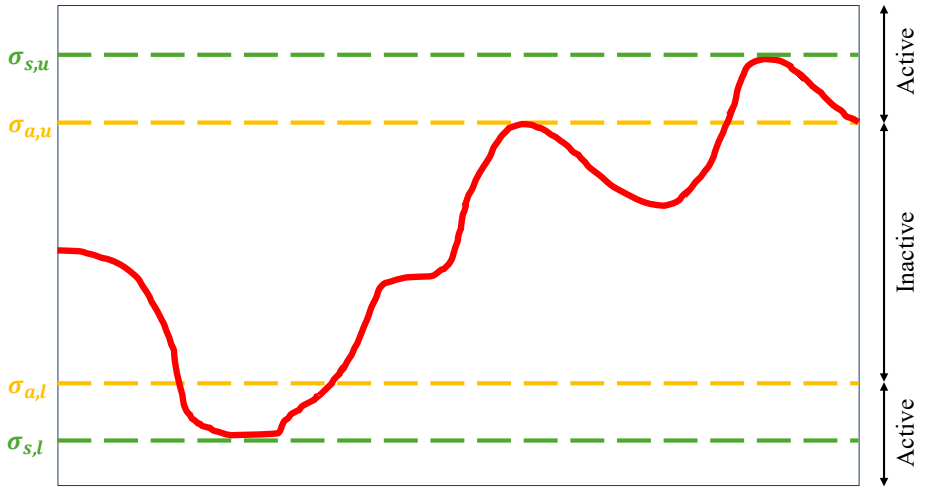


Figure 3.2: This figure presents the evolution of a generic set-based task, with particular attention to the activation thresholds ($\sigma_{a,u}$ and $\sigma_{a,l}$) and the safety thresholds ($\sigma_{s,u}$ and $\sigma_{s,l}$.)

the task priority structure using appropriate activation functions during task transitions to ensure the smoothness of the output reference trajectory. However, during transitions, the task priority is lost, and, therefore, safety constraints cannot be satisfied.

The Set-Based Task Priority Inverse Kinematics (SBTPIK) framework was developed for the purpose of addressing equality-based tasks. Thereafter, the framework has been extended to accommodate set-based tasks [172], which require the value of the task to remain within a specified range, as illustrated in Figure 3.2. This is exemplified by tasks such as the mechanical joint limits for a manipulator or obstacle avoidance for a mobile robot.

The fundamental concept is to regard a set-based task as an equality-based task that can be incorporated or excluded from the hierarchy in accordance with operational circumstances. Specifically, the set-based task must be activated when its value is near one of the *activation thresholds* ($\sigma_{a,u}$ or $\sigma_{a,l}$). It can be deactivated when the hierarchy solution, which does not contain it, would push its value toward the valid set. Once activated, the set-based task is entered into the task hierarchy as

an equality-based task with a desired value equal to a predefined *safety threshold* ($\sigma_{s,u}$ or $\sigma_{s,l}$).

The details of the activation/deactivation algorithm are provided in [173, 174]. In particular, a set-based task may be considered a one-dimensional equality-based task that is activated or deactivated in accordance with its current value. Upon exceeding the activation threshold, the value of the task should be incorporated into the task hierarchy as a novel equality-based task with the desired value:

$$\sigma_d = \begin{cases} \sigma_{s,u} & \text{if } \sigma \geq \sigma_{a,u} \\ \sigma_{s,l} & \text{if } \sigma \leq \sigma_{a,l} \end{cases} \tag{3.22}$$

The task may be deactivated when the solution of the hierarchy, which contains only the other tasks,would push its value toward the valid set. Specifically, it is possible to verify whether a generic solution $\dot{\boldsymbol{q}}$ would result in a task based on the σ_A sets exceeding the desired limit by evaluating its projection in the task space. By defining $\boldsymbol{J}_A$ as the Jacoboian matrix of σ_A, if $\boldsymbol{J}_A > 0$ the solution would increase the set-based value, otherwise if if $\boldsymbol{J}_A < 0$ the solution would decrease it. In this way, σ_A can be deactivated if

$$\sigma_A \geq \sigma_{a,u} \wedge \boldsymbol{J}_A \dot{\boldsymbol{q}} < 0 \tag{3.23}$$

or

$$\sigma_A \leq \sigma_{a,l} \wedge \boldsymbol{J}_A \dot{\boldsymbol{q}} > 0. \tag{3.24}$$

The aforementioned concept may be applied to a task hierarchy containing an arbitrary number of mixed set-based and equality-based tasks, resulting in the following four-step algorithm:

- Creation of the active task hierarchy;
- Computation of solutions;
- Projection of solutions;
- Choice of solution.

In order to create the active hierarchy $\boldsymbol{\mathcal{A}}$, which contains all equality-based tasks and all set-based tasks that exceed the activation thresholds, it is first necessary to construct the generic hierarchy $\boldsymbol{\mathcal{H}}$, which is

composed of both set-based and equality-based tasks. The pseudo-code associated with this algorithm is presented in Algorithm 1.

Algorithm 1 Create Active Task Hierarchy.

```
1: procedure CreateActiveTaskHierarchy(H)
2:     A ← ∅
3:     for each σ_k ∈ H do
4:         if σ_k is equality-based then
5:             A ← σ_k
6:         else if σ_k < MinActThresh or σ_k > MaxActThresh then
7:             A ← σ_k
8:         end if
9:     end for
10: end procedure
```

Since it is not possible to predict which solution will lead all set-based tasks to their valid ranges, it is essential to compute the solutions of all potential hierarchies that could be generated by inserting and removing all set-based tasks. Accordingly, a hierarchy comprising $\boldsymbol{h}$ active set-based tasks will yield $\mathbf{2}^h$ solutions, which can be stored in a set $\boldsymbol{\mathcal{S}}$. An example is presented in Figure 3.3.

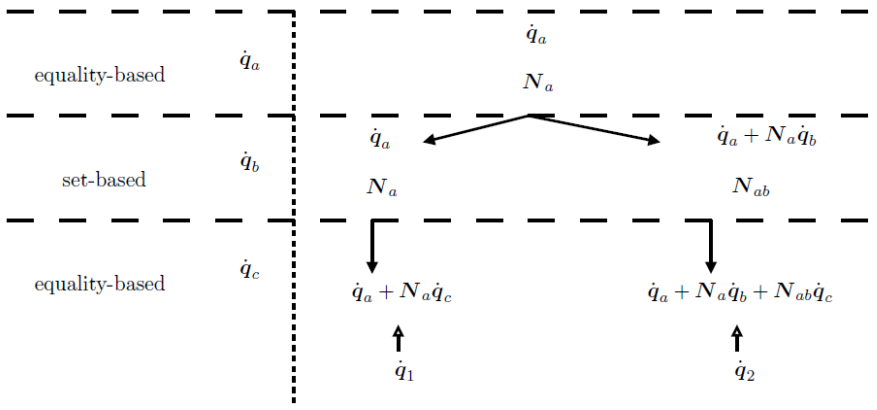


Figure 3.3: Solution tree example.

Subsequently, the solutions that satisfy conditions (3.23) or (3.24), among all the solutions in $\boldsymbol{\mathcal{S}}$, must be selected and stored in a set $\boldsymbol{\mathcal{P}}$. The procedure is outlined in Algorithm 2.

Algorithm 2 Select Solutions Satisfying Set-Based Tasks Limits.

```
1: procedure CreateActiveTaskHierarchy(A, S)
2:     P ← ∅
3:     count ← 0
4:     for each q_j ∈ S do
5:         for each σ_k ∈ A do
6:             if J_k q̇_j > 0   (J_k q̇_j < 0) then
7:                 count ← count +1
8:             else
9:                 break
10:            end if
11:        end for
12:        if count = n_a then
13:            S ← q̇_j
14:            count ← 0
15:        else
16:            count ← 0
17:        end if
18:    end for
19: end procedure
```

In the event that more than one solution is present within $\boldsymbol{\mathcal{P}}$, the highest-norm solution is selected as the desired system velocity $\dot{\boldsymbol{q}}_d$, since it is the less conservative velocity-wise. Should $\boldsymbol{\mathcal{P}}$ contain a single solution, this is chosen as the desired system velocity.
The Figure 3.4 shows the workflow of the proposed algorithm.
The green σ_i represents a generic set-based task, while the red σ_i represents a generic equality-based task and s_i represents a generic solution. The algorithm begins with a generic task hierarchy and ultimately leads to a unique solution that simultaneously accomplishes all equality-based and active set-based tasks.

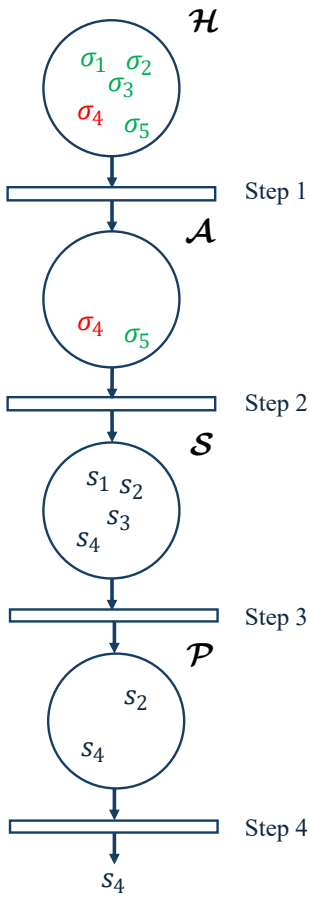


Figure 3.4: Workflow of the algorithm.

3.6 Differential Kinematics for a Non Holonomic Mobile Robot

Differential kinematics for a non-holonomic mobile robots is concerned with the study of the motion of robots with nonholonomic constraints. These are constraints that limit motion in certain degrees of freedom. To illustrate, consider a robot with two differential wheels that can move forward or backward, but not sideways (i.e., it cannot slide sideways).

The position in space of a non-holonomic robot is described by three parameters, which are defined as follows:

- x: the position along the x-axis;
- y: the position along the y-axis;
- θ: the orientation with respect to a reference axis (such as the x-axis).

Instead, the velocities associated with the robot are:

- v: linear velocity of the robot;
- ω: angular velocity of the robot.

Differential kinematics describes how the control velocities v and ω affect the variation of the robot's position and orientation over time:

$$\dot{x} = v\cos(\theta) \tag{3.25}$$
$$\dot{y} = v\sin(\theta) \tag{3.26}$$
$$\dot{\theta} = \omega. \tag{3.27}$$

The equations in (3.25) demonstrate that the motion along the two axes, x and y, is a function of its linear velocity, v, and its orientation, θ.

In the case of a two-wheel differential robot, the motion is dependent upon the angular velocities of the two wheels, designated as ω_R and ω_L, which correspond to the left and right wheels, respectively. The direct kinematics of the robot can be calculated by determining the linear and angular velocities, as follows:

$$v = \frac{r}{2}(\omega_R + \omega_L) \tag{3.28}$$
$$\omega = \frac{r}{l}(\omega_R - \omega_L). \tag{3.29}$$

whit r the radius of two wheels, and l the distance between the wheels. Thus, by employing the equations (3.28), it is possible to compute the linear and angular velocities as a function of wheel velocities. The value of v represents the average linear velocity, which is derived from the sum of the speeds of the two wheels, while the angular velocity ω is determined by the difference in velocity between the two wheels. Regarding the problem of differential inverse kinematics for a non-holonomic mobile robot, it is possible to calculate the angular velocities of the wheels by

simply inverting the equations of direct kinematics. Thus, the following expressions are obtained:

$$\dot{\theta}_L = \frac{2v - \omega L}{2r} \tag{3.30}$$

$$\dot{\theta}_R = \frac{2v + \omega L}{2r}. \tag{3.31}$$

As regard the non-holonomic constraint, it represents a restriction on motion that cannot be expressed as a relationship between the coordinates of the system. It also involves derivatives, specifically velocities. To illustrate, in the case of a robot with differential wheels that are incapable of lateral movement, the non-holonomic constraint will be expressed as follows:

$$\dot{y}\cos(\theta) - \dot{x}\sin(\theta) = 0. \tag{3.32}$$

This constraint indicates that the robot is only capable of moving in a forward or backward direction along the longitudinal axis, and it is unable to slide sideways.

3.7 Force Control in Robotic Manipulation

In the interaction between a manipulator and its environment, the ability to control the force applied by the manipulator is of critical importance for tasks that require soft manipulation, compliance, or interaction with uncertain environments. Force control enables the manipulator to adapt to external forces while maintaining a stable interaction with its environment. While traditional position control systems focus exclusively on the trajectory of motion, force control takes into account both the position of the robot and the forces exerted on it. Among the most important strategies for force control are impedance control and admittance control, which manage the dynamics of interaction between the robot and its environment. These strategies allow robots to respond to external forces by adjusting their behavior, modulating stiffness, damping, and inertia according to the level of compliance that is desired toward the external environment.

3.7.1 Impedance Control

The objective of impedance control is to adjust the dynamic relationship between the manipulator's motion (displacement, velocity, and acceleration) and the forces it experiences from the environment. The robot is represented as a mass-spring-damper system, characterized by inertia, stiffness and damping parameters and the objective of control is to emulate mechanical impedance, a property that characterizes the resistance of a system to motion when subjected to external forces. In other words, impedance control delineates the manner in which the manipulator should respond to external forces, specifying whether it should be "stiff" or "compliant". In impedance control, the desired behavior is specified in terms of mechanical impedance parameters, including:

- Stiffness ($\boldsymbol{K}_k$): this parameter governs the amount of force required to produce a given displacement. An increase in stiffness results in greater resistance to deviations from the desired trajectory, with greater resistance to displacement. Conversely, a decrease in stiffness leads to a greater degree of compliance when the manipulator interacts with an object.
- Damping ($\boldsymbol{K}_d$): adjusts the system's ability to dissipate energy, thereby facilitating smoother motion. Greater damping results in reduced oscillation.
- Inertia ($\boldsymbol{K}_m$): determines the amount of force required to change the manipulator's velocity, thereby affecting the system's dynamic response to external forces.

Thus, the general form of impedance control can be expressed by the following equation:

$$\boldsymbol{K}_m\ddot{\tilde{\mathbf{x}}} + \boldsymbol{K}_\mathbf{d}\dot{\tilde{\mathbf{x}}} + \boldsymbol{K}_\mathbf{k}\tilde{\mathbf{x}} = \boldsymbol{h} \ , \tag{3.33}$$

where $\boldsymbol{h}$ is the external force, $\boldsymbol{K}_m$ is the inertia matrix, $\boldsymbol{K}_d$ is the damping matrix, $\boldsymbol{K}_k$ is the stiffness matrix, $\tilde{\mathbf{x}} = \boldsymbol{x}_\mathbf{d} - \boldsymbol{x}$ and $\boldsymbol{x}, \boldsymbol{x}_d$ are the position of the manipulator and the desired position, respectively.

The block diagram that represents the impedance control is illustrated in the Figure 3.5.

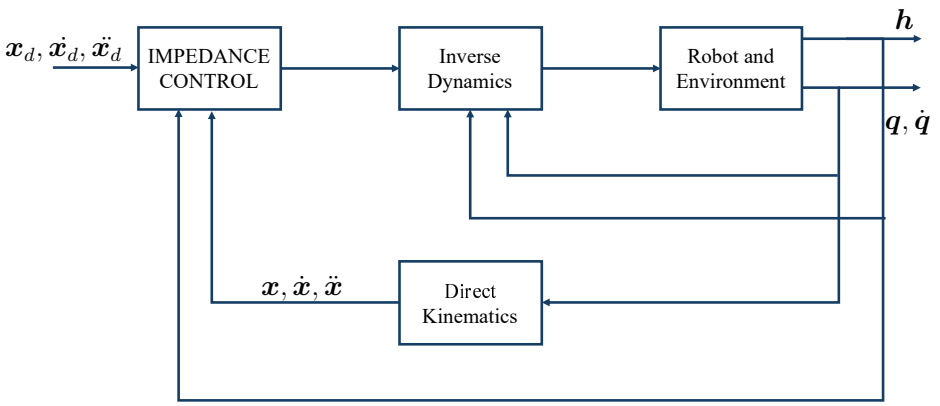


Figure 3.5: Genaral block diagram of impedance control.

One of the key challenges in implementing impedance control is the need to adjust the parameters in order to achieve an optimal balance between stability and performance. The application of high stiffness values can render the system difficult to control, particularly in the context of highly dynamic or unpredictable environments.

3.7.2 Admittance Control

Admittance control is conceptually the inverse of impedance control. Rather than modulating motion based on external forces, admittance control regulates how the manipulator responds to these forces, adjusting its position or velocity based on a desired force interaction. In this approach, the external force is employed as an input to the controller, which then calculates a corresponding motion that satisfies the desired dynamic relationship.

In admittance control, the motion of the manipulator is determined by the forces it encounters, thereby making it behave like an inertial system with compliance and damping characteristics. The motion of the system is adjusted in accordance with the external force, thereby facilitating more harmonious interactions with the external environment. At high admittance, the manipulator exhibits compliant behavior, moving significantly in response to small external forces. Conversely, at low admittance, the manipulator resists motion and requires larger external forces to induce significant motion, simulating a more rigid response.

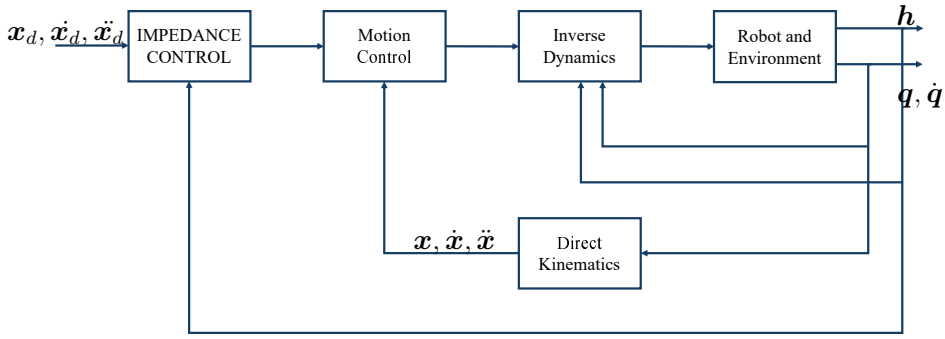


Figure 3.6: Genaral block diagram of admittance control.

The relationship governing admittance control can be expressed as follows:

$$\ddot{\tilde{\mathbf{x}}} = \boldsymbol{K}_{\mathbf{m}}^{-\mathbf{1}}(\boldsymbol{h} - \boldsymbol{K}_{\mathbf{d}}\dot{\tilde{\mathbf{x}}} - \boldsymbol{K}_{\mathbf{k}}\tilde{\mathbf{x}}) \,, \tag{3.34}$$

where $\boldsymbol{K}_m^{-1}$ is the inverse of the inertia matrix.

The block diagram that represents the admittance control is illustrated in the Figure 3.6.

Chapter 4

Shared Control Applications for Assistive Robotics

4.1 Introduction and Related works

This section focuses on the use of non-invasive brain-computer interfaces, which are headsets equipped with electrodes placed on the user's scalp that measure brain activity through EEG signals.

Several applications of assistive robotics, based on shared or supervisory control modes, involving human-machine interfaces have been studied to enable individuals with severe motor disabilities to perform daily-life activities, such as drinking or feeding [175]. The distinction between the control modes is based on the level of user involvement in the task at hand. As previously stated, in the shared control operational mode, the user is included in the control loop as an active part to generate high-frequency motion commands to be converted into low-level references for the robots, while meeting all robot and human constraints [176, 83]. On the other hand, in a supervisory control operational mode, the user has to send low-frequency commands to generate high-level directives to the system, such as start/stop actions, that are then automatically executed by the robotic system; thus, this operational mode results more suitable for applications that require a minimal involvement of the user in the direct motion control of the robot, while they require an higher level of autonomy for the robotic platform [177, 178].

The following focuses on shared control application based on the use of

motor imagery, with the aim of involving the user as much as possible. In order to achieve this, a number of approaches can be employed to control mobile robots with BCIs based on motor imagery. For instance, in [179] a BCI system based on motor imagery has been developed to allow 2D direct control of robot motions in an unknown environment. The BCI system acquires EEG signals from the human brain and classifies them into two user mental states, left-hand and right-hand motor imagery, while the robot control system provides four low-level motion commands: "going forward", "turning left", "turning right" and "stopping". A good robotic solution in assistive robotics scenarios would be a wheelchair teleoperated via brain signals. In [180], the authors have been developed a hybrid BCI architecture that combines motor imagery and P300 potential for the asynchronous operation of a brain-controlled wheelchair, in which the user can utilize up to eleven functions to control the wheelchair by sequentially performing motor imagery tasks and P300 tasks. A promising BCI application based on P300 paradigm is the teleoperation of robots, which opens the door to remote interactions and tasks that would otherwise be challenging or impossible [181]. Instead, in [182], a pioneering assistive shared control strategy for a robotic arm, directed via a motor imagery-based BCI, is presented. This architecture allows users to maintain partial control over the robotic arm while enabling the system to offer assistance when necessary.
Several studies have been conducted with the aim of improving the accuracy of motor imagery classification. For example, in [183], the authors employed a hybrid architecture comprising Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to classify motor imagery signals. To execute more complex tasks, it may be beneficial to consider a larger class number, as demonstrated in [184] and in [185], where a four class motor imagery session is adopted to control two different kind of mobile robots movements in the 3D space.

The remainder of the chapter explores how a shared control architecture that leverages motor-imagery BCIs can improve assistive robotic systems, particularly for mobile robots and robotic manipulators. By seamlessly integrating user intention with autonomous safety mechanisms, this architecture ensures user control while reducing risk. Experimental validations and performance analyses further demonstrate the feasibility and effectiveness of the proposed solutions in real-world service scenarios.

Shared control techniques have been widely explored to integrate user intent with system autonomy, particularly in assistive robotics, focusing on user autonomy, safety mechanisms, and task-specific optimizations. Several studies have introduced adaptive shared control frameworks that dynamically adjust the level of assistance based on user intent and task complexity. For example, the authors in [186] proposed an Intelligent Adaptive User Interface (iAUI) to improve the bandwidth of BCI-controlled robots. While this work includes automatic mode-switching, the adaptive tuning of assistance levels remains an area for potential improvement. Hybrid BCI systems combining multiple EEG paradigms have demonstrated improved task accuracy and flexibility. The authors in [187] employed steady-state visual evoked potentials (SSVEP) and motor imagery to control robotic arms, achieving higher success rates in pick-and-place tasks compared to pure BCI systems. Similarly, this work balances motor imagery with safety constraints, but focusing solely on this paradigm may limit its flexibility compared to hybrid approaches, which, on the other hand, are more onerous for users to handle. Vision integration has also been leveraged for context-aware assistance in shared control systems. For example, as demonstrated in [188], the use of vision-guided robotic assistance can enhance the precision of brain-machine interfaces (BMI) to control grasping tasks. Although the architecture presented in this work is primarily focused on motor imagery-based control, the incorporation of real-time vision-based feedback has the potential to enhance obstacle avoidance and user intent interpretation. Other techniques, such as fuzzy logic-based shared control, have shown robust handling of noisy BCI inputs. In [189] user intent is blended with system safety using fuzzy logic, providing adaptability for varying task complexities. While null-space-based velocity fusion effectively mitigates misclassification effects in this work, the integration of fuzzy logic might add further adaptability. The incorporation of predictive modeling of user intent has been explored in works such as [190], where predictive elements preemptively switch control modes based on environmental context or user behavior. While this work's finite-state machine effectively manages mode transitions, incorporating predictive elements could further optimize performance. Task-specific optimizations also play a critical role in shared control systems. In [191, 192] the effectiveness of customized strategies for object grasping and manipulation is demonstrated. These approaches align with the

tailored architecture for teleoperated navigation and robotic arm manipulation presented in this work. This comparison highlights how the proposed approach complements existing frameworks while identifying areas for potential enhancements, such as incorporating hybrid EEG paradigms, vision-based assistance, and predictive user intent modeling.

4.2 Null-space-based shared control of a mobile robot using motor imagery based brain-computer interface

The following section presents a shared control architecture for non-holonomic mobile robots that is commanded through a motor imagery-based brain-computer interface. The overall system is aimed at assisting people to teleoperate a mobile robot in a simulated house-like scenario through the use of two motor imagery commands [193]. The developed architecture is structured in a manner that enables the user to operate the mobile robot while safety tasks, such as obstacle avoidance, are autonomously accomplished. The proposed architecture is also responsible for automatically switching the control mode, thereby enabling the robot to adjust its configuration. This facilitates the user's ability to teleoperate the mobile robot, for instance, when the robot is required to traverse a hallway. Throughout all phases, the safety constraints tasks are managed by a task-prioritized inverse kinematic control algorithm [172, 194]. The main contributions of this work are as follows: *i*) A modification of the BCI signal filter, as presented in [195], was devised with the objective of modifying the handling of the relaxation state to make it more natural for a teleoperation task; *ii*) a shared control architecture was designed to assist the user in controlling a non-holonomic mobile robot in a scaled house-like environment, relying on a maximum of two motor imagination commands. The overall architecture has been developed by creating control modules within the ROS environment, while the OpenVibe framework has been adopted for the processing of EEG signals. The effectiveness of the proposed architecture has been validated through experiments in which a healthy user, equipped with a Unicorn g.tec BCI, performed an assisted teleoperation task through motor imagery sessions with a Turtlebot robot.

4.2.1 Shared control architecture

In the envisioned shared control strategy, the motion is controlled in a cooperative manner by both the robot and the user. The architecture foresees three different control modes:

- *free* mode;
- *guided* mode;
- *autonomous* mode.

When the system is in *free* mode, the robot autonomously moves forward at a constant predefined velocity while the user commands the direction of the motion, giving angular acceleration commands using a two-class motor imagery BCI. In this state, the robot embeds an autonomous obstacle avoidance behavior to facilitate user assistance, allowing the user to concentrate only on moving the robot toward the desired destination without worrying about damaging it and/or the environment in which it operates. The user intended motion is integrated with the necessary movement for obstacle avoidance through the use of the well-known null-space based inverse kinematics technique. Consequently, the resulting motion will satisfy all safety constraints while following the user commands *at best*. More details will be given in the following sections.
In *guided* mode, the motion direction is controlled autonomously by the robot, while the user can command the forward/backward velocity. This control mode is particularly useful in situations in which the robot has to move inside a narrow passage, e.g. a corridor, where there is not enough space to turn and a misclassification of a user command, erroneously interpreted as an angular acceleration, may result in unintended impacts.
In *autonomous* mode, the robot performs a motion in a completely autonomous manner, without any input from the user. This control mode is used as a transition state between *free* and *guided* modes. In this state, the robot may, for instance, centre its position with respect to the corridor to cross and align its forward direction with the corridor orientation.

The transition among the control modes is automatically handled following a proper finite state-machine algorithm, as illustrated in Algorithm 3. Starting from a certain initial position of the robot $\boldsymbol{p}_0$, it is assumed that

Algorithm 3 State Machine Algorithm

```
1: procedure StateMachineTransition(p_0)
2:     mode ← free
3:     k ← 0
4:     p(t_0) ← p_0
5:     while 1 do
6:         cmd ← GetBCICmd()
7:         switch mode do
8:             case free
9:                 p(t_k)← ExecFreeCmd(p(t_{k−1}),
10:                                    cmd)
11:                for each passage ∈ passages do
12:                    if ||p(t_k) − p_{passage, ini}|| ≤ Δ then
13:                        p_{passage, ini} ← passage
14:                        mode ← autonomous
15:                        break
16:                    end if
17:                end for
18:            case autonomous
19:                while !finished do
20:                    p(t_k) ← ChangeConf()
21:                    finished ← ConfReached()
22:                end while
23:                mode ← guided
24:                break
25:            case guided
26:                p(t_k) ←ExecGuidedCmd(p(t_{k−1}),
27:                                      cmd)
28:                if ||p(t_k) − p_{passage, ini}|| = 0 or
29:                    ||p(t_k)− p_{passage, fin}|| = 0 then
30:                    mode ← free
31:                    break
32:                end if
33:            k ← k + 1
34:    end while
35: end procedure
```

the initial control mode is *free*, which allows the user to control the angular acceleration of the robot, while the forward velocity is set equal to a predefined constant quantity.

When the robot is close enough to the entrance of a narrow passage (inside a sphere of radius smaller or equal than a certain threshold Δ centered at the entrance of the passage), the control mode switches in *autonomous*. At this point the robot autonomously executes a movement that changes its position, in order to reach the center of the entrance, and its orientation, in order to align with the passage direction. Upon the completion of the autonomous movement, the control mode is changed to *guided*, and the control returns to the user.
Subsequently, the operator can give forward/backward velocity commands, in order to move along the narrow passage until the exit or to go back to abort the task. In both cases the control mode is switched to *free* mode, and the user can continue moving freely in the environment controlling the motion direction giving angular acceleration commands.

BCI based Motor Imagery

When the system is in *free* or *guided* control mode, the user can use 2 commands through a motor imagery session to control the robot's movement. In particular, in *free* mode the user imagines right or left hand movements, which respectively result in a positive or negative angular acceleration. In contrast, in *guided* mode, the same imagined hand movements are associated with a forward or backward linear velocity.

All the aspects related to the acquisition, processing and classification of the brain signals are handled by resorting to the open source framework OpenVibe. In particular, a band-pass filter is applied to the raw EEG signals, with a frequency band of $8 - 30$ Hz. The filtered EEG signals are used to collect feature sets for each class, which are then sent to a classifier based on linear discriminant analysis (LDA) to distinguish between left/right hand movements. Before running an online session, it is necessary for the user to complete a training session to train the OpenVibe motor imagery classifier. This training session is composed of a randomly generated sequence of trials for each of the motor image classes, during which the user gives the related motor imagery command. The number of trials for each class has been set to 20, while the feedback

duration for each stimulus has been set to 3.75s. At the beginning of the training session, there is a one-minute relaxation phase, where no stimuli are presented, in order to let the user preparing for the training procedure, which takes around 7 minutes. OpenVibe gives in output, at a frequency of 15 Hz, a vector of probabilities associated to each one of the 2 possible classes $\boldsymbol{x}_t = [x_1\ x_2]^T$ and the classified command, which is the class associated with the highest probability.

The classified commands in output from the BCI are usually affected by noise and temporary misclassifications that potentially cause discontinuities after the translation into motion commands for a robot. In order to mitigate this undesirable effect, a filtering layer is introduced before the translation into motion commands, initially proposed in [195] for two class motor-imagery, and then extended also for 4-class motor imagery in [182]. In the following section, the original filter formulation is first recalled, and then a modification is introduced to the handling of the relaxed statement, which is more convenient for the teleoperation use case that is considered in this work.

Given a couple of possible classes $\mathcal{M} = \{C_{\text{right}}, C_{\text{left}}\}$, defining x_t^{right} as the probability of C_{right} coming from the classifier output at time t, and denoting as y_t^{right} the filter output at time t, then the filter output increment at time t, $\Delta y_t^{\text{right}}$, is obtained by weighting two forces F_{free} and F_{BCI} as follows:

$$\Delta y_t^{\text{right}} = \chi \cdot [\phi \cdot F_{\text{free}}(y_{t-1}^{\text{right}}) + (1-\phi) \cdot F_{\text{BCI}}(x_t^{\text{right}})] \tag{4.1}$$

where y_{t-1}^{right} is the filter output at time $t-1$. The free force is defined as:

$$F_{\text{free}} = \begin{cases} -\sin(\frac{\pi}{0.5-\omega} \cdot y), & \text{if } y \in [0, 0.5-\omega) \\ -\psi \sin[\frac{\pi}{\omega} \cdot (y-0.5)], & \text{if } y \in [0.5-\omega, 0.5+\omega] \\ \sin[\frac{\pi}{0.5-\omega} \cdot (y-0.5-\omega)], & \text{if } y \in (0.5+\omega, 1] \end{cases} \tag{4.2}$$

where ψ is the sin wave amplitude, and ω the range used to discriminate the filter output. The BCI force is defined as:

$$F_{\text{BCI}} = \alpha_1 \cdot (x-0.5)^3 + \alpha_2 \cdot (x-0.5). \tag{4.3}$$

where α_1 and α_2 are parameters to be properly tuned depending on the

EEG data.
The class in output from the filter is:

$$C^{\text{out}} = \begin{cases} C_{\text{right}}, & \text{if } y_t^{\text{right}} \geq 0.5 + \omega \\ C_{\text{left}}, & \text{if } y_t^{\text{left}} \leq 0.5 - \omega \end{cases} \tag{4.4}$$

being C_{right} and C_{left} mutually exclusive, while the filter does not give any output if $y_t^{\text{right}} \in (0.5 - \omega, 0.5 + \omega)$. The evolution of the two forces is shown in Fig. 4.1.

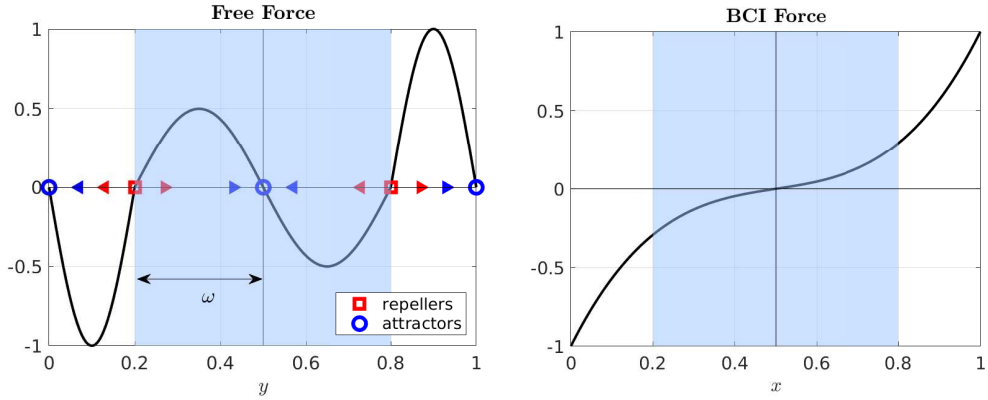


Figure 4.1: On the left: evolution of the free force F_{free} with $\psi = 0.4$, $\omega = 0.3$. On the right: BCI force F_{BCI} with $\alpha_1 = 6.4$ and $\alpha_2 = 0.4$.

The original filter formulation has a drawback in the management of the relaxed state, which occurs when the user does not want to send any commands to the system. Assuming that the output of the filter is close to one of the attractors 0 or 1, an input coming from the BCI in the range $0.5+\omega$ and $0.5-\omega$ (i.e., a relaxed state signal), does not result in a modification of the filter output. This means that in order to change the output from 1 to 0, the user has to exert a force active on the opposite direction, *strong* enough to make the output overcome the first repeller. Although this design offers a beneficial effect in terms of robustness with respect to temporary misclassifications, it is quite uncomfortable for the user when the output of the filter is used to generate a command for a robot (e.g., a velocity or an acceleration). In such cases, the relaxed state

signal may be used more conveniently and naturally for directly stopping the robot.

In order to overcome this issue, a modification in the computation of $\Delta y_t^{\text{right}}$ is proposed, adding a new component as follows:

$$\Delta y_t^{\text{right, new}} = \Delta y_t^{\text{right}} + F_{\text{restore}}(x_t^{\text{right}}) \,, \tag{4.5}$$

where $F_{\text{restore}}(x_t^{\text{right}})$ is aimed at assisting the user in managing the relaxed state. In particular, this force is designed to push the output of the filter toward the attractor placed in 0.5 (resulting in a "no command" output), when the input coming from the BCI in close proximity to 0.5. The restore force can be computed as follows:

$$F_{\text{restore}}(x_t^{\text{right}}) = \sigma(x_t^{\text{right}})\ \gamma\ (y - 0.5) \,, \tag{4.6}$$

where γ is a scalar gain and $\sigma(x_t^{\text{right}})$ is a smooth activation function defined as:

$$\sigma(x_t^{\text{right}}) = 0.5 \left[\frac{\tanh(x_t^{\text{right}} - c_1)}{\varepsilon} - \frac{\tanh(x_t^{\text{right}} - c_2)}{\varepsilon} \right] \,, \tag{4.7}$$

where $c_1 = 0.5 + \delta - \omega$ and $c_2 = 0.5 - \delta + \omega$. The parameters δ and ε are positive constants that must be properly tuned to modify the slopes and the width of the activation function. Figure 4.2 illustrates the function $\sigma(x_t^{\text{right}})$ over the filter input, with $\delta = 0.05$ and $\varepsilon = 0.02$. The effect of $\sigma(x_t^{\text{right}})$ is to modulate the restoring force in order to have an effect only when the input of the filter is close to 0.5, that is, when the user does not want to give any command to the robot, avoiding to modify the output when the user's intention is to actually give a command.

4.2.2 Null-space-based shared control strategy

Differential-drive robots kinematics and low-level controller

The pose of a frame $\boldsymbol{\mathcal{F}_R} = \{X_R, Y_R\}$ attached to the robot with respect to an inertial frame $\boldsymbol{\mathcal{F}_I} = \{X_I, Y_I\}$ can be expressed by the vector: $\boldsymbol{p} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$, where x and y represent the coordinates of the center of $\boldsymbol{\mathcal{F}_R}$ expressed in $\boldsymbol{\mathcal{F}_I}$ and θ is the angle representing the rotation between $\boldsymbol{\mathcal{F}_R}$

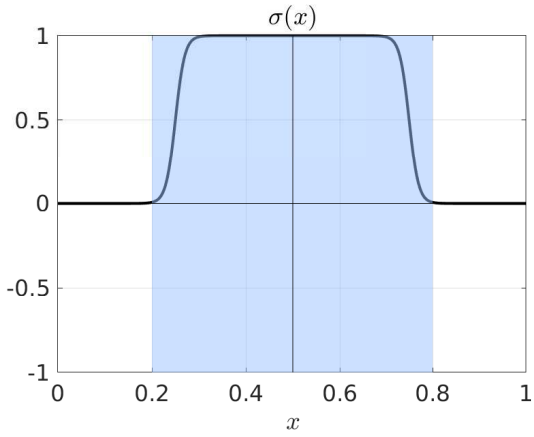


Figure 4.2: Activation function with $\delta = 0.05$ and $\varepsilon = 0.02$.

and $\boldsymbol{\mathcal{F}_I}$. The velocity of the robot, expressed in the inertial frame, is defined as follows: $\dot{\boldsymbol{p}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$, and the robot velocity expressed in robot frame as $\boldsymbol{\xi} = \begin{bmatrix} \nu & \rho & \psi \end{bmatrix}^T$, where ν is the linear velocity component along the X_R direction, ρ is the linear velocity component along the Y_R direction and ψ is the angular velocity around a vertical axis centered in $\boldsymbol{p}$. The relation between the two velocities, in the general case, can be expressed as:

$$\dot{\boldsymbol{p}} = \boldsymbol{R}(\theta)\, \boldsymbol{\xi}\ , \tag{4.8}$$

where $\boldsymbol{R}(\theta)$ is the rotation matrix expressing the elementary rotation of θ around the z-axis. Given the non-holonomic constraint, it is not possible for a differential mobile robot to translate laterally, as the velocity component ρ is null. Defining as $\boldsymbol{\eta} = [u \quad \omega]^T$ the generalized velocity vector composed by the forward and angular velocities, the kinematics of a differential mobile robot can be written as:

$$\dot{\boldsymbol{p}} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{\eta}\ . \tag{4.9}$$

In order to control the robot movement, the output of the shared control architecture is translated into a 2D velocity vector $\boldsymbol{\zeta} = [\zeta_x \quad \zeta_y]^T$ expressed in the robot frame $\boldsymbol{\mathcal{F}_R}$. This vector is obtained through a proper fusion

of two *elementary behaviors*, which are related to the user input and the obstacle avoidance, respectively. Then, the velocity $\boldsymbol{\zeta}$ is fed into a low-level controller that generates the velocity commands $\boldsymbol{\eta}$ to be sent to the robot in order to track it. More in detail, the two components are computed as follows:

$$u = k_u ||\boldsymbol{\zeta}|| \tag{4.10}$$

$$\omega = k_\omega \arctan\left(\frac{\zeta_y}{\zeta_x}\right) , \tag{4.11}$$

where k_u and k_ω are scalar gains.

Elementary behaviors

In the following, the computation of the velocities $\boldsymbol{\zeta}_i = [\zeta_{i,x} \quad \zeta_{i,y}]$ related to the two elementary behaviors and the subsequent fusion method are described.

User input

The *User input* elementary behavior generates a velocity which depends on the command sent by the user and the active control mode.

In *guided* mode, the user command the linear velocity of the robot in order to cross a narrow passage or to come back in the *free* control mode. In this case the velocity command expressed in the robot frame is:

$$\boldsymbol{\zeta}_{\text{user}} = \begin{bmatrix} V_{\text{guided}} & 0 \end{bmatrix}^T , \tag{4.12}$$

where V_{guided} is a predefined positive forward velocity in case of a *RIGHT* command received by the user, while it is negative in case of a *LEFT* command.

When the system is in *free* control mode, the classified BCI command is considered as an angular acceleration $\ddot{\phi}$ (positive in case of a *LEFT* command, negative in case of a *RIGHT* one), that is integrated twice over time to obtain an angular velocity $\dot{\phi}$ and an angle ϕ. In order to obtain the velocity vector expressed in the robot frame, the integrated ϕ is used to rotate a constant forward velocity that is automatically given

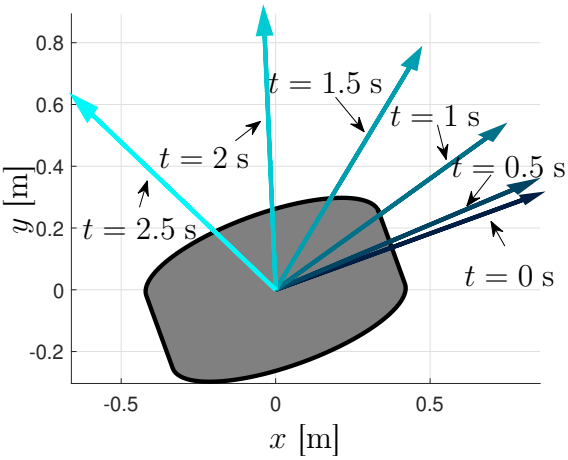


Figure 4.3: Example of velocity related to a user command given for 2.5s.

to the robot obtaining the vector:

$$\boldsymbol{\zeta}_{\text{user}} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} V_{\text{free}} \\ 0 \end{bmatrix} , \tag{4.13}$$

where V_{free} is a predefined positive velocity.
Figure 4.3 shows an example of velocity vector $\boldsymbol{\zeta}_{\text{user}}$ obtained with a user command *LEFT* given for 2.5 s in *free* control mode.

The choice to consider the command as an acceleration rather than a velocity is due to the fact that in this way the user can perform turns with different curvatures, giving him/her more control on the robot executed path compared to a velocity command.

Obstacle avoidance

The *Obstacle avoidance* behavior is needed to make the robot able to avoid potential obstacles that encounters along the path commanded by the user. The generation of this velocity is done in a completely autonomous manner by the robot in case of need, without any input form the user. Given an obstacle position expressed in the inertial frame $\boldsymbol{p}_{\text{obst}} \in \mathbb{R}^2$, it is required that the obstacle be situated at a minimum distance of D_{min} from the robot. Defining the distance between the obstacle and the robot expressed in the inertial frame as $d = ||\boldsymbol{p}_{\text{obst}} - \boldsymbol{p}||$, the differential relation

between the task velocity and the system velocity can be expressed as:

$$\frac{\partial d}{\partial \boldsymbol{p}} = -\frac{(\boldsymbol{p}_{\text{obst}} - \boldsymbol{p})^T}{||\boldsymbol{p}_{\text{obst}} - \boldsymbol{p}||} \boldsymbol{R}\, \boldsymbol{\zeta} = \boldsymbol{J}_{\text{obst}}\, \boldsymbol{\zeta}\ , \tag{4.14}$$

where $\boldsymbol{J}_{\text{obst}}$ is the Jacobian matrix of the obstacle avoidance control objective. The system velocity that fulfills the task can be computed as:

$$\boldsymbol{\zeta}_{\text{obst}} = \boldsymbol{J}^{\dagger}_{\text{obst}}\, k_{\text{obst}}\, \tilde{d}\ , \tag{4.15}$$

where $\boldsymbol{J}^{\dagger}_{\text{obst}}$ is the pseudoinverse of the obstacle avoidance Jacobian, k_{obst} is a scalar task gain and $\tilde{d} = d - D_{\text{min}}$ is the task error.

Velocities fusion

It is essential to fuse the elementary velocities required to satisfy both the user input task and the obstacle avoidance task in a way that leaves as much control as possible to the user, while still ensuring compliance with safety constraints. The aforementioned fusion is achieved through the use of the well-known null-space based inverse kinematics framework, wherein the tasks are arranged in hierarchy and the computed solution guarantees the fulfillment of the high-priority task (the obstacle avoidance) while the low-priority task (user input) is executed *at best*, leaving to the user as much control as possible to the robot motion.

Given the velocities $\boldsymbol{\zeta}_{\text{obst}}$ and $\boldsymbol{\zeta}_{\text{user}}$, computed as in Eq. (4.15) and Eq. (4.13) respectively, the overall velocity to apply to the robot can be computed as:

$$\boldsymbol{\zeta} = \boldsymbol{\zeta}_{\text{obst}} + \boldsymbol{N}_{\text{obst}}\, \boldsymbol{\zeta}_{\text{user}}\ , \tag{4.16}$$

where $\boldsymbol{N}_{\text{obst}}$ is the null space of the obstacle avoidance task, defined as:

$$\boldsymbol{N}_{\text{obst}} = \boldsymbol{I} - \boldsymbol{J}^{\dagger}_{\text{obst}} \boldsymbol{J}_{\text{obst}}\ , \tag{4.17}$$

with $\boldsymbol{I}$ the identity matrix of suitable dimensions.

It is worth noticing that this framework can be seamlessly adapted to robots with different kinematics and extended in order to handle an arbitrary number of tasks [168].

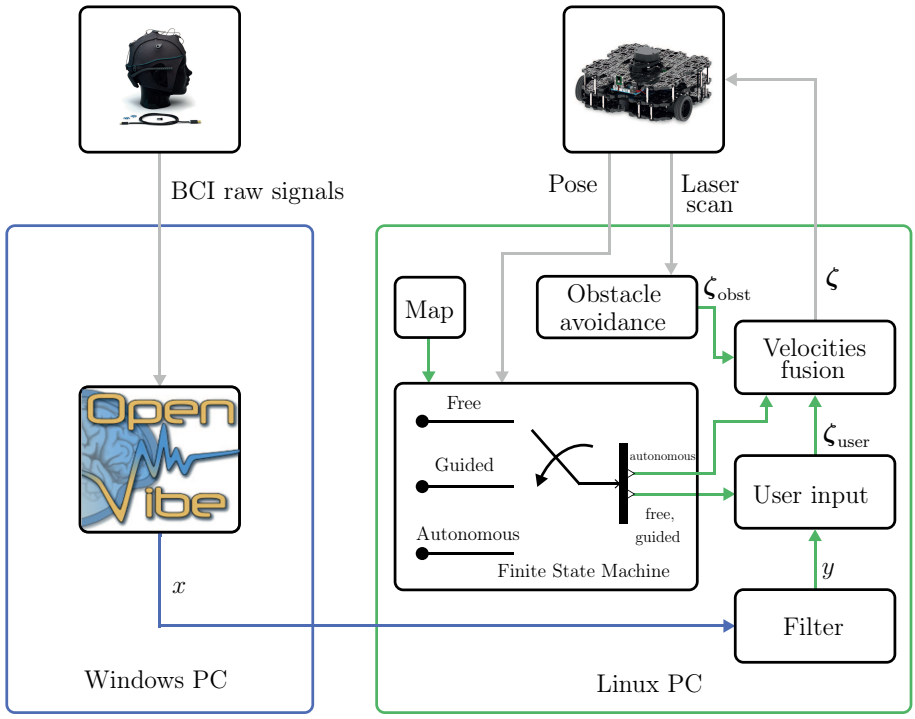


Figure 4.4: Implemented functional architecture.

4.2.3 Experiments

Implemented functional architecture

The overall architecture has been implemented and experimentally tested using a non-invasive g.tec Unicorn BCI, and a Turtlebot robot. The motor imagery BCI sessions have been conducted through the OpenVibe framework, which runs on Microsoft Windows. All control modules installed on the Turtlebot have been developed in the ROS [196] framework, which runs on Linux. To facilitate communication between the two workstations, Lab Streaming Layer (LSL) and TCP/IP protocols have been implemented. The complete scheme of the proposed solution is shown in Figure 4.4.

Experimental validation

The proposed assistive architecture has been validated through a series of experiments that aim to replicate a situation in which a person with motor disabilities wants to teleoperate a robotic system inside a house-like scenario using a BCI motor imagery paradigm. The experiments were performed by an able-bodied subject[1] (female, 27 years-old), whose goal is to drive the Turtlebot in a scaled and simulated house-like scenario.

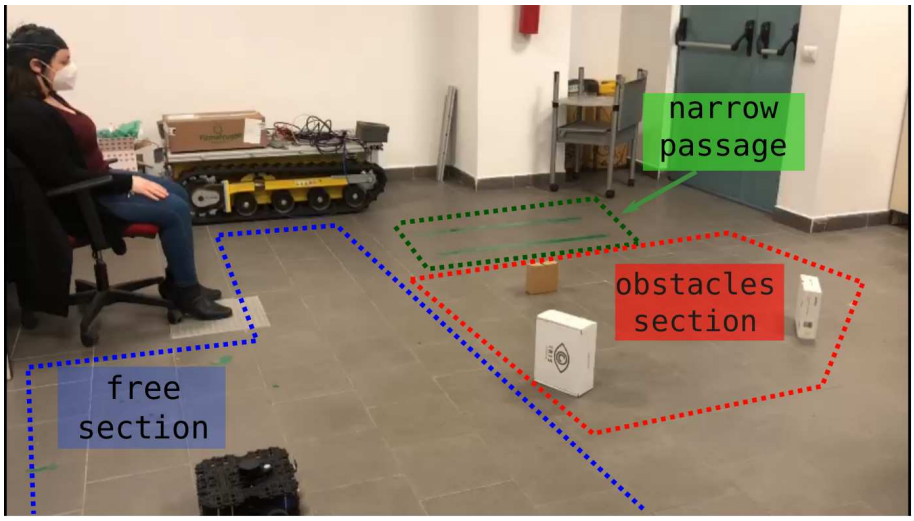


Figure 4.5: Snapshot of the built scenario for the experimental validation. The user is required to teleoperate the Turtlebot across three distinct sections: an obstacle-free area (depicted in blue), an area with a simulated narrow passage (depicted in green), and an area with three obstacles (depicted in red).

Figure 4.5 shows the considered scenario, in which three different sections can be highlighted: the first one (in blue) is an obstacle-free area; the second one (in green) is a simulated narrow passage; finally, there is an area (in red) with three obstacles. The robot starts from a predefined position and orientation, and the aim of the user is to teleoperate the Turtlebot

[1]The subject is the author of the present thesis, and the experiments were conducted in accordance with the Declaration of Helsinki. The protocol was approved by the Research Ethics Committee of the University of Cassino and Southern Lazio, and the subject provided informed consent.

across the three different areas, eventually returning in proximity of the initial position. It is assumed that in the first section of the environment, the shared control architecture is in *free* mode, and the user sends *RIGHT* or *LEFT* commands to control the robot's angular acceleration while it automatically moves forward at a constant velocity. Once the robot has reached a predefined distance from the entrance of the narrow passage, the control mode switches to *autonomous*, and the Turtlebot autonomously reaches the entrance of the passage and aligns itself along its direction. Then, the control mode switches to *guided*, whereby control is returned to the user who controls the forward/backward velocity of the robot through the two motor imagery commands in order to reach the end of the narrow passage or backward. At this point, the control mode switches back to *free*, and in the event that the user has crossed the corridor, he/she is required to go through the final section of the environment. Here, the shared control architecture facilitates obstacle avoidance, letting her to focus only on the desired location of the robot. Finally, the robot reaches the blue area again, which is the obstacle-free section, and the user teleoperates it in the proximity of the initial position.

A series of experiments was performed to validate the developed control architecture, which can be accessed via the following link: https://www.youtube.com/watch?v=dNSNhdpDQHM. The details of the first run shown in the video will present in the following. It is worth noticing that the user was able to successfully complete the desired task in all the runs.

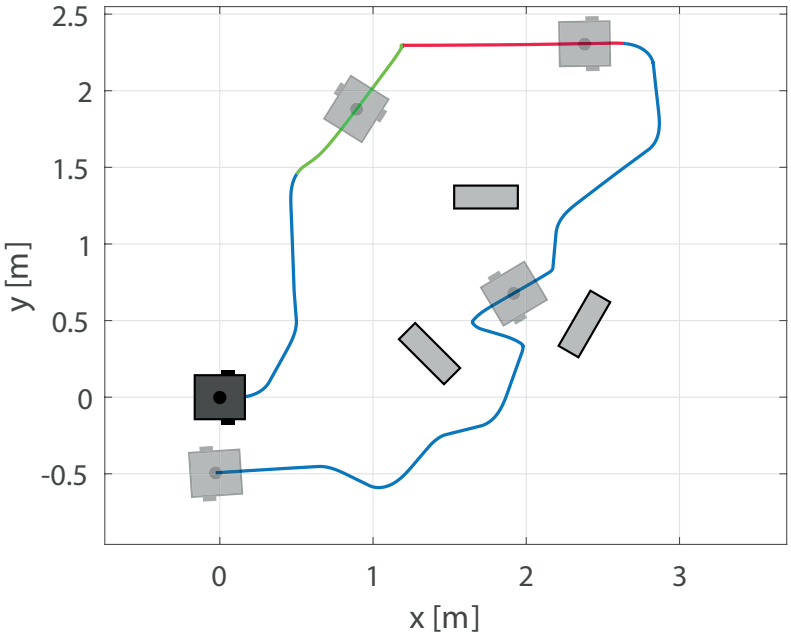


Figure 4.6: The path performed during one of the experiments highlighted according to the active control modes: in blue the path executed in *free* mode, in green in the *autonomous* mode, and in red in the *guided* mode.

Figure 4.6 shows the path executed by the Turtlebot during the considered experiment run, where its starting/ending position and the obstacle positions are highlighted.

In particular, the path followed by the robot is coloured differently depending on the corresponding active control mode. The blue path corresponds to the *free* mode, the green path to the *autonomous* mode and the red one to the *guided* mode. It is clear that the user successfully makes the robot go across all the three sections of the chosen environment, and the shared control architecture facilitated a change in the operational mode to assist the user in overcoming the narrow passage without any impact. Additionally, the null-space based velocity fusion prevented impacts in the area characterized by the presence of the three obstacles.

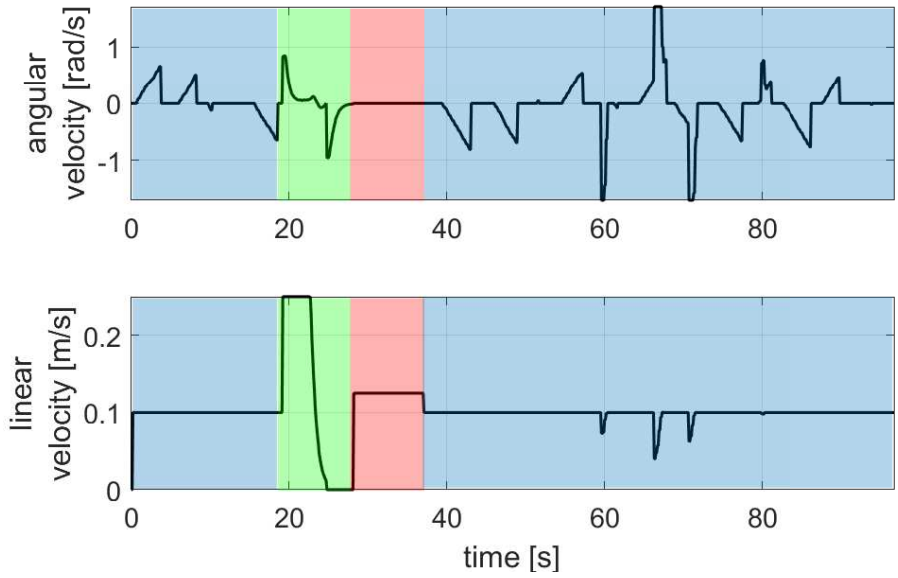


Figure 4.7: Linear and angular velocities over time during one of the runs. The background colour is indicative of the active control mode. The *free* mode is represented by the blue line, the *autonomous* mode by the green line, and the *guided* mode by the red line.

Figure 4.7 shows linear and angular velocities over time, highlighted according to the active control mode.
The section highlighted in blue corresponds to the *free* control mode, in green to the *autonomous* mode, and in red to the *guided* mode.

From $t = 0$s to $t = 19$s, the robot is in the initial obstacle-free section of the considered scenario and the system is in the *free* control mode in which the user controls the angular acceleration of the Turtlebot, which moves forward at a constant velocity; in fact, the linear velocity is equal to the default constant value (0.1 m/s), while the angular velocity increases (decreases) linearly until the output of the BCI filter is equal to the $LEFT$ ($RIGHT$) command.
When the Turtlebot is situated in close proximity to the entrance of the narrow passage ($t = 19$s), the control mode switches to *autonomous* (green background) and both the angular and linear velocities are automatically computed by the robot. From $t = 30$s to $t = 38$s the system

is in *guided* control mode, and the angular velocity is automatically kept to 0 rad/s while the user commands the forward velocity which is equal to a constant predefined value (0.13 m/s).
Then, the system switches back to *free* control mode and the robot goes across the obstacles section. In this area, the control of the linear and angular velocities is shared between the user and the robot in order to reach the desired location while avoiding collisions with the obstacles.

4.3 Shared control architecture for teleoperation of a robotic manipulator through BCI

This section presents a shared control architecture designed for assisted teleoperation tasks, which exploits the potential of brain-computer interfaces based on motor imagery. The overall system is designed to assist users in performing teleoperated manipulation tasks by adapting to their needs and providing varying degrees of autonomy at different stages of task execution. During these transitions, the architecture facilitates a seamless transition from one control mode to another. These modes encompass a spectrum from the robot's autonomous operations to the user's capacity to issue one of three distinct commands.

The low-level component of the shared control architecture is responsible for ensuring safety and facilitating operational tasks, such as collision avoidance and the management of the robot's joint limits. This is achieved through the use of a set-based task-priority inverse kinematic control algorithm, a highly versatile motion control strategy that is capable of handling both set-based and hierarchically organized tasks [172, 174].

The previously introduced architecture has been enhanced to facilitate user-guided control of a 3D robotic arm, relying on up to three user-given commands.

The overall architecture was realized by integrating control and perception software modules developed within the ROS environment, with the OpenVibe framework used to manage the BCI device. The effectiveness of the proposed architecture was validated through experiments in which a healthy user, wearing a Unicorn g.tec BCI, executed an assisted teleoperation task via motor imagery sessions with a 7-degree-of-freedom Kinova

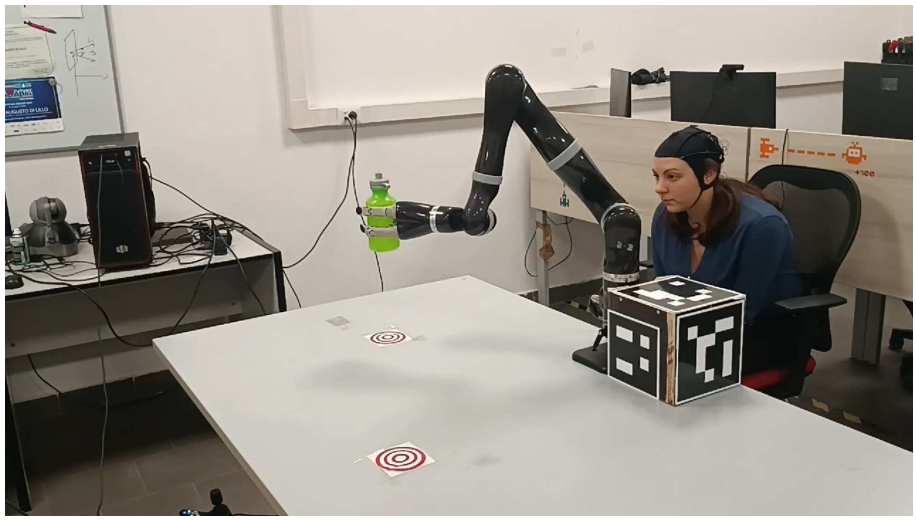


Figure 4.8: Snapshot of the built scenario for the experimental validation. The user is required to teleoperate the Jaco^2 robotic arm using BCI to manipulate objects in the space.

Jaco^2 robotic arm, as shown in the scenario in Figure 4.8.

4.3.1 Shared Control Architecture

In general, an increased number of commands that can be generated through motor imagery would provide the user with a greater degree of freedom, thereby enabling the performance of more complex tasks. In the proposed architecture, the restriction of allowing a maximum of three user commands to control the robot's end-effector is addressed, which should restrict the execution of complex tasks. This limitation is overcome through the implementation of a shared control strategy, which facilitates the user's ability to perform the desired task. This strategy is not only designed to facilitate the completion of the task at hand, but also to independently manage the requisite steps for its successful execution. To this end, the shared control strategy is organized into three distinct control modes, as follows: *user 2-D control*, *user 1-D control*, and *robot control*. The architecture is designed to intelligently switch among the modes in accordance with the prevailing operational conditions.

In the *user 2-D control* mode, the motor imagery BCI enables the user to differentiate between three classes, designated as "RIGHT", "LEFT", and "SWITCH". The output of the BCI classifier is used to adjust the desired position of the robot end-effector in two-dimensional space, such as the horizontal plane. It is worth noting that the "SWITCH" command enables the user to specify the desired movement direction (forward/backward or left/right) for the robot end-effector. In *user 1-D control* mode, the motor imagery BCI allows the user to differentiate between two categories, designated as "FORWARD" and "BACKWARD". The output of the BCI classifier is used to modify the robot's progression in a task (e.g., moving forward or backward along the relevant axis, such as the grasping direction). In *robot control* mode, the output of the BCI is disregarded, thereby enabling the robot to autonomously execute predefined operations. This approach guarantees that the architectural configuration can be tailored to the particular requirements of a given task.

4.3.2 BCI based Motor Imagery

When the system is in a *user control* mode, the user is able to teleoperate the robotic arm through a motor imagery session using two or three commands. In particular, in *user 1-D control* mode, the user imagines right or left hand movements, respectively, to go ahead and back in the task execution, while for the *user 2-D control* mode the user imagines left and right hand movements to control two pairs of directions for the robot, enabling control of the robot in right/left or forward/backward directions. The user is able to switch between pairs of directions via the "SWITCH" command. During this state, the user is in a "relaxed state" with the absence of active imagination of either right or left commands, in which the user does not want to send any commands.
To improve the reliability and continuity of the classification, the filter presented in the previous work [193] is used. The filter formulation employed in this section is illustrated in Section 4.2.1. All the aspects of the EEG signal classification process are handled by resorting to the open source OpenVibe framework, as already described in Section 4.2.1.

4.3.3 Robot end-effector control strategy

The transition between control modes is contingent upon the prevailing operational conditions and is managed through a finite state machine algorithm (shown in Algorithm 4) that adapts to the configured and achievable tasks. To provide a more comprehensive illustration of this concept, a scenario involving multiple targets within the environment, such as objects to be grasped and object release points, is considered.

Algorithm 4 State Machine Algorithm

```
procedure StateMachineTransition(P_ee,ini)
    mode ← user 2-D
    k ← 0
    P_ee(t_0) ← P_ee,ini
    while 1 do
        command ← get_bci_cmd()
        switch mode do
            case user 2-D
                p(t_k) ← exec_2d_cmd(p(t_k−1),
                                                command)
                for each target ∈ targets do
                    if ||p(t_k) − target|| ≤ Δ then
                        p_target ← target
                        mode ← robot
                        previous_mode ← user 2-D
                        break
                    end if
                end for
            case robot
                while !finished do
                    p(t_k) ← change_conf()
                    finished ← conf_reached()
                end while
                if previous_mode = user 2-D then
                    mode ← user 1-D
                else
                    mode ← user 2-D
                end if
                break
            case user 1-D
                p(t_k) ← exec_1d_cmd(p(t_k−1),
                                                command)
                if ||p(t_k) − p_target|| > Δ or
                        ||p(t_k)− p_target|| = 0 then
                    mode ← robot
                    previous_mode ← user 1-D
                    break
                end if
            k ← k + 1
    end while
end procedure
```

Starting from an initial end-effector position represented as $P_{\text{ee,ini}}$, the system's initial control mode is designated as *user 2-D control*, that allows the user to control the planar positioning of the end-effector. This procedure takes into account the initial end-effector position, $P_{\text{ee,ini}}$, and the target's position, denoted as p_{target}. The control strategy updates the end-effector position continuously based on the current control mode. When not in *robot control* mode, the strategy responds to the received BCI commands.

In the *user 2-D control* mode, the user is afforded the autonomy to maneuver the end-effector within a defined plane in order to approach the desired object. Upon bringing the end-effector into close proximity to a specific target, defined as within a sphere with a radius less than or equal to a predetermined threshold (Δ) centered on the target, the control mode switches to *robot control*. During this phase, the robot autonomously executes a movement that aligns the end-effector based on the nature of the target, such as aligning it for grasping an object. Once the autonomous movement is completed, the control mode switches to *user 1-D control* and returns to the user. In this phase, the user has the option to terminate the task by moving the end-effector away from the target, or to proceed with the task completion, such as moving the end-effector toward an object until it can be grasped autonomously. In both scenarios, the control mode shifts to *robot*, enabling the robot to autonomously relocate to a predefined position before eventually return to *user 2-D control*.

In the *user 1-D control* and *user 2-D control* modes, the class C^{out} generated by the filters at each time step is used as input to a dedicated software node. Then this node computes a reference trajectory point for the end-effector, which the robot subsequently follows.

In particular, when the user is in *2-D control* mode, each command is associated with a desired velocity at timestep k, expressed in the arm base frame as:

$$\boldsymbol{v}_{\text{RL}}(k) = \begin{bmatrix} \{V_{\text{des}}, -V_{\text{des}}\} \\ 0 \end{bmatrix} \tag{4.18}$$

$$\boldsymbol{v}_{\text{FB}}(k) = \begin{bmatrix} 0 \\ \{V_{\text{des}}, -V_{\text{des}}\} \end{bmatrix} \tag{4.19}$$

A parameter denoted as V_{des}, representing the desired amplitude of the

end-effector velocity, is introduced. Furthermore, the velocities associated with the RIGHT and LEFT commands, as well as the FORWARD and BACKWARD commands, are represented by the vectors $\boldsymbol{v}_{\mathrm{RL}}$ and $\boldsymbol{v}_{\mathrm{FB}}$, respectively. In this context, RIGHT and FORWARD signify positive directions, while LEFT and BACKWARD represent negative directions. These velocity commands are integrated over time, resulting in a reference position within the xy plane, which is subsequently transmitted to the robot:

$$\boldsymbol{p}_{\mathrm{des}}^{\mathrm{xy}}(k+1) = \boldsymbol{p}_{\mathrm{des}}^{\mathrm{xy}}(k) + \begin{bmatrix} v_{\mathrm{RL},x}(k) \\ v_{\mathrm{FB},y}(k) \end{bmatrix} T_s \ . \tag{4.20}$$

Denote T_{s} as the sampling time and retaining the z_{des} coordinate at its current value, this results in a 3D desired position, denoted as $\boldsymbol{p}_{\mathrm{des}}(k)$, computed as $[\boldsymbol{p}_{\mathrm{des}}^{xy}(k); z_{\mathrm{des}}(k)]^T$.

Upon the system transitioning to the *user 1-D control* mode, a linear trajectory is defined, tailored to the specific operation at hand. For instance, this trajectory may facilitate a linear movement for grasping an object from a pregrasp configuration or maneuvering towards a release location when the held object nears its destination. In this mode, the user could generate only two commands, FORWARD and BACKWARD, to advance or retreat within the task execution. The reference position in this scenario moves along a segment, defined as:

$$\boldsymbol{p}_{\mathrm{des}}(s(k)) = (1 - s(k))\, \boldsymbol{p}_{\mathrm{ini}} + s(k)\, \boldsymbol{p}_{\mathrm{fin}} \ , \tag{4.21}$$

where $\boldsymbol{p}_{\mathrm{ini}}$ is the initial 3D position (for instance, the pregrasp position in tasks involving object grasping), $\boldsymbol{p}_{\mathrm{fin}}$ as the final 3D position (e.g., the object's position), and $s(k) \in [0, 1]$ as a parameter. The user's command is translated into a velocity, denoted as v_{AB}, which is set to V_{des} for the FORWARD command and $-V_{\mathrm{des}}$ for the BACKWARD command. This velocity is then integrated over time to either increase or decrease the parameter s, facilitating task progression until completion ($s(k) = 1$) or task abort ($s(k) = 0$):

$$s(k+1) = s(k) + v_{\mathrm{AB}} T_{\mathrm{s}} \ . \tag{4.22}$$

In the *robot control* mode, the robot autonomously manages its end-effector's 3D position and orientation, aligning them with the ongoing task stage. For instance, this may entail positioning the end-effector at

point $\boldsymbol{p}_{\text{ini}}$ when a task commences, or maneuvering the end-effector into predefined configurations upon task completion.

In conclusion, the reference pose for the end-effector at each time step is defined by a desired position vector $\boldsymbol{p}_{\text{des}}$ and orientation $\boldsymbol{\mathcal{Q}}_{\text{des}}$ expressed in quaternion:

$$\boldsymbol{\sigma}_{\text{pose, des}} = \begin{bmatrix} \boldsymbol{p}_{\text{des}} \\ \boldsymbol{\mathcal{Q}}_{\text{des}} \end{bmatrix} \in \mathbb{R}^7. \tag{4.23}$$

4.3.4 Joints controller

The computed reference pose is fed into an inverse kinematics algorithm, which then calculates the requred joint velocities for the end-effector to be moved towards the aforementioned pose. Given the system's state the vector $\boldsymbol{q} = [q_1\ q_2\ \dots\ q_n]^T$ containing the arm joint positions for n DOF of the arm (as outlined in Section 3.2), the reference joint velocity that brings the end-effector to $\boldsymbol{\sigma}_{\text{pose, des}}$ can be computed as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}_{\text{pose}}(\dot{\boldsymbol{\sigma}}_{\text{pose, des}} + \boldsymbol{K}_{\text{pose}}\, \tilde{\boldsymbol{\sigma}}_{\text{pose}})\ , \tag{4.24}$$

where $\boldsymbol{J}^{\dagger}_{\text{pose}}$ is the pseudoinverse of the task Jacobian matrix, $\boldsymbol{K}_{\text{pose}} \in \mathbb{R}^{6\times n}$ is a gain matrix, $\dot{\boldsymbol{\sigma}}_{\text{pose, des}}$ is a feedforward term and

$\tilde{\boldsymbol{\sigma}}_{\text{pose}} = \left[(\boldsymbol{\sigma}_{\text{pos,des}} - \boldsymbol{\sigma}_{\text{pos}})^T\ \tilde{\boldsymbol{\mathcal{Q}}}^T\right]^T$ is the task error, with $\boldsymbol{\sigma}_{\text{pos}}$ is the end-effector current position, $\boldsymbol{\sigma}_{\text{pos,des}}$ is the end-effector desired position and $\tilde{\boldsymbol{\mathcal{Q}}}$ is the quaternion error.

In addition to the pose task, the implemented control algorithm allows for the definition of other tasks to be performed concurrently. In particular, it is possible to assign a priority level to each task, with the objective of computing the reference joint velocity in a manner that fulfills all of the tasks *at best*. In order to combine a hierarchy of h tasks, it is necessary to assign h levels of priority (hereafter indexed by the integer i), from the primary one ($i = 1$) to the lowest priority one ($i = h$). Then, the reference system velocity can be computed recursively as in [168]:

$$\dot{\boldsymbol{q}}_h = \sum_{i=1}^{h} (\boldsymbol{J}_i \boldsymbol{N}^A_{i-1})^{\dagger} (\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{K}_i \tilde{\boldsymbol{\sigma}}_i - \boldsymbol{J}_i \dot{\boldsymbol{q}}_{i-1})\ , \tag{4.25}$$

where $\boldsymbol{N}_i^A$ projects a vector onto the null space of the augmented Jacobian matrix, obtained by stacking all the task Jacobian matrices from task 1 to i, i.e.,

$$\boldsymbol{J}_i^A = \begin{bmatrix} \boldsymbol{J}_1^T & \boldsymbol{J}_2^T & \dots & \boldsymbol{J}_i^T \end{bmatrix}^T , \tag{4.26}$$

and is defined as

$$\boldsymbol{N}_i^A = \boldsymbol{I} - (\boldsymbol{J}_i^A)^\dagger \boldsymbol{J}_i^A , \tag{4.27}$$

with $\boldsymbol{N}_0^A = \boldsymbol{I}$ (the identity matrix of suitable dimensions), $\dot{\boldsymbol{q}}_0 = \boldsymbol{0}_n$ (the $(n \times 1)$ null vector), and $\dot{\boldsymbol{q}}_i$ is the velocity contribution of the i-th task in the hierarchy. This task priority framework has been recently extended to handle also set-based tasks, in which the control objective is to keep the task value within a certain region rather than bring it to a specific value. This behavior is obtained by a proper insertion or removal of the tasks in the hierarchy through a specific algorithm. For more details about the algorithm the reader is referred to [197, 198].

In this work such control framework is exploited to define additional tasks related to the safety of the system. The general idea is to share the control between the user and the robot also at this stage, after the trajectory generation, letting the user control the pose task while all the other safety tasks are autonomously handled by the system. In particular, the set-based safety tasks implemented for this work are joint limits and virtual walls around the arm base frame. The latter are particularly useful to mitigate the effect of a wrong classification of the BCI commands that might bring the end-effector at the boundary of the arm workspace, consequently limiting the motion and resulting in a hardly-controllable system.

4.3.5 Experimental Results

System Architecture

The proposed architecture has been successfully implemented and validated through experimental testing. For the purposes of this investigation, a non-invasive g.tec Unicorn BCI, a Kinova Jaco^2 robotic arm with 7DoF, and an Intel RealSense RGB-D sensor were employed for object detection and localization. Also in this case, motor imagery sessions were conducted using the versatile OpenVibe software system, which employed a classifier based on the Linear Discriminant Analysis (LDA) technique. The aforementioned sessions were conducted on a Microsoft Windows PC.

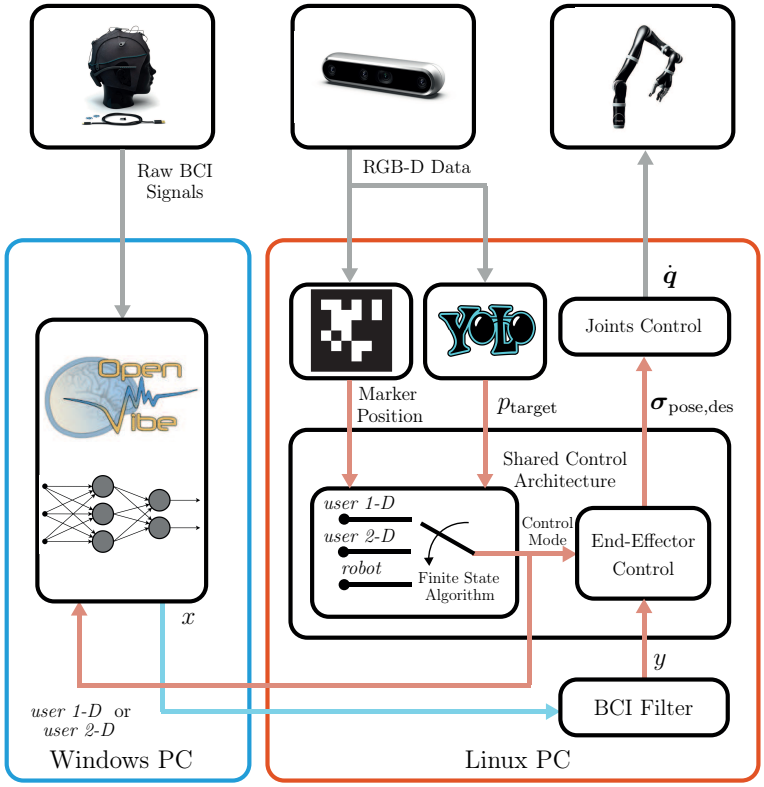


Figure 4.9: Main components of the proposed architecture.

In order to provide an overview of the architecture, Figure 4.9 illustrates the software components involved.

The software modules responsible for controlling the robotic arm, managing perception tasks, and executing BCI motor imagery filters were developed on a Linux-based PC. These modules were seamlessly integrated through the Robot Operating System. The perception module, employing RGB-D data, identified objects within the scene using a YOLO-based object recognition algorithm, accurately locating them within the 2D image plane. Their 3D space location is determined by projecting the 2D objects area on the point cloud. Their position is then transformed from the camera frame to the arm base frame composing a predefined transformation from the arm base frame to a marker and the transformation between the camera frame and the marker (computed by AprilTag [199]).

To let the two workstations to communicate, Lab Streaming Layer (LSL) and TCP/IP protocols have been implemented.

Experimental Validation

In order to validate the proposed architecture described in Section 4.3.1, a series of experiments considering a pick and place operation have been performed. The user has to teleoperate the robotic manipulator to grasp a bottle placed on the table in front of him/her, and then release it in one of the pre-assigned Release Object Areas (ROAs), as shown in Figure 4.10 that depicts select moments from the experiment's execution. In the considered scenario, two object release positions are entered.



Figure 4.10: Screenshots of the video taken during an experiment.

The results of a series of experiments, in which a healthy user (the author of this work) has to perform pick and place operations several times, are presented below, while a video of the performed experiments is accessible via the following link:https://youtu.be/b2hzrrnmr8s. Specifically, the user has to teleoperate the manipulator near the bottle, pick it up, bring it close to one of the ROAs and place it there; after that, the user has to pick up the bottle again and place it on the other ROA. For all repetitions of the experiment, the end-effector started from a default configuration and the bottle was placed in the same initial configuration. The position

of the bottle on the table could be arbitrary, because of the perception module for object recognition, it was chosen to place it in the same initial configuration so that the user always starts from the same initial conditions.

It should be recalled that in order to teleoperate the robotic arm, the user has only three commands at his disposal: LEFT, RIGHT and SWITCH. The latter is not a real command, but rather a transitional state between the left and right classes, in which the user does not think anything. Figure 4.11 illustrates the input-output of the BCI filter over time during one of the experiments.

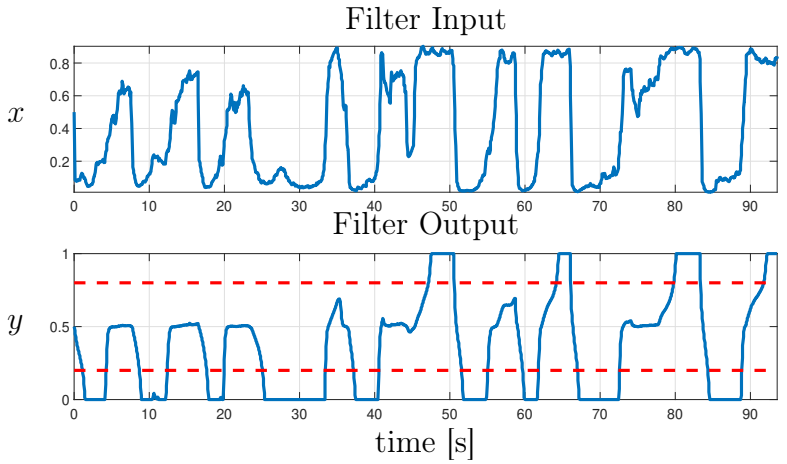


Figure 4.11: Input-output of the filter over time during the experiment.

The path followed by the manipulator end-effector in one of the experiments conducted to validate the control architecture is illustrated in Figure 4.12. The figure illustrates the initial and final points of the trajectory (Start and Stop points), the two ROAs (depicted in red and white), and the green bottle at the starting position and when it is on the two ROAs. The 3D path of the end-effector is highlighted in different colors according to the active control mode of the shared control architecture: red for *robot control* mode; green for *user 2-D control* mode; blue for *user 1-D control* mode. It is evident that the control mode changes several times during the execution of the sequence, effectively allowing the required tasks to

be performed using only three commands, one of which is a transition state.

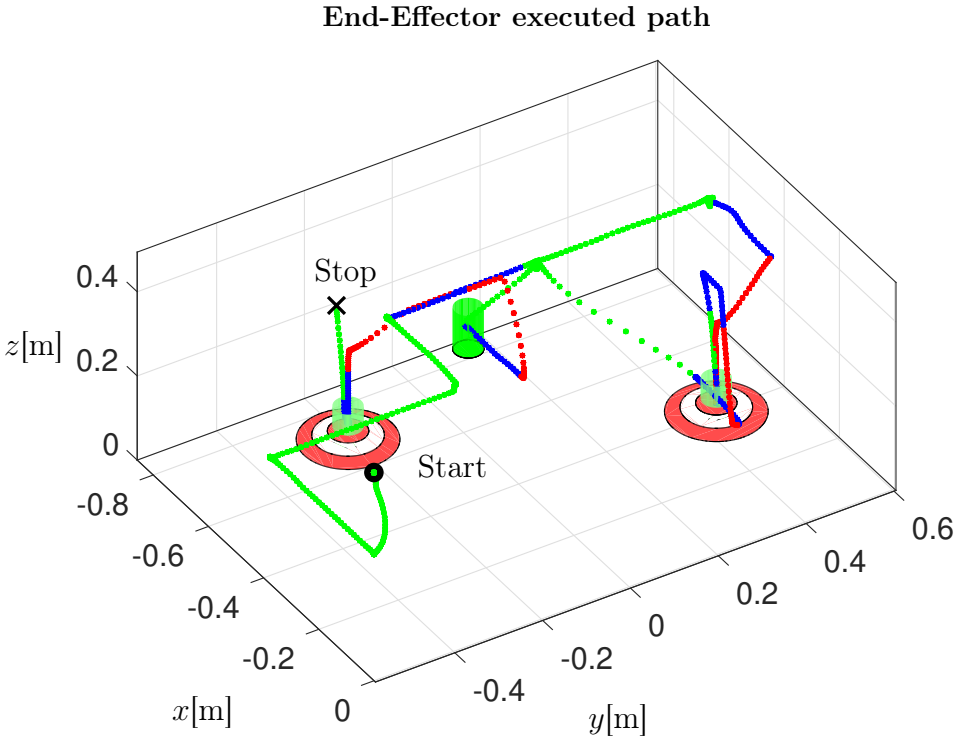


Figure 4.12: Path followed by the end-effector in one of the experiment runs. In the plot are highlighted the Start and Stop points, the two ROAs and the bottle in positions of interest.

Chapter 5

Shared Control Application for Human-Robot Interaction

This chapter presents an EMG-based shared control architecture designed for human-robot collaboration tasks. Using surface electromyographic signals, the system detects the human operator's muscle contractions and movements and adjusts the robot's operating modes through a variable admittance control strategy. The proposed structure has been experimentally validated in human-robot physical collaboration scenarios, demonstrating its potential in industrial and rehabilitation applications.

Surface ElectroMyoGraphy sensors play an important role in variable admittance control schemes to estimate the user's movement intention, as described in [132, 200] where the measurement of the operator's muscle activation are used to adapt the virtual damping online. This has led to approaches based on the integration of EMG signals and adaptive admittance control architectures for several applications including robotic manipulation, skill learning, and rehabilitation devices [201]. Since the admittance parameters determine how the robot responds to input signals, interacts with its environment, and coordinates its actions with those of the human operator, tuning control parameters is a critical aspect of designing control systems for human-robot collaboration. In [202] the authors emphasize the importance of carefully tuning the control parameters of the robotic system to optimize its performance and ensure effective collaboration with human.

Advanced implementations integrate components such as human motion prediction, adaptation algorithms, and virtual guidance to improve the

quality of assistance [203, 204]. For instance, authors in [205] proposed using sEMG signals to encode stiffness and movement patterns for robotic skill learning and adaptive behavior. Meanwhile, in [206] an sEMG-based gain-tuned compliance controller for lower limb rehabilitation robots to provide tailored assistance is developed. Additionally, the integration of sEMG and variable admittance control has shown promise in improving human-robot collaboration through compliant interaction in manufacturing tasks [136], synchronized motions in human-robot teams [207], and eliciting greater patient engagement in rehabilitation settings [208, 209]. Enabling motion intention interpretation for exoskeleton control is also a central goal [210]. In conclusion, the use of sEMG with variable admittance promises more adaptive, safe, and effective human-robot coordination in various applications [211].

Shared control frameworks in physical human-robot interaction often rely on intent detection and communication mechanisms. In [212], a comprehensive review of these frameworks is presented, highlighting the need for robust methods to ensure seamless collaboration. In contrast to these approaches, the proposed system uses EMG signals to dynamically adjust admittance parameters, providing a different approach to interpreting human intentions and improving collaboration through adaptive compliance. While in works such as [213], the authors explored advanced EMG signal processing using flexible nanogold sensors that significantly improved motion intention detection in their system, the proposed architecture uses non-invasive surface electrodes that still ensure the robustness of the signal processing despite advanced methods that could further validate its efficiency and identify opportunities for improved noise reduction. Compared to a human-in-the-loop optimization approach for shared autonomy, as shown in [214], which focuses on dynamic adaptation of control paradigms based on user preferences and task requirements, the approach used in the proposed control system could highlight its ability to generalize across users and scenarios. In addition, recent advances in EMG-guided prosthetic control [215] illustrate the potential of proportional EMG decoding to achieve fine motor control for high degree of freedom robotic hands. By drawing parallels with their technique, the adaptability and accuracy of the considered shared control system in industrial and rehabilitation applications can be further demonstrated. It is also important to emphasize the necessity of ensuring safe interaction

between humans and robots, especially in scenarios involving physical contact. By comparing the safety mechanisms employed in [216] with the stability provided by the variable admittance control strategy, the robustness of the system against unintended collisions or excessive forces can be validated.

The proposed architecture can dynamically adapt to user intentions using EMG signals, outperforming traditional heuristic or fixed parameter systems. However, the integration of advanced noise reduction techniques, user customization, and additional safety mechanisms could further enhance its effectiveness and versatility.

5.1 EMG-based shared control framework for human-robot co-manipulation tasks

This section presents a shared control architecture designed for human-robot co-manipulation tasks, that allows the human to switch among robot's operational modes through surface electromyography signals from the user's arm [217]. A support vector machine (SVM) classifier is employed to process the raw EMG data to identify two classes of contractions that are fed into a finite state machine algorithm to trigger the activation of different sets of admittance control parameters corresponding to the envisaged operational modes. The proposed architecture has been experimentally validated using a Kinova Jaco^2 manipulator, equipped with Force/Torque sensor at the end-effector, and with a user wearing Delsys Trigno Avanti EMG sensors on the dominant upper limb.

5.1.1 Problem description

This section presents a shared control architecture for human-robot co-manipulation tasks that allows the human operator to dynamically change the robot's operational mode using sEMG signals from the user's arm and hand. These signals are classified by a Support Vector Machine [218] algorithm, which detects various contractions and movements. The output of the classifier is then fed into a finite state machine algorithm, which adjusts the parameters of a variable admittance controller to allow for better human-robot interaction. Since the dynamic behavior of the manipulator

is not unique, managing different types of interactions in a unified way ensuring optimal co-manipulation experiences is not easy.

Key factors in determining admittance parameters include the robot's compliance with forces applied by human, the fluidity and intuitiveness of the manual driving experience, and the speed and accuracy of task execution. Lower damping improves maneuverability but can cause instability or inaccuracy, while higher damping enhances precision but reduces compliance, making co-manipulation more difficult and slower. Using sEMG signals to adjust admittance parameters is a novel approach that ensures stable robot behavior, contrasting with previous methods relying on heuristics [125] or specific *stable region* of parameters for specific interaction [124]. Wrench sensor readings make complex to distinguish intentional operator's gestures from unintentional collisions because the user holds the end-effector after the sensor that can not detect the forces between the contact point and the environment if the forces are exerted by the user; while integrating musculoskeletal activity allows to recognize such interactions that would be indistinguishable otherwise.

The following presents a shared control architecture that dynamically changes the variable admittance controller parameters based on the operator's EMG signals to enhance precision in task performance by reducing downtime, improving human-robot interactions for a more intuitive co-manipulation experience. The approach's effectiveness and robustness have been experimentally validated with a user wearing four sEMG sensors on his/her dominant upper limb and using a 7DOF manipulator equipped with a Force/Torque sensor for admittance control, despite not being torque-controlled.

The considered scenario is shown in Figure 5.1.



Figure 5.1: The scenario shows the experimental setup in which a human operator performs a co-manipulation task wearing four EMG sensors and using a Jaco^2 manipulator.

5.1.2 Shared control architecture

The proposed architecture includes two possible operational modes:

- *Low-Damping* mode: hand-guidance of the end-effector in the free space to allow the operator to move the end-effector, e.g., to reach a workpiece;
- *High-Damping* mode: hand-guidance of the end-effector near a surface or a workpiece to perform operations such as welding or painting.

To achieve this, the gains of the admittance controller of the manipulator are changed between two sets corresponding to each operational mode. A finite state machine algorithm handles the switching between these sets based on the movements or contractions of the operator's arm, which are recognized through an EMG-based classifier.

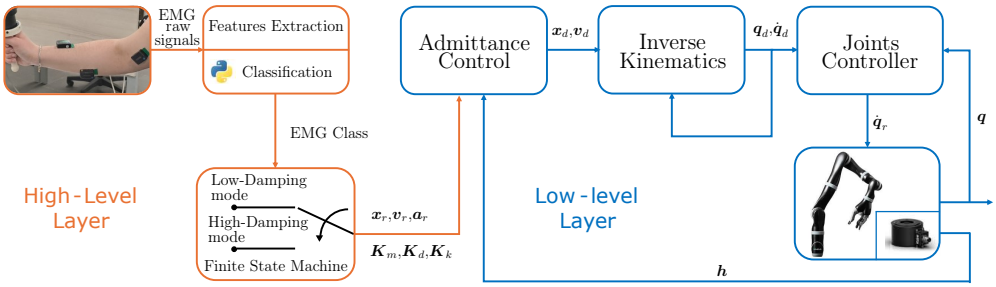


Figure 5.2: Proposed shared control architecture. The High-Level layer is responsible for the EMG signal acquisition and classification. Then, the recognized class is used into a finite-state machine algorithm that outputs a set of admittance gains. The Low-Level layer is responsible for the motion control of the manipulator, and it employs a variable admittance controller, an inverse kinematics algorithm and a joints controller.

Figure 5.2 shows the block diagram of the entire shared control architecture. The *High-level layer* includes all the functional blocks that generate specific sets of admittance gains, while the *Low-level layer* includes all the blocks that concern the motion control of the manipulator. The raw EMG signals go as input to a SVM classifier that recognizes two possible classes of motions/contractions of the operator. The finite state machine algorithm receives this information and it sets the admittance gains based on the operational mode; then, an admittance controller [174] outputs a desired trajectory for the end-effector, which is finally tracked by computing the needed joint velocity with an inverse kinematics algorithm [219]. In the following sections, details about all the functional blocks are given in a bottom-up order.

5.1.3 Low-level layer

The low-level layer implements the robot control algorithm and consists of three blocks: *i*) the joints controller, that generates actual joint velocity commands based on joint encoder readings and the inverse kinematics controller output; *ii*) the inverse kinematics controller, which computes the desired joint velocities that make the end-effector track the *desired* trajectory; and *iii*) the variable admittance controller, that modifies the

input *reference* trajectory for the end-effector, using wrench sensor readings, to achieve a desired dynamic behavior, and outputs a new *desired* trajectory for the end-effector.

Inverse kinematics and joints controller

Considering a serial manipulator with n Degrees of Freedom (DOFs), the state of the system and the inverse kinematics for the manipulator are described in Section 3.2. Assuming a redundant manipulator, i.e. $n > 6$, and the availability of a *desired* end-effector trajectory expressed as:

$$\boldsymbol{x}_d = \begin{bmatrix} \boldsymbol{p}_d \\ \boldsymbol{o}_d \end{bmatrix} \in \mathbb{R}^7 \quad \boldsymbol{v}_d = \begin{bmatrix} \dot{\boldsymbol{p}}_d \\ \boldsymbol{\omega}_d \end{bmatrix} \in \mathbb{R}^6 \quad \boldsymbol{a}_d = \begin{bmatrix} \ddot{\boldsymbol{p}}_d \\ \boldsymbol{\alpha}_d \end{bmatrix} \in \mathbb{R}^6 \,, \tag{5.1}$$

where $\boldsymbol{x}_d$ represents the desired position and quaternion, $\boldsymbol{v}_d$ gathers the desired linear and angular velocities and $\boldsymbol{a}_d$ represents the desired linear and angular accelerations. The desired joint velocity that makes the end-effector track the desired trajectory can be computed by resorting to the Closed-Loop Inverse Kinematics (CLIK) algorithm:

$$\dot{\boldsymbol{q}}_d = \boldsymbol{J}^{\dagger}(\boldsymbol{v}_d + \boldsymbol{K}_{\text{ik}}\tilde{\boldsymbol{x}}) \,, \tag{5.2}$$

where $\boldsymbol{J}^{\dagger}$ is the Moore-Penrose pseudoinverse of the Jacobian matrix, $\boldsymbol{K}_{\text{ik}} \in \mathbb{R}^{6\times6}$ is a positive-definite matrix of gains and $\tilde{\boldsymbol{x}}$ is the error vector, defined as:

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \tilde{\boldsymbol{p}} \\ \tilde{\boldsymbol{o}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q}_d) \\ \boldsymbol{o}_d^{-1} \star \boldsymbol{o}(\boldsymbol{q}_d) \end{bmatrix} \in \mathbb{R}^6 \,, \tag{5.3}$$

where $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{o}}$ are the position and quaternion errors, respectively, and "$\star$" is the quaternion product operator; $\boldsymbol{q}_d$ is the vector of desired joint position obtained by numerically integrating the desired joint velocity vector $\dot{\boldsymbol{q}}_d$, $\boldsymbol{p}(\boldsymbol{q}_d)$ and $\boldsymbol{o}(\boldsymbol{q}_d)$ are the position and orientation of the end-effector obtained by considering $\boldsymbol{q}_d$ as joint positions in the direct kinematics computation.

Finally, the desired joint velocity vector is passed to the joints controller, which computes the actual command to send to the manipulator by resorting to a proportional controller including a feedforward action as:

$$\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d + \boldsymbol{K}_{\text{jc}}\tilde{\boldsymbol{q}} \,, \tag{5.4}$$

where $\boldsymbol{K}_{\text{jc}} \in \mathbb{R}^{n \times n}$ is a positive-definite matrix of gains and $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$ is the joint position error computed with respect to the actual readings of the joint encoders.

Admittance control

The admittance controller aims to assign a virtual dynamics to the manipulator, characterized by a desired mass, damping, and stiffness (as described in Section 3.7.2). More in detail, assuming to have a wrench sensor mounted on the wrist of the manipulator and to have a certain *reference* trajectory for the end-effector, the *desired* end-effector trajectory must have the following dynamics:

$$\boldsymbol{K}_m \, \tilde{\boldsymbol{a}}_{r,d} + \boldsymbol{K}_d \, \tilde{\boldsymbol{v}}_{r,d} + \boldsymbol{K}_k \, \tilde{\boldsymbol{x}}_{r,d} = \boldsymbol{h} \ , \tag{5.5}$$

where the positive-definite matrices:

$$\boldsymbol{K}_m = \begin{bmatrix} \boldsymbol{K}_m^p & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{K}_m^o \end{bmatrix} \tag{5.6}$$

$$\boldsymbol{K}_d = \begin{bmatrix} \boldsymbol{K}_d^p & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{K}_d^o \end{bmatrix} \tag{5.7}$$

$$\boldsymbol{K}_k = \begin{bmatrix} \boldsymbol{K}_k^p & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{K}_k^o \end{bmatrix} \ , \tag{5.8}$$

represent the virtual mass, damping, and stiffness, respectively, where the position $\boldsymbol{K}_{(\cdot)}^p$ and orientation $\boldsymbol{K}_{(\cdot)}^o$ gains are highlighted, $\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{T}} & \boldsymbol{\mu}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6$ is the vector stacking the linear force and moments measured by the wrench sensor, and:

$$\tilde{\boldsymbol{x}}_{r,d} = \begin{bmatrix} \tilde{\boldsymbol{p}}_{r,d} \\ \tilde{\boldsymbol{o}}_{r,d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_r - \boldsymbol{p}_d \\ \boldsymbol{o}_r^{-1} \star \boldsymbol{o}_d \end{bmatrix} \tag{5.9}$$

$$\tilde{\boldsymbol{v}}_{r,d} = \boldsymbol{v}_r - \boldsymbol{v}_d \tag{5.10}$$

$$\tilde{\boldsymbol{a}}_{r,d} = \boldsymbol{a}_r - \boldsymbol{a}_d \ , \tag{5.11}$$

are the operational space configuration, velocity and acceleration errors computed between the *reference* and the *desired* trajectories. The desired acceleration in output from the admittance controller can be computed by

folding Eqs (5.9)-(5.11) in Eq. (5.5) and rearranging the terms, obtaining:

$$\boldsymbol{a}_d = \boldsymbol{K}_m^{-1} \left[\boldsymbol{K}_m \, \boldsymbol{a}_r + \boldsymbol{K}_d \, \tilde{\boldsymbol{v}}_{r,d} + \boldsymbol{K}_k \, \tilde{\boldsymbol{x}}_{r,d} - \boldsymbol{h} \right] . \tag{5.12}$$

The desired end-effector velocity $\boldsymbol{v}_d$ and configuration $\boldsymbol{x}_d$ to be tracked by the inverse kinematics controller in Eq. (5.2) can then be obtained by numerical integration of the desired acceleration. It is worth noticing that the virtual mass, damping and stiffness can be varied, obtaining a variable admittance controller.

Each one of the operational modes listed in Sec. 5.1.2 has a corresponding set of admittance gains that adjust the robot's behavior to successfully execute them. In detail, with reference to Eqs. (5.6)-(5.8), when in *Low-Damping* mode the $\boldsymbol{K}_k$ gain is set to $\boldsymbol{O}_{6\times6}$ to allow the operator to freely change the end-effector position and orientation. The matrix $\boldsymbol{K}_d^p$ is set as $\boldsymbol{K}_d^p = \text{diag}\{k_{d,\text{low}}^p, k_{d,\text{low}}^p, k_{d,\text{low}}^p\}$, with *low* element $k_{d,\text{low}}^p$ to assure smooth teleoperation to the user; similarly, for the orientation, the matrix $\boldsymbol{K}_d^o$ is set as $\boldsymbol{K}_d^o = \text{diag}\{k_{d,\text{low}}^o, k_{d,\text{low}}^o, k_{d,\text{low}}^o\}$, with *low* element $k_{d,\text{low}}^o$. Regarding the virtual mass, the matrix $\boldsymbol{K}_m^p$ is set to diag $= \{k_{m,\text{low}}^p, k_{m,\text{low}}^p, k_{m,\text{low}}^p\}$, while the matrix $\boldsymbol{K}_m^o$ is set to diag $= \{k_{m,\text{low}}^o, k_{m,\text{low}}^o, k_{m,\text{low}}^o\}$.

When in *High-Damping* mode the $\boldsymbol{K}_k$ gain is still set to $\boldsymbol{O}_{6\times6}$, while $\boldsymbol{K}_d^p$ is set as $\boldsymbol{K}_d^p = \text{diag}\{k_{d,\text{high}}^p, k_{d,\text{high}}^p, k_{d,\text{high}}^p\}$ with *high* element $k_{d,\text{high}}^p$ to ensure slow and precise movements, for example when painting a thin element. Regarding the orientation, the matrix $\boldsymbol{K}_d^o$ is set as $\boldsymbol{K}_d^o = \text{diag}\{k_{d,\text{high}}^o, k_{d,\text{high}}^o, k_{d,\text{high}}^o\}$, with *high* element $k_{d,\text{low}}^o$. Finally, the virtual mass, the matrix $\boldsymbol{K}_m^p$ is set to diag $= \{k_{m,\text{high}}^p, k_{m,\text{high}}^p, k_{m,\text{high}}^p\}$, while the matrix $\boldsymbol{K}_m^o$ is set to diag $= \{k_{m,\text{high}}^o, k_{m,\text{high}}^o, k_{m,\text{high}}^o\}$.

5.1.4 High-level layer

The high-level layer is composed of three main functional blocks: *i*) the EMG sensors, *ii*) the operator's motion classifier, and *iii*) the finite state machine algorithm.

EMG Signal Analysis

To detect electrical signals produced by the muscles when they contract or relax, surface or needle-based electrodes are used; in this work, non-invasive surface electrodes attached to the skin with adhesive are considered. The precise placement of these electrodes on the target muscle plays a crucial role, as the amplitude and frequency content of the EMG signals vary with electrode positioning relative to the muscle fibers. Before being processed these signals need to be amplified, digitized and filtered to remove noise and processed to extract features to evaluate muscle functionality, activation or distinction between different types of contractions. In this work, EMG signals are used to detect muscle activity and patterns related to specific movements or actions. In order to achieve this objective, it is essential to utilize a signal processing classifier. A review of the literature on classifiers and preliminary tests performed on some of them, including those based on Random Forest, revealed that a Support Vector Machine (SVM) classifier was the most suitable for the purposes of this work.

The SVM classifier, that is a widely used supervised learning algorithm for classification tasks, receives a set of features extracted from the EMG signals. In particular, three time-domain features commonly-used for EMG signals classification [204, 220] are extracted:

- The Root Mean Square (RMS) can be used to evaluate muscle activity and fatigue; it is defined as:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} y_i^2} \tag{5.13}$$

- The Mean Absolute Value (MAV) can be used to assess the intensity of muscle contraction. It's definition is:

$$MAV = \frac{1}{N}\sum_{i=1}^{N} |y_i| \tag{5.14}$$

- The Average Amplitude Change (AAC) provides information about fluctuations in muscle activity over a period of time and about the

level of muscle activation during that period. It is useful for assessing muscle fatigue, tracking changes in muscle activity, and comparing muscle involvement during different activities or conditions. It can be computed as:

$$AAC = \frac{1}{N}\sum_{i=1}^{N}|y_{i+1} - y_i| \tag{5.15}$$

For all the features, y_i is the i-th element of the vector $\boldsymbol{y}$ that contains the amplitudes of the EMG signals in a predefined epoch of 1s and N is the number of elements contained in $\boldsymbol{y}$. To train the SVM classifier, a dataset of feature vectors with associated class labels, obtained with a predetermined acquisition procedure, is used. The SVM algorithm learns a decision boundary to separate different classes by finding an optimal hyperplane in the feature space, governed by two main hyperparameters: the regularization parameter C, which prevents overfitting by limiting misclassification, and γ, which determines how much far points influence the hyperplane calculation. During the training, the SVM adjusts the parameters by solving a quadratic programming problem to identify the support vectors, i.e., the points closest to the decision boundary.
In the proposed scenario, the classifier is trained to discriminate between two classes: *free* (movements made by the operator in various directions with minimal resistance) and *contraction* (generated by the operator by tightening his hand).

Finite State Machine

The two classes in output from the SVM classifier are fed into a finite state machine algorithm to automatically switch between the operational modes, as shown in Figure 5.3.

In detail, the system starts in *Low-Damping* mode; when the operator executes the *contraction* class for at least 1.5s, the state switches to *High-Damping* mode, corresponding to a set of admittance gains that allow the operator to move the end-effector to follow a predefined path, without touching it, with lower speed and higher accuracy due to the increased damping. By generating the *contraction* class for at least 3s, the state returns to the *Low-Damping* mode, in which the operator can move the

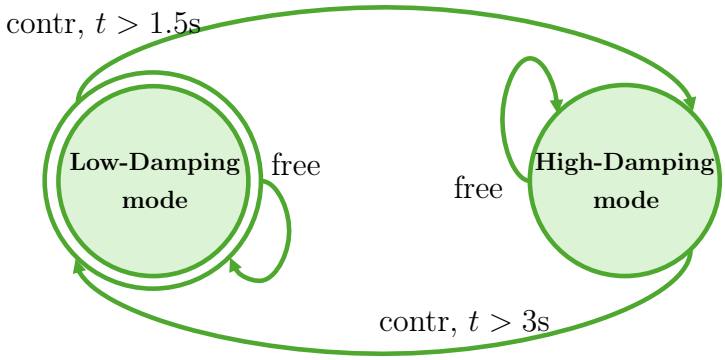


Figure 5.3: Finite State Machine.

end-effector more easily and faster towards the starting point to execute the path again.

5.1.5 Experimental validation

Experimental Setup

To validate the effectiveness of the proposed shared control architecture, experiments of human-robot co-manipulation tasks were conducted using a 7DOF Kinova Jaco[2] manipulator[1] equipped with a Bota Systems Rokubimini force/torque sensor[2]. Four non-invasive Delsys Trigno Avanti active sensors[3] with a frequency of 1.78 kHz were placed on the dominant arm of the human operator. Following Seniam guidelines[4], they were put on the *biceps* and *triceps brachii*, *flexor carpi radialis* and *extensor carpi ulnaris* muscles, as much as possible in the center of muscles. The streaming of EMG data was done through a Python GUI on a Windows PC, while the control software and the SVM classifier, developed using Python and the Scikit-learn library [221], were implemented in the ROS (Robotic Operating System) framework running on Linux. To allow the two workstations to communicate TCP/IP sockets were used.

[1]https://www.kinovarobotics.com/product/gen2-robots
[2]https://www.botasys.com/force-torque-sensors/rokubi
[3]https://delsys.com/trigno-avanti/.
[4]http://seniam.org

Training and test of the classifier

To train the SVM classifier described in section 5.1.4, EMG signals from a human operator's arm were recorded while performing specific movements or contractions interacting with a manipulator. Two interactions were considered for data collection: in the first, the robot's behavior was made compliant, using the *Low-Damping* mode parameters, and the operator moved the end-effector in random directions to collect data related to the *free* class; in the second, the robot had a rigid behavior, obtained by referring to a standard position controller, and the operator tightened the hand on the end-effector handle to collect data related to the *contraction* class. A Matlab GUI told the operator which movements or contractions to perform and automatically labeled the data to obtain two datasets: one for training and one for testing the classifier. For each of the four EMG sensors, three time-domain features were extracted, resulting in a 12-element feature vector, with an associated class label. The SVM model is trained using the hyperparameters in Table 5.1.

The confusion matrix for binary problems, where the rows represent the expected class distribution and the columns the predicted distribution by the classifier, was used to evaluate classifier performance and to compute class metrics such as *A*ccuracy, *P*recision, *R*ecall, and *F1*-score, yielding the following indexes:

$$A = \frac{TP}{TP + TN + FP + FN} \tag{5.16}$$

$$P = \frac{TP}{TP + FP} \tag{5.17}$$

$$R = \frac{TP}{TP + FN} \tag{5.18}$$

$$F1 = \frac{TP}{TP + \frac{FN+FP}{2}}, \tag{5.19}$$

where TP, TN, FP and FN are true positives/negatives and false positives/negatives taken from the binary problem confusion matrix, respectively. True positives and negatives are the correctly identified positive/negative classes, while false positives and negatives are the incorrectly classified values. The accuracy measures how many instances were correctly classified, the precision measures the accuracy of the positive

predictions, the recall is the ratio of positive instances that are correctly detected, and the F1-score is the harmonic mean of precision and recall.

In the on-line use of the classifier, the statistical mode of the predicted classes calculated over the last 1.5s is sent to the finite state machine to increase the robustness in terms of misclassifications.

Experimental results

To validate the overall architecture by triggering all the envisioned state transitions, a series of experiments were performed.

Control parameters		
Parameter	**Value**	**Description**
$\boldsymbol{K}_{\text{ik}}$	$\text{diag}\{20\ \boldsymbol{I}_3, 15\ \boldsymbol{I}_3\}$	Inverse kinematics gain
$\boldsymbol{K}_{\text{jc}}$	$\text{diag}\{3\ \boldsymbol{I}_4, 2\ \boldsymbol{I}_3\}$	Joints controller gain
$k^p_{d,\text{low}}$	40	Low-damping position gain
$k^o_{d,\text{low}}$	2	Low-damping orientation gain
$k^p_{m,\text{low}}$	3	Low-mass position gain
$k^o_{m,\text{low}}$	0.1	Low-mass orientation gain
$k^p_{d,\text{high}}$	120	High-damping position gain
$k^o_{d,\text{high}}$	4	High-damping orientation gain
$k^p_{m,\text{high}}$	5	High-mass position gain
$k^o_{m,\text{high}}$	0.1	High-mass orientation gain
Classifier parameters		
Parameter	**Value**	**Description**
C	1	Regularization parameter
γ	*'scale'*[5]	Parameter of a Gaussian Kernel
Kernel	'RBF'	Kernel used in the SVM Classifier

Table 5.1: Experimental parameters used in the proposed architecture.

[5] 'scale' is a parameter calculated as 1 divided by the product of the features number and the variance of the feature vector.

A subject[6] (female, aged 29) wearing EMG sensors had to guide the robot end-effector several times along a predefined path, as if she were performing a painting operation on a thin element, as shown in the scenario in Figure 5.1. The path consists of a cylindrical filament whose shape is similar to a square wave. The user repeated the designed experiment 15 times to test the reproducibility of the approach. Table 5.1 shows the parameters for the controllers and the classifier used in the experiments, where all the admittance parameters were experimentally determined; while a video showing the experiments is provided at the following link: https://youtu.be/93u5i8HmsPY.

In the following, the results of the classifier performances and the experimental execution are discussed. In particular, the confusion matrix is shown in Figure 5.4, while Table 5.2 contains the classifier indexes.

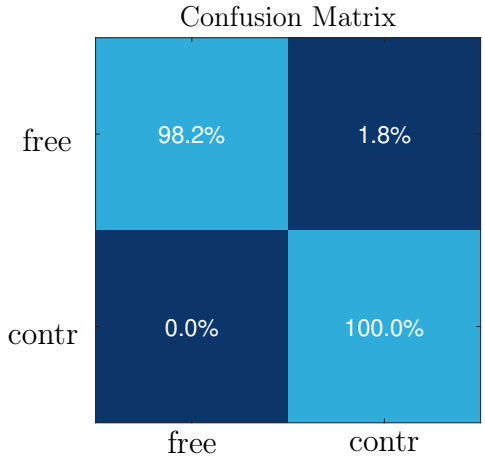


Figure 5.4: The confusion matrix obtained after training the classifier.

All performance indexes exceed 99%, as indicated in the confusion matrix, which has only 1.8% of misclassifications for *contraction* class. This means that, in addition to the high classification performance attested by the high accuracy value, the number of false positives is also very low. The

[6]The experiments were taken in accordance with the Declaration of Helsinki, the protocol has been approved by the Research Ethics Committee at the university where the study is conducted and the subject gave informed consent.

precision and recall also indicate a solid balance confirmed by the F1-score, whose value is close to the accuracy. Thus, the numerical results attest a reliable and accurate classification capability, also supported by the use of only two classes.

Accuracy (%)	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
99.20	99.28	99.12	99.19

Table 5.2: Table containing performance evaluation metrics.

Figure 5.5 shows the path followed by the end-effector during one of the experiments performed, including the start and end points.

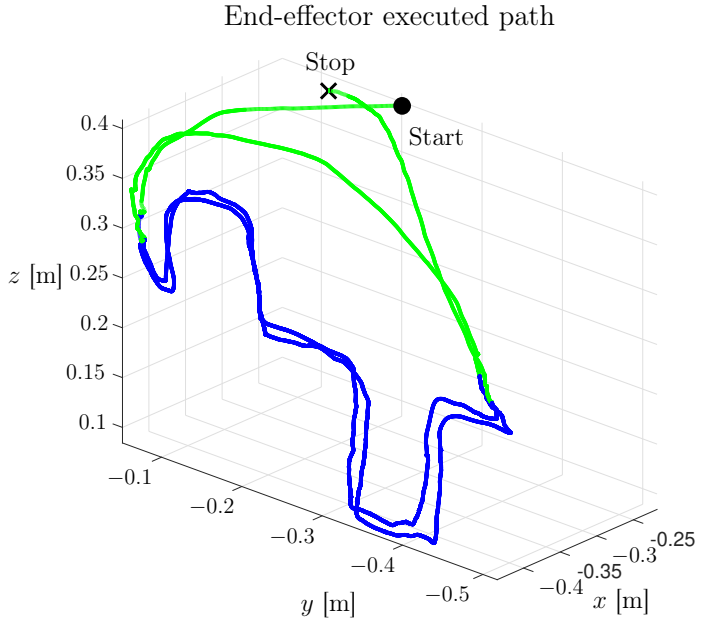


Figure 5.5: The path followed by the manipulator's end-effector in one of the experiments performed. The path is highlighted with two colors based on the two operational modes: *Low-Damping* mode is marked in green and *High-Damping* mode is highlighted in blue.

Two different colors were used to highlight the path followed by the manipulator, depending on the corresponding active operational mode: green and blue paths correspond to *Low-Damping* and *High-Damping* modes, respectively.

The user drives the end-effector near the beginning of the filament to be followed, generates the *contraction* class for 1.5s to switch to *High-Damping* mode, and moves the end-effector along the filament without touching it using the *free* class. By generating *contraction* for at least 3s, the user switches the FSM back to *Low-Damping* mode and quickly moves the end-effector back to start point with the *free* class (as can be seen by the green edge connecting the ends of the blue lines). Then, the user generates *contraction* for 1.5s, returning the FSM to *High-Damping* mode, and uses the *free* class to follow the filament again; in fact, there are two blue lines in the shape of a square wave. Finally, the user switches the FSM back to *Low-Damping* mode using *contraction* for 3s and moves the manipulator to the endpoint.

The user was able to complete the experiment without difficulty by easily changing the operational modes, as outlined in the NASA-TLX[7] questionnaire she completed.

Figure 5.6 shows the time evolution of the classifier output, the force measured by the F/T sensor, and the linear velocity of the manipulator's end-effector.

Similarly to what was highlighted in the performed path, the colors green and blue indicate the *Low-Damping* and *High-Damping* modes, respectively. In particular, Figure 5.6*a* shows the time evolution of the classifier output, which provides information about how it was able to distinguish between *free* and *contraction* classes. The classifier output determines the transition between the control modes, as highlighted by the different colored areas; in fact, after the first 6s (or after 83s) in which the operator is in *free* class, she generates *contraction* for 1.5s and the FSM switches to *High-Damping* mode; the FSM remains in the same state until the operator stops following the filament; then, the operator generates a 3s *contraction* (after 72 or 146s) and the FSM returns to *Low-Damping* mode.

[7]https://humansystems.arc.nasa.gov/groups/TLX/

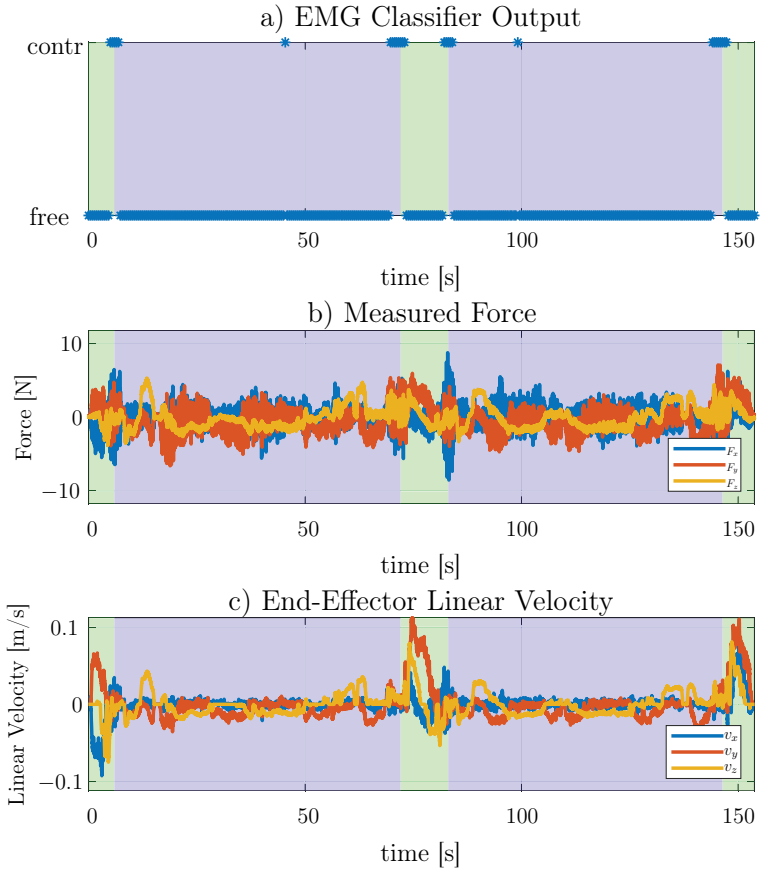


Figure 5.6: Experimental results: a) EMG Classifier Output; b) Force measured by the F/T sensor; c) End-Effector Linear Velocity. The different colored areas on all graphs correspond to the two different control modes: green and blue represent the *Low-Damping* and the *High-Damping* modes, respectively.

Instead, the time evolution of the force and linear velocity of the end-effector are shown in Figures 5.6*b* and 5.6*c*, respectively. The velocity increases significantly in the green areas, corresponding to a mode with low damping values, compared to the blue areas where, since the damping is higher, the manipulator becomes less compliant and moves more slowly during the co-manipulation.

Oscillations in the force trend also increase in the green areas corresponding to *Low-Damping*, especially when the user generates the *contraction*

class to switch modes. In the transition from *Low-Damping* to *High-Damping* this happens because, before the transition occurs, the force sensor perceives the user's hand as a rigid external environment; while in the transition from *High-Damping* to *Low-Damping* the operator does not stop contracting immediately after the transition because of his reflexes, and the lower damping is not sufficient to reduce the oscillations due to residual contractions.

5.2 EMG-driven shared control architecture for human-robot co-manipulation tasks

The following section presents a shared control strategy that allows a human operator to physically guide the end-effector of a robotic manipulator to perform different tasks, possibly in interaction with the environment [222]. To switch among different operational modes referring to a finite state machine algorithm, ElectroMyoGraphic signals from the user's arm are used to detect muscular contractions and to interact with a variable admittance control strategy. Specifically, a Support Vector Machine classifier processes the raw EMG data to identify three classes of contractions that trigger the activation of different sets of admittance control parameters corresponding to the envisaged operational modes. The proposed architecture has been experimentally validated using a Kinova Jaco^2 manipulator, equipped with Force/Torque sensor at the end-effector, and with a group of users wearing Delsys Trigno Avanti EMG sensors on the dominant upper limb.

5.2.1 Problem Description

In the following, a shared control architecture suitable for human-robot co-manipulation tasks is proposed. In detail, the human operator can dynamically change the robot operational mode by performing different contractions with his/her arm, which are recognized and classified online by monitoring sEMG signals and referring to a Support Vector Machine (SVM) algorithm [218].

The classifier output is fed into a finite-state machine algorithm that

changes the variable admittance controller parameters to allow human-robot collaboration in different kinds of interaction between the robot and the environment. Devising a control algorithm that can handle different kinds of interactions in a unified manner is challenging due to the manipulator's dynamic behavior not being unique. On one hand, the end-effector should be compliant with the forces exerted by the human operator to provide a smooth and intuitive hand-guiding experience. On the other hand, low damping can cause undesirable resonance and unstable motions when the robot is in contact with stiff environments. The use of EMG sensors to modify the admittance parameters despite the type of interaction forces is a novel approach to ensure stable robot behavior with respect to approaches that only consider robot motion, such as the one in [125], which estimates unstable robot behavior through heuristics, and the one in [124], which only identifies a *stable region* in terms of parameters for a specific type of interaction.

It is worth noticing that a possible alternative would have been to steer the admittance gains based on the wrench sensor readings instead of EMG signals, but this choice would have had two main drawbacks: first, it would be complex to distinguish intentional operator's gestures from unintentional collisions; second, since the operator usually holds the end-effector right after the wrench sensor mounted on the robot wrist, when the end-effector is brought in contact with a surface by the user, the wrench sensor is not able to properly detect the interaction forces between the handler and the environment due the presence of not measurable forces applied by the user. Using information about the user's musculoskeletal activity, instead, allows to properly manage also this kind of interaction, which would have been indistinguishable otherwise.

Therefore, the main contribution of this work can be found in the design of a shared control architecture that, by processing the EMG signals of the operator, dynamically changes the variable admittance controller parameters in order to overcome resonance and robot instability issues and to allow different human-robot and robot-environment interactions. The effectiveness and the robustness of the approach have been experimentally validated employing a 7DOFs manipulator equipped with a Force/Torque sensor, and with a group of four users wearing four sEMG sensors on their dominant upper limb. Since the manipulator used to test the proposed architecture can not be torque-controlled, a variable admittance control



Figure 5.7: The scenario shows the experimental setup in which a human operator performs a co-manipulation task using a Jaco^2 manipulator. On the dominant arm of the human operator, four Delsys Trigno Avanti EMG sensors are placed on the biceps and triceps brachii, flexor carpi radialis and extensor carpi ulnaris muscles.

strategy has been chosen. The considered scenario is shown in Figure 5.7.

5.2.2 Shared control architecture

In the proposed architecture, three possible operational modes are taken into account:

- *no-contact* mode: hand-guidance of the end-effector in the free space to allow the operator to move the end-effector towards, e.g., a workpiece;
- *contact* mode: hand-guidance of the end-effector while in contact with a stiff surface to allow performing operations like, e.g., carving, welding or drawing;
- *hold* mode: in case of external disturbances, the manipulator is designed to elastically return the end-effector to a fixed position;

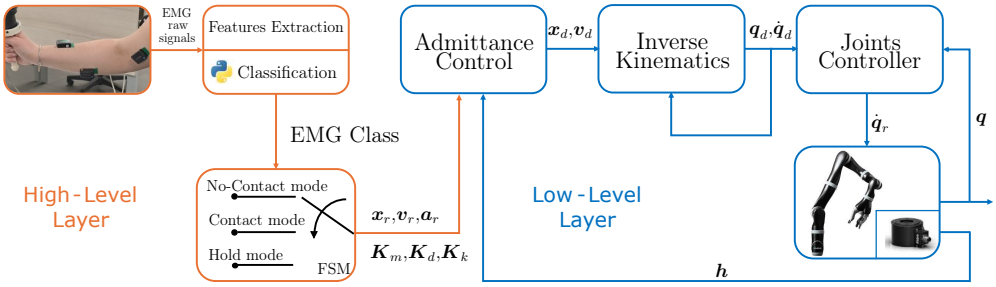


Figure 5.8: Proposed shared control architecture. The High-Level layer is responsible for the EMG signal acquisition and classification. Then, the recognized class is used into a finite-state machine algorithm that outputs a set of admittance gains. The Low-Level layer is responsible for the motion control of the manipulator, and employs a variable admittance controller, an inverse kinematics algorithm and a joints controller.

this position is the same at which the switch to the *hold* mode occurs. This allows for the temporary relocation of the robot away from the point of interest, facilitating the placement of a workpiece that the robot is then responsible for maintaining in that position.

The proposed approach allows for the tackling of a wide range of common tasks in industrial scenarios through the combination of these three operational modes. To achieve this, the gains of the admittance controller of the manipulator are changed among three sets corresponding to each operational mode. A finite state machine algorithm handles the switching among these sets based on the movements or contractions of the operator's arm, which are recognized through an EMG-based classifier. Figure 5.8 shows the block diagram of the entire shared control architecture. The *High-level layer* includes all the functional blocks that address the generation of a certain set of admittance gains, while the *Low-level layer* includes all the blocks that concern the motion control of the manipulator. The raw EMG signals go as input to a SVM classifier that recognizes three possible classes of motions/contractions of the operator. The finite state machine algorithm receives this information and configures a set of admittance gains on the base of the operational mode; then, an admittance controller [174] outputs a desired trajectory for the

end-effector, which is finally tracked by computing the needed joint velocity with an inverse kinematics algorithm [219]. In the following sections, details about all the functional blocks are given in a bottom-up order.

5.2.3 Low-level layer

The low-level layer implements the robot control algorithm and it is composed of three blocks: *i*) the joints controller, *ii*) the inverse kinematics controller and *iii*) the variable admittance controller. The variable admittance controller takes in input a certain *reference* trajectory for the end-effector and the readings of the wrench sensor mounted on the wrist of the manipulator, modifies it to comply with a desired variable dynamic behavior and outputs a new *desired* trajectory for the end-effector. The inverse kinematics controller is responsible for computing the desired joint velocities that make the end-effector track the *desired* trajectory, while the joints controller generates the actual joint velocities command to the manipulator based on the joint encoder readings and on the inverse kinematics controller output.

Inverse kinematics and joints controller

The shared control architecture proposed in this section employs the same *Inverse kinematics controller* and *Joints controller* modules that were previously described in Section 5.1.3.

Admittance control

As previously stated in Section 5.1.3, the objective of the admittance controller is to assign a virtual dynamic to the manipulator, which is defined by a desired mass, damping, and stiffness. The theoretical formulation of the admittance controller remains consistent with that presented in the previous section. The following delineates how to assign this dynamic in accordance with the actual goal of the control architecture.

Each one of the operational modes listed in Sec. 5.1.3 has a corresponding set of admittance gains and potentially *requested* trajectory ($\boldsymbol{x}_r$, $\boldsymbol{v}_r$, $\boldsymbol{a}_r$) that adjust the robot's behavior to successfully execute them. In detail, with reference to Eqs. (5.6)-(5.8), when in *no-contact* mode the $\boldsymbol{K}_k^p$ gain is set to $\boldsymbol{O}_{3\times3}$ to allow the operator to freely move the end-effector and

$\boldsymbol{K}_d^p$ is set as $\boldsymbol{K}_d^p = \text{diag}\{k_{d,\text{low}}, k_{d,\text{low}}, k_{d,\text{low}}\}$, with *low* element $k_{d,\text{low}}$ to assure smooth teleoperation to the user; when in *contact* mode the $\boldsymbol{K}_k^p$ gain is still set to $\boldsymbol{O}_{3\times3}$, while $\boldsymbol{K}_d^p$ is set as $\boldsymbol{K}_d^p = \text{diag}\{k_{d,\text{low}}, k_{d,\text{low}}, k_{d,\text{high}}\}$ with *high* element $k_{d,\text{high}}$ to guarantee both a stable contact with a stiff surface parallel to the xy plane of the arm base frame (e.g. a table where the workpieces are placed) and a smooth hand-guiding experience on the other two directions; finally, when in *hold* mode, the $\boldsymbol{K}_k^p$ gain is set to $\boldsymbol{K}_k^p = \text{diag}\{k_k, k_k, k_k\}$ with $k_k > 0$ and $\boldsymbol{x}_r$ is set to the end-effector position at the time of the state switch, while $\boldsymbol{K}_d^p$ is set as $\boldsymbol{K}_d^p = \text{diag}\{k_{d,\text{low}}, k_{d,\text{low}}, k_{d,\text{high}}\}$ as in *contact* mode, to make the end-effector hold the position while being in contact with a stiff environment on the z coordinate and compliant with external forces coming from the other two directions. In all the possible states, the orientation of the end-effector is handled by letting the operator always adjust it with low damping, i.e. $\boldsymbol{K}_k^o = \boldsymbol{O}_{3\times3}$ and $\boldsymbol{K}_d^o = \text{diag}\{k_{d,\text{low}}, k_{d,\text{low}}, k_{d,\text{low}}\}$. Finally, the virtual mass in Eq. (5.6) is kept constant in all the states.

5.2.4 High-level layer

The high-level layer is composed of three main functional blocks: *i*) the EMG sensors, *ii*) the operator's motion classifier, and *iii*) the finite state machine algorithm.

EMG signal processing and classification

As described in Section 5.1.4, EMG signals are collected using non-invasive surface electrodes attached to the skin with adhesive to detect muscle activity and patterns related to specific movements and contractions. Similarly, a support vector machine classifier was employed, and the three time-domain features *Root Mean Square* (RMS), *Mean Absolute Value* (MAV) and *Average Amplitude Change* (AAC) are extracted using the expressions provided in (5.13)- (5.15). The SVM algorithm employs the identical training procedure delineated in Section 5.1.4.

In the proposed scenario, the classifier is trained to discriminate among three classes: *free*, which includes all movements made by the operator moving the arm in various directions in the absence or with limited resisting forces; *contraction*, generated by the operator when he/she tightens

his hand; and *up*, detected when the operator pushes upwards in the presence of resisting forces.

Finite State Machine

The three classes in output from the SVM classifier are fed into a finite state machine algorithm to automatically switch among the three operational modes, as shown in Figure 5.9.

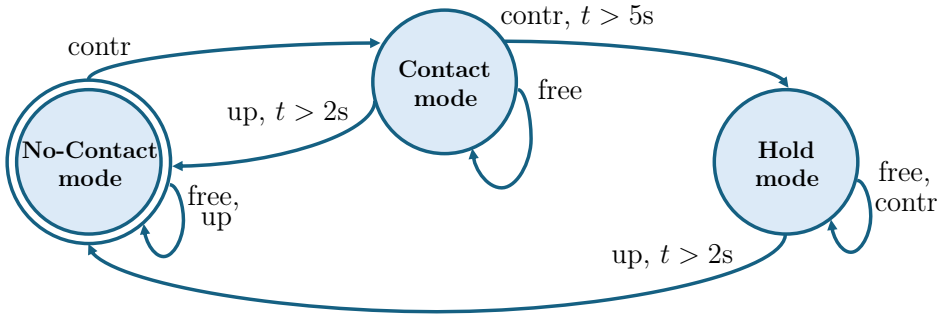


Figure 5.9: Finite State Machine. Starting from the *no-contact* mode, the FSM switches to a different control mode based on the class triggered by the human operator and the time it is maintained.

In detail, the system starts in *no-contact* mode and stays in this state until the classifier recognizes the *contraction* class from the operator. When this happens, the state is changed to *contact*, giving in output a set of admittance gains that allows to safely put the end-effector in contact with a stiff environment without any unstable behavior. From this state, the operator can choose to switch back to *no-contact* mode, by triggering the *up* class for at least 2s, or to go into *hold* mode by generating again the *contraction* class for at least 5s. In this state, the end-effector holds the current position while the operator can still physically interact with it, and it remains in this state until the operator generates the *up* class for at least 2s, switching back to *no-contact* mode.

5.2.5 Experimental validation

Experimental Setup

To validate the effectiveness of the proposed shared control architecture, experiments of human-robot co-manipulation tasks were conducted using a 7DOF Kinova Jaco^2 manipulator equipped with a Bota Systems Rokubimini force/torque sensor. Four non-invasive Delsys Trigno Avanti sensors with a frequency of 1.78 kHz, were also used; these are active sensors that already have pan integrated pre-amplification circuit that allows the reduction of input noise, such as parasitic voltages due to capacitive couplings or the possible displacement of the electrodes. Following Seniam guidelines, sensors were placed on the dominant arm of the human operator, as much as possible in the center of the following muscles: the *biceps* and *triceps brachii*, *flexor carpi radialis* and *extensor carpi ulnaris* muscles, as shown in the scenario in Fig. 5.7.

The streaming of EMG data was done through a Python GUI on a Windows PC, while the control software and the SVM classifier, developed using Python and the Scikit-learn library [221], were implemented in the ROS (Robotic Operating System) framework running on Linux. To allow the two workstations to communicate TCP/IP sockets were used.

Training and test of the classifier

As outlined in the section on SVM classification 5.1.4, the SVM classifier requires two data sets for classification purposes: a training data set and a test data set. The two data sets are collected separately using the same procedure, in which the human operator performs a sequence of predefined movements and contractions while interacting with the manipulator. Each movement or contraction is performed for 5s and, considering a frequency of 1.78 kHz, approximately 8900 samples will be obtained for each movement. The two data sets contain the same number of movements or contractions and the same number of samples.

Two distinct behaviors of the robot are considered in the data collection procedure. In the first case the robot is compliant: this behavior is obtained with the robot in admittance control using the *no-contact* mode parameters set, and it is used for acquiring data regarding the *free* class by moving the end-effector in random directions. In the second case, the

robot is rigid: this behavior is obtained by referring to a standard position controller and is used to gather data regarding the *contraction* (by tightening the hand on the end-effector handle), and *up* (by pushing the end-effector upwards) classes. In the process of acquiring training and test data, a Matlab GUI was utilized to indicate to the operator what movements or contractions to perform and for how long, and to automatically label the data with the appropriate class.

Both the training and test datasets obtained with the aforementioned acquisition procedure are processed by extracting the time domain features listed in Sec. 5.1.4 using Python for the classifier training. These three features are extracted for each of the four EMG sensors and transformed into a feature vector of 12 elements with an associated label class needed to train the classifier.

The SVM model is trained using the hyperparameters in Table 5.3 and these parameters were not modified for individual participants, as one of the objective of the study was to evaluate the generalizability of the proposed method across different users. The use of predefined parameters eliminates the need for manual tuning and demonstrates the robustness of the architecture even with different users. The experimental results confirmed the reliability of the method, as evidenced by the consistent performance of the classifier and the successful execution of the task across subjects, although automatic calibration techniques could be explored to improve the adaptability to individual user characteristics or specific applications.

The classifier performance are evaluated using confusion matrix for multiclass problems and class-wise metrics *A*ccuracy, *P*recision, *R*ecall, and *F1*-score calculated according to the indices in (5.16)- (5.19). In the on-line use of the classifier, the statistical mode of the predicted classes calculated over the last 1.5s is sent to the finite state machine to increase the robustness in terms of misclassifications.

Experimental results

To validate the overall architecture by triggering all the envisioned state transitions, the following representative sequence of operations characterized by several human-robot and human-environment interactions is requested to the user: *i*) drive the end-effector closer to a surface using

no-contact mode; *ii*) switch to *contact* mode to enter in contact with the surface; *iii*) follow a predefined circular path drawn on the surface; *iv*) lift the end-effector to switch to *no-contact* mode; *v*) bring the end-effector in contact with the center of the circumference; *vi*) change the robot operational mode to *hold* and push the end-effector to prove that it returns to the center; *vii*) lift the end-effector to return in *no-contact* mode again.

To test the generalization and the reproducibility of the approach, the designed experiment has been performed by a group of four participants (one female, aged 28, and three males, aged between 27 and 35), each of whom repeated it for 10 times.
In accordance with the Declaration of Helsinki, all the participants provided signed informed consent, and the research protocol was approved by the Research Ethics Committee of the University of Cassino and Southern Lazio. The EMG sensors were placed on the same muscles of the dominant arm for all the subjects; however, considering the differences in the physical structure of the subjects, there were small displacements in the positioning of the sensors from one person to another to ensure that the sensors were placed as much as possible in the center of the considered muscles.

Each of the four subjects performed the acquisition procedure described in the Subsection 5.2.5 to obtain the training and test sets for classifier training. Subsequently, each subject obtained their trained SVM model, and the performance of the classifier was evaluated for each participant's model using a confusion matrix and the global indices computed with it.

Control parameters		
Parameter	**Value**	**Description**
$\boldsymbol{K}_{\text{ik}}$	diag$\{20\ \boldsymbol{I}_3, 15\ \boldsymbol{I}_3\}$	Inverse kinematics gain
$\boldsymbol{K}_{\text{jc}}$	diag$\{3\ \boldsymbol{I}_4, 2\ \boldsymbol{I}_3\}$	Joints controller gain
$k_{d,\text{low}}$	80	Low-damping gain
$k_{d,\text{high}}$	1000	High-damping gain
k_k	100	Stiffness gain in *hold* mode
Classifier parameters		
Parameter	**Value**	**Description**
C	1	Regularization parameter
γ	*'scale'*	Parameter of a Gaussian Kernel
Kernel	'RBF'	Kernel used in the SVM Classifier

Table 5.3: Experimental parameters used in the proposed architecture.

Table 5.3 shows the parameters for the controllers and the classifier used in the experiments, where all the admittance parameters were empirically determined in such a way that both the resonance phenomena can be dealt with and the co-manipulation can be made easier.

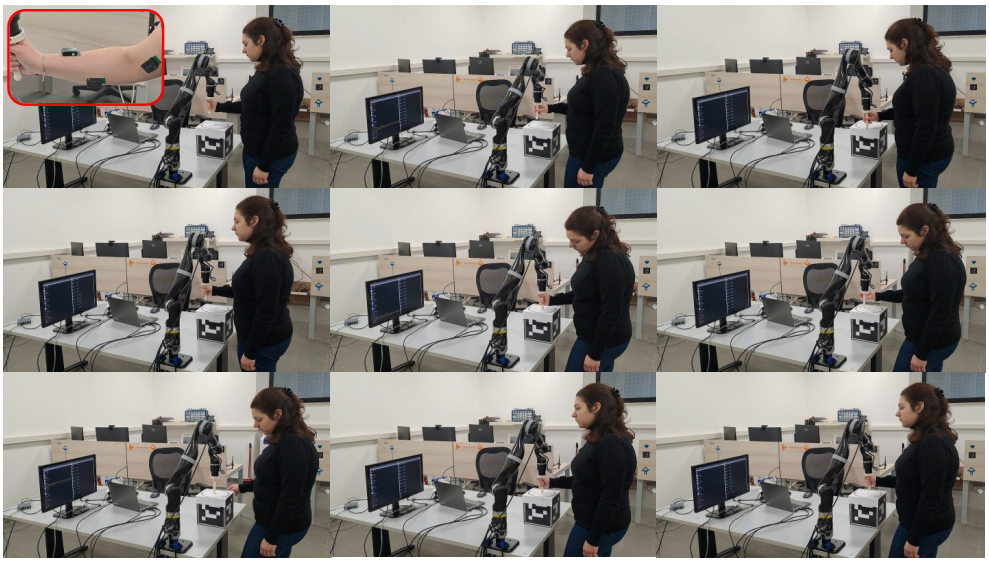


Figure 5.10: The series of images displayed above illustrates one of the experiments conducted by Subject 1.

In particular, the confusion matrices of the SVM classifier trained for each subject are shown in Figure 5.11, while the global classifier indexes calculated using the formulas presented in Equation (5.16) through Equation (5.19) are shown in Table 5.4.

A video showing some of the experimental runs of all participants is also provided at the link: https://youtu.be/4YZ64SgFEro, and a selection of snapshots from the video of the experiment performed by Subject 1 is presented in Figure 5.10.

In the following, the results of the classifier performances and experimental execution are discussed.

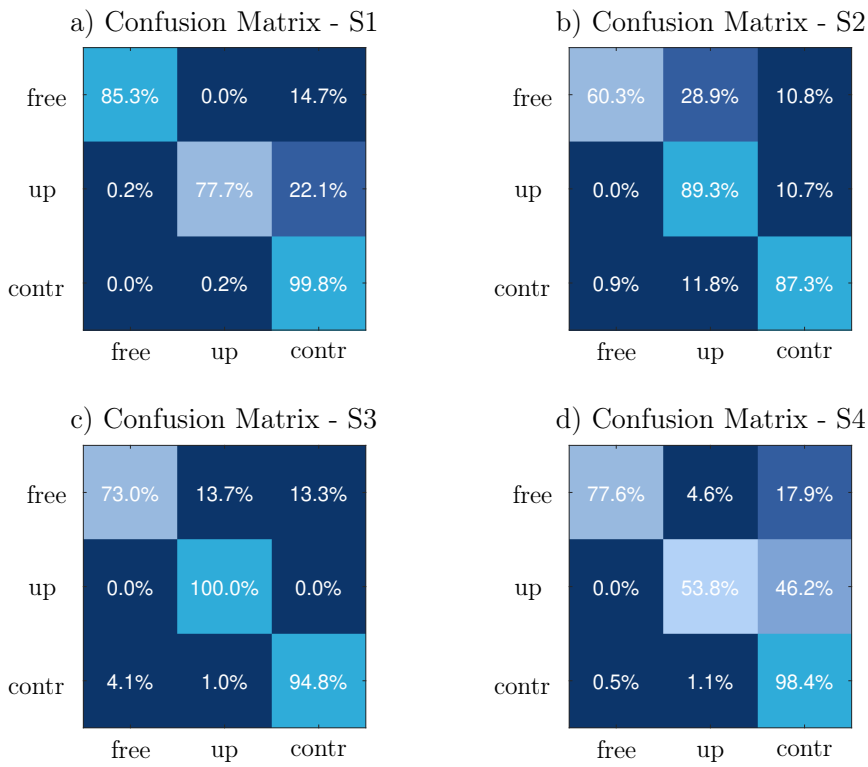


Figure 5.11: The confusion matrices obtained following the training of the classifier for all participants: a) the confusion matrix obtained for the Subject 1; b) the confusion matrix obtained for the Subject 2; c) the confusion matrix obtained for the Subject 3; d) the confusion matrix obtained for the Subject 4.

Accuracy scores range from 81.69% to 91.40% across the four subjects, indicating a strong overall classification performance, meaning that the majority of examples are classified correctly.
Precision indexes show high values, from 82.16% to 95.18%, indicating few false positives, while the recall, generally good but lower than precision (from 76.61% to 89.28%), suggests that the classifier is too conservative in predicting positive labels.
Finally, F1 scores range from 78.11% to 90.59%, confirming a solid precision/recall balance with values close to accuracy.
The numerical results demonstrate a reliable and accurate classification

capability, although further analysis would be useful to improve recall by reducing false negatives.

Subject	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
S1	91.40	95.18	87.59	90.59
S2	81.69	82.16	78.97	78.11
S3	90.84	89.22	89.28	88.81
S4	84.13	88.41	76.61	80.45

Table 5.4: Table containing performance evaluation metrics

As illustrated in Table 5.4, the performance indexes are, on average, higher for Subject 1 than for the other subjects, which can be attributed to the fact that she is more familiar with the procedure of acquiring datasets for training the classifier during interaction with the robot. This is due to the fact that she has carried out previous tests with the aforementioned setup, whereas the other subjects has merely followed the acquisition instructions without performing any preliminary tests. Despite this, the performance indexes are very high for all subjects and very close to the values obtained by Subject 1, who is to be considered the expert subject, and this attests that the adopted system is highly user-friendly, as also indicated in the NASA-TLX questionnaires completed by users. Indeed, the mean workload score across all dimensions for the task in question is approximately 34%, indicating a relatively low overall workload. This suggests that participants generally found the task to be manageable and not onerous.

Figure 5.12 shows the path followed by the end-effector during one of the experiments performed by Subject 1, where the start and end points of the path are reported.

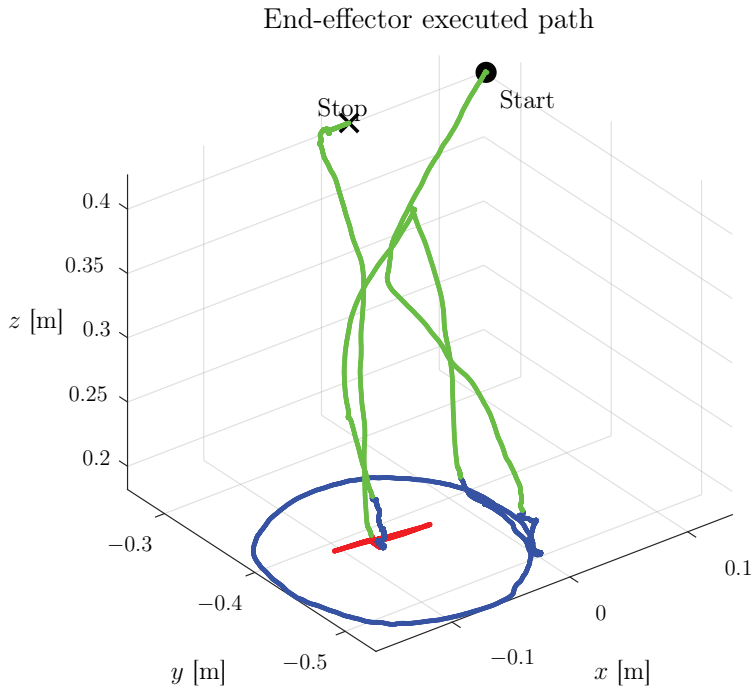


Figure 5.12: The path followed by the manipulator's end-effector in one of the experiments performed by the Subject 1. The path is highlighted with three colors based on the three operational modes: *no-contact* mode is marked in green, *contact* mode is highlighted in blue and *hold* mode is colored in red.

Three different colors are used to highlight the path followed by the manipulator guided by the human operator, depending on the corresponding active operational mode: the paths executed in *no-contact* mode are highlighted in green; those corresponding to *contact* mode are highlighted in blue; finally, the path followed by the end-effector in *hold* mode is colored in red. It can be seen that the subject managed to complete the experiment without any difficulties by quickly changing the operational mode. Similar results were obtained by the other subjects involved, as shown in the video provided as supplementary material for the present work.

Instead, Figure 5.13 shows the time evolution of the classifier output, the force measured by the F/T sensor, and the linear velocity of the

manipulator's end-effector.

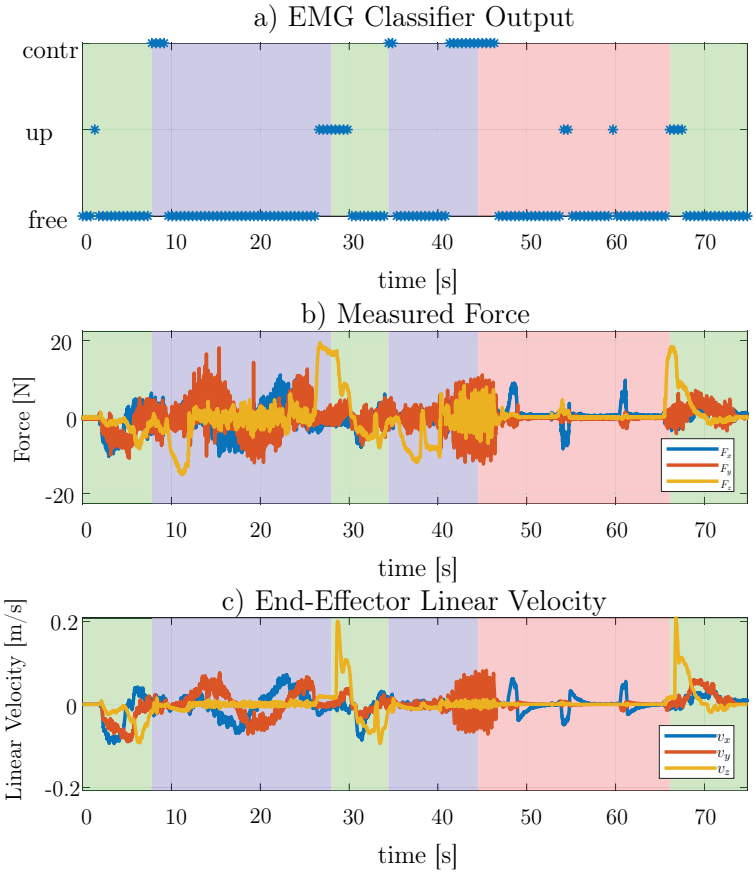


Figure 5.13: Results of one of the experiments of the Subject 1: a) EMG Classifier Output, i.e. the statistical mode of the last 1.5s of predicted classes; b) Force measured by the F/T sensor; c) End-Effector Linear Velocity. The different colored areas on all graphs correspond to the three different control modes: green, blue and red represent the *no-contact*, *contact*, and *hold* modes, respectively.

In particular, Figure 5.136*a* shows the time evolution of the classifier output, which provides information about how it was able to distinguish among the three classes *free*, *contraction* and *up* during the experiment. Similarly to what was highlighted in the performed path, the colors green,

blue, and red indicate the *no-contact*, *contact*, and *hold* modes, respectively. The output of the classifier determines the transition among the control modes, as demonstrated by the different colored areas; in fact, after the first 7s in which the human is in *free* class, he/she generates *contraction* and FSM switches to *no-contact* mode, while when he/she generates *up* for at least 2s as happens between 28 and 30s or after 66s the mode returns to *no-contact*; even when the operator generates *contraction* for at least 5s as happens after 40s FSM switches to *hold* mode.

The time evolution of the force (Figure 5.136*b*) and the end-effector linear velocity (Figure 5.136*c*) are highlighted with the same colors according to the operational modes. To trigger the transition from *no-contact* to *contact* mode, i.e. from a low damping value to a high damping value, the operator needs to use *contraction* class by contracting the hand around the end-effector. This causes oscillations in the force and velocity plots caused by the contraction itself, which also triggers the resonance phenomena, even if for a short period of time. This behavior is more evident when the operator generates *contraction* for at least 5s to switch to *hold* mode; in fact, even if he/she is already in a control mode characterized by high damping, the high K_d is set only along the z component to avoid resonance phenomena in case of prolonged contact with the external environment, while low values are left on the x and y components to allow hand guidance and interaction with the operator in general.

Chapter 6

Conclusions and Future Works

This thesis presents a comprehensive investigation of shared control techniques for assistive robotics and human-robot interaction applications, with a particular emphasis on the utilization of bioelectric signals, including electroencephalography and electromyography, to facilitate enhanced human-robot collaboration. The integration of these signals into shared control structures enables users to communicate their intentions to robotic systems in a more intuitive and natural manner. This is of particular significance in applications involving individuals with motor disabilities or in scenarios where precise and adaptive control is required.

A significant contribution of this research is the development of shared control architectures that integrate human input with autonomous robotic behaviors. This hybrid control approach guarantees that both the human operator and the robotic system play a pivotal role in task execution. In particular, the use of brain-computer interfaces and EMG-based controls has demonstrated that bioelectric signals can be effectively exploited to handle complex robotic tasks, including teleoperation, co-manipulation, and mobility assistance.

In the field of assistive robotics, the capacity for individuals with severe motor disabilities to operate devices such as robotic arms or wheelchairs through non-invasive bioelectric interfaces represents a substantial advancement. The experimental results demonstrate that shared control structures can compensate for the user's limitations, thereby enabling the robot to assist the user in precision tasks, such as obstacle avoidance

or fine motor control, as well as in manipulating objects. This is accomplished by maintaining the user's involvement in the control process, thereby preserving their autonomy.

In the context of human-robot collaboration, this thesis has demonstrated how shared control facilitates seamless interaction between human operators and robots, particularly in industrial and medical settings. By combining the strengths of human cognitive abilities, such as decision-making and task adaptation, with the precision and strength of robots, shared control facilitates the execution of tasks that neither humans nor robots could accomplish efficiently on their own. The integration of null-space-based control techniques with bioelectric signals has demonstrated efficacy in enabling robots to assist in co-management tasks, thereby ensuring the safe and efficient completion of tasks without the excessive physical strain on the user.

One of the principal findings of this thesis is the impact of dynamic control systems, such as variable admittance control, which allow the robot to adapt its response in real time based on input from the human operator and changes in the external environment. This adaptability is crucial in human-robot interaction scenarios where environmental conditions can change in an unpredictable manner, such as during rehabilitation exercises or in dynamic industrial environments. The results demonstrate that enabling robots to adapt their behavior based on muscle activity facilitates a more seamless and effective interaction, reducing the probability of errors or accidents and enhancing user satisfaction.

Furthermore, the thesis underscored the significance of safety and reliability in shared control systems. By maintaining human control and ensuring that the robot can accurately discern the user's intentions, the shared control architectures developed in this work can significantly mitigate the risks associated with robotic assistance, particularly in applications such as healthcare and rehabilitation. Furthermore, the utilisation of bioelectric signals in shared control presents novel opportunities for enhancing the accessibility of robotic technologies for individuals with disabilities, thereby facilitating greater independence and quality of life.

In conclusion, this thesis has demonstrated the efficacy of shared control techniques that integrate bioelectric signals into human-robot interaction structures. By leveraging the capabilities of both humans and robots,

these systems facilitate the execution of tasks in a more efficient, safer, and smoother manner, both in assistive applications and in collaborative industrial settings. The findings of this study provide a foundation for future advancements in both domains, with the possibility of extensive integration in healthcare, industrial automation, and beyond.

Although the research presented in this thesis has enhanced the comprehension and implementation of shared control systems in assistive robotics and human-robot interaction, a number of promising avenues for future investigation emerge from the findings. These areas of future exploration have the potential to enhance the performance, adaptability, and scope of shared control systems, thereby increasing their effectiveness in a variety of real-world settings. One of the principal areas for future research is the enhancement of the acquisition and processing of bioelectric signals, including electroencephalography and electromyography. These signals are frequently corrupted by noise, interference, and variability among users, which can diminish the accuracy and responsiveness of shared control systems. Future work should concentrate on the development of more robust signal filtering and denoising algorithms to guarantee the reliability of control signals, even in less than optimal conditions. Enhanced real-time signal processing would facilitate more prompt responses from the robot, thereby enhancing the user experience by minimizing latency between intention and action. The application of sophisticated machine learning methodologies, including deep learning and neural networks, has great potential in this area. These techniques could be employed to automatically classify and interpret bioelectric signals with greater precision, even among users with disparate signal profiles. Moreover, transfer learning could be utilized to reduce the necessity for extensive training sessions, thereby enabling shared control systems to rapidly adapt to new users or disparate tasks without requiring exhaustive recalibration.

Furthermore, shared control systems frequently rely on a single type of bioelectric signal, such as electroencephalography (EEG) or electromyography, to interpret user intention. Nevertheless, there is a mounting interest in multimodal human-machine interfaces that integrate diverse forms of input, including voice commands, gaze tracking, haptic feedback, and even gesture recognition. Future research could investigate the potential of combining these disparate modalities into a unified control

system. For instance, a robotic system could utilize EEG signals for high-level decision-making, while EMG data could be employed to regulate finer movements, and voice commands could be used to initiate specific tasks. This multimodal approach would not only enhance the flexibility and usability of shared control systems but also improve access for users with varying degrees of physical ability. Furthermore, this approach could enhance robots' ability to discern complex user intentions, a capability that is particularly crucial in contexts necessitating rapid adaptation to dynamic scenarios. Examples of such scenarios include emergency medical response and high-precision manufacturing tasks. While this thesis has concentrated on assistive robotics for individuals with disabilities and human-robot interaction in collaborative tasks, the shared control systems developed here have the potential to be applied in a wide range of fields. One promising avenue of application is the integration of shared control systems in advanced healthcare scenarios, such as robot-assisted surgeries and physical therapy. For instance, robotic surgery systems could benefit from shared control, allowing surgeons to conduct delicate operations with precise accuracy while the robot autonomously manages tasks such as stability or navigation within the surgical environment.

In the context of industry, the implementation of shared control systems could prove advantageous in collaborative robots (cobots) operating in conjunction with human workers in manufacturing environments. As Industry 4.0 continues to develop, the integration of shared control systems has the potential to significantly enhance worker productivity and safety, particularly in tasks that necessitate a combination of human cognitive flexibility and robot mechanical precision. In practical applications, human-robot interactions frequently extend beyond the scope of isolated collaborations. In numerous industrial and healthcare settings, multiple individuals and robots operate concurrently in intricate environments. Further research is required to investigate the potential for extending shared control techniques to multi-agent systems, in which multiple robots and human operators collaborate to complete tasks. This would necessitate the creation of novel algorithms capable of managing task allocation, communication, and coordination among numerous agents in real time. To illustrate, in a manufacturing environment, a team of robots could work alongside human operators to assemble a product, with each agent contributing its unique capabilities to the task. A shared control system

could distribute control dynamically based on the requirements of the task. In this way, robots could be assigned routine or physically demanding tasks, while humans focus on decision-making and problem-solving. Furthermore, these systems must consider team dynamics to ensure that all agents, whether human or robotic, work together in a conflict-free and efficient manner.

By addressing these future research directions, shared control systems can be further refined and expanded, thereby revolutionizing both assistive technologies and collaborative robotics. This will be achieved by providing more intuitive, adaptive, and safe interactions between humans and robots.

Bibliography

[1] R. Francos and A. Bruckstein, "On the role and opportunities in teamwork design for advanced multi-robot search systems," *Frontiers in robotics and AI*, vol. 10, p. 1089062, 04 2023.

[2] A. Dzedzickis, J. Subaciute-Zemaitiene, E. Šutinys, U. Prentice, and V. Bučinskas, "Advanced applications of industrial robotics: New trends and possibilities," *Applied Sciences*, vol. 12, p. 135, 12 2021.

[3] M. W. Gifari, H. Naghibi, S. Stramigioli, and M. Abayazid, "A review on recent advances in soft surgical robots for endoscopic applications," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 15, no. 5, p. e2010, 2019.

[4] Z. Yuan, "Current status and prospects of actuator in robotics," *Applied and Computational Engineering*, vol. 11, pp. 181–191, 09 2023.

[5] F. Siepel, B. Maris, M. Welleweerd, V. Groenhuis, P. Fiorini, and S. Stramigioli, "Needle and biopsy robots: a review," *Current Robotics Reports*, 2021, springer deal.

[6] Y.-L. Lee, K.-J. Chiang, C.-K. Lin, T.-K. Chao, M.-H. Yu, Y.-L. Liu, and Y.-C. Wang, "Robotic hysterectomy with ureter identification and uterine artery ligation for benign gynecological conditions: An early-year single-center experience," *Diagnostics*, vol. 13, p. 1809, 05 2023.

[7] H.-Y. Cheng, Y.-J. Chen, P.-H. Wang, H.-W. Tsai, Y.-H. Chang, N.-F. Twu, C.-M. Juang, H. Wu, M.-S. Yen, and K. Chao, "Robotic-assisted laparoscopic complex myomectomy: A single medical center's experience," *Taiwanese journal of obstetrics & gynecology*, vol. 54, pp. 39–42, 02 2015.

[8] S. Goel, "Teaching robots to interact with humans in a smart environment," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 6434–6435.

[9] Y. Zhang and M. Lu, "A review of recent advancements in soft and flexible robots for medical applications," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 16, no. 3, p. e2096, 2020.

[10] G. Ambaye, E. Boldsaikhan, and K. Krishnan, "Soft robot design, manufacturing, and operation challenges: A review," *Journal of Manufacturing and Materials Processing*, vol. 8, p. 79, 04 2024.

[11] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon, "A survey of robots in healthcare," *Technologies*, vol. 9, no. 1, 2021.

[12] T. Tanioka, R. Locsin, F. Betriana, Y. Kai, K. Osaka, M. E. Baua, and S. Schoenhofer, "Intentional observational clinical research design: Innovative design for complex clinical research using advanced technology," *International Journal of Environmental Research and Public Health*, vol. 18, p. 11184, 10 2021.

[13] X. Wei and W. Rey, "Advancements in substation inspection robots: A review of research and development," *E3S Web of Conferences*, vol. 528, 05 2024.

[14] V. De Silva, J. Roche, X. Shi, and A. Kondoz, "Iot driven ambient intelligence architecture for indoor intelligent mobility," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 2018, pp. 451–456.

[15] M. J. Johnson, K. Bui, and N. Rahimi, *Medical and Assistive Robotics in Global Health*. Springer International Publishing, 2021.

[16] T. A. Lawrie, H. Liu, D. Lu, T. Dowswell, H. Song, L. Wang, and G. Shi, "Robot-assisted surgery in gynaecology," *Cochrane Database of Systematic Reviews*, no. 4, 2019.

[17] J. Troccaz, G. Dagnino, and G.-Z. Yang, "Frontiers of medical robotics: From concept to systems to clinical translation," *Annual Review of Biomedical Engineering*, vol. 21, 06 2019.

[18] I. Alkatout, L. Mettler, N. Maass, and J. Ackermann, "Robotic surgery in gynecology," *Journal of the Turkish German Gynecological Association*, vol. 17, pp. 224–232, 12 2016.

[19] R. Cooper, G. Smolinski, J. Candiotti, S. Satpute, G. Grindle, T. Sparling, M. Nordstrom, X. Yuan, A. Symsack, C. Lee, N. Vitiello, S. Knezevic, T. Sugar, U. Schneider, V. Kopp, M. Holl, I. Gaunaurd, R. Gailey, P. Bonato, and P. Pasquina, "Current state, needs, and opportunities for wearable robots in military medical rehabilitation and force protection," *Actuators*, vol. 13, p. 236, 06 2024.

[20] A. Goia, V. Gilard, R. Lefaucheur, M.-L. Welter, D. Maltête, and S. Derrey, "Accuracy of the robot-assisted procedure in deep brain stimulation," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 15, 08 2019.

[21] W. H. Chang and Y.-H. Kim, "Robot-assisted therapy in stroke rehabilitation," *Journal of stroke*, vol. 15, no. 3, p. 174, 2013.

[22] N. Norouzi-Gheidari, P. Archambault, and J. Fung, "Effects of robot-assisted therapy on stroke rehabilitation in upper limbs: Systematic review and meta-analysis of the literature," *Journal of rehabilitation research and development*, vol. 49, pp. 479–96, 06 2012.

[23] J. Mas-Santillán, F. J. Acevedo-Rodríguez, and R. J. López-Sastre, "Embedding ros and ai-based perception capabilities in a novel low-cost assistive robotic platform," *Engineering Proceedings*, vol. 7, no. 1, p. 12, 2021.

[24] E. Martinez-Martin and A. P. del Pobil, "Vision for robust robot manipulation," *Sensors*, vol. 19, no. 7, 2019.

[25] J. Hirth, K. Berns, and K. Mianowski, "Designing arms and hands for the humanoid robot roman," *Advanced Materials Research*, vol. 463-464, 02 2012.

[26] M. Pan, C. Yuan, X. Liang, T. Dong, T. Liu, J. Zhang, J. Zou, H. Yang, and C. Bowen, "Soft actuators and robotic devices for rehabilitation and assistance," *Advanced Intelligent Systems*, vol. 4, p. 2100140, 11 2021.

[27] I. N. Weerarathna, D. Raymond, and A. Luharia, "Human-robot collaboration for healthcare: A narrative review," *Cureus*, vol. 15, no. 11, 2023.

[28] G. Averta, D. Caporale, C. D. Santina, A. Bicchi, and M. Bianchi, "A technical framework for human-like motion generation with autonomous anthropomorphic redundant manipulators," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3853–3859.

[29] A. Hameed, A. Ordys, J. Możaryn, and A. Sibilska-Mroziewicz, "Control system design and methods for collaborative robots," *Applied Sciences*, vol. 13, no. 1, p. 675, 2023.

[30] C. Schou and O. Madsen, "A plug and produce framework for industrial collaborative robots," *International Journal of Advanced Robotic Systems*, vol. 14, p. 172988141771747, 07 2017.

[31] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati, "Human-robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, p. 100, 2019.

[32] R. Bloss, "Collaborative robots are rapidly providing major improvements in productivity, safety, programing ease, portability and cost while addressing many new applications," *Industrial Robot: An International Journal*, vol. 43, pp. 463–468, 08 2016.

[33] D. Rodríguez-Guerra, G. Sorrosal, I. Cabanes, and C. Calleja, "Human-robot interaction review: Challenges and solutions for modern industrial environments," *IEEE Access*, vol. 9, pp. 108 557–108 578, 2021.

[34] S. Sladić, R. Lisjak, L. Runko Luttenberger, and M. Musa, "Trends and progress in collaborative robot applications," *Politehnika*, vol. 5, pp. 32–37, 06 2021.

[35] Y. Liu and H. Jebelli, "Intention-aware robot motion planning for safe worker-robot collaboration," *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 15, pp. 2242–2269, 2024.

[36] S. You, J.-H. Kim, S. Lee, V. Kamat, and L. P. Robert, "Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments," *Automation in Construction*, vol. 96, pp. 161–170, 2018.

[37] P. Maurice, V. Padois, Y. Measson, and P. Bidaud, "Digital human modeling for collaborative robotics," in *DHM and Posturography*. Elsevier, 2019, pp. 771–779.

[38] A. Castro, F. Silva, and V. Santos, "Trends of human-robot collaboration in industry contexts: Handover, learning, and metrics," *Sensors*, vol. 21, no. 12, p. 4113, 2021.

[39] R. Caccavale and A. Finzi, "A robotic cognitive control framework for collaborative task execution and learning," *Topics in Cognitive Science*, vol. 14, no. 2, pp. 327–343, 2022.

[40] A. M. Titu, V. Gusan, M. Dragomir, A. Pop, and S. Ţîţu, "Cost calculation and deployment strategies for collaborative robots in production lines: An innovative and sustainable perspective in knowledge-based organizations," *Sustainability*, vol. 16, p. 5292, 06 2024.

[41] A. Hanna, K. Bengtsson, P.-L. Götvall, and M. Ekström, "Towards safe human robot collaboration - risk assessment of intelligent automation," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 424–431.

[42] G. Pang, G. Yang, and Z. Pang, "Review of robot skin: A potential enabler for safe collaboration, immersive teleoperation, and affective interaction of future collaborative robots," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 3, pp. 681–700, 2021.

[43] P. Gao and H. Zhang, "Bayesian deep graph matching for correspondence identification in collaborative perception," in *Robotics Science and Systems (RSS)*, 07 2021.

[44] S. B. i. Badia, P. A. Silva, D. Branco, A. Pinto, C. Carvalho, P. Menezes, J. Almeida, and A. Pilacinski, "Virtual reality for safe testing and development in collaborative robotics: challenges and perspectives," *Electronics*, vol. 11, no. 11, p. 1726, 2022.

[45] M. Á. Mariscal Saldaña, J. González Pérez, A. Khalid, J. M. Gutiérrez Llorente, S. García Herrero *et al.*, "Risks management and cobots. identifying critical variables," in *Proceedings of the 29th European Safety and Reliability Conference (ESREL)*, 2019.

[46] S. Heydaryan, J. Bedolla, and G. Belingardi, "Safety design and development of a human-robot collaboration assembly process in the automotive industry," *Applied Sciences*, vol. 8, p. 344, 02 2018.

[47] A. Sathyan and O. Ma, "Collaborative control of multiple robots using genetic fuzzy systems," *Robotica*, vol. 37, no. 11, pp. 1922–1936, 2019.

[48] J. Kildal, J. Molina, and U. Andrés, "User-centred human-robot collaborative handling of small parts in a mim process," in *Advances in Manufacturing Technology XXXV*. IOS Press, 2022, pp. 113–118.

[49] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human-robot collaboration," *Autonomous robots*, vol. 42, pp. 957–975, 2018.

[50] S. Kim, "Working with robots: human resource development considerations in human-robot interaction," *Human Resource Development Review*, vol. 21, no. 1, pp. 48–74, 2022.

[51] M. Faccio, I. Granata, A. Menini, M. Milanese, C. Rossato, M. Bottin, R. Minto, P. Pluchino, L. Gamberini, G. Boschetti *et al.*, "Human factors in cobot era: a review of modern production systems features," *Journal of Intelligent Manufacturing*, vol. 34, no. 1, pp. 85–106, 2023.

[52] F. Lima, C. Carvalho, M. Acardi, E. Santos, G. Miranda, R. Maia, and A. Massote, "Digital manufacturing tools in the simulation of collaborative robots: Towards industry 4.0," *Brazilian Journal of Operations & Production Management*, vol. 16, pp. 261–280, 05 2019.

[53] M. R. Mia and J. Shuford, "Exploring the synergy of artificial intelligence and robotics in industry 4.0 applications," *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, vol. 1, no. 1, 2024.

[54] 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.

[55] S. Moon, S. Rhim, Y.-J. Cho, K. Park, and G. Virk, "Summary of recent standardization activities in the field of robotics," *Robotica*, vol. 31, 03 2013.

[56] F. Ahmed, M. Waqas, B. Jawed, A. M. Soomro, S. Kumar, A. Hina, U. Khan, K. H. Kim, and K. H. Choi, "Decade of bio-inspired soft robots: A review," *Smart Materials and Structures*, vol. 31, no. 7, p. 073002, 2022.

[57] M. Jung, D. Difranzo, B. Stoll, S. Shen, A. Lawrence, and H. Claure, "Robot assisted tower construction - a method to study the impact of a robot's allocation behavior," *ACM Transactions on Human-Robot Interaction*, 05 2020.

[58] M. Zorman, B. Žlahtič, S. Stradovnik, and A. Hace, "Transferring artificial intelligence practices between collaborative robotics and autonomous driving," *Kybernetes*, vol. 52, no. 9, pp. 2924–2942, 2022.

[59] S. Islam and W. Lughmani, "A connective framework for social collaborative robotic system," *Machines*, vol. 10, p. 1086, 11 2022.

[60] Y. Rizal, *Computer Simulation of Human-Robot Collaboration in the Context of Industry Revolution 4.0.* IntechOpen, 2019.

[61] F. Barravecchiaa, M. Bartolomeib, L. Mastrogiacomoc, and F. Franceschinid, "A bio-inspired reinterpretation of symbiotic

human-robot collaboration in assembly processes," in *Italian Manufacturing Association Conference: XVI AITeM*, vol. 35. Materials Research Forum LLC, 2023, p. 2.

[62] L. Lu, Z. Xie, H. Wang, L. Li, E. P. Fitts, and X. Xu, "Measurements of mental stress and safety awareness during human robot collaboration-review," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 66, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2022, pp. 2273–2277.

[63] J. Arents, V. Abolins, J. Judvaitis, O. Vismanis, A. Oraby, and K. Ozols, "Human-robot collaboration trends and safety aspects: A systematic review," *Journal of Sensor and Actuator Networks*, vol. 10, no. 3, p. 48, 2021.

[64] K. P. Hawkins, N. Vo, S. Bansal, and A. F. Bobick, "Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 499–506.

[65] T. Degeorges and G. Sziebig, "Communication between human and robots within a collaborative workspace," in *Advanced Manufacturing and Automation X*, 2021, pp. 652–657.

[66] G. Pang, J. Deng, F. Wang, J. Zhang, Z. Pang, and G. Yang, "Development of flexible robot skin for safe and natural human-robot collaboration," *Micromachines*, vol. 9, no. 11, p. 576, 2018.

[67] J. Slovák, M. Melicher, M. Šimovec, and J. Vachálek, "Vision and rtls safety implementation in an experimental human—robot collaboration scenario," *Sensors*, vol. 21, no. 7, p. 2419, 2021.

[68] I. Salehi, G. Rotithor, and A. Dani, "Safe adaptive trajectory tracking control of robot for human-robot interaction using barrier function transformation," *Collaborative and Humanoid Robots*, p. 129, 2021.

[69] C. Zeng, C. Yang, and Z. Chen, "Bio-inspired robotic impedance adaptation for human-robot collaborative tasks," *Science China Information Sciences*, vol. 63, pp. 1–10, 2020.

[70] T. Williams, D. Szafr, T. Chakraborti, and H. B. Amor, "Report on the first international workshop on virtual, augmented, and mixed reality for human-robot interaction," *AI Magazine*, vol. 39, no. 4, p. 64–66, 2018.

[71] S. Schleidgen and O. Friedrich, "Joint interaction and mutual understanding in social robotics," *Science and Engineering Ethics*, vol. 28, no. 6, p. 48, 2022.

[72] M. Kyrarini, M. A. Haseeb, D. Ristić-Durrant, and A. Gräser, "Robot learning of industrial assembly task via human demonstrations," *Autonomous Robots*, vol. 43, pp. 239–257, 2018.

[73] W. Wang, Y. Chen, R. Li, and Y. Jia, "Learning and comfort in human-robot interaction: A review," *Applied Sciences*, vol. 9, no. 23, p. 5152, 2019.

[74] H. Abidi and M. Cianchetti, "On intrinsic safety of soft robots," *Frontiers in Robotics and AI*, vol. 4, p. 5, 2017.

[75] D. Zurlo, T. Heitmann, M. Morlock, and A. De Luca, "Collision detection and contact point estimation using virtual joint torque sensing applied to a cobot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7533–7539.

[76] S. A. Bowyer and F. Rodriguez y Baena, "Dissipative control for physical human–robot interaction," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1281–1293, 2015.

[77] A. Q. Keemink, H. Van der Kooij, and A. H. Stienen, "Admittance control for physical human-robot interaction," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1421–1444, 2018.

[78] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Safe physical human-robot interaction: measurements, analysis and new insights," in *Robotics Research: The 13th International Symposium ISRR*, 2010, pp. 395–407.

[79] R. Alami, A. Albu-Schaeffer, A. Bicchi, R. Bischoff, R. Chatila, A. De Luca, A. De Santis, G. Giralt, J. Guiochet, G. Hirzinger, F. Ingrand, V. Lippiello, R. Mattone, D. Powell, S. Sen, B. Siciliano, G. Tonietti, and L. Villani, "Safe and dependable physical

human-robot interaction in anthropic domains: State of the art and challenges," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1–16.

[80] C. Wang and J. Zhao, "Based on human-like variable admittance control for human-robot collaborative motion," *Robotica*, vol. 41, no. 7, pp. 2155–2176, 2023.

[81] D. A. Abbink, T. Carlson, M. Mulder, J. C. F. de Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control systems—finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018.

[82] M. Zolotas and Y. Demiris, "Transparent intent for explainable shared control in assistive robotics," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 5184–5185.

[83] G. Quere, A. Hagengruber, M. Iskandar, S. Bustamante, D. Leidner, F. Stulp, and J. Vogel, "Shared control templates for assistive robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1956–1962.

[84] D. Abbink, M. Mulder, and E. Boer, "Haptic shared control: Smoothly shifting control authority?" *Cognition, Technology & Work*, vol. 14, pp. 19–28, 03 2012.

[85] K. Qian, D. Niu, F. Fang, and X. Ma, "Learning spatial constraints using gaussian process for shared control of semi-autonomous mobile robots." in *ICINCO (2)*, 2017, pp. 201–208.

[86] L. Olivi, R. Souza, E. Rohmer, and E. Cardozo, "Shared control for assistive mobile robots based on vector fields," in *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2013, pp. 96–101.

[87] R. Rahal, F. Abi-Farraj, P. R. Giordano, and C. Pacchierotti, "Haptic shared-control methods for robotic cutting under nonholonomic constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 8151–8157.

[88] M. Zolotas and Y. Demiris, "Towards explainable shared control using augmented reality," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3020–3026.

[89] H. J. Jeon, D. P. Losey, and D. Sadigh, "Shared autonomy with learned latent actions," *arXiv preprint arXiv:2005.03210*, 2020.

[90] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. Mussa-Ivaldi, and B. Argall, "Assistive robotic manipulation through shared autonomy and a body-machine interface," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2015, pp. 526–531.

[91] S. Jain and B. Argall, "Probabilistic human intent recognition for shared autonomy in assistive robotics," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.

[92] H. Zeng, Y. Shen, X. Hu, A. Song, B. Xu, H. Li, Y. Wang, and P. Wen, "Semi-autonomous robotic arm reaching with hybrid gaze–brain machine interface," *Frontiers in neurorobotics*, vol. 13, p. 111, 2020.

[93] A. C. Lopes, G. Pires, L. Vaz, and U. Nunes, "Wheelchair navigation assisted by human-machine shared-control and a p300-based brain computer interface," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2438–2444.

[94] A. Frolov, O. Mokienko, R. Lyukmanov, E. Biryukova, S. Kotov, L. Turbina, G. Nadareyshvily, and Y. Bushkova, "Post-stroke rehabilitation training with a motor-imagery-based brain-computer interface (bci)-controlled hand exoskeleton: A randomized controlled multicenter trial," *Frontiers in Neuroscience*, vol. 11, p. 400, 2017.

[95] A. F. Salazar-Gomez, J. DelPreto, S. Gil, F. H. Guenther, and D. Rus, "Correcting robot mistakes in real time using eeg signals," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6570–6577.

[96] Y. Mishchenko, M. Kaya, E. Ozbay, and H. Yanar, "Developing

a three- to six-state eeg-based brain–computer interface for a virtual robotic manipulator control," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 4, pp. 977–987, 2019.

[97] Y. Blokland, L. Spyrou, D. Thijssen, T. Eijsvogels, W. Colier, M. Floor-Westerdijk, R. Vlek, J. Bruhn, and J. Farquhar, "Combined eeg-fnirs decoding of motor attempt and imagery for brain switch control: An offline study in patients with tetraplegia," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, pp. 222–229, 2014.

[98] Y. Kwak, W. Song, and S.-E. Kim, "Fganet: fnirs-guided attention network for hybrid eeg-fnirs brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 329–339, 2022.

[99] J. Chen, Y. Xia, X. Zhou, E. V. Rosas, A. Thomas, R. Loureiro, R. J. Cooper, T. Carlson, and H. Zhao, "fnirs-eeg bcis for motor rehabilitation: A review," *Bioengineering*, vol. 10, 2023.

[100] G. Moreno, N. De La Cruz, J. Ortiz, and V. Andaluz, *Human-Robot Collaborative Control for Handling and Transfer Objects.* Springer International Publishing, 04 2021, pp. 96–110.

[101] E. Kyrkjebø, "Inertial human motion estimation for physical human-robot interaction using an interaction velocity update to reduce drift," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 163–164.

[102] A. Jonnavittula and D. P. Losey, "Learning to share autonomy across repeated interaction," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1851–1858.

[103] H. Kong, C. Yang, G. Li, and S.-L. Dai, "A semg-based shared control system with no-target obstacle avoidance for omnidirectional mobile robots," *IEEE Access*, vol. 8, pp. 26 030–26 040, 2020.

[104] G. Antonelli, P. Beomonte Zobel, F. Durante, and M. Zeer, "Modeling-based emg signal (mbes) classifier for robotic remote-control purposes," *Actuators*, vol. 11, p. 65, 02 2022.

[105] L. Chen, H. Wu, S. Duan, Y. Guan, and J. Rojas, "Learning human-robot collaboration insights through the integration of muscle activity in interaction motion models," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 491–496.

[106] J. Luo, C. Yang, N. Wang, and M. Wang, "Enhanced teleoperation performance using hybrid control and virtual fixture," *International Journal of Systems Science*, vol. 50, pp. 1–12, 2019.

[107] M. Semprini, A. V. Cuppone, I. Delis, V. Squeri, S. Panzeri, and J. Konczak, "Biofeedback signals for robotic rehabilitation: Assessment of wrist muscle activation patterns in healthy humans," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 7, pp. 883–892, 2017.

[108] C.-C. Liu, B. B. Atitallah, R. Ramalingame, R. Barioul, A. Djemal, H. Hellara, and O. Kanoun, "A hybrid measurement system for hand signs recognition based on emg-fmg measurements," *2022 IEEE 9th International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1–6, 2022.

[109] C. Ahmadizadeh, L.-K. Merhi, B. Pousett, S. Sangha, and C. Menon, "Toward intuitive prosthetic control: Solving common issues using force myography, surface electromyography, and pattern recognition in a pilot case study," *IEEE Robotics & Automation Magazine*, vol. 24, pp. 102–111, 2017.

[110] H. Wang, Q. Ding, Y. Luo, Z. Wu, J. Yu, H. Chen, Y. Zhou, H. Zhang, K. Tao, X. Chen, J. Fu, and J. Wu, "High-performance hydrogel sensors enabled multimodal and accurate human-machine interaction system for active rehabilitation." *Advanced materials*, 2023.

[111] F. Auat Cheein, N. López, C. Soria, F. Sciascio, F. Pereira, and R. Carelli, "Slam algorithm applied to robotics assistance for navigation in unknown environments," *Journal of neuroengineering and rehabilitation*, vol. 7, p. 10, 2010.

[112] J. DelPreto and D. Rus, "Sharing the load: Human-robot team lifting using muscle activity," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7906–7912.

[113] A. Úbeda, B. Zapata-Impata, S. Puente, P. Gil, F. Candelas Herias, and F. Torres, "A vision-driven collaborative robotic grasping system tele-operated by surface electromyography," *Sensors*, vol. 18, p. 2366, 2018.

[114] G. Trzmiel, D. Kurz, and W. Smoczyński, "The use of the emg signal for the arm model control," in *ITM Web of Conferences*, vol. 28. EDP Sciences, 2019, p. 01024.

[115] A. L. Gebregiorgis and E. E. Brown, "Using hidden markov models to track upper extremity arm motions for surface electromyographic based robot teleoperation," in *5th ISSNIP-IEEE Biosignals and Biorobotics Conference (2014): Biosignals and Robotics for Better and Safer Living (BRC)*, 2014, pp. 1–6.

[116] A. Al-Yacoub, M. Flanagan, A. Buerkle, T. Bamber, P. Ferreira, E.-M. Hubbard, and N. Lohse, "Data-driven modelling of human-human co-manipulation using force and muscle surface electromyogram activities," *Electronics*, vol. 10, no. 13, p. 1509, 2021.

[117] M. Maimeri, C. D. Santina, C. Piazza, M. Rossi, M. G. Catalano, and G. Grioli, "Design and assessment of control maps for multi-channel semg-driven prostheses and supernumerary limbs," *Frontiers in neurorobotics*, vol. 13, 2019.

[118] M. Pesenti, Z. Alkhoury, M. Bednarczyk, H. Omran, and B. Bayle, "Linear parameter-varying identification of the emg–force relationship of the human arm," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, pp. 1–6.

[119] F. Bian, D. Ren, R. Li, P. Liang, k. Wang, and L. Zhao, "An extended dmp framework for robot learning and improving variable stiffness manipulation," *Assembly Automation*, vol. ahead-of-print, 2019.

[120] G. Lentini, A. Settimi, D. Caporale, M. Garabini, G. Grioli, L. Pallottino, M. G. Catalano, and A. Bicchi, "Alter-ego: A mobile robot

with a functionally anthropomorphic upper body designed for physical interaction," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 94–107, 2019.

[121] J. Cacace, R. Caccavale, A. Finzi, and V. Lippiello, "Variable admittance control based on virtual fixtures for human-robot comanipulation," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 1569–1574.

[122] F. Zahedi, J. Arnold, C. Phillips, and H. Lee, "Variable damping control for phri: Considering stability, agility, and human effort in controlling human interactive robots," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 5, pp. 504–513, 2021.

[123] S. Lyu and C. C. Cheah, "Adaptive robot control for human-dominant interactions using a general task function," in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2016, pp. 39–44.

[124] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

[125] F. Ferraguti, C. Talignani Landi, L. Sabattini, M. Bonfe, C. Fantuzzi, and C. Secchi, "A variable admittance control strategy for stable physical human-robot interaction," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 747–765, 2019.

[126] H. Kim and W. Yang, "Variable admittance control based on human–robot collaboration observer using frequency analysis for sensitive and safe interaction," *Sensors*, vol. 21, no. 5, p. 1899, 2021.

[127] S. M. M. Rahman and R. Ikeura, "Weight-perception-based fixed and variable admittance control algorithms for unimanual and bimanual lifting of objects with a power assist robotic system," *International Journal of Advanced Robotic Systems*, vol. 15, 2018.

[128] G. Raiola, C. A. Cardenas, T. S. Tadele, T. de Vries, and S. Stramigioli, "Development of a safety- and energy-aware impedance controller for collaborative robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1237–1244, 2018.

[129] H. Wang, F. Patota, G. Buondonno, M. Haendl, A. Luca, and K. Kosuge, "Stability and variable admittance control in the physical interaction with a mobile robot," *International Journal of Advanced Robotic Systems*, vol. 12, p. 173, 2015.

[130] S. Gopinathan, S. Ötting, and J. Steil, "A user study on personalized stiffness control and task specificity in physical human–robot interaction," *Frontiers in Robotics and AI*, vol. 4, 2017.

[131] Y. Wang, Y. Yang, B. Zhao, X. Qi, Y. Hu, B. Li, L. Sun, L. Zhang, and M. Meng, "Variable admittance control based on trajectory prediction of human hand motion for physical human-robot interaction," *Applied Sciences*, vol. 11, p. 5651, 2021.

[132] S. Grafakos, F. Dimeas, and N. Aspragathos, "Variable admittance control in phri using emg-based arm muscles co-activation," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 001 900–001 905.

[133] F. Ferraguti, N. Preda, A. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, "An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, 2015.

[134] C. T. Landi, F. Ferraguti, L. Sabattini, C. Secchi, and C. Fantuzzi, "Admittance control parameter adaptation for physical human-robot interaction," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2911–2916.

[135] Y. Shen, B. Li, W. Tian, J. Duan, and M. Liu, "Human-robot collaborative assembly of movable airfoils using adaptive admittance control," *Journal of Intelligent Manufacturing and Special Equipment*, vol. 3, no. 2, pp. 204–216, 2022.

[136] X. Chen, N. Wang, H. Cheng, and C. Yang, "Neural learning enhanced variable admittance control for human–robot collaboration," *IEEE Access*, vol. 8, pp. 25 727–25 737, 2020.

[137] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. Molinari Tosatti, and N. Pedrocchi, "Model-based reinforcement learning

variable impedance control for human-robot collaboration," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 2, pp. 417–433, 2020.

[138] A.-N. Sharkawy, P. N. Koustournpardis, and N. Aspragathos, "Variable admittance control for human-robot collaboration based on online neural network training," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1334–1339.

[139] F. Dimeas and N. Aspragathos, "Fuzzy learning variable admittance control for human-robot cooperation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4770–4775.

[140] D. L. Schomer and F. H. Lopes da Silva, *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Oxford University Press, 11 2017.

[141] F. Martini, J. Nath, and E. Bartholomew, *Fundamentals of Anatomy & Physiology, Global Edition*. Pearson Education, 2017.

[142] J. Cui, L. Su, R. Wei, G. Li, H. Hu, and X. Dang, "Eeg authentication based on deep learning of triplet loss," *Neural Network World*, vol. 32, no. 5, p. 269, 2022.

[143] J. Luo, C. Sun, B. Chang, Y. Jing, K. Li, Y. Li, Q. Zhang, H. Wang, and C. Hou, "Mxene-enabled self-adaptive hydrogel interface for active electroencephalogram interactions," *ACS nano*, vol. 16, no. 11, pp. 19 373–19 384, 2022.

[144] P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, "Chapter 2 - technological basics of eeg recording and operation of apparatus," in *Introduction to EEG- and Speech-Based Emotion Recognition*, P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, Eds. Academic Press, 2016, pp. 19–50.

[145] S. Luck, *An Introduction to The Event-Related Potential Technique*. MIT press, 2014.

[146] T. W. Picton *et al.*, "The p300 wave of the human event-related potential," *Journal of clinical neurophysiology*, vol. 9, pp. 456–456, 1992.

[147] J. Polich, "Updating p300: an integrative theory of p3a and p3b," *Clinical neurophysiology*, vol. 118, no. 10, pp. 2128–2148, 2007.

[148] D. Zhu, J. Bieger, G. Garcia-Molina, and R. Aarts, "A survey of stimulation methods used in ssvep-based bcis," *Computational Intelligence and Neuroscience*, vol. 2010, p. 702357, 01 2010.

[149] S. Hayashi, K. Shimura, and T. Kasai, "Rapid plastic changes of human primary motor cortex with repetitive motor practice and transcranial magnetic stimulation," *Perceptual and Motor Skills*, vol. 101, no. 2, pp. 575–586, 2005.

[150] C. A. Porro, V. Cettolo, M. P. Francescato, and P. Baraldi, "Ipsilateral involvement of primary motor cortex during motor imagery," *European Journal of Neuroscience*, vol. 12, no. 8, pp. 3059–3063, 2000.

[151] K. Kilteni, B. Andersson, C. Houborg, and H. Ehrsson, "Motor imagery involves predicting the sensory consequences of the imagined movement," *Nature Communications*, vol. 9, 2018.

[152] R. Merletti and D. Farina, *Surface Electromyography: Physiology, Engineering and Applications*. John Wiley & Sons, Ltd, 2016.

[153] D. I. Rubin, "Chapter 16 - needle electromyography: Basic concepts," in *Clinical Neurophysiology: Basis and Technical Aspects*, ser. Handbook of Clinical Neurology, K. H. Levin and P. Chauvel, Eds. Elsevier, 2019, vol. 160, pp. 243–256.

[154] M. Mabrouk and O. Diab, "Surface multi-purposes low power wireless electromyography (emg) system design," *International Journal of Computer Applications*, vol. 41, pp. 10–16, 03 2012.

[155] G. Durandau, D. Farina, G. Asín-Prieto, I. Dimbwadyo-Terrer, S. Lerma-Lara, J. L. Pons, J. C. Moreno, and M. Sartori, "Voluntary control of wearable robotic exoskeletons by patients with paresis via neuromechanical modeling," *Journal of NeuroEngineering and Rehabilitation*, vol. 16, 2019.

[156] J. Zhang, T. Wang, Z. Qingyu, and S. Liu, "Impacts of a lower limb exoskeleton robot on the muscle strength of tibialis anterior muscle

in stroke patients," *E3S Web of Conferences*, vol. 185, p. 03036, 01 2020.

[157] V. Khoshdel and A. Akbarzadeh, "An optimized artificial neural network for human-force estimation: consequences for rehabilitation robotics," *Industrial Robot: An International Journal*, vol. 45, pp. 416–423, 2018.

[158] R. d. J. Portillo-Velez, A. Rodriguez-Angeles, and C. A. Cruz-Villar, "An optimal admittance approach for physical human-robot interaction," in *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control*, 2011, pp. 1–6.

[159] J. Sun, G. Liu, Y. Sun, K. Lin, Z. Zhou, and J. Cai, "Application of surface electromyography in exercise fatigue: A review," *Frontiers in Systems Neuroscience*, vol. 16, p. 893275, 2022.

[160] J. Pazos and P. Petromilli Nordi Sasso Garcia, "Using electromyography to assess postural load in dentistry: A literature review," *Journal of Advances in Medicine and Medical Research*, pp. 23–35, 2020.

[161] I. Eyad, E. I. Abbas, S. Khaleda, K. Rijab, A. Fuad, and A. Ahmed, "Optimal wavelet filter for de-noising surface electromyographic signal captured from biceps brachii muscle," *Engineering and Technology Journal*, vol. 33, 2015.

[162] S. Linderman, D. Scarborough, R. Aspenleiter, H. Stein, and E. Berkson, "Assessing quadriceps muscle contraction using a novel surface mechanomyography sensor during two neuromuscular control screening tasks," *Sensors (Basel, Switzerland)*, vol. 23, 06 2023.

[163] A. Kimoto, H. Fujiyama, and M. Machida, "A wireless multi-layered emg/mmg/nirs sensor for muscular activity evaluation," *Sensors*, vol. 23, p. 1539, 01 2023.

[164] T. Roland, K. Wimberger, S. Amsuss, M. Russold, and W. Baumgartner, "An insulated flexible sensor for stable electromyography detection: Application to prosthesis control," *Sensors*, vol. 19, p. 961, 2019.

[165] F. Caccavale, S. Chiaverini, and B. Siciliano, "Second-order kinematic control of robot manipulators with jacobian damped least-squares inverse: theory and experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 3, pp. 188–194, 1997.

[166] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2010.

[167] S. Chiaverini, B. Siciliano, and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp. 123–134, 1994.

[168] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, 1991, pp. 1211–1216 vol.2.

[169] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[170] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

[171] E. Simetti and G. Casalino, "A novel practical technique to integrate inequality control objectives and task transitions in priority based control," *Journal of Intelligent & Robotic Systems*, vol. 84, 12 2016.

[172] S. Moe, G. Antonelli, A. Teel, K. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, 04 2016.

[173] P. Di Lillo, F. Arrichiello, G. Antonelli, and S. Chiaverini, "Safety-related tasks within the set-based task-priority inverse kinematics framework," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6130–6135.

[174] P. Di Lillo, E. Simetti, F. Wanderlingh, G. Casalino, and G. Antonelli, "Underwater intervention with remote supervision via satellite communication: Developed control architecture and experimental results within the dexrov project," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 108–123, 2021.

[175] N. C. M. Nickelsen, "Imagining and tinkering with assistive robotics in care for the disabled," *Paladyn, Journal of Behavioral Robotics*, vol. 10, no. 1, pp. 128 – 139, 2019.

[176] H. Wang and X. P. Liu, "Adaptive shared control for a novel mobile assistive robot," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1725–1736, 2014.

[177] A. De Luca and F. Flacco, "Integrated control for phri: Collision avoidance, detection, reaction and collaboration," in *4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2012, pp. 288–295.

[178] D. Achanccaray, J. M. Chau, J. Pirca, F. Sepulveda, and M. Hayashibe, "Assistive robot arm controlled by a p300-based brain machine interface for daily activities," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2019, pp. 1171–1174.

[179] M. Aljalal, R. Djemal, and S. Ibrahim, "Robot navigation using a brain computer interface based on motor imagery," *Journal of Medical and Biological Engineering*, vol. 39, no. 4, pp. 508–522, 2019.

[180] Y. Yu, Z. Zhou, Y. Liu, J. Jiang, E. Yin, N. Zhang, Z. Wang, Y. Liu, X. Wu, and D. Hu, "Self-paced operation of a wheelchair based on a hybrid brain-computer interface combining motor imagery and p300 potential," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 12, pp. 2516–2526, 2017.

[181] G. Gillini, P. Di Lillo, F. Arrichiello, D. Di Vito, A. Marino, G. Antonelli, and S. Chiaverini, "A dual-arm mobile robot system performing assistive tasks operated via p300-based brain computer interface," *Industrial Robot*, 2020.

[182] G. Gillini, P. D. Lillo, and F. Arrichiello, "An assistive shared control architecture for a robotic arm using eeg-based bci with motor imagery," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4132–4137.

[183] T. Mwata-Velu, J. Ruiz-Pinales, H. Rostro-Gonzalez, M. A. Ibarra-Manzano, J. M. Cruz-Duarte, J. G. Avina-Cervantes *et al.*, "Motor imagery classification based on a recurrent-convolutional architecture to control a hexapod robot," *Mathematics*, vol. 9, no. 6, p. 606, 2021.

[184] A. M. Batula, Y. E. Kim, and H. Ayaz, "Virtual and actual humanoid robot control with four-class motor-imagery-based optical brain-computer interface," *BioMed research international*, 2017.

[185] K. LaFleur, K. Cassady, A. Doud, K. Shades, E. Rogin, and B. He, "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface," *Journal of neural engineering*, vol. 10, no. 4, p. 046003, 2013.

[186] V. Gandhi, G. Prasad, D. Coyle, L. Behera, and T. Mcginnity, "Eeg-based mobile robot control through an adaptive brain–robot interface," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, pp. 1278–1285, 2014.

[187] L. Cao, G. Li, Y. xu, H. Zhang, X. Shu, and D. Zhang, "A brain-actuated robotic arm system using non-invasive hybrid brain-computer interface and shared control strategy," *Journal of Neural Engineering*, vol. 18, 04 2021.

[188] J. Downey, J. M. Weiss, K. Muelling, A. Venkatraman, J.-S. Valois, M. Hebert, J. Bagnell, A. B. Schwartz, and J. Collinger, "Blending of brain-machine interface and vision-guided autonomous robotics improves neuroprosthetic arm performance during grasping," *Journal of NeuroEngineering and Rehabilitation*, vol. 13, 2016.

[189] R. Liu, K. Xue, Y. Wang, and L. Yang, "A fuzzy-based shared controller for brain-actuated simulated robotic system," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 7384–7387, 2011.

[190] K. Muelling, A. Venkatraman, J.-S. Valois, J. Downey, J. M. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. Collinger, and J. Bagnell, "Autonomy infused teleoperation with application to brain computer interface controlled manipulation," *Autonomous Robots*, vol. 41, pp. 1401–1422, 2017.

[191] F. Arrichiello, P. D. Lillo, D. D. Vito, G. Antonelli, and S. Chiaverini, "Assistive robot operated via p300-based brain computer interface," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6032–6037, 2017.

[192] Y. Xu, H. Zhang, L. Cao, X. Shu, and D. Zhang, "A shared control strategy for reach and grasp of multiple objects using robot vision and noninvasive brain–computer interface," *IEEE Transactions on Automation Science and Engineering*, vol. 19, pp. 360–372, 2022.

[193] F. Patriarca, G. Gillini, P. D. Lillo, and F. Arrichiello, "Null-space-based shared control of a mobile robot using motor imagery based brain-computer interface," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022, pp. 2263–2268, ©2024 Patriarca Francesca.

[194] P. Di Lillo, S. Chiaverini, and G. Antonelli, "Handling robot constraints within a set-based multi-task priority inverse kinematics framework," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7477–7483.

[195] L. Tonin, F. C. Bauer, and J. d. R. Millán, "The role of the control framework for continuous teleoperation of a brain-machine interface-driven mobile robot," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 78–91, 2019.

[196] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[197] P. Di Lillo, F. Arrichiello, D. Di Vito, and G. Antonelli, "Bci-controlled assistive manipulator: Developed architecture and experimental results," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 91–104, 2021.

[198] E. Cataldi, F. Real, A. Suarez, P. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, G. Heredia, and A. Ollero, "Set-based inverse kinematics control of an anthropomorphic dual arm aerial manipulator," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2960–2966.

[199] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[200] Z. A. Adeola-Bello and N. Z. Azlan, "Power assist rehabilitation robot and motion intention estimation," *International Journal of Robotics and Control Systems*, vol. 2, no. 2, pp. 297–316, 2022.

[201] A. Gonzalez-Mendoza, I. Quinones-Uriostegui, S. Salazar-Cruz, A. I. Perez Sanpablo, R. López, and R. Lozano, "Design and implementation of a rehabilitation upper-limb exoskeleton robot controlled by cognitive and physical interfaces," *Journal of Bionic Engineering*, vol. 19, 06 2022.

[202] Y. Aydin, D. Sirintuna, and C. Basdogan, "Towards collaborative drilling with a cobot using admittance controller," *Transactions of the Institute of Measurement and Control*, vol. 43, no. 8, pp. 1760–1773, 2021.

[203] J. Bae, K. Kim, J. Huh, and D. Hong, "Variable admittance control with virtual stiffness guidance for human–robot collaboration," *IEEE Access*, vol. 8, pp. 117 335–117 346, 2020.

[204] J. Li, G. Li, Z. Chen, and J. Li, "A novel emg-based variable impedance control method for a tele-operation system under an unstructured environment," *IEEE Access*, vol. 10, pp. 89 509–89 518, 2022.

[205] C. Zeng, C. Yang, H. Cheng, Y. Li, and S.-L. Dai, "Simultaneously encoding movement and sEMG-based stiffness for robotic skill learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1244–1252, 2021.

[206] J. Tian, H. Wang, S. Zheng, Y. Ning, X. Zhang, J. Niu, and L. Vladareanu, "sEMG-based gain-tuned compliance control for the

lower limb rehabilitation robot during passive training," *Sensors*, vol. 22, no. 20, 2022.

[207] Y. Zhuang, S. Yao, C. Ma, and R. Song, "Admittance control based on EMG-driven musculoskeletal model improves the human–robot synchronization," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1211–1218, 2019.

[208] Q. Wu, B. Chen, and H. Wu, "Adaptive admittance control of an upper extremity rehabilitation robot with neural-network-based disturbance observer," *IEEE Access*, vol. 7, pp. 123 807–123 819, 2019.

[209] Y. Zhuang, Y. Leng, J. Zhou, R. Song, L. Li, and S. W. Su, "Voluntary control of an ankle joint exoskeleton by able-bodied individuals and stroke survivors using emg-based admittance control scheme," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 2, pp. 695–705, 2021.

[210] A. Villa-Parra, D. Delisle Rodriguez, T. Botelho, J. Villarejo Mayor, A. Delis, R. Carelli, A. Frizera, and T. Freire, "Control of a robotic knee exoskeleton for assistance and rehabilitation based on motion intention from sEMG," *Research on Biomedical Engineering*, vol. 34, 2018.

[211] L. Han, L. Zhao, Y. Huang, and W. Xu, "Variable admittance control for safe physical human–robot interaction considering intuitive human intention," *Mechatronics*, vol. 97, 2024.

[212] D. P. Losey, C. G. Mcdonald, E. Battaglia, and M. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human-robot interaction," *Applied Mechanics Reviews*, vol. 70, pp. 010 804–010 804, 2018.

[213] P. Huang, H. Wang, Y. Wang, Z. Liu, O. W. Samuel, M. Yu, X. Li, S. Chen, and G. Li, "Identification of upper-limb movements based on muscle shape change signals for human-robot interaction," *Computational and Mathematical Methods in Medicine*, vol. 2020, 2020.

[214] D. Gopinath, S. Jain, and B. Argall, "Human-in-the-loop optimization of shared autonomy in assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, pp. 247–254, 2017.

[215] F. Khadivar, V. Mendez, C. Correia, I. Batzianoulis, A. Billard, and S. Micera, "Emg-driven shared human-robot compliant control for in-hand object manipulation in hand prostheses," *Journal of Neural Engineering*, vol. 19, 2022.

[216] F. Rezazadegan, J. Geng, M. Ghirardi, G. Menga, S. Murè, G. Camuncoli, and M. Demichela, "Risked-based design for the physical human-robot interaction (phri): an overview," *Chemical engineering transactions*, vol. 43, pp. 1249–1254, 2015.

[217] F. Patriarca, P. Di Lillo, and F. Arrichiello, "Emg-based shared control framework for human-robot co-manipulation tasks," in *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO*, INSTICC. SciTePress, 2024, pp. 46–53.

[218] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[219] P. Di Lillo, D. D. Vito, and G. Antonelli, "Merging global and local planners: Real-time replanning algorithm of redundant robots within a task-priority framework," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 1180–1193, 2023.

[220] H. Manjunatha, S. S. Jujjavarapu, and E. T. Esfahani, "Classification of motor control difficulty using EMG in physical human-robot interaction," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 2708–2713.

[221] O. Kramer, "Scikit-learn," *Machine learning for evolution strategies*, pp. 45–53, 2016.

[222] F. Patriarca, P. Di Lillo, and F. Arrichiello, "Emg-driven shared control architecture for human-robot co-manipulation tasks," *Submitted to IEEE Transactions on Human-Machine Systems*, 2024.