Active Training and Assistance Device for an Individually Adaptable Strength and Coordination Training

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Denis Štogl

aus Bjelovar (Kroatien)

Tag der mündlichen Prüfung: 29. Januar 2021

Erster Gutachter: Prof. Dr.-Ing. habil. Björn Hein

Zweiter Gutachter: Prof. Dr. Sc. Ivan Petrović

Institute for Anthropomatics and Robotics (IAR) -
Intelligent Process Automation and Robotics Lab (IPR)
KIT Department of Informatics
Karlsruhe Institute of Technology
Engler-Bunte-Ring 8
76131 Karlsruhe

First of all, I would like to express my sincere gratitude to Prof. Dr.-Ing. habil. Björn Hein for the possibilities and support he has given me throughout the research conducted for this thesis. I especially fondly remember the time back in December 2011 when I, as an exchange student speaking "broken" German, asked him for an opportunity for a student job at the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics (IAR-IPR). Without those nine months of working under his supervision, I probably would have never ended up being a part of the scientific staff in Prof. Hein's research group in September 2013. I hope to return the trust and support he gave to me during seven years at the IAR-IPR with the research presented in this thesis. Nevertheless, the leadership of Prof. Hein shaped me into the roboticist I am today.

Secondly, I would like to thank Prof. Dr. Sc. Ivan Petrović from the Department of Control and Computer Engineering, University of Zagreb Faculty of Electrical Engineering and Computing, who supervised me since my start in the field of robotics. Especially, I would like to thank him for his support to participate in the Erasmus exchange program and for supervising my bachelor and master theses. I was thrilled when he accepted to be an examiner for this Ph.D. Thesis.

Thanks to all my colleagues from IAR-IPR for providing a supportive and thriving working environment and for all the help and comments they offered to improve my research. Many thanks to the "RoboTrainer-safety-team", Dennis Hartmann, Tom Huck, Christoph Ledermann, and Michael Mende, who provided invaluable support for developing and evaluating the safety concept. Especially, thanks to Michael Mende for all his hours spent and all the knowledge he unselfishly shared to solve challenges when designing and developing both versions of RoboTrainer. All of the work would be at least much harder without his ingenious solutions and broad experience with mechanics and electronics. Furthermore, thanks to all the students I supervised during my time at IAR-IPR and especially those that contributed bits and pieces to my research.

I am deeply grateful to the team from Central Institute of Mental Health (CIMH), Prof. Dr. Patric Meyer, Michael Hoppstädter, and Regine Bader, for the organization and excellent cooperation during the pilot study of RoboTrainer Prototype. The team from German Sport University Cologne (DSHS), Prof. Dr. Kirsten Albracht, Dr. Björn Braunstein and Fabian Göll, didn't contribute anything less to this thesis by organizing the evolution study of RoboTrainer v2.

A big "thank you" also goes to rob@work-team, specifically Mike Siee and Mojin Robotics GmbH, especially Tim Fröhlich, for all their time and support on hardware repairs.

A special thanks goes to my parents, Zdenka and Željko, who supported me in all my "crazy" ideas in my life that finally led to this work. Thank you for not stopping me

from going to Music school aside from the regular gymnasium. Thank you for letting me go to summer schools in Višnjan and selflessly supporting my education and studies. Thank you for being strong and standing behind me when I decided to go to Karlsruhe for student exchange and finally letting me move away from home to pursue my ideas. Now I understand that it has not always been easy for you.

Last but not least, I would like to thank my wife, Luana, and our two dogs, Botan and Jara, for occupying me with "normal" things in life. Without them, my thesis might have been finished faster but definitely, it would have been less fun. Thank you for keeping balance in my life and being there for me in all challenging and despairing moments.

Abstract

Active Training and Assistance Device for an Individually Adaptable Strength and Coordination Training

The aging of the world's population, especially evident in western countries, is a great challenge for humankind. A significant impact is expected on the healthcare sector confronted with an increasing number of people needing individual care because of the aging-related decline of physical and cognitive capabilities. Therefore, especially in the last century, many research efforts were invested in order to understand the causes and developments of aging-related diseases, comprehend their progress, and find possible treatments.

The current models show that the most relevant variable for the progress of such diseases is the lack of sensory and motor input resulting from decreased mobility and the lack of novel experiences. This is confirmed by many studies showing positive effects of physical activity on the overall state of older adults with mild cognitive impairment and people around them. This thesis aims to enrich older adults' possibilities to engage in physical activity in a self-determined and safe manner considering their individual needs.

In the last two decades, the research on robotized walking support devices, called Smart Walkers, focused on sensory and cognitive assistance for older or disabled persons. From those endeavors a variety of human-walker interaction interfaces and methods for locomotion and navigation support emerged. However, training possibilities for motor activation of people using a Smart Walker are not yet investigated. In contrast to a few Smart Walkers examining rehabilitation possibilities by focusing on mitigation of advanced disease, this thesis introduces a device to address cognitive impairments at an early stage for prolonging users' physical and mental fitness.

To validate the idea of such training, a prototype device, called RoboTrainer Prototype, was developed by extending the research mobile platform with a force-torque sensor and a bike-handlebar as input interface. The training consists of predefined training paths marked on the floor along which users have to navigate the device. The prototype uses an admittance equation to generate its velocity from user's input and introduces control actions, i.e., behavior modifiers to configure the training more challenging. The pilot study with ten older adults with mild cognitive impairment showed a significant increase in users' interaction performance with the device. It also showed the usefulness of control actions to adjust training complexity.

Although the pilot showed the feasibility of the training, the RoboTrainer Prototype's footprint and mechanical robustness were suboptimal. Therefore, the second part of the thesis focuses on designing a novel device to overcome the prototype's drawbacks. Besides higher mechanical stability, the RoboTrainer v2 provides adjustment of its footprint, i.e., users' support area, as a unique feature compared to other Smart Walkers. This enables agile training with healthy users and, at the same time, rehabilitation scenarios where physical support is necessary.

The control approach for RoboTrainer v2 extends the admittance control of the prototype by implementing three adaptive strategies. The first is the adaption of users' input's sensitivity depending on the user-walker system's stability, which avoids oscillations when users stiff their hands. The second adaption includes a novel non-linear velocity-based alteration of admittance parameters to increase the device's performance in agile training. The third adaption is utilized in a pre-training parameterization process, where interaction forces are calculated per-user to fine-tune individual controller's constants.

The control actions are behavior modifiers that serve as building blocks of supportive and challenging training with RoboTrainer. They use the virtual force field concept to influence the device's movement within the training environment. The movement is influenced by global control actions (entire environment) or spatial control actions (limited areas). The control actions keep user's intention intact by implementing an independent admittance dynamic to calculate velocity influence to RoboTrainer. This enables the crucial separation of controller states to achieve passive and safe modification of the device's behavior during training.

The contributions are validated separately and in two user evaluations with twenty-two and thirteen young, healthy adults. The evaluations allow insights on coherence between developed functionalities and their influence on users. They also validate the overall approach and confirm some of the assumptions done when designing individual RoboTrainer's components.

This thesis' contributions result in a novel device for research of physical human-robot interaction in training with adults. The future research with this device opens a path for Smart Walkers to support societies in dealing with upcoming demographic changes.

Keywords: *Smart Walker, RoboTrainer, physical human-robot interaction, safety for smart walkers, mechanical adaption, adaptive admittance control, individual control, parameterization process, control actions, behavior modifiers, motor activation, training for people with MCI, mild congnitive impairment, robot development*

Zusammenfassung

Aktives Trainings- und Assistenzsystem für ein an Menschen individuell anpassbares Kraft- und Koordinationstraining

Das Altern der Weltbevölkerung, insbesondere in der westlichen Welt, stellt die Menschheit vor eine große Herausforderung. Zu erwarten sind erhebliche Auswirkungen auf den Gesundheitssektor, der im Hinblick auf eine steigende Anzahl von Menschen mit altersbedingtem körperlichem und kognitivem Abbau und dem damit erhöhten Bedürfnis einer individuellen Versorgung vor einer großen Aufgabe steht. Insbesondere im letzten Jahrhundert wurden viele wissenschaftliche Anstrengungen unternommen, um Ursache und Entwicklung altersbedingter Erkrankungen, ihr Voranschreiten und mögliche Behandlungen, zu verstehen.

Die derzeitigen Modelle zeigen, dass der entscheidende Faktor für die Entwicklung solcher Krankheiten der Mangel an sensorischen und motorischen Einflüssen ist, diese wiederum sind das Ergebnis verringerter Mobilität und immer weniger neuer Erfahrungen. Eine Vielzahl von Studien zeigt, dass erhöhte körperliche Aktivität einen positiven Effekt auf den Allgemeinzustand von älteren Erwachsenen mit leichten kognitiven Beeinträchtigungen und den Menschen in deren unmittelbarer Umgebung hat. Diese Arbeit zielt darauf ab, älteren Menschen die Möglichkeit zu bieten, eigenständig und sicher ein individuelles körperliches Training zu absolvieren.

In den letzten zwei Jahrzehnten hat die Forschung im Bereich der robotischen Bewegungsassistenten, auch Smarte Rollatoren genannt, den Fokus auf die sensorische und kognitive Unterstützung für ältere und eingeschränkte Personen gesetzt. Durch zahlreiche Bemühungen entstand eine Vielzahl von Ansätzen zur Mensch-Rollator-Interaktion, alle mit dem Ziel, Bewegung und Navigation innerhalb der Umgebung zu unterstützen.

Aber trotz allem sind Trainingsmöglichkeiten zur motorischen Aktivierung mittels Smarter Rollatoren noch nicht erforscht. Im Gegensatz zu manchen Smarten Rollatoren, die

den Fokus auf Rehabilitationsmöglichkeiten für eine bereits fortgeschrittene Krankheit setzen, zielt diese Arbeit darauf ab, kognitive Beeinträchtigungen in einem frühen Stadium soweit wie möglich zu verlangsamen, damit die körperliche und mentale Fitness des Nutzers so lang wie möglich aufrechterhalten bleibt.

Um die Idee eines solchen Trainings zu überprüfen, wurde ein Prototyp-Gerät namens RoboTrainer-Prototyp entworfen, eine mobile Roboter-Plattform, die mit einem zusätzlichen Kraft-Momente-Sensor und einem Fahrradlenker als Eingabe-Schnittstelle ausgestattet wurde. Das Training beinhaltet vordefinierte Trainingspfade mit Markierungen am Boden, entlang derer der Nutzer das Gerät navigieren soll. Der Prototyp benutzt eine Admittanzgleichung, um seine Geschwindigkeit anhand der Eingabe des Nutzers zu berechnen. Desweiteren leitet das Gerät gezielte Regelungsaktionen bzw. Verhaltensänderungen des Roboters ein, um das Training herausfordernd zu gestalten.

Die Pilotstudie, die mit zehn älteren Erwachsenen mit beginnender Demenz durchgeführt wurde, zeigte eine signifikante Steigerung ihrer Interaktionsfähigkeit mit diesem Gerät. Sie bewies ebenfalls den Nutzen von Regelungsaktionen, um die Komplexität des Trainings ständig neu anzupassen. Obwohl diese Studie die Durchführbarkeit des Trainings zeigte, waren Grundfläche und mechanische Stabilität des RoboTrainer-Prototyps suboptimal. Deswegen fokussiert sich der zweite Teil dieser Arbeit darauf, ein neues Gerät zu entwerfen, um die Nachteile des Prototyps zu beheben. Neben einer erhöhten mechanischen Stabilität, ermöglicht der RoboTrainer v2 eine Anpassung seiner Grundfläche. Dieses spezifische Merkmal der Smarten Rollatoren dient vor allem dazu, die Unterstützungsfläche für den Benutzer anzupassen. Das ermöglicht einerseits ein agiles Training mit gesunden Personen und andererseits Rehabilitations-Szenarien bei Menschen, die körperliche Unterstützung benötigen.

Der Regelungsansatz für den RoboTrainer v2 erweitert den Admittanzregler des Prototypen durch drei adaptive Strategien. Die erste ist die Anpassung der Sensitivität an die Eingabe des Nutzers, abhängig von der Stabilität des Nutzer-Rollater-Systems, welche Schwankungen verhindert, die dann passieren können, wenn die Hände des Nutzers versteifen. Die zweite Anpassung beinhaltet eine neuartige nicht-lineare, geschwindigkeitsbasierende Änderung der Admittanz-Parameter, um die Wendigkeit des Rollators zu erhöhen. Die dritte Anpassung erfolgt vor dem eigentlichen Training in einem Parametrierungsprozess, wo nutzereigene Interaktionskräfte gemessen werden, um individuelle Reglerkonstanten fein abzustimmen und zu berechnen.

Die Regelungsaktionen sind Verhaltensänderungen des Gerätes, die als Bausteine für unterstützende und herausfordernde Trainingseinheiten mit dem RoboTrainer dienen. Sie nutzen das virtuelle Kraft-Feld-Konzept, um die Bewegung des Gerätes in der Trainingsumgebung zu beeinflussen. Die Bewegung des RoboTrainers wird in der Gesamtumgebung durch globale oder, in bestimmten Teilbereichen, durch räumliche Aktionen beeinflusst. Die Regelungsaktionen erhalten die Absicht des Nutzers aufrecht, in dem sie eine

unabhängige Admittanzdynamik implementieren, um deren Einfluss auf die Geschwindigkeit des RoboTrainers zu berechnen. Dies ermöglicht die entscheidende Trennung von Reglerzuständen, um während des Trainings passive und sichere Interaktionen mit dem Gerät zu erreichen.

Die oben genannten Beiträge wurden getrennt ausgewertet und in zwei Studien mit jeweils 22 bzw. 13 jungen, gesunden Erwachsenen untersucht. Diese Studien ermöglichen einen umfassenden Einblick in die Zusammenhänge zwischen unterschiedlichen Funktionalitäten und deren Einfluss auf die Nutzer. Sie bestätigen den gesamten Ansatz, sowie die gemachten Vermutungen im Hinblick auf die Gestaltung einzelner Teile dieser Arbeit.

Die Einzelergebnisse dieser Arbeit resultieren in einem neuartigen Forschungsgerät für physische Mensch-Roboter-Interaktionen während des Trainings mit Erwachsenen. Zukünftige Forschungen mit dem RoboTrainer ebnen den Weg für Smarte Rollatoren als Hilfe für die Gesellschaft im Hinblick auf den bevorstehenden demographischen Wandel.

Stichwörter: *Smarter Rollator, RoboTrainer, physische Mensch-Roboter Interaktion, funktionelle Sicherheit für smarte Rollatoren, mechanische Anpassung, adaptive Admittanzregelung, individuelle Regelung, Parametrisierungsvorgehen, Regelungsaktionen, Verhaltensmodifikation, motorische Aktivierung, Training für Menschen mit beginnender Demenz, Roboterentwicklung*

Contents

Abstract vii

Zusammenfassung xi

1. Introduction 1

1.1. Motivation and Problem Description 1
1.2. Main Contributions and Goal of this Thesis 4
1.3. Outline 7
1.4. Nomenclature 8

2. Background and Related Work 11

2.1. From Gait Dysfunction and MCI to Neuromuscular Training 12
2.2. Conventional Assistive and Training Devices for Mobility 14
2.3. Smart Walkers – More than Physical Gait Assistance Devices 17
2.3.1. From Passive Navigation Assistance to Active Rehabilitation Devices 18
2.3.2. Commercial Smart Walkers 30
2.3.3. Design of Smart Walkers 31
2.3.4. Control of Smart Walkers 35
2.3.5. High-Level Control Approaches 41
2.3.6. Tracking User's Legs during Interaction with a Smart Walker . . 42
2.4. Comparison of Relevant SotA-Smart Walkers' Features 43
2.5. Relevant Control Theory Background 46

3. Proof of Concept of a Smart Walker for Training 49

3.1. A Neuromuscular Training with a Mobile Robotic Device 50
3.2. Design of the RoboTrainer Prototype 54
3.2.1. Hardware 54
3.2.2. Software and Control of RoboTrainer Prototype 55
3.3. Evaluation of the Concept 60
3.3.1. Experiment 62

3.3.2. Results and Discussion 70
3.4. Conclusions 83
3.5. Lessons Learned 84

4. Design of a Device for Active Training 87
4.1. Use of the RoboTrainer 88
4.2. Technical Requirements for RoboTrainer v2 90
4.2.1. Requirements from Walkers-Related Norms 90
4.2.2. Requirements from Norms for Machines and Robots 91
4.2.3. Functional Requirements for Mechatronic Design 94
4.3. Mechatronic Design 94
4.4. Safety Concept for RoboTrainer v2 102
4.4.1. Concept 103
4.4.2. Realization 105
4.4.3. Discussion on Further Safety Measures 109
4.5. Conclusion on the Device Design 110

5. Control of Devices for Active Training 113
5.1. Control Architecture and Strategy 114
5.1.1. Details on the Implementation of the Control Strategy for RoboTrainer 116
5.2. Description of the User-RoboTrainer Interaction for Control 117
5.3. Stability of the User-Walker System in Active Training 125
5.3.1. Reasons for Unstable Behavior of the User-Walker System 125
5.3.2. Stabilization of the User-Walker System 129
5.4. Velocity-Dependent Adaption 136
5.5. Per-User Specific Control 140
5.5.1. Parameterization Strategy for Measuring of Maximal User's Interaction Forces 143
5.5.2. Parameterization Strategy for Measuring Velocity-Dependant-Adaption Parameters 144
5.6. Implementation of the Control Approach using ros_control Framework . 145
5.7. Conclusion on the RoboTrainer's Control 148

6. Control Actions as Modifiers of Smart Walkers' Behavior 149
6.1. Concept of Control Actions 150
6.1.1. Global Control Actions 151
6.1.2. Spatial Control Actions 153
6.2. Passivity and Safety for Control Actions 162
6.3. Implementation of the spatial control actions 169
6.4. Conclusions on Control Actions for Training 173

7. User Evaluations 175
7.1. Evaluation of Adaptive and Individualized Control 176
7.1.1. Evaluation Setup 176
7.1.2. Results and Discussion 180
7.1.3. Findings from the Evaluation 189
7.2. Evaluation of the RoboTrainer v2 193
7.2.1. Evaluation Setup 193
7.2.2. Results and Discussion 198
7.2.3. Findings from the Evaluation of RoboTrainer v2 217

8. Conclusions 219

Bibliography 223

Appendix 245
A. Overview of Smart Walker 245
B. RoboTrainer's Hardware Technical Specification 266
B.1. Safety Hardware 266
B.2. Drive Steer Modules 266
B.3. Force Torque Sensor 266
C. Data representation 267
C.1. Boxplots 267
C.2. Data significance 267
D. Documents for Evaluation of Individual Parameters and Adaptive Control 268
D.1. Task description and clarification of use of user's data (in german) 268

Glossary 271

Acronyms 273

List of Figures 280

List of Tables 283

Listings 285

1 Introduction

1.1. Motivation and Problem Description

One of the primary objectives of modern society is to provide an environment which enables a fulfilled life to each individual. Expecting an increasingly aging world's population in the next decades [165], this becomes an ever-growing challenge. Until 2050, United Nations [166] expect an increase in older adults of 197.9 % worldwide, with immense growth in the developing countries. This growth could become one of the most significant social transformations of the twenty-first century [165], with implications for many sectors, from healthcare and labor to financial markets.

The prolonged life span bears many risks for individuals as aging negatively influences their physical and cognitive capabilities. Between age 40 and 80, the muscle power reduces 30-40 % [1, 100], and 40-80 % of older adults need rehabilitation [132, 143]. These age-related processes significantly impact walking abilities and lead to increased fall risk and consequently fall frequency. Therefore, older persons often suffer from serious injuries, followed by a loss of autonomy and life quality [54, 142]. According to Park and Reuter-Lorenz [128], a vicious circle starts for the affected persons since reduced sensorimotor input leads to partial "disuse" of the brain and thus even faster cognitive decline. The loss of movement-autonomy is the most fatal for people with age-related cognitive deterioration, e.g., mild cognitive impairment (MCI) or dementia. These people also have fewer prospective outcomes of medical interventions and rehabilitation [168].

Dementia, and its pre-stage MCI, are defined as "...an umbrella term for a number of progressive disorders of which Alzheimer's Disease (AD) is the most common, affecting memory, thinking, behavior and ability to perform everyday activities" [17]. On the neurophysiological level, this means loss of connectivity, reduced growth of new connections, and a lack of further learning capacity resulting in maladaptive behaviors such as motor instability, inactivity, and social isolation [128]. As emphasized by neuroscientific models, the most relevant variable for the progress of such diseases is the lack of sensory and motor input. This is confirmed by many researchers [9, 13, 126] as well as meta-analyses from Heyn, Abreu, and Ottenbacher [62] and Bowes et al. [17], showing the positive effect of physical activity to people with dementia and comparable cognitive impairments. The effects include improvements in cognition, mood, behavior, and physical fitness. Even studies with healthy persons show significant improvements in crucial neuro-cognitive processes that generally decline with age if the person is engaged in sensorimotor training to reverse cognitive plasticity [106]. Fleiner et al. [45] also show that physical exercise of older adults with dementia positively impacts caregivers and probably even all affected persons in their environment, like family and doctors.

Since the number of people who have dementia is estimated to triple worldwide by 2050 [181], many researchers investigate possibilities to prolong the persons' autonomy and reduce the burden on social system. A part of the solution could be smart assistive technologies. Such devices could delay individuals' need to enter a nursing home or reduce the burden on caretaking personnel [185]. Numerous authors show the need for such new-generation assistive devices to reduce the risk of falls, e.g., Bateni and Maki [12] and M. Bradley and R. Hernandez [104]. Also, the development of the market confirms that need. The 2015 report of *International Federation of Robotics* (IFR) states that: "Handicap assistance robots have taken off to the anticipated degree in the past few years. In 2014, a total of 4,416 robots were sold, up from 699 in 2013 - an increase of 542%. This increase is partly due to a more complete coverage. Numerous national research projects in many countries concentrate on this huge future market for service robots. In contrast to the household and entertainment robots, these robots are high-tech products.". [76].

Walking is the basis of human mobility and probably the most important competence for an independent and fulfilled life. Therefore, since more than two decades, there is very active research on smart walking assistive devices [114]. Such devices are usually designed as wheeled walkers and called Smart Walkers (SWs). Additionally to physical support, Smart Walkers provide sensorial and locomotion assistance for guidance, health monitoring, and cognitive support, e.g., medication reminders. They should compensate impaired users' weaknesses to control their (lower) limbs and increase their confidence and safety [110].

The Smart Walkers are still a topic of research[1] and, unfortunately, rarely used outside research labs. Nevertheless, some devices with specialized purposes are on their way to the users, if not utilized already. For example, ASBGo++ Smart Walker, designed to rehabilitate Ataxia patients, left the research facility after six years and will be commercialized [4]. The second example is GUIDO Smart Walker, walking assistance for

[1] Detailed overview on SWs' research is given in section 2.3.

visually-impaired older adults for use in elderly-care facilities [90, 134]. The GUIDO is, or at least was, commercially available. The Smart Walkers' limited success is probably due to their high price compared to conventional wheeled walkers. Another example of a fascinating but unsuccessful commercial SW is the LEA Smart Walker [136], which tried to tackle a broad spectrum of functionalities for daily use. Therefore, it was costly. This thesis focuses on a device used in a controlled environment for a particular use.

The knowledge about the influence of physical and cognitive activity on the overall state and independence of persons with mild cognitive impairment on the one side and the possibilities of smart autonomous assistive devices on the other side raise the following question: "Is it possible to use these devices for targeted activation of cognitive reserves of people with MCI, or elderly in general?". With this question in mind, the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics from Karlsruhe Institute of Technology (KIT) and Central Institute of Mental Health Mannheim from University Heidelberg started an interdisciplinary project called *"Technical system for physical activation of people with early-stage dementia"* [156]. The project investigated "how" and "if at all" robot-based mobile training systems could be beneficial in a therapeutic context for people with mild cognitive impairment. The main challenges were, first, to investigate if suitable training can be developed and how users accept it, and second, what is an appropriate design and which functionalities are required. This project serves as the groundwork for this thesis with details presented in chapter 3.

The results of the pilot-study showed the feasibility of utilizing such a training device with measurable influence on the participants' performance in interacting with a prototype device – the RoboTrainer Prototype. The device's mechanical design was not optimal for the targeted scenario and cumbersome in some situations. Further, another relevant issue was insufficient stability and safety of the used mobile robotic base. The available literature mainly presents devices designed for walking assistance but not suitable for training where higher forces, up to 200 N, are expected. Also, considering the training flexibility, an omnidirectional device is the only option regardless of mechanical and control complexity, whereas most commercial devices and devices from literature have differential kinematics [148]. A force-based interaction with a Smart Walker is the most intuitive [114]. Therefore, most devices use a force-torque sensor to capture the users' input. There is very little information regarding Smart Walkers' safety concepts, especially when a device influences users' behavior by acting with forces against their intentions. Those situations need safety-certified concepts, which are mostly missing in the literature.

When focusing on Smart Walkers (SWs)' control, the suitable method depends on the interaction approach and available sensors. In the case of force-torque sensors, open-loop admittance-dynamic is used to generate the device's velocities. The main drawback of this approach is that the users' physical effort depends on the controller's parameters, and considering the different physical fitness of users, their experience can vary enormously.

This was also confirmed in the pilot-study. An approach to reduce users' effort when strolling with a SW decreases the controller's damping factor linearly with the device's increasing velocity [185]. The disadvantage of the approach is longer stopping distance, which can be dangerous. Also, the approach does not change the device's dynamics for different users. In the literature on SWs, there is no sensible strategy found to parameterize an admittance-controlled SW for individual users' needs. Another controller related challenge is the physical coupling between device and user, resulting in oscillatory behavior if the user changes the stiffness of his arms. Here, the literature proposes to use an energy monitor to detect such situations and to damp users' input if needed [27]. Concerning RoboTrainer, performance of this method was not satisfactory.

The virtual force fields (VFFs) are the most common concept to influence the Smart Walker's behavior with control software [114]. Usually, Smart Walkers generate VFFs from their sensors online. Afterwards, they are superposed to the user's input ahead of the device's controller [29]. When considering training with the SW, a VFF representation is needed in advance to understand the functioning. The RoboTrainer Prototype uses a SLAM algorithm to build a representation of its environment, i.e., map and generates a VFF environment from a training description. When utilizing the virtual force fields as done in literature, open issues are dangerous situations caused by VFF override of user commands and device's movement without users' intention. As proposed in literature [29], using a passive approach is not feasible for active and challenging training. Therefore, novel methods need to be investigated.

1.2. Main Contributions and Goal of this Thesis

During the last decades, significant research efforts have been spent investigating how robotics can support older adults' mobility. Despite the considerable achievements, it is unclear if and how walking assistive devices equipped with robotic technology can be applied for targeted training to preserve and potentially improve users' physical and mental state. This thesis aims to explore the technical perspective for active training with a Smart Walker (SW). The proposed ideas are intended to examine requirements and functionalities for a suitable device and their influence on the training.

Multiple research questions arise when imagining active training with an autonomous robotic device:

1. How can training with a mobile robotic device look like and which form of human-robot-interaction should be used?

2. How should a prospective device be designed and which mechatronic properties are relevant for physical support, intuitive interaction and user safety?

3. How to control such a device, and how individual human-robot interaction parameters, e.g. users' fitness, can be accommodated?

4. How to achieve variability of training to support various complexity levels and provoke users' engagement.

This thesis makes four contributions to address these questions and realize combined strength and coordination training (cf. figure 1.1). The proposed training can even challenge the physical fitness and motor skills of younger and healthy persons.

The first contribution is proof of concept, including a conclusion on training feasibility and device capabilities. The initial concept presents the training design, a prototype device called RoboTrainer Prototype, and proposes a method to estimate users' performance and training complexity. The training utilizes the device's localization capabilities to compare user's interaction with the ground-truth and its actuation to provide unexpected disturbances during the exercises. Although the fundamental concepts are shown in literature, this kind of specific combination of purposeful challenges is not investigated yet. The following contributions exploit a pilot study's experiences, where ten older adults evaluated the training-concept and the prototype device.

The second contribution is a mechatronic design of a novel three-wheeled SW-like device for the active training. The device – called RoboTrainer v2 – addresses mechanical, structural, and safety drawbacks of the RoboTrainer Prototype. It also considers relevant standards and norms, implementing them if they do not limit functionalities required to realize the training. Besides the mechanical handle-height adjustment, common in many Smart Walkers, RoboTrainer v2 allows modification of its footprint and, thus, modification of users' support area. The modification changes the position of the device's rear wheels in two degrees of freedom. Such an approach is not found in any Smart Walker known from literature. Regardless of this flexibility, RoboTrainer v2 has a considerably more robust structure than RoboTrainer Prototype. From a safety perspective, RoboTrainer v2 implements many measures to protect users from injuries with the help of high-performance safety hardware.

The third contribution are the control concepts for SWs in active training. The contribution proposes novel and extends existing methods for individual and adaptive SW's control. First, a detailed insight on the concrete case of physical human-robot interaction is given. It provides an extended model that reveals a connection between users' sensorimotor performance and the user-walker system's stability. These insights enhance state-of-the-art by avoiding oscillations in user-walker interaction. The comprehensive approach also considers users' specific intentions when commanding a SW. The control approach integrates a novel non-linear velocity-based input-force adaption enabling users' effort regulation when moving at a certain speed. This is done to support agile training scenarios with young, healthy adults. A superordinate process implements

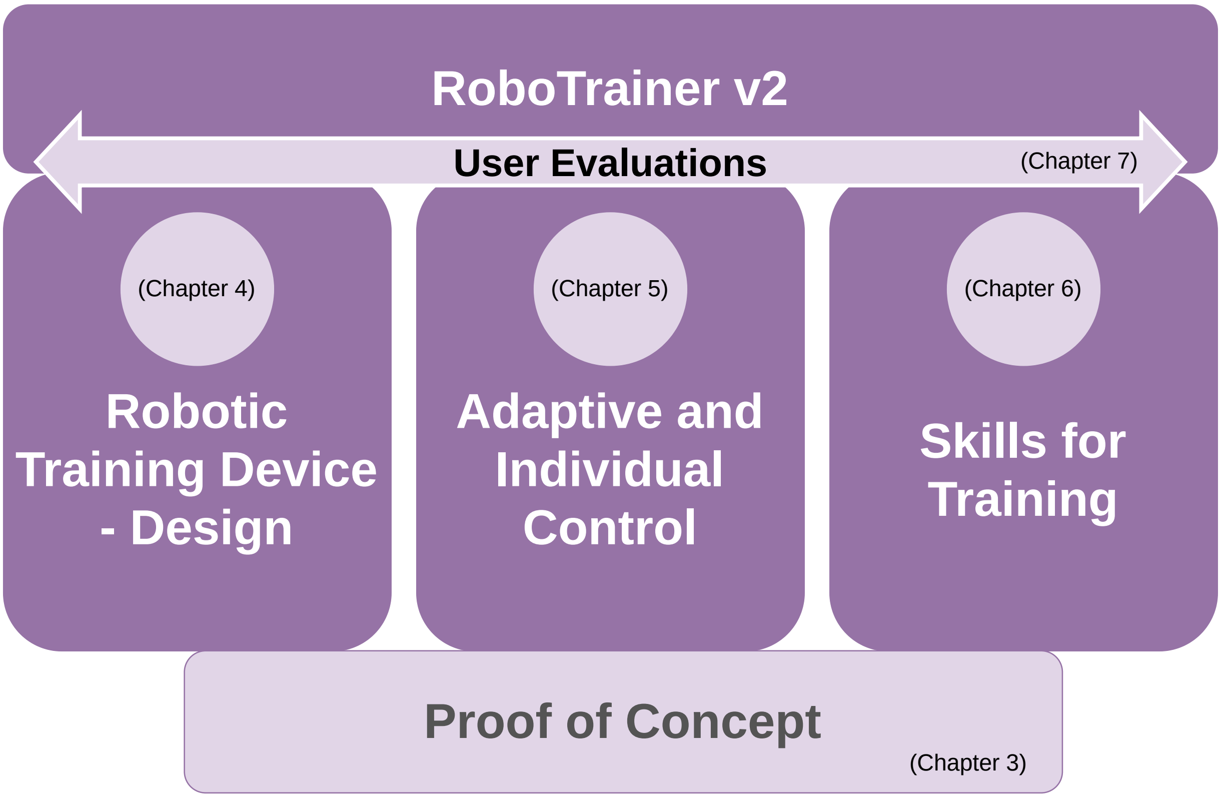


Figure 1.1.: This thesis proposes a robot device for individual strength and coordination training. The endeavors start in chapter 3 with the proof of concepts developing a prototype device and testing the interaction skills in a pilot study. Chapter 4 proposes a design of a novel device specifically for training purposes. Next, adaption and individualization methods for device control are discussed in chapter 5. Chapter 6 focuses on presenting skills for active and manifold training. Those three main pillars confluence together in a novel device for research of physical human-robot interaction for training with adults, called RoboTrainer v2. Chapter 7 presents user evaluations of the whole concept.

environment-independent and easy-to-use methods for automatic controller parameterization in order to realize per-user individualization. Those methods adjust the parameters of the main admittance controller and the limits of velocity-based adaption.

The fourth contribution are behavior modifiers used as building blocks for training, called control actions. Those modifiers are using the virtual force field-concept to create a potential field representation of the training environment. This potential field representation influences the Smart Walker when moving through it. Depending on how they influence the device, control actions (CAs) are categorized into two classes: global control actions (GCAs) and spatial control actions (SCAs). The GCAs equally modify the SW's behavior within the whole training environment and SCAs do this locally, in predefined places. Similar approaches are used in literature for SW's collision avoidance and navigation assistance. The difference is that the CAs define static virtual training environment and do not rely on sensor measurements. Using CAs in the controller as proposed in literature might have inconvenient effect on SW's behavior resulting in dangerous movements for the user. Therefore, two additional methods, called *passivity* and *safety*, are introduced to prevent such situations.

The contributions are evaluated separately from each other and in focused experiments with multiple users to acquire insights regarding their overall influence on the user-walker interaction.

1.3. Outline

This thesis is further organized as follows: In the first part of the next chapter (chapter 2), an overview of related work regarding conventional assistive devices, Smart Walkers, and physical and cognitive activation is given. In the second part of the chapter, individual SWs' characteristics, i.e., design, control and high-level interaction, are specified and discussed. Chapter 2 ends with a short overview of the control theory regarding admittance control. Next, chapter 3 presents the training concept, the RoboTrainer Prototype, and its evaluation in a pilot study with ten older adults having mild cognitive impairment. The chapter is rounded off with a discussion about drawbacks and possible improvements to the training approach and the device. Chapters 4 to 6 offer a detailed overview of this thesis' main contributions regarding design, control, and behavior modifiers for RoboTrainer v2. Accomplished user studies are presented in chapter 7, displaying study protocols and their results concerning the contributions. Finally, the thesis finishes with a recapitulation of the main achievements and an outlook in chapter 8.

1.4. Nomenclature

This section explains the common terms and wording used throughout the thesis. Compared to the publications, some terms were renamed and, therefore, also mentioned in the following list.

RoboTrainer The name *RoboTrainer* is short for *RoboTrainer Prototype* (cf. figure 3.2) or *RoboTrainer v2* (cf. figure 4.1). Its concrete meaning depends on the context. However, this does not make any difference in most cases since both devices' core functionalities are the same. Also, sometimes *RoboTrainer* generally describes a group of *Smart Walkers* with similar functionalities, and these thesis concepts would apply to them.

RoboTrainer Prototype The device also had the following names in related publications: *RoboTrainer Prototype*, *RoboTrainer v1*, *Heika device*, *prototype device*.

control actions (CAs) The term *control actions* describe RoboTrainer's behavior modifiers used to configure individual training tasks. The terms *control concepts*, *high-level control concepts*, *training building blocks* have the same meaning. The term *control actions* is introduced recently in [155] and therefore used only in the most recent publications. Before that, terms like *high-level control concepts* and *control modalities* were used.

input force – interaction force The terms "input force" and "interaction force" usually mean the same. The only exception is when observing the user-walker interaction in detail in section 5.2. In this case, "interaction force" is composed of the user's "input force" and disturbance caused by the RoboTrainer's movement.

user – participant The words "user" and "participant" are used for a person using RoboTrainer. Usually, the word "user" is used. These two words have the same meaning.

task – exercise The words "task" and "exercise" have the same meaning in the context of RoboTrainer's evaluations. They represent the smallest training unit defined by a specific users' assignment and a specific, usually unique, device's configuration.

repetition – trial – attempt The words "repetition", "trial", and "attempt" have the same meaning in the context of RoboTrainer's evaluations. They represent the smallest unit in the measurement context, i.e., one execution of a task.

Vectors and Matrices The bold font in formulas, e.g., $\boldsymbol{F}$, underlines that a variable is a matrix or vector. If not explicitly written, the matrixes' size is 3×3 with elements on their diagonals only, and the typical column-vector length is 3. In general, the RoboTrainer's degrees of freedom DOFs are considered independent of each other if not stated differently.

Equations with element-wise vector and matrix calculations Many equations consider independent calculation for each degree of freedom separately. Those are written in matrix form to emphasize their multidimensionality but represent element-wise calculations, i.e., they should be read as matrices of equations. For multiplication, the "$*$" operator underlines this, e.g.,

$$\mathbf{F} = \mathbf{M} * \mathbf{A} \text{ represents } \begin{bmatrix} F_x \\ F_y \\ F_\omega \end{bmatrix} = \begin{bmatrix} M_x \cdot A_x \\ M_y \cdot A_y \\ M_\omega \cdot A_\omega \end{bmatrix}$$

The equations with matrices and fractions should be read as follows

$$\mathbf{T} = \frac{\mathbf{M}}{\mathbf{D}} \text{ represents } \begin{bmatrix} T_x \\ T_y \\ T_\omega \end{bmatrix} = \begin{bmatrix} \frac{M_x}{D_x} \\ \frac{M_y}{D_y} \\ \frac{M_\omega}{D_\omega} \end{bmatrix}$$

See also section 2.5 for more details.

2 Background and Related Work

The application of robotics technology to mobility assistance devices gained traction in research around twenty-five years ago. The first research cases were robotized canes [16] and walkers [92] to assist older adults or visually impaired persons. Since then, technology has made a giant leap in miniaturization and computational power. However, the broader use of robotized assistance systems is still missing. The main reasons for this are technological complexity and the high costs of those devices. Still, the last decades provide rich knowledge gained from research in aging, physical and cognitive impairments, and influence and use of technology to enhance the lives of older people and those with impairments.

This chapter provides an overview of conventional and robotized assistive devices. The presented research and technology are used as groundwork and inspiration for this thesis. The chapter begins with a short portrayal of medical research on gait dysfunction and its relation to mild cognitive impairment and mental state. Section 2.1 is rounded off by explaining the concept of neuromuscular training and its implementation in research. This section broadens the motivation chapter and characterizes the purpose and long-term goals of the RoboTrainer.

An overview of conventional assistive devices for use by elderly persons is given in section 2.2. Section 2.3 overviews the most relevant robotized assistive devices, their features, mechanical designs, and control possibilities. Those are mainly based on a wheeled walker and therefore called Smart Walkers (SWs). After a per-device overview of research (section 2.3.1) and commercial (section 2.3.2) devices, the comparative overviews from design- (section 2.3.3), control- (section 2.3.4), high-level features- (section 2.3.5) and user's tracking- section 2.3.6) perspective is given. Each of those sections closes with a short description of RoboTrainer's properties in the corresponding field. Section 2.4

gives an overview of the most relevant functionalities of the RoboTrainer. Those functionalities are mapped to the most relevant Smart Walkers from literature in table 2.1. Chapter 2 concludes with a short overview of the control theory regarding admittance dynamics used in this work (section 2.5). In appendix section A, a tabular overview of the relevant SWs, their features, corresponding publications as well as design and control properties are given.

2.1. From Gait Dysfunction and mild cognitive impairment to Neuromuscular Training

The main clinical reason for using assistive mobility devices is gait dysfunction [21, 114]. A gait dysfunction is indicated by a decrease in speed and stride length, which increases efficiency in body motion[1], and maximizes balance and stability [114]. According to Duxbury [42], this is a natural way to reduce fall risk. Gait dysfunction and mobility disorders, in general, are often perceived as an inevitable consequence of aging. However, they are instead a reflection of the presence of an age-related disease or a multitude of such [147]. Many gait disorders have their source in cognitive diseases or impairments. Those are caused by loss of safety and efficiency of walking, which rely on the sensorimotor system and interaction between executive control (actions), cognitive (e.g., navigation, perception), and affective dimension (e.g., cautiousness, risk-taking) [147]. Especially dementia, as a type of aging-related cognitive impairment, is associated with a radical decline in sensory-perception and motor skills, as well as with worsening of cognition and memory functions [149]. The term “mild cognitive impairment (MCI)” is used in neuropathology for impairments less severe than dementia. MCI’s associated cognitive decline might be less irreversible and better respond to treatment [98].

Another issue of aging is a loss of muscle mass. This process starts at the age of 40, and, until the age of 80, people lose 30 % to 40 % of their muscle strength [100, 1]. This, again, leads to loss of motor performance in everyday life and increases the risk of chronic diseases and, finally, to a loss of autonomy and quality of life [54].

A common situation where a person’s confidence in walking becomes visible is walking while performing a secondary task. Lundin-Olsson, Nyberg, and Gustafson [102] were the first who investigated this and introduced the “Stops walking when talking”-parameter as a predictor of falls for elderly people. Hyndman and Ashburn [72] tried to use the same measure to predict falls in people with stroke, but they questioned its use as a sole indicator. Nevertheless, the authors state that persons who stopped walking were significantly more disabled[2] and dependent in daily living [72]. Irrespective of this

[1] Using less energy per stride and less force to decent on any particular joint.

[2] The overall sample presented mild to moderate cognitive disabilities and no impairment.

research, Camicioli et al. [21] showed that probable Alzheimer's disease (AD) patients also slow down when talking, i.e., executing another task. Except for those situations, called "dual-task", Dodge et al. [39] show that decreasing walking speed and its daily variability may be an early marker for mild cognitive impairment.

The above-presented research raises a question: is there a possibility to stop or at least slow down the mentioned effects without medication, or how can medication be supported best. Promising solutions are neuromuscular activation programs. A 12-week dual-task training program with 61 seniors with dementia and a mean age of 81.9 years using a randomized controlled study is presented by Schwenk et al. [140]. The results show significant improvement in dual-task performance under complex conditions and minor improvement in less challenging tasks. Schwenk et al. [139] show that specific and intensive training can significantly improve dementia patients' motor performance. In general, a positive impact of a specific exercise and occupational training for a person with mild cognitive impairments, dementia, and Alzheimer's disease is shown in many studies, for example, by Hofmann et al. [70], Heyn, Abreu, and Ottenbacher [62], Kattenstroth et al. [86], and Bowes et al. [17].

When considering neuromuscular training with elderly persons with mild cognitive impairment (MCI), Alzheimer's disease (AD), or Parkinson's disease (PD), significant impacts on the patient's gait are expected. Those impacts are usually measured by investigating speed, stride time, and length of usual gait, endurance gait, dual-task performance, or the Time Up & Go (TUG) test. If the training utilizes obstacles, the time for obstacle negotiation is also relevant [131]. Various clinical measurement methods are used in the literature to evaluate the severity of cognitive impairment and the patient's suitability for the training, e.g., Trail Making Test A and B, Montreal Cognitive Assessment (MOCA), Consortium to Establish a Registry for Alzheimer's Disease (CERAD-Plus) [121], Cambridge Neuropsychological Test Automated Battery (CANTAB), and Wechsler Memory Scale – Revised (WMS-R) [177].

Recent studies, like [119], [141], and [149], evaluate combined cognitive-motor training programs using focused attention on elderly persons with MCI. Sooyeon [149] compares a combined cognitive-motor learning program with a focused attention learning program. The intervention group (IG) showed a significant learning effect compared to the control group (CG) on Mini-Mental States Examination-Korea (MMSE-K) and Time Up & Go (TUG) tests. In the two other tests, *flexibility* and *pegboard*, there was no difference. The authors conclude that cognitive-motor training with focused attention can improve perception and registration of incoming sensory stimuli, and refinement of motor response and movements. Schwenk et al. [141] use sensory feedback in motor training with persons with MCI. The users needed to fulfill specific tasks on the monitor by moving their lower limbs equipped with inertial sensors. The training significantly reduced fear of falling and improved balance, i.e., the center of mass sway was reduced during standing with opened and closed eyes. The authors concluded that the training is well accepted in the target population and is beneficial for postural control. Mirelman et al. [119] presented a concept of using virtual reality technologies for gait training. They deployed intensive treadmill training combined with visual obstacles in a virtual reality world projected on the wall in front of the patients. Each patient received 18 training sessions over six weeks. The results show that patients' gait speed under normal and dual-task conditions can significantly improve cognitive and functional performance. The functional improvement was still measurable four weeks after the training.

In an aging society, many persons inevitably develop physical or cognitive impairment which presents a significant burden for the person itself, its closest relatives, and society as a whole. For many years, these diseases have been researched regarding their development and treatments with or without medication. Nevertheless, most of the treatments are resource-intensive, especially in terms of personnel. Therefore, they are hardly usable in our current elderly care. Moreover, many measuring techniques to investigate a patient's state can be improved and automated using novel technologies, like arrays of inertial sensors or even more complex devices, e.g., Smart Walkers. The impact of dual-task and focused attention training is very promising, so that using novel technologies for those are worth investigating. The technology could provide older people with a sufficient amount of training and at the same time, measure their condition for diagnosis purposes and adapt intensity and complexity. This would give more time to caregivers for precious social contact with their patients. Therefore, the main question is how such technology, i.e., robotic systems, should be designed and used.

2.2. Conventional Assistive and Training Devices for Mobility

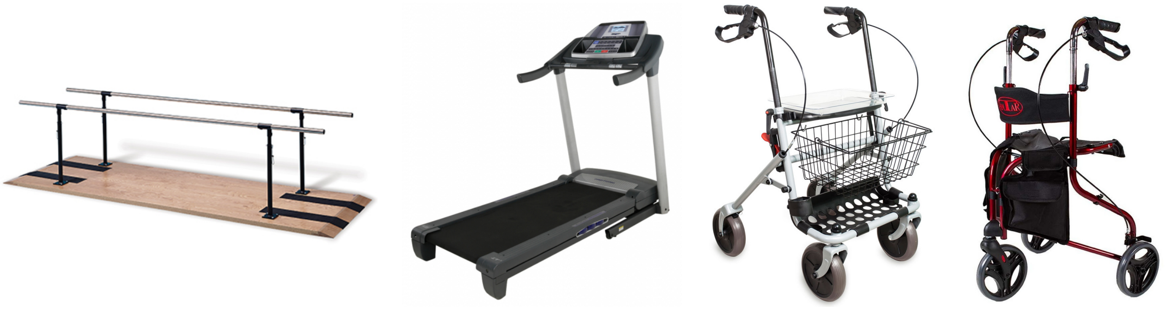
An overview of assistive mobility devices is given by Martins et al. [114]. The authors classify assistive devices into two groups, alternative devices that replace total or partial mobility and augmentative devices that support and advance a person's residual mobility capacities. An example of an alternative device is a wheelchair, and the primary purpose of such devices is to restore a person's mobility. The alternative devices could cause health problems, like loss of bone mass, degradation of blood circulation, or skin sores caused by the user's sitting position [91]. Therefore it is always better to use augmentative devices if possible [114, 91]. The augmentative devices are mobility-training devices, e.g., parallel bars, treadmill-training devices, self-ported devices, e.g., orthoses and exoskeletons, and external devices, like canes and walkers.

The most basic mobility training device are parallel bars (figure 2.1a) [114]. They provide good rehabilitation results and can be used ambulatory. However, the patients have to support themselves with upper limbs, and two or three therapists are usually involved in the training. Moreover, parallel bars limit the natural arm swing, essential for gait stability and efficiency [127]. Treadmill (figure 2.1b) training is used to improve walking capabilities, like with patients suffering from incomplete spinal cord injury. At the beginning of this training, a therapist helps the patient move his legs on the treadmill [34], which significantly burdens therapists. The essential part of motor training and rehabilitation are the repetitive movements, leading to high time- and personnel- expenses.

Scientific literature proposes to use robotic devices like orthosis, i.e., lower-limb exoskeletons [34], or specialized devices, as presented in [127], to overcome those central issues. Colombo et al. [34] developed a size-adjustable orthosis to support the patient's lower-limb movement when doing treadmill training. Novandy, Yoon, and Manurung [127] present a novel robotic system for the patient's support during gait rehabilitation. The device provides partial body-weight support, excitation for natural arm swings, and simulation of different ground properties, e.g., slopes. Still, the main issue in gait therapy is to keep patients engaged during training. Otherwise, they may lose interest, become passive, and may not want to participate anymore [103].

According to Martins et al. [114], walkers are relevant assistive devices in practice because of their simple use and rehabilitation potential. The persons use their locomotion abilities, avoiding the immediate deteriorative use of alternative mobility devices [91]. The walkers are usually prescribed to improve patients' mobility and maintain balance [35, 12]. The goal is to increase patients' confidence, sense of safety, activity, and independence level.

For this thesis, a subgroup of walkers with three or four wheels is interesting. These types of walkers are called *wheeled walkers* or *Rollators* (figures 2.1c and 2.1d). Most patients prefer those walkers because they provide the most natural gait patterns and are usually equipped with a shopping basket and a resting seat [35, 104]. Since wheeled walkers are easy to use and drive, they are helpful for elderly who are in reasonably good physical shape. Even though they have squeezing or releasing brakes to stop the device, users have to learn how to use them properly [91]. When used by persons with significant balance problems or cognitive impairment, they may roll away and increase fall risk [12, 104]. If a person with low grip strength, speed or agility is using a rollator, it is recommended to use a padded bar across the front of a walker, which initiates braking if pressed or if a user falls onto it [35]. Also, there are other situations where the fall risk increases, e.g., walker wheels can catch the user's foot or catch objects like carpets, furniture, or door frames, and a walker could slip out of the user's hands and even tip over [107, 12, 23]. Despite their potential benefits, 30 %–50 % of people do not use their walkers shortly after receiving them [137]. The literature states a multitude of reasons for that. According to Martins et al. [114], the following reasons for not using wheeled walkers are the most prominent: (1) need for more maneuvering space - compared to canes; (2) difficulties on carpets; (3) difficulties in crossing obstacles; (4) results in a slower gait speed; and (5) requires considerably more energy and cardiovascular fitness.



(a) Parallel Bars. Source: Web[3] **(b)** Treadmill. Source: Web[4] **(c)** Rollator 4-wheeled. Source: Web[5] **(d)** Rollator 3-wheeled. Source: Web[6]

Figure 2.1.: Examples of conventional assistive and training devices for mobility.

Lindemann et al. [101] focus their research on an evaluation of general problems older people have when using wheeled walkers and on specific problems with opening a door against walking direction. The authors report that the main issues when using a wheeled walker are walking down- and uphill, walking on uneven ground, and obstacle crossing. Other minor problems are not-rotatable back wheels, walking through the door, and turning on the spot. Also, a major problem was that elderly persons often forget to fix the brakes. Their results show that walking through the door was faster without a walker, but walking speed, in general, was faster with a walker. Lindemann et al. [101] conclude that using an omnidirectional smart walker, i.e., a robotized wheeled walker, could increase the quality of life for many older people. The main help would be environment recognition, e.g., doors, obstacles, or slopes and give support in those situations.

To summarize, using robotic technology could resolve the issues with conventional training and assistive mobility devices. When considering rehabilitation, exoskeleton-based devices solve the issue that multiple therapists are indispensable at the beginning of the training. Many modern exoskeletons have interfaces to other technologies like augmented or virtual reality, making the training more attractive. A comprehensive and up-to-date overview of exoskeletons is provided on the "Exoskeleton Report" website[7]. The literature shows that conventional wheeled walkers have many issues like bulkiness, nav-

[3] Hausmann Industries: `https://www.hausmann.com/product_pages/t109_pb7175/Model_1300.html` (accessed Nov. 4, 2020)

[4] Fitness Superstore: `https://www.fitness-superstore.co.uk/pro-form-700-zlt-folding-treadmill-northampton-ex-display-model.html` (accessed Nov. 4, 2020)

[5] AktivShop: `https://www.aktivshop.de/rollator-rfm-standard` (accessed Nov. 4, 2020)

[6] RehaTechnik Löesch: `https://rehatechnik-loesch.de/Antar-Gehrad-3-Raede` (accessed Nov. 4, 2020)

[7] `https://exoskeletonreport.com/`

igation limitations, or breaks which prevent efficient interaction and increase the user's risk of injury. Therefore, adding robotic technologies to conventional walkers could increase safety and value for a user. These technologies enable high-level functionalities like health monitoring, guidance, or communication tools. They can support locomotion and provide greater stability during walking, especially on slopes. The following section presents an overview of Smart Walkers in research and industry trying to realize those high-level functionalities and bypass the conventional wheel walkers' shortcomings.

2.3. Smart Walkers – More than Physical Gait Assistance Devices

The term *Smart Walker (SW)* is used in literature for a wheeled walker equipped with sensors and actuators for functionalities beyond physical support. Smart Walkers (SWs) are utilizing robotics knowledge to provide better body weight support [91, 51, 159, 110], gait assistance [122, 69] and monitoring [6, 22], and collision avoidance and navigation [105, 174, 120]. The research on Smart Walkers was started by Lacey and M. Dawson-Howe [91], where a novel robotic mobility aid for the elderly blind, called PAM-AID, is presented. The predecessor devices were *Smart Canes* [16, 40] and *Smart Wheelchairs* [170]. Smart wheelchairs have the same issues as conventional alternative devices, e.g., loss of muscle mass. On the other side, the smart canes' main drawback is limited gait assistance and physical support. Since the first Smart Walkers were introduced at the end of the 1990s and early 2000s [91, 92, 125, 105, 185, 174], more than sixty (60) devices have been found in the literature. The general criteria were that the devices provide physical support and have integrated at least one sensor or actuator. Great resources to overview the field of SWs are review studies from Dune, Gorce, and Merlet [41], Martins et al. [114], Martins et al. [112], Solenne et al. [148], and Alves et al. [4].

Martins et al. [114] provide the first review on walking aids, focusing on SWs [148]. They analyze and classify SWs according to their functions, focusing on human-robot interaction (HRI). This classification is set up into five categories of functionalities defined by Frizera et al. [51]: (1) physical support; (2) sensorial support; (3) cognitive assistance; (4) health monitoring; and (5) advanced human-machine interface. The control of a SW depends on the used HRI. Therefore, [114] is also a valuable source to analyze used control methods in respective SWs. In their later work, Martins et al. [113] review motorized SWs regarding their interaction and navigation strategies, i.e., manual guidance, shared-control, and autonomous navigation. Additionally, they provide a comprehensive list of Smart Walkers' features for user-state monitoring and user safety.

When focusing on the design of SWs, Solenne et al. [148] classify thirty-eight SWs from literature into seven categories depending on their kinematics and wheels' setup. The authors provide guidelines for technological choices when designing SW, investigating them from the module-oriented perspective. So for each SW as a physical and functional device, the following elements are examined:

(1) locomotion;

(2) support and stabilization of the user (patient);

(3) cognitive support systems, e.g., obstacle avoidance;

(4) integration into daily life of users; and

(5) approaches for evaluation.

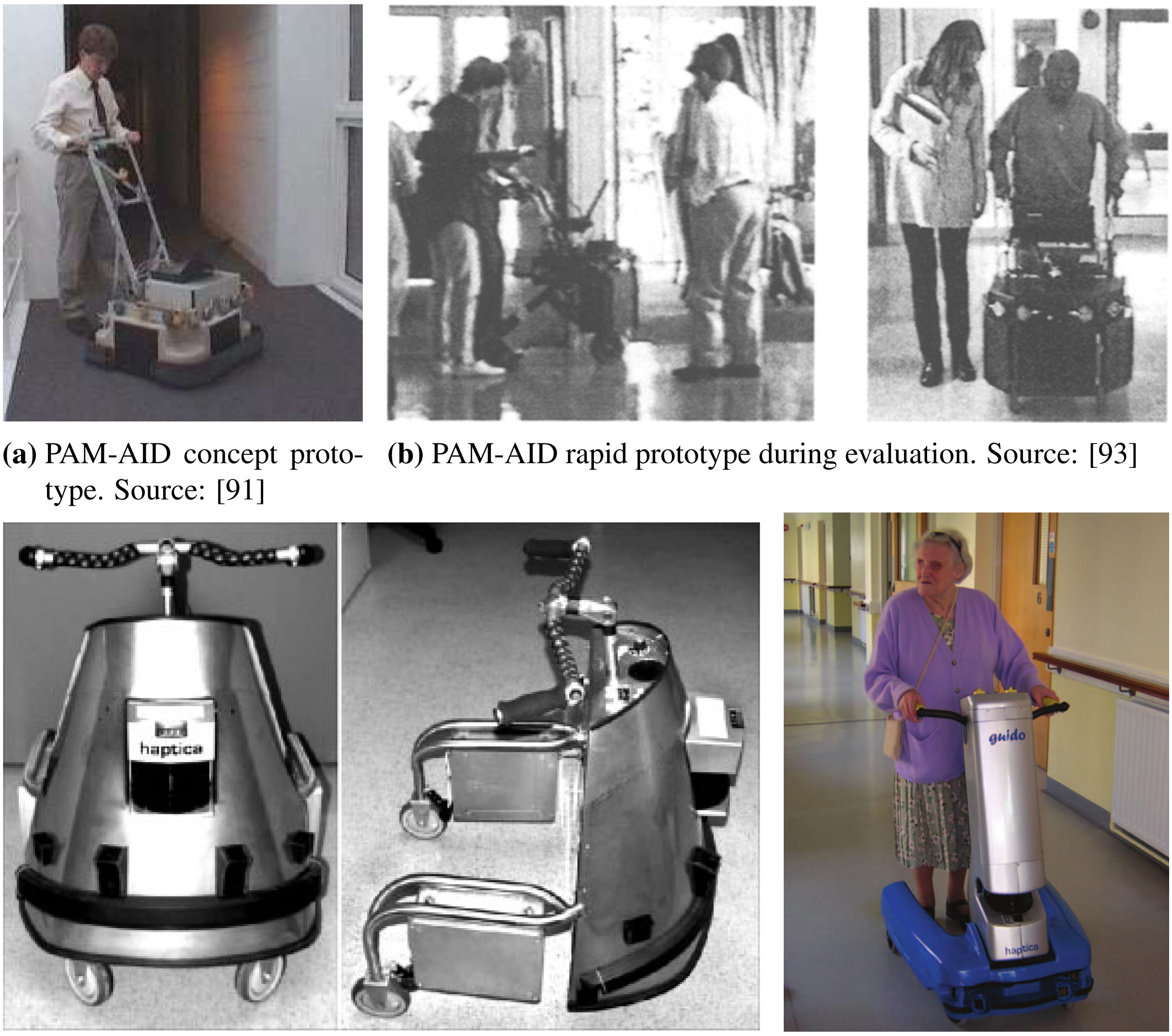
The first three modules provide criteria regarding mechanical design to achieve specific functionality. They include a description of SW's geometry, kinematics, sensors, and input interfaces. On the control side, Solenne et al. [148] focus on strolling, i.e., motion synchronization between the user and a SW, and methods to compensate for the user's loss of balance. The authors do not provide specific criteria for evaluating the control quality but suggest that the user-walker distance is crucial. Furthermore, the authors focus on obstacle avoidance methods utilizing different sensors, navigation, user-walker communication channels, monitoring users' physiological status, and long-term interaction. Finally, Solenne et al. [148] propose a three-step evaluation method for SWs: first with healthy persons, second with the target users, and finally in long-term studies comparing SWs against conventional walkers.

The remainder of this section presents the last thirty years of research on Smart Walkers. It gives examples for different types of them, beginning with the early days, when mostly passive devices were used, e.g., [105, 174, 65, 134], to the recently developed active, commercial devices, e.g., [4, 56, 55, 136]. This overview of representative devices builds on the analysis from Martins et al. [114] and Solenne et al. [148]. The following sections provide details on those devices, review their functionalities, design (section 2.3.3), and control (sections 2.3.4 and 2.3.5). Their influence on the development of the RoboTrainer v2 is discussed at the end of each section. Appendix section A contributes a tabular overview of the most relevant Smart Walker (SW) from the literature.

2.3.1. From Passive Navigation Assistance to Active Rehabilitation Devices

Personal Adaptive Mobility Aid (PAM-AID) devices developed at Trinity College in Dublin, Ireland, are the first broadly known Smart Walkers (SWs). Under the name PAM-AID, multiple prototypes were developed, resulting in the commercial *GUIDO* smart walker [134, 90]. The first concept prototype [91] was based on a *Labmate* mobile robot base with a handrail and joystick to control it (figure 2.2a). The development was evaluated in a laboratory environment with healthy interest groups, i.e., institution owners and professional caregivers, working at residential homes for visually impaired people. The second, also active, version [92, 93] modifies a commercially available rollator to provide physical support and obstacle avoidance (figure 2.2b). The device uses a force-torque sensor (FTS) as the input interface since a joystick may result in oscillatory behavior [92, 105]. The device was tested with eight (8) subjects from the target group. MacNamara and Lacey [105] present the first passive PAM-AID enabling two operation modes. The first is manual control with audio feedback about a potential collision with the environment. The second mode is an assistive mode for obstacle avoidance via steering the front caster wheels. During the evaluation, the users accepted very well the audio feedback with spoken messages [105]. The two commercial versions, Veterans Affairs Personal Adaptive Mobility Aid (figure 2.2c) [135] and *GUIDO* SW (figure 2.2d) [134], follow the passive approach, where only the steering is actively controlled. Both devices have an FTS to measure users' intention, i.e., torque, and multiple switches to change a control mode. The participants accepted the devices very well. Nevertheless, many of them would feel uncomfortable when using *GUIDO* in the presence of other people. The experimental results with visually impaired people show reliable collision avoidance, whereas general walking speed slows down when using *GUIDO* [134].

***Personal Aids for Mobility and Monitoring* (PAMM)** are also a series of robotic aids developed by the *Field and Space Robotics Laboratory* at Massachusetts Institute of Technology (MIT) to support elderly in assisted living facilities and delay their transition to nursing homes [40, 185]. Working with several assisted living facilities, the authors established the PAMM concept with goals on the system's performance regarding (1) potential users, (2) environment, (3) physical stability, (4) guidance and obstacle avoidance, and (5) health monitoring. Two systems, *SmartCane* [40] and *SmartWalker* [152, 150], were developed (figure 2.3). The cane configuration provides one-sided physical support and guidance to seniors living independently and in assisted living facilities. The main advantage of the cane is its small footprint, i.e., dimensions, and subsequent agility. To manage narrow spaces with the PAMM SW, Spenko [152] developed an Active Split Offset Castor (ASOC)-wheel used on the front side of the SW combined with passive caster rear wheels to provide omnidirectional kinematics. Both devices use six-axis FTS mounted between the user's handles and the device's frame. Yu, Spenko, and Dubowsky [185] state that using an FTS provides users with a natural and intuitive interface. However, the authors also state that using force signals may directly result in unstable motion due to signals' fluctuations, as shown by [57, 58]. Therefore, Yu, Spenko, and Dubowsky [185] provide a profound analysis of admittance-control for their devices and propose an adaptive approach to increase stability when the device is standing or moving at low velocities, and to reduce the users' effort when moving faster. This approach is used as the basis for velocity-adaptive control done in this thesis. More details can be found



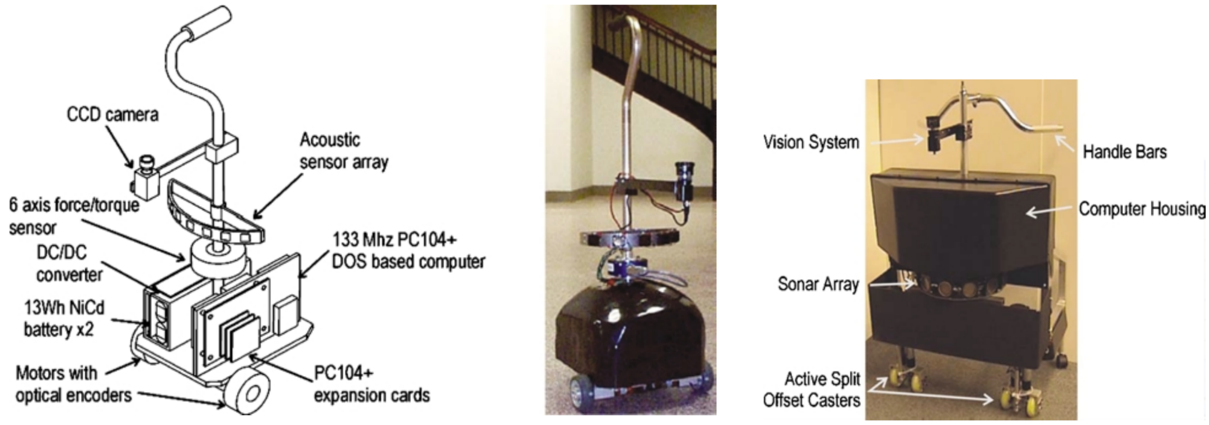
(a) PAM-AID concept prototype. Source: [91]

(b) PAM-AID rapid prototype during evaluation. Source: [93]

(c) Veterans Affairs Personal Adaptive Mobility Aid (VA-PAMAID). Source: [135]

(d) Guido Smart Walker. Source: [134]

Figure 2.2.: Research and commercial versions of PAM-AID Smart Walker.



(a) PAMM Cane Prototype. Source: [185]

(b) PAMM SmartWalker prototype. Source: [185]

Figure 2.3.: Versions of *Personal Aids for Mobility and Monitoring* (PAMM).

in section 2.3.4. The PAMM devices also implement multiple control modes: manual – user-driven admittance control; autonomous – device plans a path and leads the user along with it; and shared-control where the user can influence the pre-planned device's path [185, 151].

Medical Automation Research Center (MARC) or *CO-Operative Locomotion (COOL) Aide* [174, 71] is a passive SW modified from a conventional three-wheeled walker, adding a steering motor at its front wheel (figure 2.4). The device's primary purpose is to provide obstacle and stair detection and help elderly persons stay away from them. For this functionality, a laser range finder (LRF), infrared (IR) sensors, sonars, and encoders for the dead-reckoning were added. Huang et al. [71] apply the virtual force field (VFF) concept [87, 15] to influence direction of the shared-controlled MARC SW. This helps users to avoid obstacles using a heuristic logic controller based on the walker's dynamic model. The user's movement goal is determined from the interaction forces and moments measured on MARC's handles using probabilistic Dempster-Shafer Theory (DST) to calculate valid passages. The heuristic logic prevents the user and the controller from competing if they have different beliefs for the best path. The path change during shared control is done based on the walker's speed, i.e., lower velocity results in paths closer to obstacles. The MARC SW is evaluated on healthy subjects in a laboratory environment at different development phases [173, 7]. The RoboTrainer also uses the virtual force field (VFF) concept to modify its behavior based on the training setup (see chapter 6).

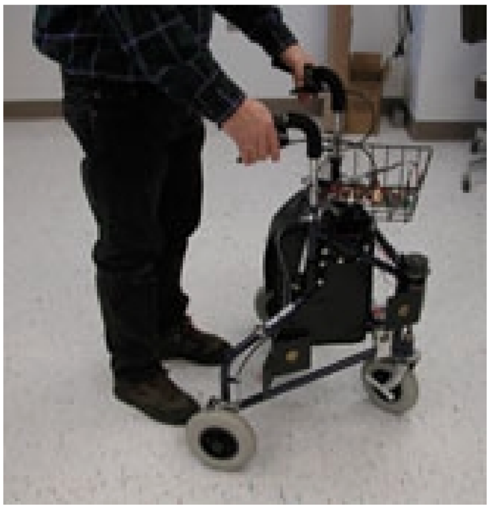
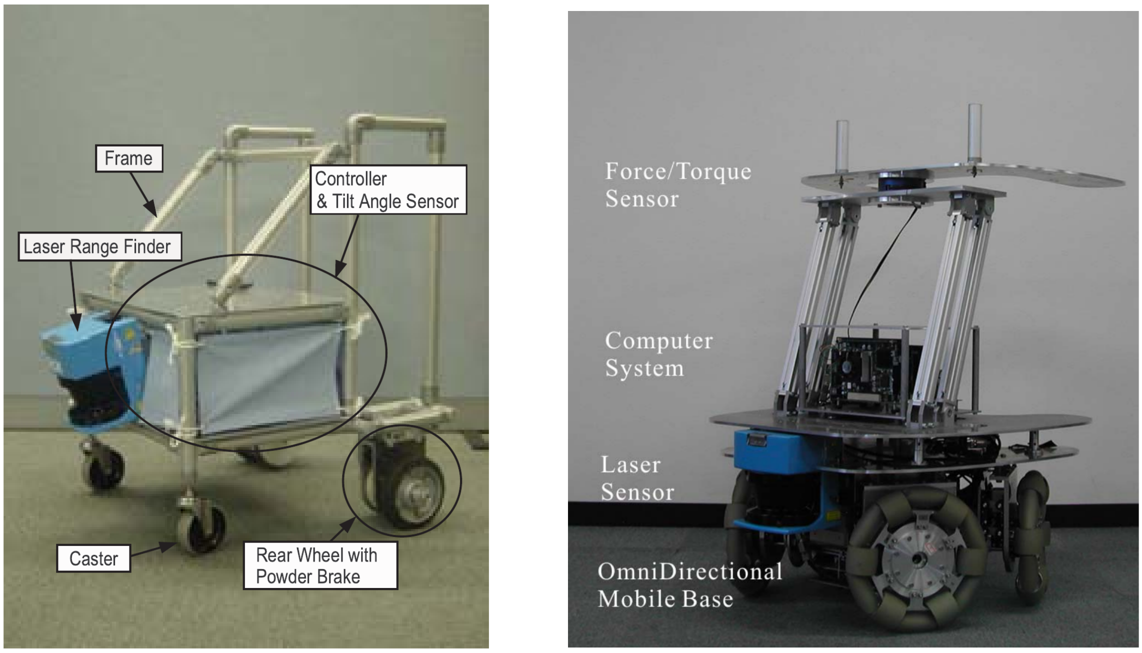


Figure 2.4.: MARC Smart Walker. Source: [174]

RT-Walker is a passive, break-controlled SW developed at Tohoku University, Japan (figure 2.5a). The walker is expected to help older adults, disabled, or blind persons to avoid collisions, dangerous situations like stairs and run-away or run-over situations on slopes [65, 64, 63]. Hirata, Komatsuda, and Kosuge [66] further introduce a fall prevention method using *RT Walker*. The method uses a laser range finder oriented vertically towards the user, measuring their vertical cross-section to calculate their center of mass. Unique about this SW are computer-controlled brakes on the two fixed rear wheels, which steer the walker's direction without adding energy into the user-walker system. Using this approach, the authors eliminated all safety concerns associated with active SWs. Collision avoidance uses an artificial potential field, i.e., the virtual force field (VFF) concept, to describe the environment. Hirata et al. [68] realized the walker's control using an admittance model with the variable dynamic. The SW prototype was evaluated in the laboratory experiments with healthy blindfolded persons [65, 63].

Walking Helper is an active, omnidirectional SW from Tohoku University, Japan (figure 2.5b). The device uses an FTS placed under the forearm support as the users' input interface. This concept is often implemented in SWs with forearm support, e.g., [51, 50, 110, 85]. Besides the pure role as input interface, detecting contact between a user and SW is also relevant from the safety perspective. The fascinating part of *Walking Helper* is not its design but its control. Oscar Chuy's work between 2004 [28] and 2007 [27] focused on passive SW's behavior and the adaption to an individual user by adjusting the walker's center of rotation (CoR). Chuy, Hirata, and Kosuge [26] propose training to detect dis-balance in the user's left and right arm when applying force to a SW, and automatic adjustment of the walker's CoR to compensate it. For this thesis, essential work was published in 2007 [31], where a concept of passive behavior for active



(a) RT Walker. Source: [64]

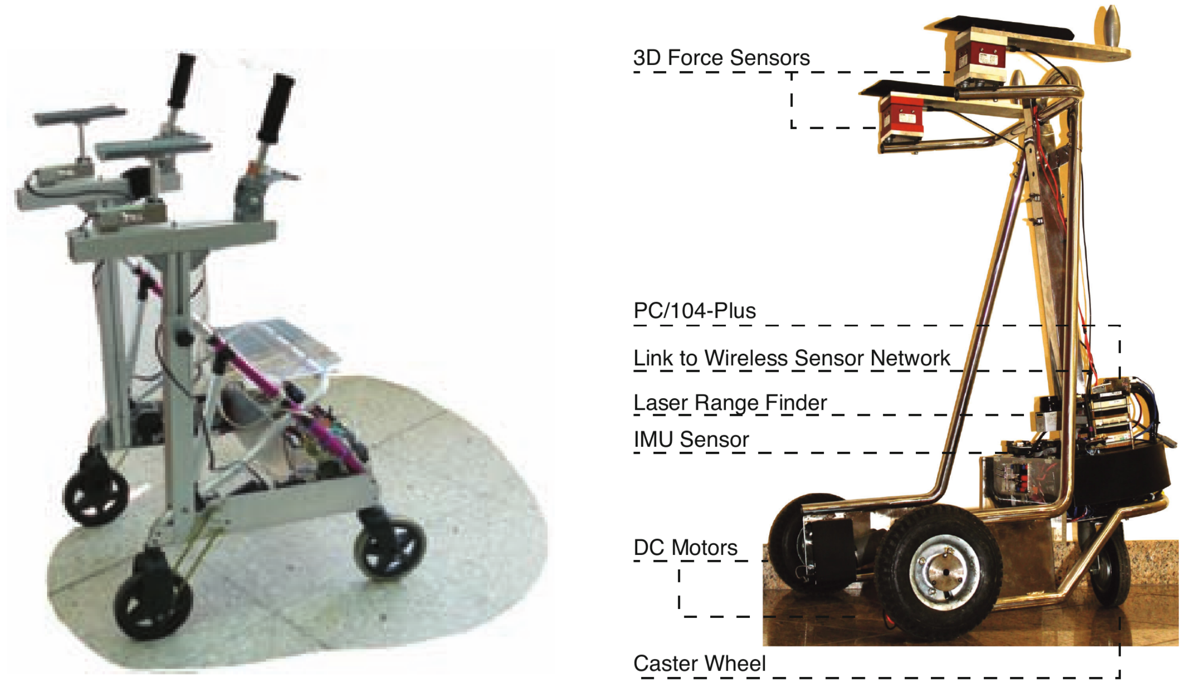
(b) Walking Helper. Source: [27]

Figure 2.5.: Smart Walkers developed at Tohoku University, Japan.

SWs is presented. This approach is adapted and used extensively to control RoboTrainer Prototype and RoboTrainer v2 (see chapter 5) and served as an inspiration to increase the control actions' safety (see section 6.2). Therefore, section 2.3.4 provides a detailed overview of the approach from Chuy, Hirata, and Kosuge [27].

SIMBIOSIS Walker is a four-wheeled walker with actuated fixed rear wheels and passive caster wheels in front (figure 2.6a). *SIMBIOSIS* was developed at the Instituto de Automática Industrial – CSIC, Spain, and uses force-loading cells in its forearm support to detect the user's input and presence [51]. Its primary purpose is to provide physical support and gait monitoring using force sensors, laser-scanner, and sonars [52, 53]. Much of the work with *SIMBIOSIS* focuses on estimating gait characteristics from force measurements on the walker's handles, as done by Alwan et al. [5] and Alwan et al. [6]. The walker's navigation commands are generated by a fuzzy controller, which applies the extracted user's intention from the force inputs after removing the gait-related force-oscillations. *SIMBIOSIS Walker* was clinically validated with eight patients at the Spinal Cord Injury Hospital of Toledo, Spain [53].

UFES Walker is the continuation of Anselmo Frizera's work with *SIMBIOSIS* Walker [51] at the Federal University of Espirito Santo (UFES), Brazil (figure 2.6b). Frizera et al. [51] present a new three-wheeled SW with differential kinematics and actuated



(a) SIMBIOSIS Walker. Source: [51]

(b) UFES Walker. Source: [50]

Figure 2.6.: Smart Walkers developed by Anselmo Frizera [51, 50].

non-rotatable rear wheels. *UFES Walker* is extended with an Inertial Measurement Unit (IMU) to determine slopes and walker's movement. Its primary purpose, aside from motion support, is biomechanical monitoring and gait assessment. Therefore, it provides wireless connections of external sensors placed on the user and shares their data over a cloud [94]. The control approach of *UFES Walker* is similar to those implemented in *SIMBIOSIS* Walker [33, p 57].

AZIMUT-3 Smart Walker is an omnidirectional, quasi-holonomic SW based on a mobile robot base (figure 2.7) that measures users' interaction forces through specially designed wheels [48]. The wheels, called *AZIMUT wheels*, are 2 DOF orientable wheels with axis shift. Each wheel's axis has an integrated torque-sensor, measuring external influence on the SW. Those measurements are loaded into the robot's direct-dynamics model to calculate external force, i.e., disturbance. This method makes the design of a SW somewhat simpler because input sensors are not needed. Nevertheless, as shown by the authors, the exact calculation of disturbance force is complex and, in some cases, even not possible [48]. *AZIMUT-3*'s primary purpose is to develop a natural interface for physical human-robot interaction (pHRI).

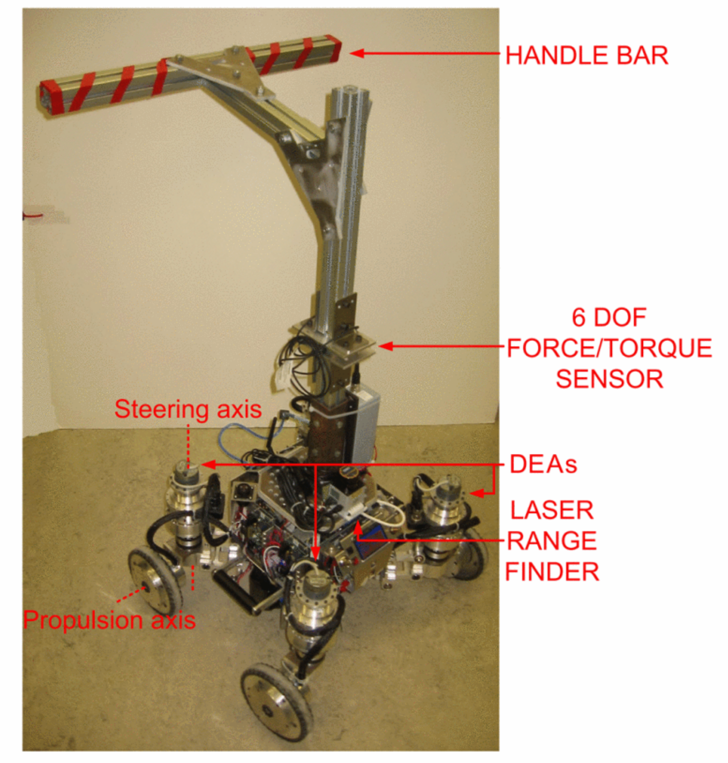


Figure 2.7.: AZIMUT-3 Smart Walker. Source: [49]

JAIST active robotic walker (JARoW) focuses on the natural, easy-to-learn, and simple-to-use interaction without manual or force sensor-based controls (figure 2.8) [97, 96]. The authors' central assumption is that elderly persons have slower reflexes, delayed reaction times, and are usually unfamiliar with electronic or mechanical controls. Therefore, the JARoW's interaction interface localizes users' lower limbs and calculates users' intent from their position. The first version of the algorithm uses infra-red sensors for users' leg detection [97]. In later works [95, 96], the authors use laser range finder (LRF) and Kalman filter-based tracking to determine positions more accurately. JARoW's control algorithm uses those positions to calculate the angle between users' legs and relative distances to the walker's frame in order to generate motion. Data from LRF are also used to build potential fields around obstacles in the environment to reduce the JARoW's velocity before a collision. Regarding the mechanical design, JARoW is unique. It has a custom-made circular frame with forearm support and the user positioned at its center (figure 2.8). Three omni-wheels assure omnidirectional movement. Lee et al. [96] evaluate the JARoW with five elderly subjects in controlled, everyday situations. The participants were mostly satisfied with *JARoW*'s functionalities and provided some ideas for future versions, e.g., the ability to ascend/descend stairs.

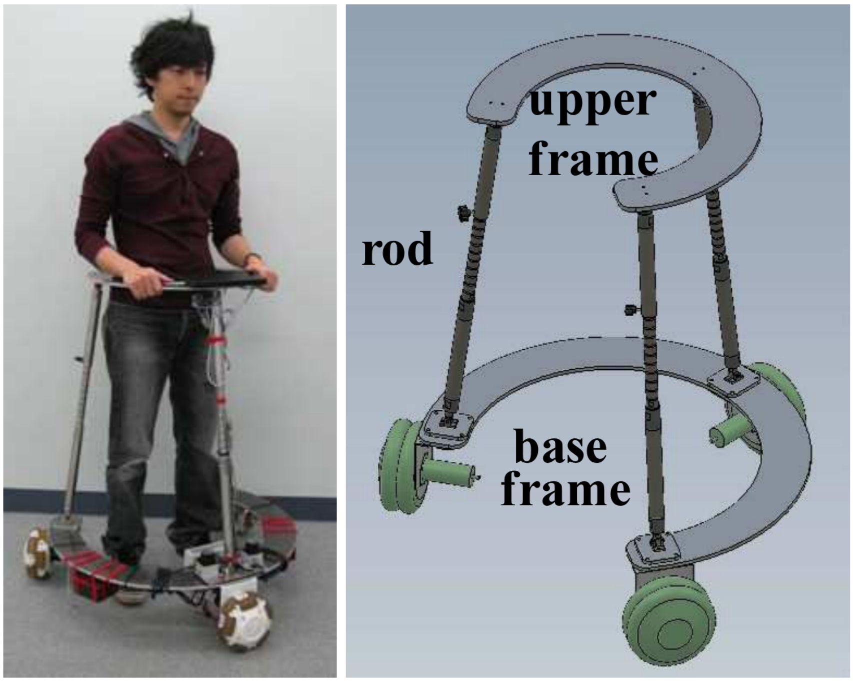


Figure 2.8.: JAIST active robotic walker (JARoW) [95]

Omni-Directional Walker (ODW) is an omnidirectional walker with four omni-wheels developed at Kochi University of Technology, Japan (figure 2.9) [85]. The ODW is designed for physical support and walking rehabilitation where pre-defined paths have to be followed. Its input interface consists of four force sensors placed in the corners beneath the user's forearm support [85]. Because of non-uniform load distribution, i.e., the user exerts more weight on the elbows than on wrists, the measured forces can not be used directly. Therefore, Jiang and Wang [85] use a fuzzy controller, which rules are based on the interaction data from eight young and healthy male volunteers aged 20 to 25 years. Parallel to this work, Tan et al. [159] propose to use an adaptive admittance controller and compare it in simulation to a Proportional-plus-Integral-plus-Derivative (PID) controller. In other papers about ODW, the authors develop strategies for compensation of center-of-gravity shifts, load changes [161], and disturbance forces caused by users' movement. These strategies aim to provide an accurate following of rehabilitation programs' paths [158]. The approaches were evaluated in simulations where authors showed that they are feasible. Nevertheless, an evaluation with the physical device would have been more relevant.

ASBGo is a SW for support of elderly persons and rehabilitation of ataxia patients developed by Adaptive System Behaviour Group (ASBGo) at Minho University, Portugal [108]. The whole development between 2012 and 2017 includes four versions of the device (figure 2.10). The main goal of ASBGo is to create low-cost prototypes that

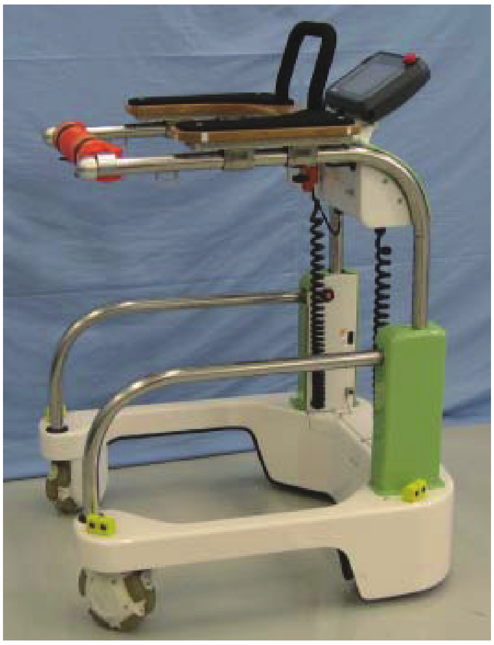


Figure 2.9.: The Omni-directional walker from Kochi University of Technology. Source: [159]

can be easily turned into a product. Since using a joystick as an input device sometimes results in unstable or jerky motion and force-torque sensors are expensive and bulky [108], Martins et al. [109] develop novel handlebars as an input interface to control the ASBGo. Those handlebars are placed on the forearm support and provide two types of operation: (1) vertical – for users with extension problems on their arms; and (2) horizontal – for users with shoulder problems. The novel interface has two potentiometers to detect handles' rotation and linear movement to move the walker forward and rotate it [110]. Integrated springs automatically return the handles to their origin. Martins et al. [110] also propose a calibration strategy for the novel user-walker interface, which adjusts minimum and maximum amplitudes and resistance of the handles. The ASBGo has integrated information about the user's body motion [19], gait analysis [20], and a fall prevention strategy [130]. Martins et al. [112] evaluate their novel interface and ASBGo in a three-week-long clinical study with an ataxia patient.

ASBGo++ is the fourth and latest prototype developed by the same group from the Minho University, Portugal (figures 2.10c and 2.10d). Ahead of its development, Alves et al. [3] provided a detailed discussion on mechatronic drawbacks and potential modifications of the previous version – ASBGo third version (figure 2.10b). Alves et al. [4] introduce the ASBGo++ and compares it to similar Smart Walkers developed in an academic environment. The authors state that the well-centralized center of mass is an advantage regarding the studied walkers. The ASBGo++ is developed in a partnership with a company for orthopedic devices, Orthos XXI. The walker is equipped with sensors to detect forthcoming falls [130], user's movement intentions, and gait patterns [19, 20]. Those functionalities are ported from the earlier versions of the walker. Like the earlier version, it has two passive caster wheels in front and two motorized rear wheels. This version provides comfort features like a wider gait area (width: 58 cm, length: 69 cm), electrical height adjustment of the forearm support, and better materials' strength. From the control perspective, ASBGo++ has four operating modes:

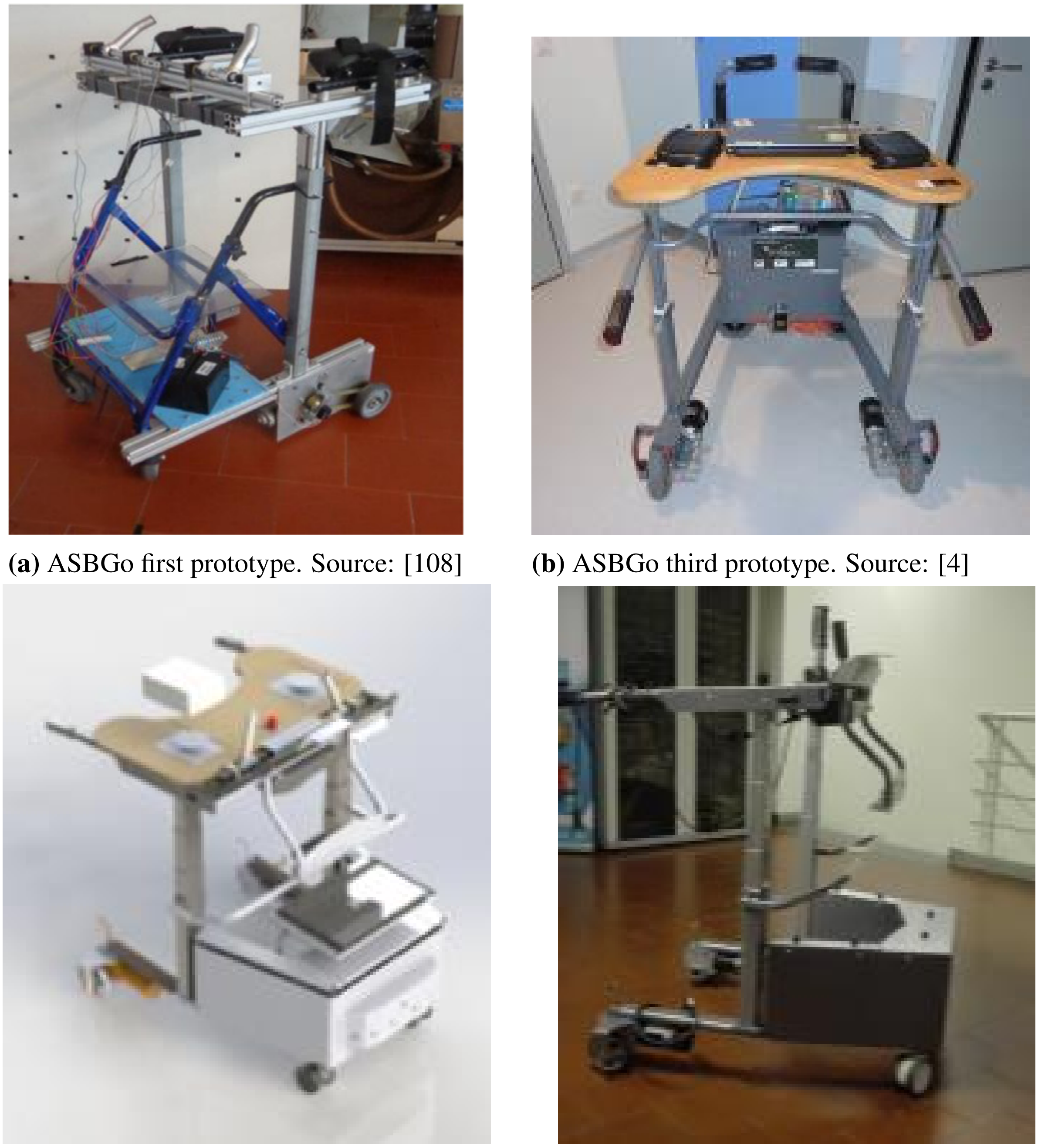
(1) manual mode – handlebars are the input interface providing commands to start, accelerate, slow down, and turn left or right;

(2) remote control mode – used by physiotherapists to monitor the user's behavior, compensations, and reactions against changes in speed and orientations;

(3) autonomous mode – using navigation;

(4) safety mode – warns if a dangerous situation is detected.

SMARTWALKER is a high-tech extension of a conventional walker developed by Eidgenössische Technische Hochschule Zürich (ETH Zürich), Switzerland (figure 2.11). Shin et al. [145] integrate robotic technology into a walker to enable their temporary removal from social situations, e.g., lunchtime in elderly care facilities or retirement homes. In those situations, walkers are not needed since their users sit, but they rather clutter the environment and represent a potential danger for the persons coming and going. The authors utilize localization and navigation functionality with gesture recognition to enable elderly persons to "send-away" and "call-back" their wheeled walkers [145]. The real-time capable gesture recognition is based on k-nearest neighbors (k-NN) classifier with dynamic time warping (DTW). SMARTWALKER is built upon a conventional walker's frame, adding sensors, actuators, and appropriate hardware and software to control those. It is a three-wheel walker with a steerable front wheel and two motorized rear wheels, i.e., it has differential kinematics. An RGBD camera for gesture recognition is mounted under the handlebars on a motor for 360° rotation (figure 2.11). The environment registration is done by using a laser sensor at the bottom of the frame. For computation, SMARTWALKER uses a single-board computer *BeagleBone* for low-level control and a tablet-PC for running algorithms and graphical user interface. SMARTWALKER provides two modes: (1) assistive mode – functionality of conventional wheeled walker with assistance on the slopes, and (2) autonomous mode – functionality of an autonomous mobile platform.The control algorithm uses ground inclination, brakes' status, and the user's leg distance (mean over 90 s) [146]. The prototype was evaluated with twenty-three [145] and twenty-one [146] residents at five different retirement homes in Zürich, Switzerland. The users found the assistive mode relatively comfortable compared to moving the walker without assistance. The authors conclude that the residents found SMARTWALKER exciting and useful, but they had difficulties with the gesture-based



(a) ASBGo first prototype. Source: [108]

(b) ASBGo third prototype. Source: [4]

(c) ASBGo++ CAD model. Source: [4]

(d) ASBGo++ Physical model. Source: [4]

Figure 2.10.: Versions of the Adaptive System Behaviour Group (ASBGo) Smart Walker from Minho University, Portugal.

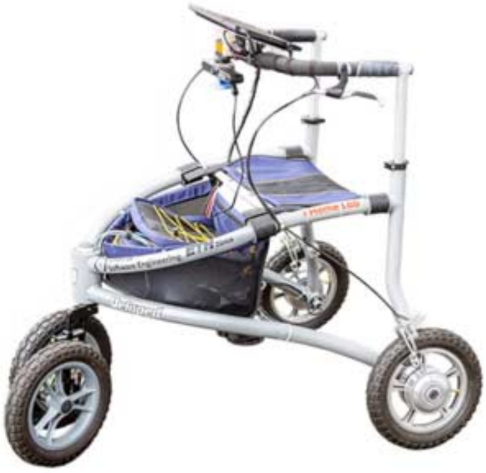


Figure 2.11.: SMARTWALKER. Source: [146]

interface and are reluctant to switch to a robotic walker. The authors assume that the device's size and weight, compared to the traditional walker, and unfamiliarity with the technology negatively influence participants' experience [146].

2.3.2. Commercial Smart Walkers

ello - Der elektrische Rollator is a Smart Rollator (figure 2.12a) developed by German startup e-Movements GmbH, Stuttgart, Germany, in 2017. In 2019, the company went bankrupt, and WMT GmbH, located in Stuttgart, now owns the *ello* [56]. *ello* provides physical support and assistance on slopes. It is controlled using buttons to set the speed. Its kinematics is the same as a conventional walker, and it can be folded for more comfortable transport and storage. Additionally, *ello* is equipped with a GPS tracker and the possibility to initiate an emergency call. Its battery supports a running time of 3 h.

Lean Elderly Assistant (LEA) was developed in 2014 by the company Robot Care Systems from Delft, Netherlands [136]. *LEA* is the most advanced SW from this overview (figure 2.12b). Its primary purpose is to be all-around geriatric assistance providing:

- gait and sit-to-stand assistance;
- activation and training functionalities, e.g., dancing partner;
- cognitive assistance, e.g., navigation and collision avoidance;
- user monitoring; and

- personal assistance, e.g., reminding about personal medication.

These functionalities are possible due to various onboard sensors: IMU, sonars, 3D camera, and integrated tablet. The exact user-walker interface is unknown, but the videos that are available[8], allow the assumption that the device probably has a force-based interface. *LEA* also enables single-arm use and configuration of rehabilitation programs. Unfortunately, at the end of 2019, Robot Care Systems filed for bankruptcy, and the development and production of *LEA* stopped. At the present moment, it is very hard to find any information about *LEA*.

beactive+e E-Rollator is a SW developed in 2013 by the German company BEMOTEC GmbH from Reutlingen, Germany [55]. *beactive+e* is a velocity controlled wheeled walker for geriatric assistance and rehabilitation. Integrated IMU enables assistance on slopes and curbs. *beactive+e* has a removable battery in three sizes to provide three, six, and nine hours of walking support. The removable battery enables the device to be folded for storage and transportation purposes. The company behind *beactive+e* mentions that individual velocity for a user can be configured and that velocity profiles for after-stoke rehabilitation are programmable [9]. This commercial SW is still actively developed and available.

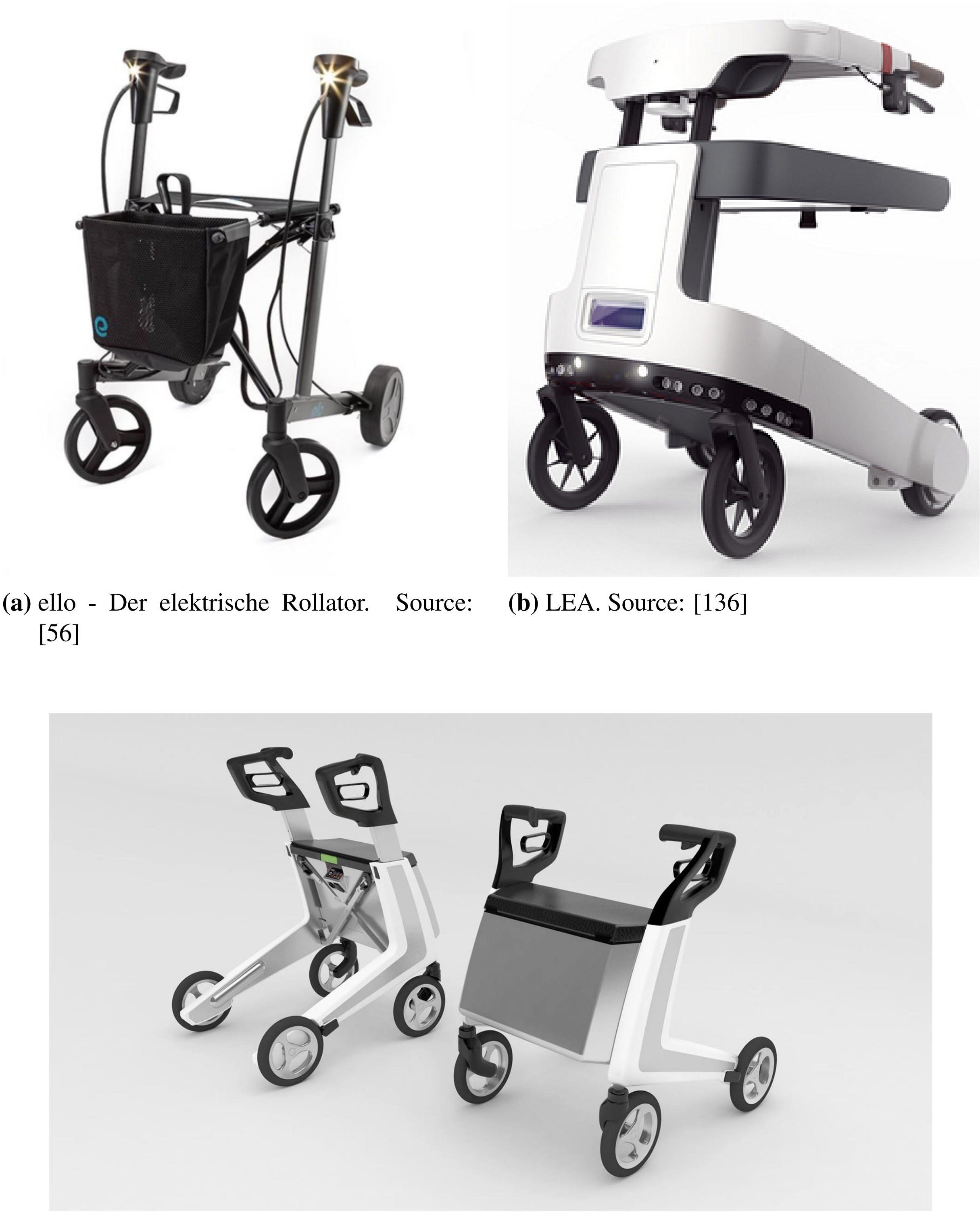
2.3.3. Design of Smart Walkers

SWs are predetermined to support physically and follow commands from their users. Therefore, users' disabilities and needs at sensory, cognitive, and motor levels have to be considered to enable safe and comfortable use [180]. The overview of relevant SWs in section 2.3.1 shows that their designs vary depending on the walker's purpose and targeted user group. This section presents general principles and concepts for designing a novel SW and their influence on the RoboTrainer Prototype and the RoboTrainer v2. An in-depth analysis of RoboTrainer v2's design is given in chapter 4.

The first part of the application-oriented overview of Smart Walkers by Solenne et al. [148] focuses on their mechatronic design regarding locomotion abilities and kinematics. More specifically, the authors evaluate SWs using the following requirements: (1) locomotion, (2) environment, and (3) choice of wheels. According to them, locomotion optimization aims to realize omnidirectional movement using the lowest possible number of motors. When considering environmental criteria, the size and maneuverability of a SW

[8] LEA Videos: `https://www.youtube.com/watch?v=Z60i0C6Bye8&t=2s` and `https://www.youtube.com/watch?v=Z60i0C6Bye8&t=2s` (accessed: Nov. 2, 2020)

[9] Explanation of *beactive+e*'s functionalities: `https://www.youtube.com/watch?v=ySg8gB8vsgA` (accessed: Nov. 2, 2020)



(a) ello - Der elektrische Rollator. Source: [56]

(b) LEA. Source: [136]

(c) beactive+e E-Rollator. Source: [55]

Figure 2.12.: Commercial Smart Walkers

are essential. Therefore, it is relevant to define if SW is intended for use in a hospital or nursing home facility – the most adapted environments, or private home – a cluttered and narrow environment. If SW is designed for use in the household environment, its width should respect the local door-width standards, e.g., Germany, 800 mm [36]. Moreover, if used outdoors, its mechanical design should resist different weather conditions. When reviewing the literature, Solenne et al. [148] found that six types of wheels are used with SWs: fixed wheels, centered orientable wheels, caster wheels, Swedish wheels, spherical wheels as in Wada and Asada [170], and Active Split Offset Castor (ASOC) [152]. Those are evaluated based on multiple criteria, like enabling omnidirectionality, tolerance to ground irregularities, load capacity, and complexity. Finally, Solenne et al. [148] classify the SWs into seven different footprint-kinematics-wheels categories, evaluating their maneuverability and operability.

The rectangular frame with four wheels is the most common form used for wheeled walkers and Smart Walkers (see sections 2.3.1 and A). The main reason for this is guaranteed stability in many situations and, in an optimal case, the user is positioned on the SW's rear axis. The three-wheeled walkers' stability depends on their frame construction since they could also flip over in the forward direction, e.g., *MARC SW* is probably less stable than *JARoW SW* (cf. figures 2.4 and 2.8). To improve the stability in different situations, Ye et al. [183] automatically minimize the SW's footprint in narrow spaces and enlarge it when possible, and Shi et al. [144] reconfigure the SW during the sit-to-stand transition. Wada and Asada [170] proposed a similar approach for undercarriage autonomous wheelchairs using four spherical wheels. Such reconfigurable devices are seldom used because of their technical complexity and high development costs. Therefore, most of the SWs from the literature use four-wheeled frames, especially in cases where users need additional physical support, e.g., ataxia patients [4]. The SW's center of gravity should also be as low as possible, meaning that the heavy components should be placed closer to the ground during design process. Solenne et al. [148] propose to use the force-angle criteria defined by Alwan et al. [7] for mechanical-stability evaluation. Nevertheless, the *ISO 11199-2:2005 Walking aid manipulated by both arms – Requirements and test methods – Part 2: Rollators* [79] defines the minimal turnover angle for conventional walkers, providing more legal safety for developers and producers of SWs.

The most used wheels' configuration for SWs is with fixed and motorized rear wheels and rotatable (caster) front wheel(s). This configuration is classified as type "A" by Solenne et al. [148]. The authors counted twenty-one SW using it. The configuration has differential kinematics with the center of rotation between the two rear wheels. The reasons to use this configuration for a SW are: (1) ease of control; (2) simple motorization and wheel mechanics; (3) physically assured left-right stability; (4) simplified safety concept since SW can not drive over the user's feet. The main issues older people have with their wheeled walkers are connected to this kinematics [101]. Also, non-omnidirectional walkers could represent a risk of falling, especially at the beginning and the end of use, when users try to reach or park them [148].

Therefore, many SWs realize omnidirectional kinematics using different approaches, e.g., passive caster and Active Split Offset Castor (ASOC) wheels [150], center-orientable wheels [57], or omnidirectional wheels [28, 159, 97]. The choice of wheel-type plays a crucial role in this case. If a SW has passive wheels, they have to be caster-type. Possibilities and discussion on active wheels enabling SW's omnidirectionality are discussed in [148]. Center-orientable wheels are the best choice to get the highest precision and minimize the SW's movement when changing wheels' orientation. They could also be in ASOC configuration without axis shift.

To conclude, the optimal locomotion-design would have: (1) a rectangular footprint placing the users on the rear wheels-axis, (2) four steerable wheels, (3) two of those active – center-orientable wheels preferred, and (4) size of the wheels adapted to the outdoor environment.

Solenne et al. [148] discuss the further importance of the user's support surface in the sense of its contact with a SW. The most commonly used surfaces are hands for handles and handlebars and arms for forearm support. When considering handles, the user's weight support depends on handles placed in the coronal or sagittal plane. The coronal handles (perpendicular to the walker's movement direction) provide the most straightforward interface to push the SW, where sagittal (parallel to the SW's movement direction) provide more support [148]. In the literature overview presented in section 2.3.1, both cases can be found. The forearm support is used when users are expected to have frail upper or lower limbs or unstable gait, like Ataxia patients [112].

RoboTrainer's Design in Context of the State of the Art

RoboTrainer Prototype is built upon the existing omnidirectional mobile robot base, attaching a force-torque sensor as an interaction interface and a bike handlebar. This is an often used approach to realize the prototype to investigate the interaction concepts, e.g., [91, 28, 48]. The FTS is chosen as the input interface because it provides intuitive and straightforward interaction with a SW, and is successfully utilized in many SWs from the literature, e.g., [117, 27, 187, 174, 6, 151, 51, 50]. Also, as many research SWs [110, 48, 28], the RoboTrainer Prototype uses standardized, off-the-shelf components, enabling fast development and modular design for potential extensions. The used mobile robot base is an adapted base of the *Care-o-bot 3* service robot, called *rob@work* [47]. The RoboTrainer Prototype's base has four center-orientable wheels, i.e., drive-steer modules, intended for indoor use. The user stands behind the robot's rear axis which could lead to the flipping-over if they support all their weight on the devices' handlebar. Nevertheless, this configuration has the advantage that movements to the side are possible without collision between the SW's frame and the user's feet. Still, the used base's footprint is suboptimal for the intended scenario (cf. section 3.3). The handlebar choice was good, especially the ergonomic grips that provided comfortable use of the RoboTrainer Prototype during evaluation (section 3.1). The handlebar's fixture provides limited height change, which was never used since the users did not have significant issues interacting with the device. The handlebar is somewhat bent from the coronal plane, providing rather force-based interaction with the walker than physical support.

RoboTrainer v2 has a custom-made mobile base optimized for training functionality. It uses the same FTS as the RoboTrainer Prototype and bike handlebar as the user's input interface. Except for the limited height adaption due to the reused bike fixture, the handlebar can also be moved along the device's vertical rod. The base has three drives (drive-steer modules) from *Care-o-bot 4* service robot [46] providing omnidirectional movement. These new-generation drives are much more robust than those used in the RoboTrainer Prototype, so it is expected to avoid mechanical issues that happened before because of high-force interaction with users. The choice of drives is also influenced by the possibility of reusing the software stack for low-level control from the previous RoboTrainer version. The main design feature of RoboTrainer v2 is the reconfiguration of its footprint (figure 4.4). This reconfiguration is done manually and enables agile training with sideway movements in closed configurations and large support area in opened configurations. The size of RoboTrainer v2 is optimized to pass through the standard-doors defined by *DIN 18040-1:2010 Construction of accessible buildings - Design principles - Part 1: Publicly accessible buildings* [36] norm.

2.3.4. Control of Smart Walkers

The application of Smart Walker (SW) is a typical example of human-in-the-loop control [91]. Therefore, it is difficult to separate the user from the control system. For RoboTrainer, the user is a part of the control loop since it navigates it to solve tasks. Still, the strolling control must work correctly before creating any challenging task for the user where RoboTrainer behaves "strangely". This section begins with an overview of control approaches used with SWs. Later, it provides an in-depth analysis of the admittance control approach and specific functionalities used in RoboTrainer.

Usually, SWs need to regulate their forward and turning movements [148] since backward walking poses many risks, especially for older adults or people with disabilities. The control approach is inherent to the used hardware, especially the user's input interface for commands. Another factor is the interface toward the SWs' actuators. The most widely used commands are velocities, like by Yu, Spenko, and Dubowsky [185], Chuy, Hirata, and Kosuge [27], Frizera et al. [51], and Martins, Santos, and Frizera [108]. The first SW prototype from Lacey and M. Dawson-Howe [91] uses a joystick as the controller's input controlling the device's velocity. In this case, the joystick's deviation from zero-position is used to generate wheels' velocities multiplying it with some proportional factor [91]. In another case presented by Martins et al. [110], virtual forces inside the controller are generated, and the resulting SW's velocity is calculated using the admittance equation.

In the first decade of the 21st century, the progress in hardware performance enabled more calculation-intensive control methods using various interfaces. *AZIMUT-3* SW [48] uses force-sensitive load cells in custom-made wheels as input in its dynamic model to generate the walker's movement. The goal is to create a natural and straightforward interface by making the whole SW's body responsive to external forces [48, 49]. Nevertheless, the measurement of input forces is not possible in every wheels' configuration, which is merely caused by the physical properties of the SW. Frizera et al. [50] propose another inverse kinematics controller for a SW with differential kinematics. The controller minimizes user-walker orientation differences and distances between them. Input data are the user's legs positions measured by a laser range finder (LRF) and Inertial Measurement Unit (IMU)-data from the user and the walker. This approach enables precise user tracking, but the drawback is that external wireless sensors are needed. The issues with gait-based interfaces are non-linearity and variation of the gait across users. This makes them hard to use for reliable control.

Everyday challenges for persons using a wheeled walker are walking uphill, where additional moving assistance is needed, and walking downhill, where stability and braking are the main issues. Tani et al. [163] propose a generalized internal model control (GIMC) design in combination with a Linear-Quadratic (LQ) control to realize this functionality. The approach is used in a SW where the walker's velocity is set, like in *ello* [56] and *beactive+e* [55] commercial SWs.

The most common and robust interface to detect the user's intentions is by measuring interaction forces [114], e.g., [175, 185, 68, 27, 51]. The interfaces like buttons, switches, and touch screens confuse elderly persons, and joysticks are sensitive to vibrations due to the foot strike or uneven terrain. Wasson et al. [175] are the first to investigate user-walker interaction forces to determine users' navigation intent. They provide fundamental knowledge about mapping input forces to the SW's movement-directions and interference between those. Alwan et al. [6] and Frizera Neto et al. [52] focus on separating the user's navigation commands and gait-caused oscillations from the force signal. To do this, the methods used are somewhat different depending on the user-walker contact type, i.e., handles or forearm support. The main gain from this approach is that the same force-torque sensor (FTS) can be used as the input interface and for gait monitoring.

Yu, Spenko, and Dubowsky [185] and Chuy, Hirata, and Kosuge [27] worked extensively on applying admittance-model control when the user's interaction force is measured and the walker's velocity is controlled. Yu, Spenko, and Dubowsky [185] propose a damping adaption method based on the walker's movement velocity. The method should stabilize the walker on the lower and make it easier to move on higher velocities. Chuy, Hirata, and Kosuge [27] investigate the passive behavior of an active type SW in a series of publications. The main idea is to provide inherited safety in the controller so that the energy never flows from the walker toward the user since this could endanger them. According to the authors, passive support systems have the following characteristics [29]:

1. They are user-powered – users should push the support system in order to move;
2. They are inactive without the user's intentions – the support system does not move if there is no intention applied;
3. The actuators are used for steering and braking.

RoboTrainer's Control in Context of the State of the Art

RoboTrainer uses a force-torque sensor as the user's input interface and velocity-controlled wheels. Therefore, the admittance-model-based control utilizing the mass-damper system as the desired dynamics is used. The used multi-dimensional model is shown in equation (2.1), where $\mathbf{M} \in \mathbf{R}^{n \times n}$ and $\mathbf{D} \in \mathbf{R}^{n \times n}$ are respectively inertia (in kg) and damping (in kg/s) matrices. The $\mathbf{F}_\mathrm{h} \in \mathbf{R}^{n \times 1}$ is input force (in N), or in this concrete case, the force representing the user's intention.

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) = \mathbf{F}_\mathrm{h}(t) \tag{2.1}$$

For the specific case of SWs, the system's dimension is $n = 3$, and the matrices M and D are diagonal. This implies independence between degrees of freedom (DOFs), i.e., forward and sideward movement and rotation. This is the most used approach in the literature, e.g., [185, 64, 27].

Passive Behavior

For active training with a SW, as proposed in this thesis, stability and passivity of the walker's behavior's are essential. This is already mentioned by Yu, Spenko, and Dubowsky [185], where Chuy et al. [31] provide detailed passivity analysis and a solution to keep the system passive in critical situations. For this, Chuy et al. [31] examine one-dimensional control dynamics, defined as SW's desired movement with desired mass M_d and damping D_d:

$$M_\mathrm{d}\ddot{x}(t) + D_\mathrm{d}\dot{x}(t) = F_\mathrm{h}(t) \tag{2.2}$$

Since the system is in a stationary state at time $t = 0$, the initial conditions are $\ddot{x}(0) = 0$, $\dot{x}(0) = 0$ and $F_\mathrm{h}(0) = 0$. The actual movement of a SW in one dimension is described with:

$$M_a\ddot{x}(t) = F_\mathrm{h}(t) + F_{acc}(t) \tag{2.3}$$

The M_a is the actual mass of the SW, F_h applied force, and F_{acc} SW actuators' force. The initial states are again assumed to be zero.

Combining the equations (2.2) and (2.3), it follows the relationship between the actuator's force F_{acc} and user's input F_h:

$$G_\mathrm{SW}(s) = \frac{F_{acc}(s)}{F_\mathrm{h}(s)} = \frac{M_\mathrm{a} - M_\mathrm{d}}{M_\mathrm{d}} \left(\frac{s - \frac{D_\mathrm{d}}{M_\mathrm{a} - M_\mathrm{d}}}{s + \frac{D_\mathrm{d}}{M_\mathrm{d}}} \right) \tag{2.4}$$

Equation (2.4) provides stable behavior for the user's input, assuming $M_\mathrm{d}, D_\mathrm{d} > 0$. According to Chuy, Hirata, and Kosuge [27], the SW's transfer functionequation (2.4) is passive since it only consumes the user's energy.

Still, depending on the choice of parameters M_d and D_d, Yu, Spenko, and Dubowsky [185] reported oscillatory behavior of the SW during the interaction, i.e., the user-walker system is at its stability limits. Chuy, Hirata, and Kosuge [27] found the cause for this in a disturbance force that appears during physical interaction. They describe it as a user's reaction force at handles. This reaction force creates a positive feedback loop (figure 2.13a) to the walker's input. Closing the open-loop transfer function (equation (2.4)) over a gain-element $H(s) = h$ and resolving its poles, the stability criteria become [27]:

$$M_\mathrm{d} > \frac{hM_\mathrm{a}}{h + 1} \tag{2.5}$$

Therefore, the user-walker system's stability depends on the amount of user's reaction force ($\boldsymbol{F}_\mathrm{dis}$) measured by the FTS, i.e., depends on the stiffness of the users' arms' during the interaction.

To solve this issue, Chuy, Hirata, and Kosuge [27] add a damping-gain for the user's input (figure 2.13b). This modifies the stability criteria from equation (2.5) to:

$$M_\mathrm{d} > \frac{khM_\mathrm{a}}{1 + kh} \tag{2.6}$$

The authors report that reducing the factor $k < 1$ also stabilized the system in their experiments. Chuy, Hirata, and Kosuge [27] propose further monitoring the system's passive behavior. The approach calculates the energy of the system (equation (2.7)). If this energy is negative, the energy flows from the walker toward the user. Therefore, the factor k has to be reduced to remove potential danger.

$$E_{system} > \int_0^t \mathbf{F}_h \cdot \dot{\mathbf{x}} dt \tag{2.7}$$

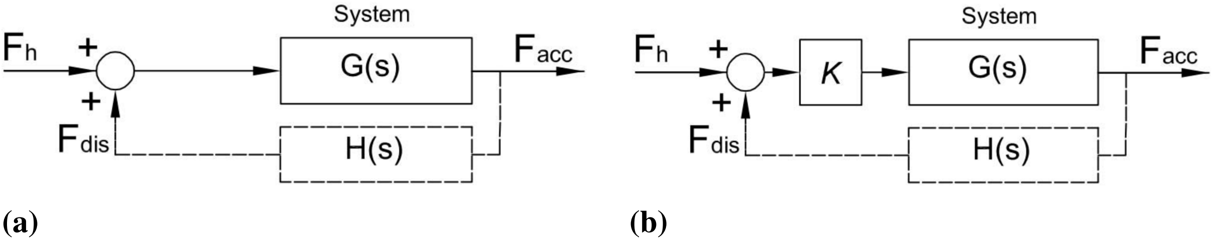


Figure 2.13.: The SW's representation in block diagram form with a disturbance force F_{dis} that appears during physical interaction as defined by Chuy, Hirata, and Kosuge [27]. F_{h} is the input force, F_{acc} resulting actuators force, and $H(s)$ is the disturbance's generalized transfer function. (a): analyzed system with disturbance. (b): system with stabilization factor K. Source: [27, p 167]

In general, the energy calculation in equation (2.7) will be positive if the user's force and the walker's velocity have the same direction. The authors state that the main drawback of the approach is that the system's energy is summed over time, and when the system becomes unstable, energy needs time to get negative. In this thesis, the approach from Chuy, Hirata, and Kosuge [27] is applied to RoboTrainer, evaluated, and extended to be more performant (see section 5.3.2).

Adaptive Admittance Control

From the admittance model (equation (2.1)), it is clear that more vital users will move faster with a SW, whereas weaker users will potentially be overwhelmed with the effort needed to move the device. This issue is also observed by Yu, Spenko, and Dubowsky [185] when testing the SmartCane [40] with elderly persons. They investigated different combinations of desired mass M_{d} and damping D_{d} parameters (equation (2.2)) and characterized five areas with a specific SmartCane's behavior, depicted in figure 2.14. The authors observed that for too responsive models, i.e., with a small mass, the motion is oscillatory due to high-frequency noise in the force signal, which is not absorbed by the controller's time constant. For models with large mass and small damping, the motion is too difficult to control since the inertia, i.e., the system's time constant, is considerable. The models with large mass and damping appear heavy since the effects of input force are reduced.

Another interesting finding is that Yu, Spenko, and Dubowsky [185] observed that users have different requirements on the admittance model for different motion phases. Older people feel less assured when they walk from a standstill with a very responsive device as they are afraid of falling. The users would also like their mobility aid to react fast when they want to stop or slow down. On the other hand, they would like to stroll with

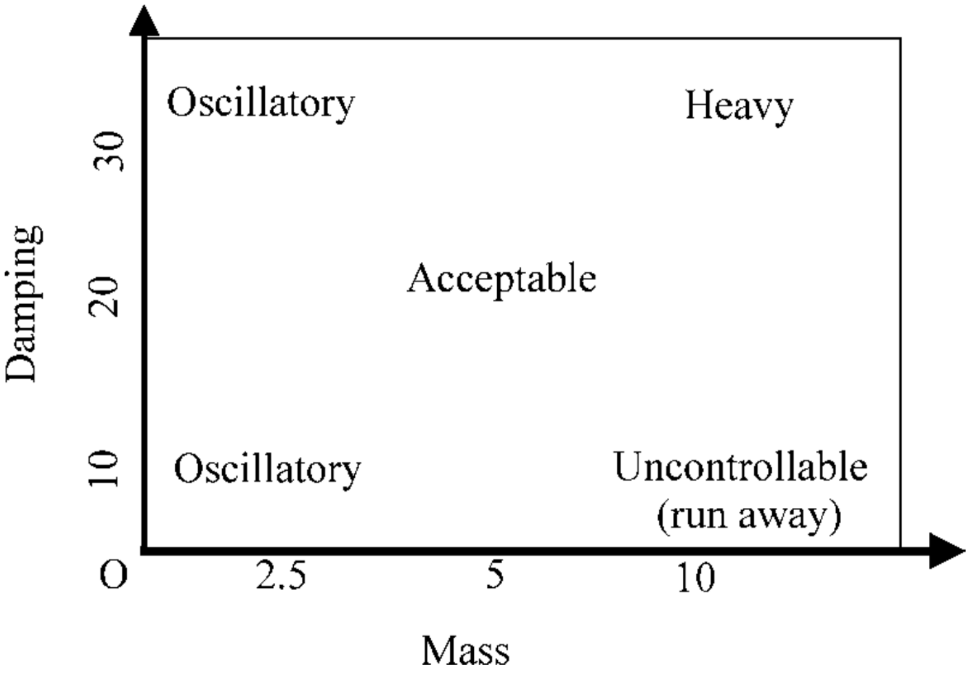


Figure 2.14.: Effect of the mass and damping parameters on SW's behavior. The results for the SmartCane [40]. Too responsive models, i.e., with a small mass, culminate in oscillatory behavior. The models with large mass and small damping result in uncontrollable motions, and those with large mass and damping appear heavy. Source: [185].

almost no effort using their natural walking speed. Therefore, the system should be less responsive at the start with higher mass and damping. The model with the low mass and higher damping should be used when the user wants to stop, and the model with small damping when walking with constant speed. To achieve this, Yu, Spenko, and Dubowsky [185] propose the following velocity-dependent damping adaption model:

$$b = b_m - \frac{b_m - b_0}{V_m}|V| \tag{2.8}$$

Variable b_m is the maximal damping, b_0 is the minimal damping, V is the walking aid speed, i.e., *PAMM*, and V_m is the maximum speed allowed. The model from equation (2.8) was tested with more than ten users who agreed that the aid with the variable damping model is easier to use. Nevertheless, more details about the users and test conditions are not given.

The adaptive approach from Yu, Spenko, and Dubowsky [185] inspired the approach for velocity-dependent adaption implemented in this thesis (section 5.4). Their method is also compared to the method from this thesis during the evaluation (section 7.2).

Parameterization Strategies

Per-user parameters in the admittance equation are necessary to achieve satisfactory interaction for individual users. Yu, Spenko, and Dubowsky [185] already mention this, but they do not provide any approach to determine those parameters. Only a few authors addressed this topic in the literature, e.g., [30, 26, 110]. Chuy, Hirata, and Kosuge [30] propose a circular path along which the user needs to navigate the SW to calculate an individual offset for controlling the torque. The initial approach needs a path in the SW's environment and localization to determine the offset. The process has two steps, one for each rotation direction. In the following work, Chuy, Hirata, and Kosuge [26] show a simpler approach. The users now have to drive the SW along the straight wall and keep a constant distance. Meanwhile, the wall-following precision is calculated from LRF data, and a new center of rotation is estimated. This approach's advantage is usability in almost any environment without preparation and higher precision since the SW's localization is not used. Martins et al. [114] propose another offline method to determine the meaning of the user's input on SW's handles. The user moves SW freely, i.e., with decoupled motors and their interaction forces are measured. These forces are used to calculate signal gains to adjust the user's input before the admittance dynamics. This approach enables a determination of natural interaction forces without influence from the SW's control.

For RoboTrainer v2, a two-stage parameterization approach is presented in section 5.5.

2.3.5. High-Level Control Approaches

Almost all SWs from the literature provide cognitive or sensorial support to their users. The common use-cases are obstacle avoidance, navigational assistance, and execution of rehabilitation programs. This is done using methods like the virtual force field (VFF) [71, 65, 27] initially developed for mobile robots' path planning [87, 171]. The calculated VFFs are superposed to the user's input directly or by using a weighted sum (equation (2.9)) as by Yu, Spenko, and Dubowsky [185].

$$\mathbf{M} \cdot \ddot{\mathbf{x}}(t) + \mathbf{D} \cdot \dot{\mathbf{x}}(t) = \alpha \mathbf{F}_{\mathrm{h}}(t) + (1-\alpha)\mathbf{F}_{\mathbf{system}}(t) \tag{2.9}$$

The $\mathbf{F}_{\mathbf{system}}$ is the control system's intention, and α the scaling factor of shared control. This factor is derived from the quality criteria of the user's interaction with the SW, and it is application-specific [185]. The issue with this approach is its non-passivity, i.e., the SW's control system modifies the user's intention before the admittance model. This means that the system can move even if the user is not holding it or it can override the user's intention and endanger the user. This approach is used in the RoboTrainer Prototype controller [154] for activation of the *artificial-forces* concept.

Chuy, Hirata, and Kosuge [29] propose a passive approach to influence SW's movement using VFFs. They propose a change of dynamic parameters based on the environment's feedback, i.e., obstacle or path configuration. According to equation (2.10) [29], this alters the user's intention. In the equation, $\mathbf{M_{env}} \in \mathbf{R}^{n \times n}$ and $\mathbf{D_{env}} \in \mathbf{R}^{n \times n}$ are inertia and damping caused by the environment.

$$(\mathbf{M} + \mathbf{M_{env}}) \cdot \ddot{\mathbf{x}}(t) + (\mathbf{D} + \mathbf{D_{env}}) \cdot \dot{\mathbf{x}}(t) = \mathbf{F}_{\mathrm{h}}(t) \tag{2.10}$$

Using this approach, an active SW's passivity is kept and the user's input can be influenced. The authors also provide stability analysis and guidelines on how to vary these parameters. Unfortunately, the approach does not correspond to RoboTrainer's scenario, where some SW's activity is desirable. This means, on the low velocities, users would like to have the SW's behavior as described by equation (2.10), but on the higher velocities and in scenarios with younger persons, challenging disturbances should be allowed. Moreover, letting each control action change the main controller dynamic would result in a very complex implementation, hard to debug and evaluate. Therefore, a velocity- and direction-aware approach, presented in section 6.2, is implemented for the RoboTrainer v2.

A further adaption for the RoboTrainer's behavior is modifying the user's input to shift its center of rotation. Chuy, Hirata, and Kosuge [28] first proposed this approach, and its purpose is to balance strength differences in the user's left and right arm. This is done by moving the point of applied dynamic (equation (2.11)) using a transformation matrix ${}^{coa}\mathbf{T}_r(X_{coa}, Y_{coa})$. The matrix changes the torque influence on the system and thus the SW's heading angle. The authors also propose a training method to calibrate the transformation matrix to enable optimal SW's movement for individual users [30, 26].

$$\mathbf{M} \cdot \ddot{\mathbf{x}}(t) + \mathbf{D} \cdot \dot{\mathbf{x}}(t) =^{coa} \mathbf{T}_r(X_{coa}, Y_{coa})^r \mathbf{F}_h(t) \tag{2.11}$$

This approach is also used as a control action in RoboTrainer v2 (see chapter 6).

2.3.6. Tracking User's Legs during Interaction with a Smart Walker

RoboTrainer v2 uses the position of the user's legs to determine individual velocity-dependent adaption. This section gives a short overview of the methods for leg tracking used in scenarios with SWs.

Recent methods for tracking the user's leg position during interaction with a SW use laser range finder (LRF) data [95, 111, 50, 146]. The main issue is to separate the left and right leg measurements, for which different classifiers were used, e.g., Frizera et al. [50] uses *k*-nearest neighbors (*k*-NN) and Shin, Rusakov, and Meyer [146] an expectation-maximization (EM) algorithm. An algorithm from Lee et al. [95] also calculates the center between the legs as the user's body center's projection. In their earlier work, the authors propose the use of infrared (IR) sensors to calculate the user's leg segments' positions as a low-cost solution [97]. Nevertheless, the approach was abandoned because of IR sensors' insufficient precision and the lack of possibility to distinguish between the legs reliably. Frizera-Neto et al. [53] propose using Inertial Measurement Units (IMUs) fixed to the user's legs to determine their motion and transfer those data using wireless communication to *SIMBIOSIS* SW. The approach has the advantage that a precise user's gait modeling is possible. Still, additional hardware and preparations are needed before using the SW.

Since the RoboTrainer v2 already has laser range finders for the safety and registration of its environment, their data are used for user's leg tracking. During the Bachelor's Thesis of Azanov [11], an algorithm using the k-NN approach for leg segmentation and Kalman filter for extracting dynamic parameters for each leg is implemented. This algorithm is used for measuring the user's distance during the evaluations described in chapter 7.

2.4. Comparison of Relevant SotA-Smart Walkers' Features

A detailed overview of features from Smart Walkers from the literature is given in Appendix section A. This section shows the most relevant features and compares them to those implemented by the RoboTrainer Prototype and RoboTrainer v2. The features are chosen concerning the goals of this thesis. The Smart Walker is selected based on active actuation, relevance, and amount of information in the literature.

The features are categorized into three main classes corresponding to the structure of this thesis:

I **Design** – mechatronic design (chapter 4);

II **Control** – control strategies and their functionalities (chapter 5);

III **Behavior Modification** – support functionalities to modify SW's behavior for a scenario (chapter 6).

The features to compare different Smart Walkers are defined as criteria from the above categories:

I - User inside Footprint (UiF) User is placed inside the device footprint, minimizing the tip-over risk.

I - Omni Smart Walker can move in any direction with an arbitrary orientation.

I - M.Adapt. Smart Walker provides mechanical adjustments to support users' individual needs.

I - Safety There is a safety concept for the device using, if applicable, safety-certified components.

II - Passive The control approach provides passive behavior, i.e., the device can react if the energy flow is toward the user.

II - Adaptive The control approach provides online parameter adaption.

II - Individual The control approach implements a parameterization strategy to determine users' parameters.

III - Passive Implemented behavior modifiers will not move the device if a user has no intention.

III - Reactive Device can influence user's intention by executing programs or evaluating data from the environment or user monitoring.

III - Multiple There are multiple types of behavior modifiers used for different goals.

Table 2.1.: Overview of the most relevant Smart Walkers from the state of the art compared to the RoboTrainer Prototype and RoboTrainer v2. The explanation of the fields are given in section 2.4.

SW	**Design**				**Control**			**Behavior Modification**		
	UiF	Omni	M.Adapt.	Safety	Passive	Adaptive	Individual	Passive	Reactive	Multiple
PAM-AID 2nd [92]	✓		✓	✓					✓	
PAMM SmartWalker [184]	✓		✓			✓			✓	
Walking Helper [28]		✓			✓		✓	✓	✓	
SIMBIOSIS [51]	✓		✓							
JARoW [97]	✓	✓	✓						✓	
ODW [85]	✓	✓	✓	✓		✓			✓	
ASBGo [108]	✓		✓				✓		✓	
UFES Walker [50]	✓					✓			✓	
SMARTWALKER [145]	✓								✓	
ASBGo++ [4]	✓		✓	✓				✓	✓	
Smart Walkers from this Thesis										
RoboTrainer Prototype [156]		✓				✓	✓		✓	✓
RoboTrainer v2 [153]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Commercial Smart Walkers										
ello [56]	✓		✓						✓	
beactive+e [55]	✓		✓	✓			✓		✓	
LEA [136]	✓		✓	✓			✓		✓	✓

2.5. Relevant Control Theory Background

This section presents the overview of the control theory regarding the admittance model used in the following chapters of this thesis.

The general form of the admittance model for controlling a SW is given in equation (2.12). The $\mathbf{M}$ represents the mass, $\mathbf{D}$ damping, and $\mathbf{v}$ velocity of a physical system when influenced by force $\mathbf{F}$. Another form, also used in this thesis, is given in equation (2.13), where the velocity variable $\mathbf{v}$ is substituted with the position derivative $\dot{\mathbf{x}}$.

$$\mathbf{M} \cdot \dot{\mathbf{v}}(t) + \mathbf{D} \cdot \mathbf{v}(t) = \mathbf{F}(t) \tag{2.12}$$

$$\mathbf{M} \cdot \ddot{\mathbf{x}}(t) + \mathbf{D} \cdot \dot{\mathbf{x}}(t) = \mathbf{F}(t) \tag{2.13}$$

Using Laplace-transformation [18, p 18], the system from equation (2.12) can be written for one degree of freedom in the form:

$$M_{\text{i}} \cdot sV_{\text{i}}(s) + D_{\text{i}} \cdot V_{\text{i}}(s) = F_{\text{i}}(s) \tag{2.14}$$

from which the input-output transfer function is calculated:

$$Gi(s) = \frac{V_{\text{i}}(s)}{F_{\text{i}}(s)} = \frac{1}{M_{\text{i}}s + D_{\text{i}}} \tag{2.15}$$

To simplify the nomenclature in the reminder of this work, the equations are "matrix of equations", where each, axis, i.e., degree of freedom is independent. To emphasize its multidimensionality, the equation is then written using matrix-nomenclature:

$$\mathbf{G}(s) = \frac{\mathbf{V}(s)}{\mathbf{F}(s)} = \frac{1}{\mathbf{M}s + \mathbf{D}} \tag{2.16}$$

which should be read as:

$$\begin{bmatrix} G_{\text{i}} \\ \dots \\ G_{\text{k}} \end{bmatrix} = \begin{bmatrix} \frac{V_{\text{i}}(s)}{F_{\text{i}}(s)} \\ \dots \\ \frac{V_{\text{k}}(s)}{F_{\text{k}}(s)} \end{bmatrix} = \begin{bmatrix} \frac{1}{M_{\text{i}}s+D_{\text{i}}} \\ \dots \\ \frac{1}{M_{\text{k}}s+D_{\text{k}}} \end{bmatrix}$$

For the analysis of the transfer function, the standard form of PT1-component [89, p 183] is helpful:

$$\mathbf{G}(s) = \frac{\mathbf{K}}{\mathbf{T}s + 1} = \frac{\frac{1}{\mathbf{D}}}{\frac{\mathbf{M}}{\mathbf{D}}s + 1} \tag{2.17}$$

with the pole vector $\mathbf{p} = \frac{\mathbf{D}}{\mathbf{M}}$.

The standard form from equation (2.17) helps to understand the effects of its parameters on the system's response. The step response of the system in the time domain is:

$$\mathbf{v}(t) = \mathbf{K}\left(1 - e^{-\frac{t}{\mathbf{T}}}\right)\mathbf{f}(t) = \frac{1}{\mathbf{D}}\left(1 - e^{-\frac{t}{\frac{\mathbf{M}}{\mathbf{D}}}}\right)\mathbf{f}(t) \tag{2.18}$$

For this response, the following is true [89, p 184]:

1. The system's steady-state ($t \to \infty$) is equal to the input multiplied with $\mathbf{K} = \frac{1}{\mathbf{D}}$;
2. After time $\mathbf{T} = \frac{\mathbf{M}}{\mathbf{D}}$, the system produces an output of $0.632\mathbf{K}$ multiplied with the input.

To implement an admittance controller in a digital computer, zero-order hold (ZOH) transformation [18, p 34] is used to keep transition characteristics in sampling moments the same as for a continuous system [89, p 650]. The discrete transfer function can then be then calculated using Z-transformation [25, p 188]:

$$G(z) = Z\left\{G_{ZOH}(s) \cdot G(s)\right\} = Z\left\{\frac{1 - e^{-s\Delta t}}{s} \cdot G(s)\right\} = (1 - z^{-1})\, Z\left\{\frac{G(s)}{s}\right\} \tag{2.19}$$

where G_{ZOH} is the transfer function of the ZOH element, and Δt is the sampling period.

Equation (2.16) can be written and its variables substituted as:

$$G(s) = \frac{\frac{1}{\mathbf{M}}}{s + \frac{\mathbf{D}}{\mathbf{M}}} = \frac{b}{s + a} = \frac{b}{a}\frac{a}{s + a} \tag{2.20}$$

where $b = \frac{1}{\mathbf{M}}$ and $a = \frac{\mathbf{D}}{\mathbf{M}}$.

Now, the ZOH discretization from equation (2.19) can be directly applied, giving the discrete transfer function of the admittance rule (equation (2.12)) [89, p 650]:

$$G(z) = \frac{\mathbf{V}(z)}{\mathbf{F}(z)} = \frac{1}{\mathbf{D}} \cdot \frac{1 - e^{-\frac{\mathbf{D}}{\mathbf{M}}\Delta t}}{z - e^{-\frac{\mathbf{D}}{\mathbf{M}}T}} \tag{2.21}$$

The admittance controller is implemented using its canonical form [18, p 95][25, p 189]:

$$\mathbf{V}(k) = \frac{1}{\mathbf{D}} \cdot \mathbf{F}(k-1) \cdot (1 - e^{-\frac{\mathbf{D}}{\mathbf{M}}\Delta t}) + \mathbf{V}(k-1) \cdot e^{-\frac{\mathbf{D}}{\mathbf{M}}\Delta t} \tag{2.22}$$

The steady-state of equation (2.22) is calculated for $t \to \infty \implies z \to 0$:

$$V(0) = \frac{\mathbf{F}}{\mathbf{D}} \tag{2.23}$$

3 Proof of Concept of a Smart Walker for Training

The Smart Walkers (SWs) out of literature focus on physical support and condition monitoring for elderly and disabled persons. The goal is to support and prolong the mobility and independence of users. Only a handful of devices provide any training or rehabilitation functionalities for their users. Most devices are also highly optimized for only one purpose, e.g., ASBGo Smart Walker for Ataxia rehabilitation [112]. There is a lack of devices targeting still healthy persons with specific training to slow down the negative aging impacts. A promising, sensor-supported training is presented by Schwenk et al. [141] for persons in the early stages of dementia, like mild cognitive impairment (MCI). Similarly, robot-supported training is introduced by Stogl et al. [156], whose concept and results are presented in this chapter. The used device is a kind of Smart Walker developed to provide neuromuscular training for people with MCI.

The research presented here was done in the *HEiKA*[1] project: *Technical system for physical activation of persons with early-stage dementia* in cooperation with Central Institute of Mental Health (CIMH) in Mannheim, Germany. The goal was to investigate "how" and "if at all" a robotic system could be beneficially used in a therapeutic context for people with MCI. The work focused on developing the RoboTrainer Prototype and its evaluation in a study with the target group [156, 154]. This chapter presents details on the RoboTrainer Prototype, i.e., its hardware and software architecture, and the targeted training design for persons with mild cognitive impairment using high-level control concepts.

The first section (section 3.1) presents the concept of neuromuscular training developed in this work. The section details the scientific background, the training purpose, the

[1] Heidelberg Karlsruhe Strategic Partnership: `https://www.heika-research.de`

human-robot interaction approach, technical constraints, and requirements. Based on these, the training approach is reasoned. Section 3.2 presents the RoboTrainer Prototype developed for the proposed training. The prototype's mechanical design, control stack, and implemented control concepts, also called control actions, are discussed. In the following section 3.3, the evaluation of the proposed training and RoboTrainer Prototype is presented. The evaluation was done by ten older adults who used the device in five individual one-hour sessions and provided feedback on its functionality and the training tasks. The chapter closes with a short conclusion in section 3.4 and discusses the device's advantages and disadvantages in section 3.5. This discussion and the improvements mentioned therein are the basis for RoboTrainer v2 and further development of the control software. Those are presented in the following sections.

The work presented in this section is partially published in Stogl et al. [156] and Stogl et al. [154] and described and analyzed in the Master's Theses of Armbruster [10] and Wang [172].

3.1. A Neuromuscular Training with a Mobile Robotic Device

The current state of the art on combined cognitive motor training utilizes sensory feedback [141] and virtual reality technologies [119] to engage people in exercise and provide information on the user's state (see section 2.1). The research on Smart Walkers (SWs) focuses on physical support and users' state monitoring rather than on novel and engaging training approaches for sensorimotor activation. Therefore, there is no suitable training approach in the literature that could be "just used" with a SW-like device or a mobile robot. This fact leads to questions on the feasibility of such training and its concrete benefits. A Smart Walker, i.e., an intelligent autonomous device, can extend existing methods by providing focused and individualized treatment. Some advantages of using such devices with conventional training and activities are the following: (1) continuous observation, documentation, and evaluation of the training process and progress; (2) detailed measurements of a person's state during the interaction with a training device; (3) self-determination for the user by adapting session-lengths and training times; (4) safety of the user by providing physical support and controlled environment for the training; (5) long-term-oriented training including task variability and repeatability of relevant patterns for a specific user; and (6) reduction of work-load for caregivers during training, thus saving them precious time for social engagement with their patients.

The studies mentioned above, and other literature, have a holistic view of physical activity's influence on persons with mild cognitive impairment and dementia. So, there is a lack of research regarding the impact of the specific training patterns which provides input about type, structure and design of training itself. Without such studies, it is impossible to develop sensible technical systems for a specific training purpose. Therefore, in this work, a training concept with a reasonably simple robotic system is presented. It utilizes direct physical human-robot interaction based on measured force and moments. Such interaction is easily understandable for elderly persons independent of their experience with technical and robotic systems. Furthermore, the training is designed so that no previous knowledge or specific skills are needed. The exercises are especially designed to provoke motor and mental effort by its users. The following paragraphs discuss the reasons and methods for such training.

What is the goal of the training?

The purpose of Neuromuscular Training is to engage a person in physical activity and, at the same time, provide a cognitive challenge. Although not the same, typical examples are dual-task activities, such as walking and simultaneously executing some secondary tasks (see section 2.1). In this work, walking is used because it is the most straightforward physical activity. Therefore, it integrates into many scenarios, even in training with a mobile robotic device. To realize the cognitive stimulus, users would have to solve specific tasks by directly interacting with such a device. There are two possibilities; first, users act passively during training, i.e., they follow what the device is doing and react to it; and second, the users are actively controlling the device. In the literature, a combination of both is usual. For example, in [141], the users control a virtual lower-limb along predefined trajectories by the training organizers. This means that users are "controllers", reacting to the given reference based on the training's state. A similar approach would be feasible with a mobile robotic device, where users should control it and react to its behavior changes and the changes in a training environment. Commonly, elderly or disabled people use some walking assistance device, so it is sensible to provide physical support by the training device. This means that physical contact between user and device is desired. Furthermore, the training should be simple and easy to understand so that no previous knowledge or skills are needed. However, more challenging tasks must be possible for keeping motor and mental reserves active as users improve.

Above discussion shows the main task of the training. Users have to navigate a SW in the training environment and, at the same time, the device modifies its behavior, so users are cognitively challenged. The user should guide the device along predefined lines, which serves as a ground-truth reference for the users and estimates the interaction quality and individual performance.

How should the user and robot interact?

The literature on Smart Walkers suggests that the most intuitive Human-Robot Interface for SWs is to measure user's intention forces, e.g., via force-torque sensor (FTS). The measured forces and torques are typically utilised to generate SW's velocity through admittance dynamics (cf. equation (2.12)). This dynamics simulates the movement of rolling and sliding real-world objects, e.g., rolling a shopping cart or sliding a box on the floor. As a result, the interaction with such Smart Walkers feels natural and intuitive. Furthermore, the literature proposes to use the *virtual force field* concept to deviate the walker's behavior from the user's input for assistance purposes. The same concept could be used to modify the device's behavior for training. To summarize, the main technical requirements for the interaction are a FTS-based interface, velocity controlled robot, and control based on the admittance dynamics.

Are there any technical requirements and constraints?

This section defines technical requirements and constraints for a mobile robot and environment. Those have to be considered when establishing the exact training concept.

1. A conventional mobile robot should be used as a proof-of-concept training device.
2. The device has an input interface with a force measuring tool, e.g., a force-torque sensor.
3. The device can modify the user's input to provide challenging training and avoid dangerous situations, e.g., collision with the environment.
4. The mobile robot should be omnidirectional to empower versatile movements and a variety of training tasks.
5. The robot should have some safety mechanism to prevent collisions with the environment and with persons nearby, e.g., training supervisors.
6. The device should store its internal and interaction data for later analysis and evaluation.
7. The training length is limited to the robot's autonomy, i.e., its batteries' capacity.
8. The robot should localize itself in the environment to evaluate the human-robot interaction (HRI) quality.
9. The training environment should be indoor and barrier-free.
10. The floor material is irrelevant, but it should not be too slippery.
11. There should be a possibility to mark reference positions or paths within the environment.

12. The training area should be large enough for users to navigate the device around, approximately 4×5m.
13. In the training environment, necessary infrastructure such as electricity and network connection should be provided.

The proposed system could be similar to Smart Walkers from the literature which use the force-torque sensor on their handles, e.g., [175, 6, 52, 77].

How could training look like?

During the training with a SW, the user navigates it along the markings on the floor. For easier orientation, there are also reference points on the robot. The deviation from those reference points could be one of the criteria for users' performance and quality of the HRI. The following proposition should be investigated to provide variety and fine-tuning to the training and the complexity of it by demanding different dexterity. The training's paths could have distinct geometrical forms, e.g., straight lines, circles, and curvatures with various radii. Different movement-directions of the robot should also be considered, i.e., forward, backward, and sideward. The device's manifold control strategies can also be included, e.g., changing apparent dynamics of the device, include minor disturbances, or unintuitive control. This is possible thanks to the interaction approach where the device's controller and actuators solely generate the device movement. Therefore, the device's controller has a decisive influence on users' input and manipulates itin order to achieve a specific interaction type. For example, the generated disturbances influence the device's movement by pulling it to a specific direction or providing additional resistance for the user. The disturbances have to be strong enough to be experienced by the user but sufficiently weak to not endanger them. In general, the unintuitive control means that users' input is transformed unexpectedly.

What are the expectations from the training?

A distinct feature of the training with RoboTrainer lies in the physical interaction, when users navigate freely within the training environment. This requires a certain amount of motor and spatial orientation skills from users. However, since the RoboTrainer is an autonomous robot, it could guide its user to train and increase these skills. The RoboTrainer can even provide a specific cognitive load to its user by reducing support and even make the solving of a task more complicated. In general, such a device could provide targeted physical and mental load during the interaction, trying to provoke long-term positive effects to the symptoms entailed by preconditions of dementia.

Another reason for performing training via direct contact with a robotic device is to increase the participants' engagement and motivate them to do it on a regular basis. It is anticipated that users' are keener on trying out and using the device because it acts in a physical world, unlike virtual tasks like in *serious gaming*. At best, users develop the ambition to learn and optimize the robot's precise control throughout their interaction.

Also, as the training advances and the users are confronted with more complex tasks, it is expected that their dexterity regarding the device's handling increases. This means that the tasks from the beginning of the training should be experienced less complex after some time. Hopefully, users' motor skills increase which leads to a performance increase in neuropsychological tests, as shown in research with holistic physical training [62, 140, 126].

3.2. Design of the RoboTrainer Prototype

This section describes the RoboTrainer Prototype as a framework used to realize the above described training approach. The RoboTrainer enables training evaluation in a pilot study. First, the device's hardware is presented in section 3.2.1, followed by the software and concepts used to fulfill the training in section 3.2.2.

3.2.1. Hardware

The basis of the RoboTrainer Prototype is a *rob@work 3* mobile platform (figure 3.1) developed by Fraunhofer-Institut für Produktionstechnik und Automatisierung (Fraunhofer IPA) in Stuttgart, Germany. The platform is a modified *Care-o-bot 3*®[2] base. It is used to research indoor logistics and as a mobile base for investigating industrial production scenarios. The platform is extended with a fixture for the user's input device consisting of a force-torque sensor and bike handles with ergonomic grips. The complete device as used in the pilot study is shown in figure 3.2.

The RoboTrainer Prototype has four independently controllable drive-steer modules that enable quasi-holonomic behavior. Strictly speaking, the kinematics is not holonomic because the wheels first have to change their orientation to drive in a particular direction. Nonetheless, the 360°-turnable wheels enable movement in every direction with any platform's orientation. Also, simultaneous rotation and translation movements in any direction are possible. The wheels' top speed is around 1.4 m/s which is more than the average comfortable walking speed of healthy people in their fifties and above [14]. Also, as classified by Solenne et al. [148], the device movement is almost non-existing

[2] Fraunhofer IPA: *Care-O-bot 3* website: `https://www.care-o-bot.de/en/care-o-bot-3.html` (accessed: 12.11.2020)

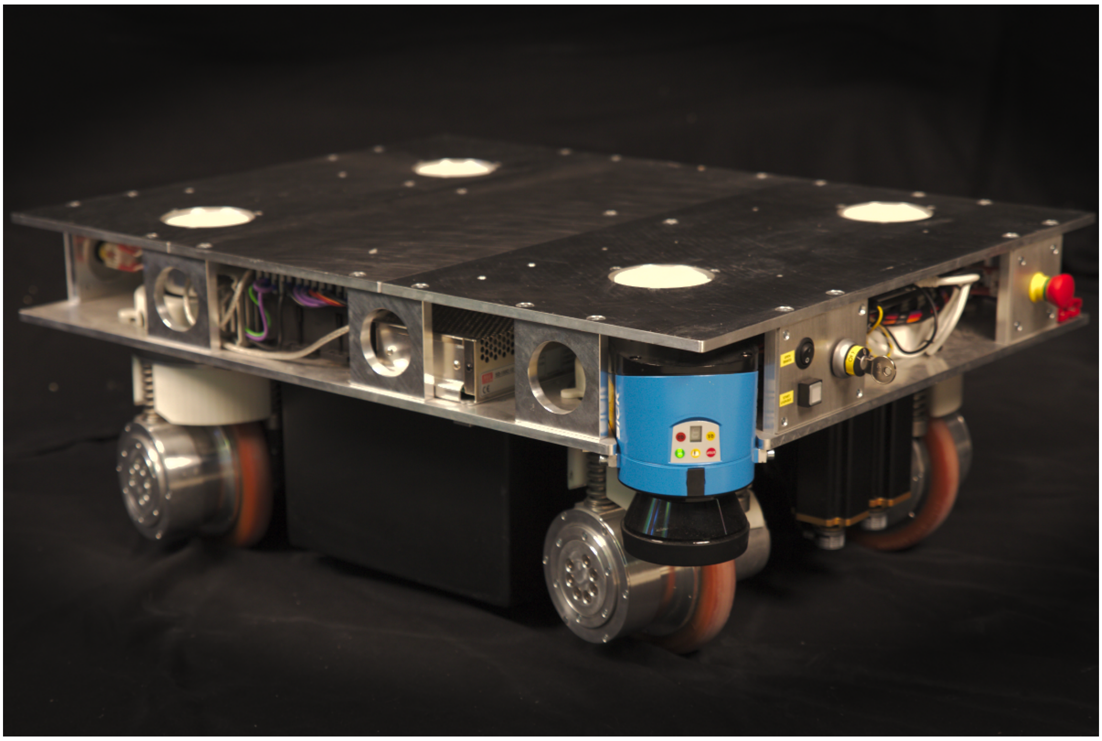


Figure 3.1.: *rob@work 3* mobile platform – the base for the RoboTrainer Prototype.

when changing the wheels' orientation using center-rotatable wheels The device's autonomy is supported with a battery pack of four lead-acid 12 V, 28 A h batteries assuring approximately four training hours. The two *SICK LMS-100* laser range finders on the base's diagonal corners provide a 360° view around the device. They are used to map and localize the training environment and for collision avoidance, i.e., emergency stop if the distance to an obstacle gets below a predefined threshold. The main parts of the user's input device are *ATI Mini58* force-torque sensor and bike handles. They are mounted on the base platform using aluminum strut profiles. The technical details of the FTS are given in Appendix section B.3. The sensor mechanically couples the base platform with the bike handles where the user has physical contact with the RoboTrainer.

The control of the RoboTrainer Prototype is done by an onboard PC, concretely, an *Apple Mac mini* computer with an Intel® Core™ i7−2635QM CPU @ 2.00 GHz and 4 GB of RAM. A solid-state drive of 250 GB maintains the control software and recorded data during training.

3.2.2. Software and Control of RoboTrainer Prototype

The software of the RoboTrainer Prototype is realized using the Robot Operating System (ROS)-framework.[133], which is supported natively by the *rob@work*. ROS provides

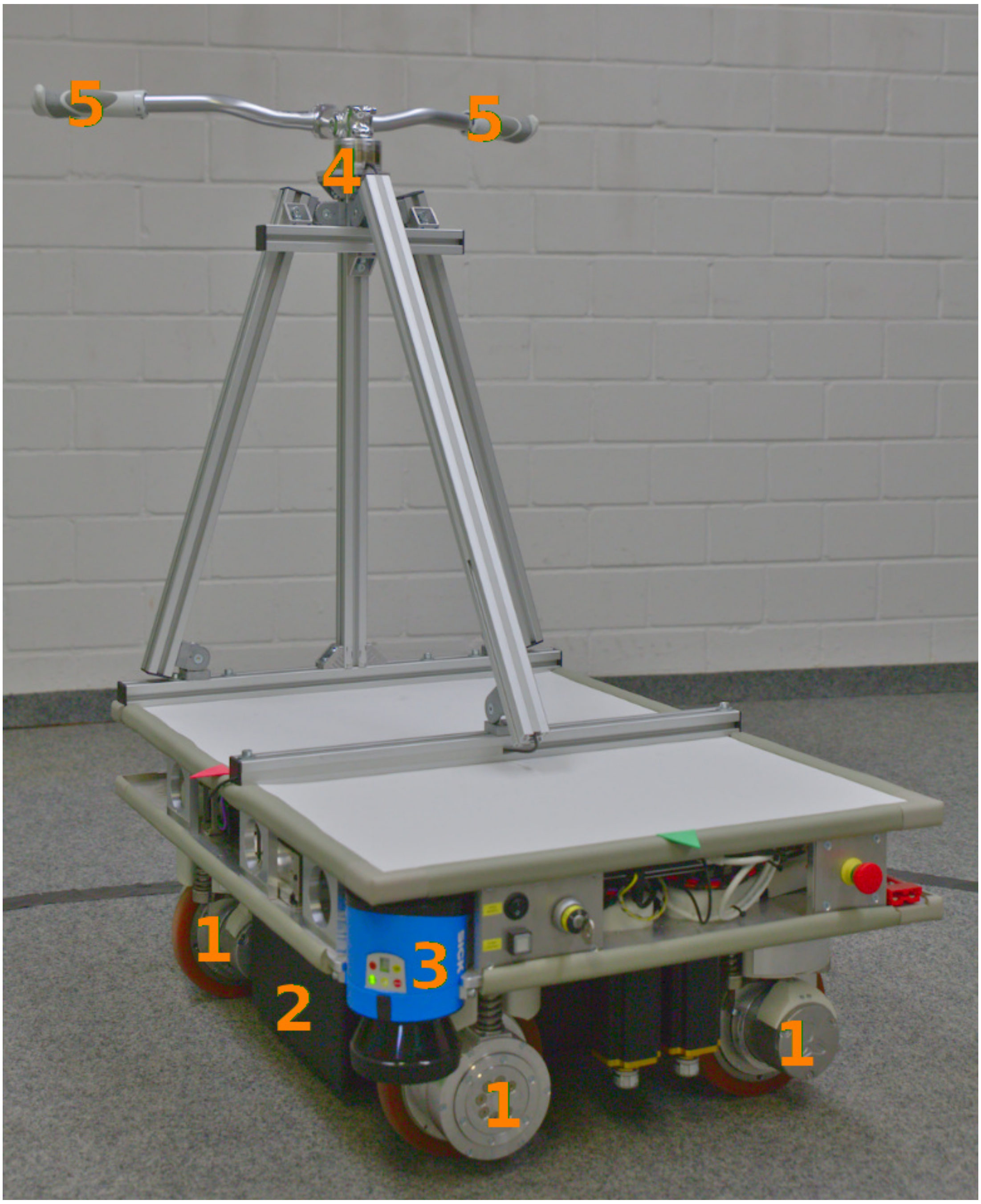


Figure 3.2.: The RoboTrainer Prototype as used in the HEiKA pilot study. Its main components are independently controllable drive-steer modules (1), a rechargeable battery pack (2), SICK LMS100 laser range scanners (3), a force-torque sensor (4), and a bike handlebar for controlling the robotic device (5).

various tools for prototyping robotic applications and standardized communication between sensors and actuators in a robotic system. The ROS is an open-source software and defacto standard software in robotic research, enabling easy collaboration between research institutions and simple reuse of scientific achievements from other researchers. This empowers users to focus on their algorithms and applications.

The RoboTrainer Prototype's controller, based on admittance dynamics, and high-level training functionalities are realized using ROS. Besides that, the ROS-tools *rviz* and *rosbag* are used for online visualization and storage of training data.

Control of RoboTrainer Prototype

The admittance control model is used for RoboTrainer Prototype because of its simplicity and intuitiveness. As described in section 2.5, the model provides naturally stable and passive behavior of first-order dynamics, simply adjustable by using only two parameters. The RoboTrainer's three degrees of freedom (DOFs), i.e., linear movement on the 2D-ground plane and rotation around the vertical axis, are controlled independently using different parameters. RoboTrainer Prototype uses an open-loop architecture from the control theory's perspective. To implement the controller on a PC, the time-continuous form is discretized using ZOH transformation to preserve its dynamic properties (see section 2.5). The initial controller is parameterized using a *MATLAB* [115] simulation to achieve the desired dynamic behavior. The fine-tuning is done directly on the device during the testing phase before the study. In ROS terms, the controller is implemented as an independent process, a *node*, communicating via ROS-*topics* with the sensor driver and the low-level controller. Although this architecture does not guarantee real-time constraints and data may be lost in the control loop, the approach was feasible and robust enough for the study. The controller's parameters are configured using ROS-*Parameters* infrastructure. Figure 3.3 shows the RoboTrainer Prototype's control architecture.

The user's input force $\mathbf{F}_\mathrm{h}$ arrives from the ATI-FTS-driver node over ROS-topics to the controller. The force is then clamped and normalized with $\mathbf{F}_{\mathrm{h}_\mathrm{max}} = 100\,\mathrm{N}$. The influence of the *artificial force* concept is superposed on the user's input before applying the admittance equation. After the admittance rule, the *inverted controls* concept is applied if activated. Before publishing the reference velocity to a ROS-topic for the low-level, *rob@work*-platform's controller, the output velocity is denormalized and clamped with $V_{max} = 1.2$ m/s. The admittance rule equation is calculated using the time-discrete form presented in equation (2.22) in section 2.5. For the localization, *amcl*[3]-ROS package is used. More details about *artificial force* and *inverted controls* concepts are given in the next section.

[3] ROS wiki: AMCL documentation: `https://wiki.ros.org/amcl` (visited: 15.11.2020)

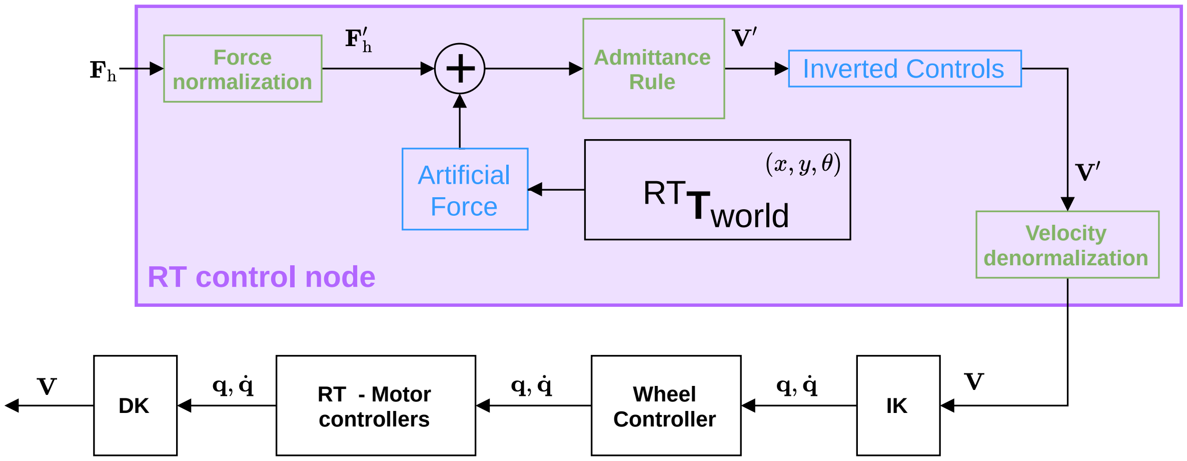


Figure 3.3.: The RoboTrainer Prototype's control architecture. The user's input force F_h arrives from the external node using ROS-topics to the control node. After entering the controller, the input force is clamped and normalized. Before applying the admittance rule, the input force is modified by the *artificial force* concept. The *inverted controls* concept is applied if so configured after calculating the RoboTrainer's velocity using admittance dynamics. The resulting output velocity is then denormalized, clamped, and sent to the output ROS-topics for the low-level *rob@work* controller.

High-Level Control Concepts

Ahead of the study, a small environment, shown in figure 3.4, was built to enable initial tests with the RoboTrainer. The environment is bounded by fake walls visible to the laser range finders (LRFs). This environment allows tests with consistent LRF-scans which would not be possible in IAR-IPR's robot laboratory. In the test environment, a training reference path is defined.

Figure 3.5 shows the virtual representation of the test environment from figure 3.4. The visualization is done using ROS'es 3D visualization tool – *rviz*. *rviz* enables visualization of data in the ROS system, like robot models, sensor data, and user-defined data, e.g., disturbance forces. The presented virtual path has to correspond to the actual environment in order to determine the user's precision when navigating RoboTrainer. The following approach is used to ensure this: The RoboTrainer is manually driven via a joystick along the predefined path in the real environment. The points where a path changes its direction are recorded, and the course is linearly interpolated between them. Therefore, in curves, the distances between recorded points are smaller which helps to achieve an appropriate curvature approximation.

The high-level control concepts, later called control actions (see chapter 6), extend the base admittance-dynamics control to enable versatile training tasks. The first used con-

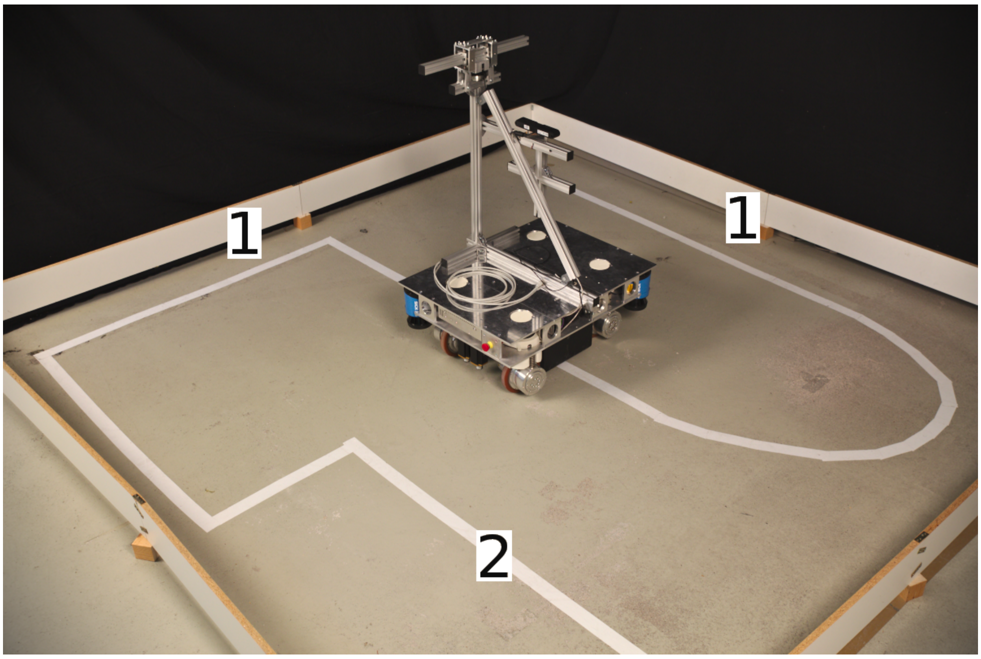


Figure 3.4.: The RoboTrainer Prototype's test environment at the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics (IAR-IPR). The environment is enclosed by the "walls" (1) visible to the laser range finders. A reference training path is marked with white tape on the floor (2). The device shown in the figure is one of the versions before the final prototype shown in figure 3.2.

cept is called *inverted controls*, which reverses the device's sideward controls. This means that the input force given to the left results in the RoboTrainer's movement to the right. The main parameter for this behavior is maximal sideward speed. The parameter should be set according to the user's skills and it scales the admittance rule's parameters to adapt the device's dynamics appropriately. A lower parameter value for maximal sideward velocity produces higher damping and mass of the admittance-controlled system, i.e., slower dynamics. Parameter tuning is done ahead of the training to keep the parameters equal for all participants.

The second concept is called *artificial forces*. It enables the utilization of a disturbance force which modifies the RoboTrainer Prototype's behavior, for example by pulling it away from the predefined path. An *artificial force* is defined in the virtual environment (see figure 3.5) and is therefore not visible for users. The goal is to provide unanticipated behavior of the RoboTrainer, which users need to detect and correct. The functionality of the *artificial force* concept is depicted in figure 3.6. An *artificial force* is defined with a force vector and its influence area. The area is used to increase the effects and simplify the calculation of the force's influence. One can imagine it as a slope in the virtual force

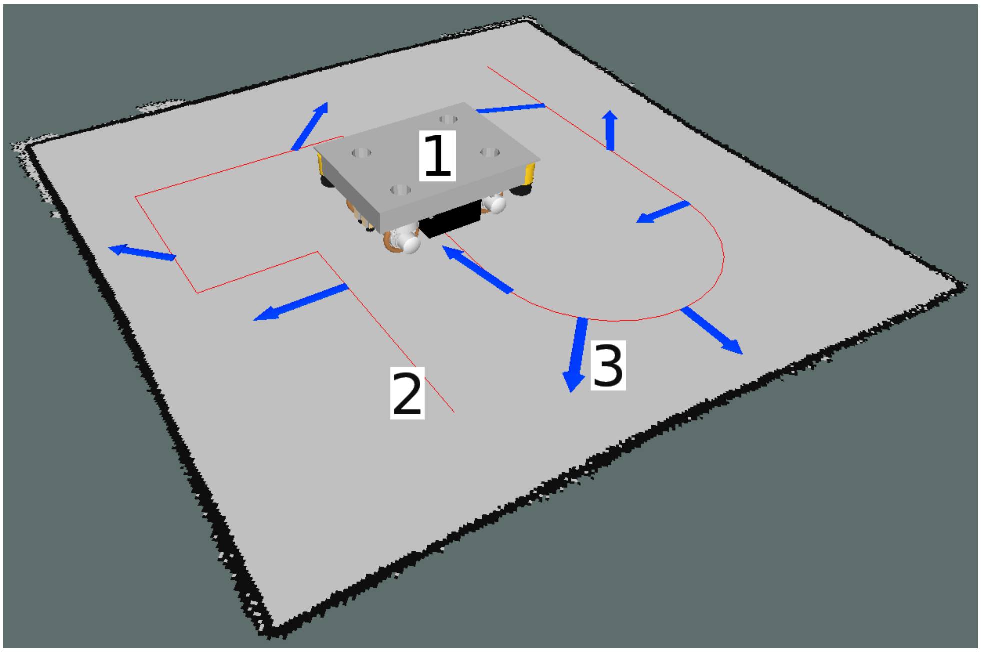


Figure 3.5.: The virtual representation of the test environment at the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics (cf. figure 3.4). The figure presents the robot model (1), representation of the virtual path (2), and markers for disturbance forces (3).

field. As long as the user is inside the influence radius, the *artificial force* is superposed on the user's input using equation (3.1).

$$\mathbf{F}'_{\mathrm{h}}(t) = \mathbf{F}_{\mathrm{h}}(t) + \alpha \mathbf{F}_{\mathrm{a}}(t) \tag{3.1}$$

The $\mathbf{F}'_{\mathrm{h}}$ is the user's input force $\mathbf{F}_{\mathrm{h}}$ modified by the *artificial force* $\mathbf{F}_{\mathrm{a}}$. The factor $\alpha \in [0, 1]$ controls the artificial force's influence, i.e., the ratio between the user's input and the artificial force. The *artificial force*'s amplitude has a trapezoidal profile, as shown in figure 3.6 (blue line), to reduce the jerk when entering and exiting its influence radius. A detailed discussion about that is given in section 6.1.2.

3.3. Evaluation of the Concept

The above-presented training and device concepts were evaluated in a pilot study with ten elderly participants. In the following, the pilot is also referred to as the *HEiKA experiment*. The main goal was to investigate the feasibility of the proposed training as well as users' reactions to the training device. The concrete investigated questions are the following [156, 154]:

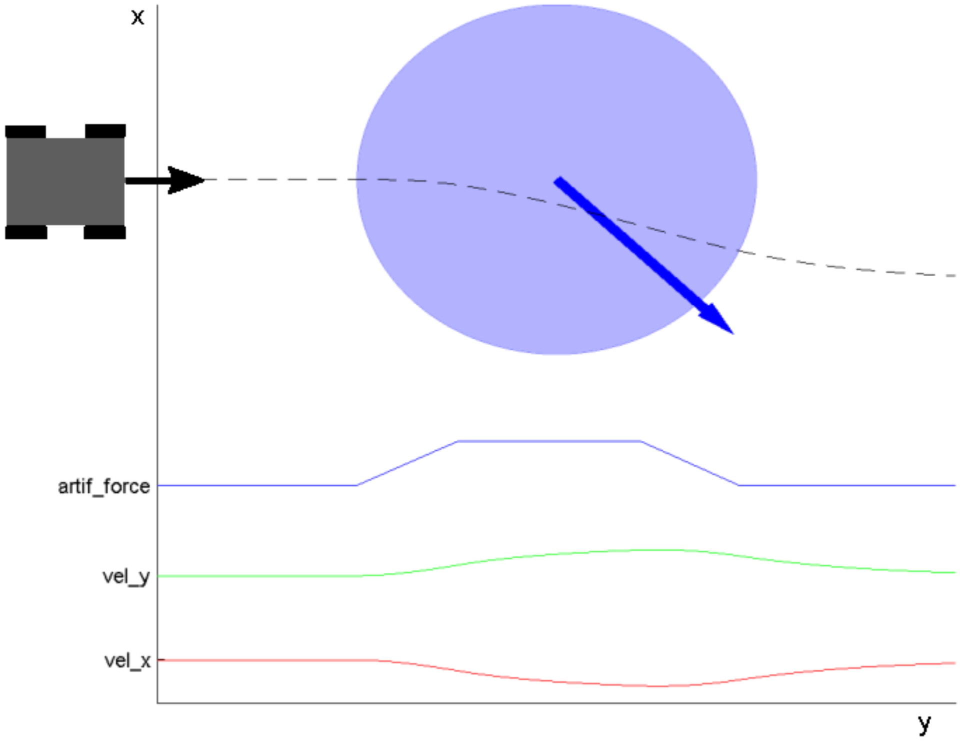


Figure 3.6.: The concept of *artificial forces* developed for the RoboTrainer Prototype. An *artificial force* is represented by the blue arrow, whereas the direction is defined by the orientation and the strength by its length. The blue circle represents the influence radius of the force. The graphs represent the trapezoidal profile of the artificial force (blue line) and changes of the device's velocity in x- (green line) and y-axis' (red line) direction by a constant input force. Author: Oliver Armbruster [10, 154].

1. How willing are people with MCI to use a robotics-based training device and how could the device be improved to increase users' acceptance?
2. Could possible effects on the participants' motor level be detected with such a training device?
3. How would users react to control concepts for adjustment of the complexity of their training?
4. What knowledge about users and their performance can be gained from the device's onboard sensors?
5. What can be said about the robustness of the device and user's safety during real training situations and how can this be further optimized?

In the long term, the experiment investigates the potential to use such robotic devices, i.e., Smart Walkers, in treatments for persons with MCI. Therefore, the participants went through comprehensive neurological and psychological screenings before and after the training sessions. Those screenings are not part of this thesis, and thus their results are not presented.

3.3.1. Experiment

The main challenge for the users is to navigate, i.e., maneuver the RoboTrainer along the paths marked on the floor. RoboTrainer is controlled by applying forces onto its handlebar, which demands precision and coordination to move it in the desired direction. Consequently, the user's physical activity is stimulated by walking and cognitive efforts are stimulated by demanding and versatile tasks, e.g., counter-intuitive controls.

The training consisted of multiple one-hour training sessions with ascending complexity. Further, each session consisted of various tasks or exercises, which are repeated three times each. An exercise is defined by a specific path and any combination of the device's capabilities:

(i) forward and backward movements;

(ii) sideward movements;

(iii) rotation along any curve in space;

(iv) inversion of sideward controls;

(v) placing the *artificial forces* along a path.

The procedure of the whole experiment is shown in figure 3.7. The following sections describe it in detail.

Study population

The ten participants, eight males and two females, were recruited by the Central Institute of Mental Health (CIMH). The participants were between 55 to 78 years of age, with an average of 70.7 ± 6.34. All participants were diagnosed with mild cognitive impairment (MCI). Their initial suitability was verified using telephone screening. The participants had a negative history of medical conditions, neurological brain diseases, or mental disorders. Testing the RoboTrainer and choosing rather healthy elderly persons for the test training reduces the risk of accidents and, potentially, provides better feedback regarding interaction with the robotic device.

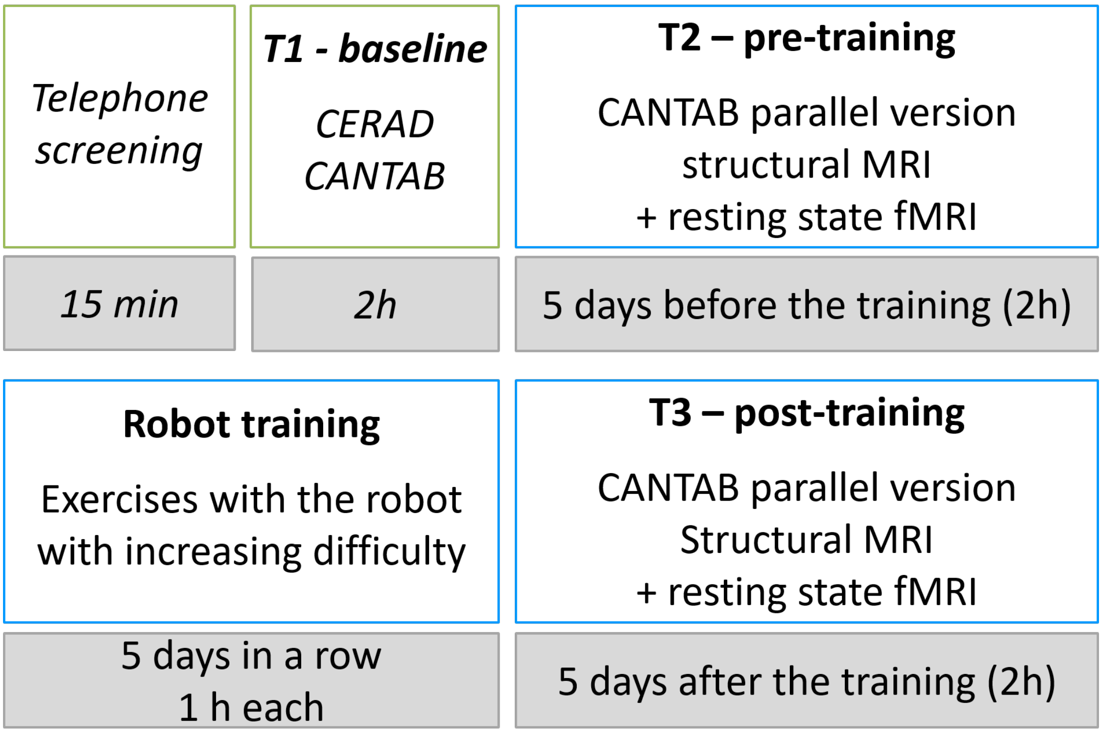


Figure 3.7.: Individual steps of the *HEiKA experiment*. The first two phases were finding and filtering suitable participants for the training with RoboTrainer. The tests *T2 - pre-training* and *T3 - post-training* were used to compare participants' states before and after the training with the RoboTrainer. Source: [154].

The *T1-baseline* tests provide an objective evaluation of suitability for the training, i.e., the occurrence of MCI only, not any severer condition or disease. During this phase, the participants were tested with the following procedures: Consortium to Establish a Registry for Alzheimer's Disease (CERAD-Plus) [121], Cambridge Neuropsychological Test Automated Battery (CANTAB), and Wechsler Memory Scale – Revised (WMS-R) [177]. The general MCI criteria are used according to Winblad et al. [179]. The participants are diagnosed with MCI if they are tested positively against the following criteria: (i) subjective memory complaint (corroborated by an informant if possible); (ii) objective memory impairment (abnormal for their age); (iii) essentially preserved general cognition for age; (iv) intact functional activities of daily living; and (v) no dementia present.

Training with RoboTrainer

The training with RoboTrainer was conducted in the basement of CIMH. Those premises allowed the training without disturbance from people not involved in the pilot study. In

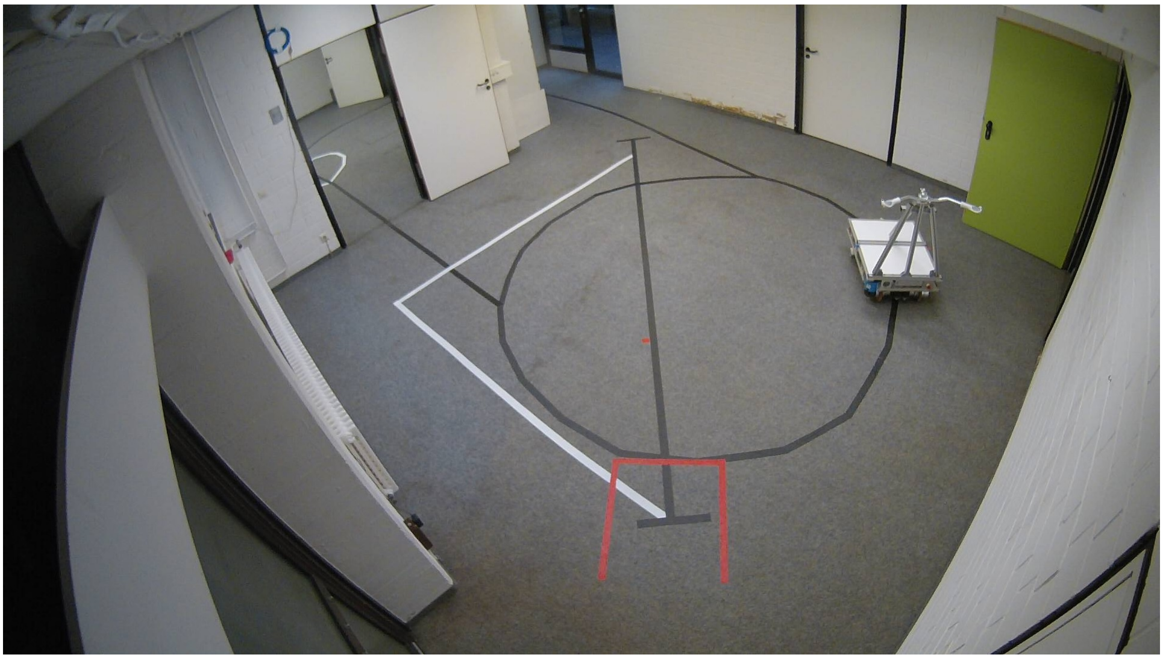


Figure 3.8.: The premises at CIMH where the training with the RoboTrainer is undertaken. This is the view of the main room. The second room is partially visible in the upper part of the image. The training paths are marked with black and white tape on the floor. The red markings represent the RoboTrainer's start position. Source: [154].

the weeks before (*T2-pre-training*) and after (*T3-post-training*) the experiment, additional neuropsychological screenings were conducted to compare the users' condition before and after the training. Those medical results are not presented and discussed here because they are out of this thesis's scope.

The training area is depicted in figure 3.8. It consisted of two rooms and a hallway connected with doors. On the floor of the premises, five training paths were defined. Those are depicted in figure 3.9, shown in the environment's virtual representation. The view on the main room depicted in figure 3.8 is from the corner where the number "1" is placed in figure 3.9. The virtual representation was created by mapping the environment using *gmapping*-ROS package. The marked paths were then recorded, navigating the RoboTrainer with a joystick using the procedure explained in section 3.2.2.

The training with the RoboTrainer consisted of multiple exercises, i.e., tasks, with ascending complexity. The complete overview of them is given in table 3.1. The individual paths and *artificial forces'* setup are shown in figures 3.10 and 3.11. The training began with more straightforward tasks where only one-dimensional movement was necessary, e.g., a line. The second path was a circle where a combination of linear movement and rotation was required. At the end of the first training session, *artificial forces* along the

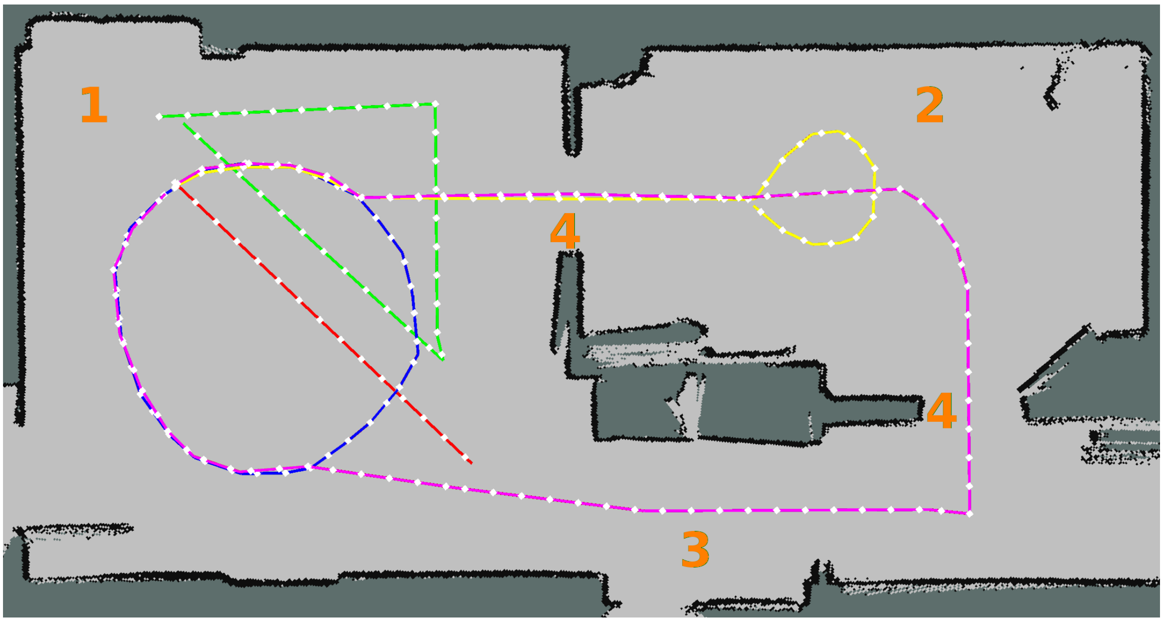


Figure 3.9.: Virtual representation of the paths used in the pilot study in a 2D floor plan. The orange numbers mark two rooms (a larger (1) and a smaller one (2)), a corridor (3) and doors between them (4) The virtual paths are named as follows: *c1* line (red), *c2* circle (blue), *c3* triangle (green), *c4* short path (yellow), *c5* long path (violet). Source: [154].

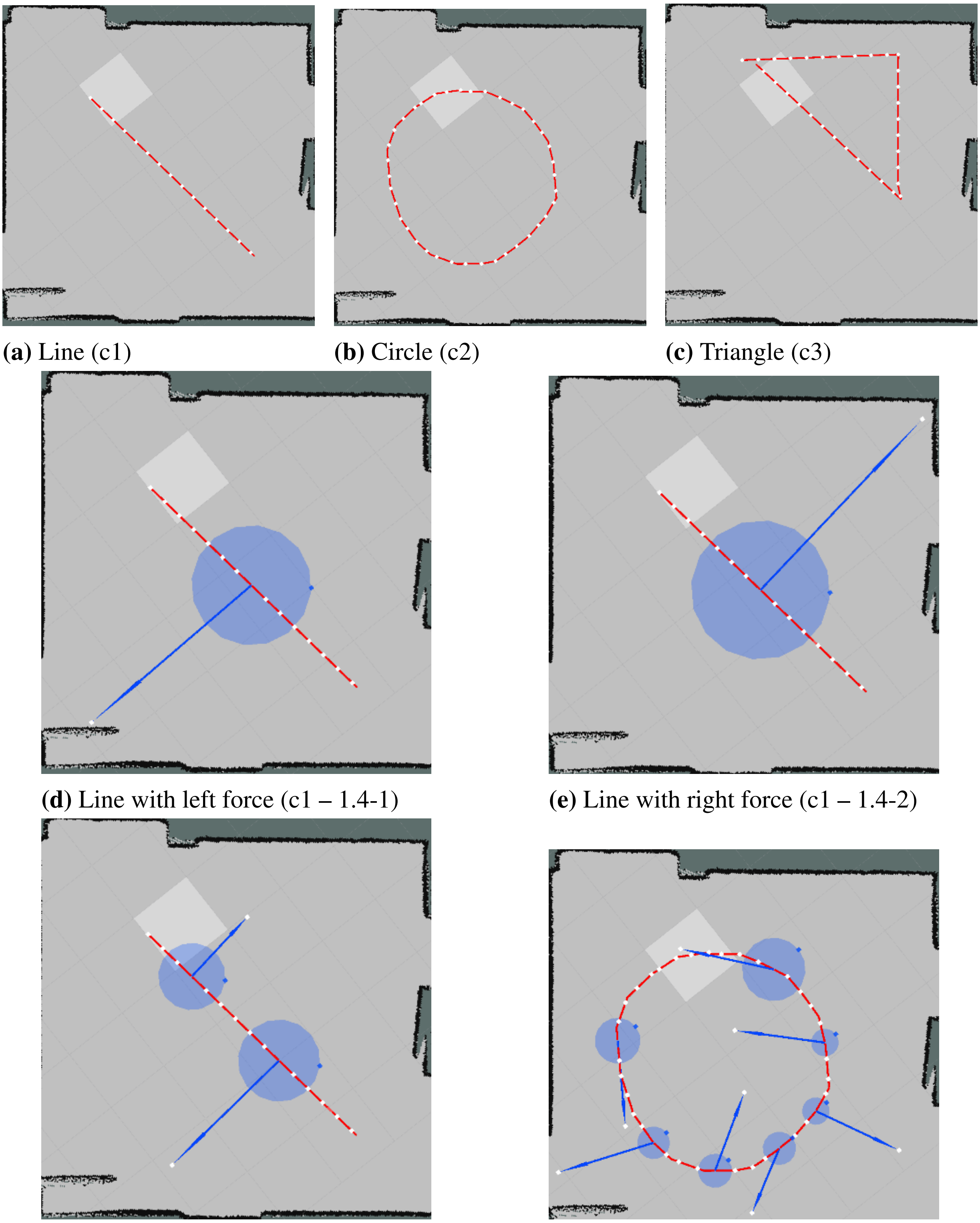
path "line" (c1) were introduced. On the second day of training, users had to go through the door with RoboTrainer and maneuver it sidewards. On the third day, the users, for the first time, did the long path with and without disturbance forces. At the end of the session, *inverted controls* concept was introduced to them. On the fourth day, the participants accomplished all the exercises with reversed controls. On the fifth day, the last day, participants repeated the most relevant exercises from the training week. This repetition day compared users' performance during the same exercise at the different training stages.

Each training session, except the first and the last, began with a ten-minute-long repetition of exercises from the previous day. In the following 40 minutes, the participants repeated each exercise at least three times. A training session is finalized with the ten-minute-long repetition of all exercises from that day. The first task repetition is done for users' to warm up and get used to it. The last repetition block is done to compare the users' performance change within a training day.

Table 3.1.: Overview of the training with RoboTrainer Prototype. The paths are numbered as in Fig. 3.9. The *"Direction"* column abbreviations are fwd - forward, bwd - backward, swd - sideward. The *"# Forces"* column represents the number of *artificial forces* distributed on a path. The *"Inverted"* column marks if the *inverted controls* concept was activated for the left/right control (y-axis). Source: [154].

ID	index	description	path	direction	# forces	inverted
Day 1						
	general instructions and getting to know the device					
1.1	1	line (no turning)	c1	fwd/bwd	0	no
1.2	2	line	c1	fwd	0	no
1.3	3	circle	c2	fwd	0	no
1.4-1	4	line w/ forces	c1	fwd	1★	no
1.4-2	5	line w/ forces	c1	fwd	1♦	no
1.4-3	6	line w/ forces	c1	fwd	2♢	no
1.5	7	line fwd & bwd w/ forces	c1	fwd/bwd	2♢	no
	repetition of the tasks of day 1					
Day 2						
	repetition of the tasks of day 1					
2.1	8	2-room path	c4	fwd	0	no
2.2	9	triangle	c3	swd/fwd/bwd	0	no
2.3	10	circle sidewards	c2	swd	0	no
2.4	11	reversed training		swd	0	yes
	repetition of the tasks of day 2					
Day 3						
	repetition of the tasks of day 2					
3.1	12	full path	c5	fwd	0	no
3.2	13	full path w/ forces	c5	fwd	5	no
3.3	14	8-shape around obstacles		fwd	0	no
3.4	15	line fwd & bwd reversed	c1	fwd/bwd	0	yes
	repetition of the tasks of day 3					
Day 4						
	repetition of the tasks of day 3					
4.1	16	line sidewards reversed	c1	swd	0	yes
4.2	17	circle sidewards reversed	c2	swd	0	yes
4.3	18	triangle reversed	c3	swd/fwd/bwd	0	yes
4.4	19	360° turning			0	yes
	repetition of the tasks of day 4					
Day 5						
	repeat the full path as warm-up					
5.1		line fwd & bwd w/ forces	c1	fwd/bwd	2♢	no
5.2		2-room path	c4	fwd	0	no
5.3		triangle	c3	swd/fwd/bwd	0	no
5.4		circle sidewards	c2	swd	0	no
5.5		full path w/ forces	c5	fwd	5	no
5.6		8-shaped around obstacles		fwd	0	no
5.7		line sidewards reversed	c1	swd	0	yes
5.8		circle sidewards reversed	c2	swd	0	yes
5.9		triangle reversed	c3	swd/fwd/bwd	0	yes
5.10		circle w/ forces	c2	fwd	7	no

★ forces from figure 3.10d; ♦ forces from figure 3.10e; ♢ forces from figure 3.10f



(a) Line (c1)

(b) Circle (c2)

(c) Triangle (c3)

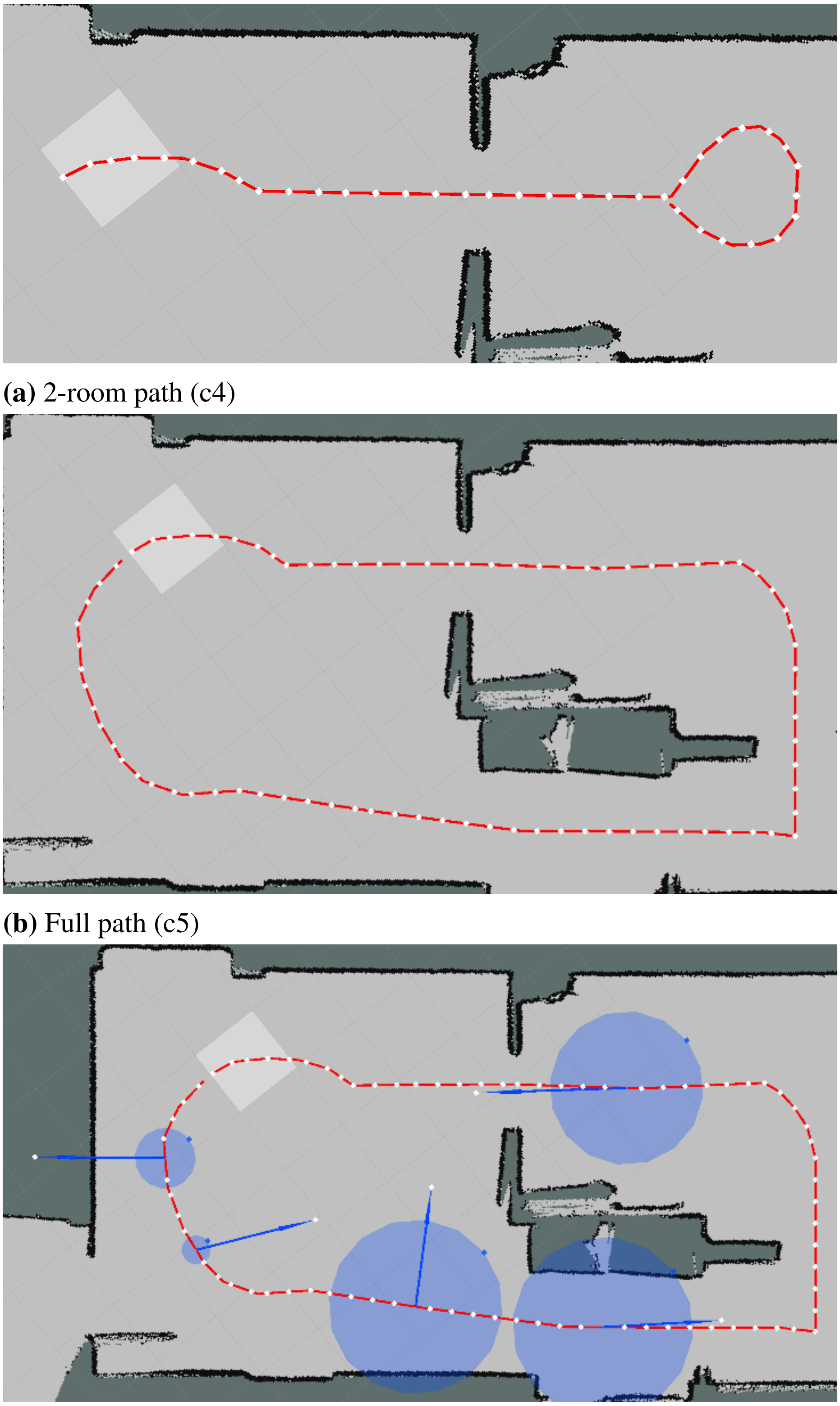
(d) Line with left force (c1 – 1.4-1)

(e) Line with right force (c1 – 1.4-2)

(f) Line with two forces (c1 – 1.4-3)

(g) Circle with forces (c2 – 5.10)

Figure 3.10.: Path configurations in the pilot study (1/2). Path's name and ID are given. If suitable, also the task's ID is noted. The red lines represent the paths, the blue arrows represent the directions, and the blue circles represent the *artificial forces*' influence-radii.



(a) 2-room path (c4)

(b) Full path (c5)

(c) Full path with forces (c5 – 3.2)

Figure 3.11.: Path configurations in the pilot study (2/2). Name and path's ID are given. If suitable, also the task's ID is noted. The red lines represent the paths, the blue arrows represent the directions, and the blue circles represent the *artificial forces*' influence-radii.

Measurements

During the experiment, all data available to the RoboTrainer were recorded using the ROS logging tool *rosbag*. This means that each user's interaction for every exercise was saved and could be playbacked later for analysis. The most important of those are RoboTrainer's position to calculate the users' navigation precision, RoboTrainer's velocity, and interaction forces. Besides the precision of navigation of the RoboTrainer, users' time to finish an exercise was measured manually, and users had to answer a questionnaire after each exercise.

The participants' main task was to navigate the RoboTrainer along the predefined paths as precisely as possible. Therefore, the deviation from those paths was the main evaluation criteria. The deviation was calculated periodically as the shortest distance between the markers on the RoboTrainer (green and red arrows in figure 3.2) and the virtual representation of the paths. The deviation for a task was calculated as a weighted sum of all deviations during the task according to equation (3.2). The calculation was done between start t_1 and end time t_2 using time step Δt. The velocity correction factor $\frac{v_t}{v_{max}}$ normalizes the actual velocity v_t with the device's maximal possible velocity v_{max} for the specific direction. A lower value represents a better control performance.

$$deviation_{t_1-t_2} = \sum_{t \in T} d_t \cdot \frac{v_t}{v_{max}} \tag{3.2}$$

$$T = [t_1, t_1 + \Delta t, t_1 + 2\Delta t, \ldots, t_2 - \Delta t]$$

Participants' time performance for a task was measured during the second repetition of the first attempt, the attempt at the end of the same day, and the last day. Although the task execution time was not set as a goal for the participants, it provided valuable insights into users' performance, especially its alteration. The users naturally wanted to finish the task fast.

Besides presented objective variables, the users were asked to evaluate each task's complexity and personal performance. The answers were recorded at a Likert-type scale with the values: 1 - very easy, very good up to 5 - very complex, very bad, and 6 - not solvable, unsatisfied.

At the end of each training day, the users were additionally asked the following three questions:

1. How easy was the handling of the device for you today?
2. How safe did you feel while handling the device?
3. How fast could you accustom to handling the device?

The rating was also done using a Likert-type scale with values: 1 - very easy, very safe, very fast, up to 5 - very complex, very unsafe, very slow, and 6 - not solvable, extremely unsafe, not at all.

Statistical analysis of the results was done for the participants' deviation and time performance, where mean values and standard deviation for each exercise and its repetitions were calculated. The statistical significance was determined by comparing the first attempt of the task (those with time measurements, i.e., second repetition) and the attempt on the last day using the ANOVA method. The distribution of the variables was tested using Kolmogorov-Smirnov-Test. A p-value ≤ 0.05 was considered statistically significant. The Python Library *SciPy v1.1.0*[4] was used for this analysis. The users' answers were analyzed by calculating the median and range for each question.

3.3.2. Results and Discussion

The pilot study's results are evaluated in four main categories:

(1) precision of controlling the RoboTrainer Prototype, i.e., deviation;

(2) time performance, i.e., the time needed to accomplish the task;

(3) user experience from questionnaires;

(4) influence of high-level control concepts, i.e., *artificial force* and *inverted controls*.

This section is structured according to those criteria and finalized with an overall evaluation and discussion.

Measurements of the first attempt (second repetition), the attempt at the end of the same day, and the last attempt at the last training day are analyzed to compare users' performance. Those values are compared with regard to the deviation and time measurements, whereas the questionnaire's answers are compared for the tasks' first attempts and the attempts on the last day.

The Precision of Controlling the RoboTrainer Prototype – Deviation

For control precision, per-task mean values of the users' deviation from the predefined paths are evaluated. The data are shown in three different forms in figures 3.12 to 3.14 for emphasizing various aspects of the training and the results. Figure 3.12 shows the average deviation for all participants and training tasks, i.e., the exercises. Figure 3.13

[4] https://www.scipy.org

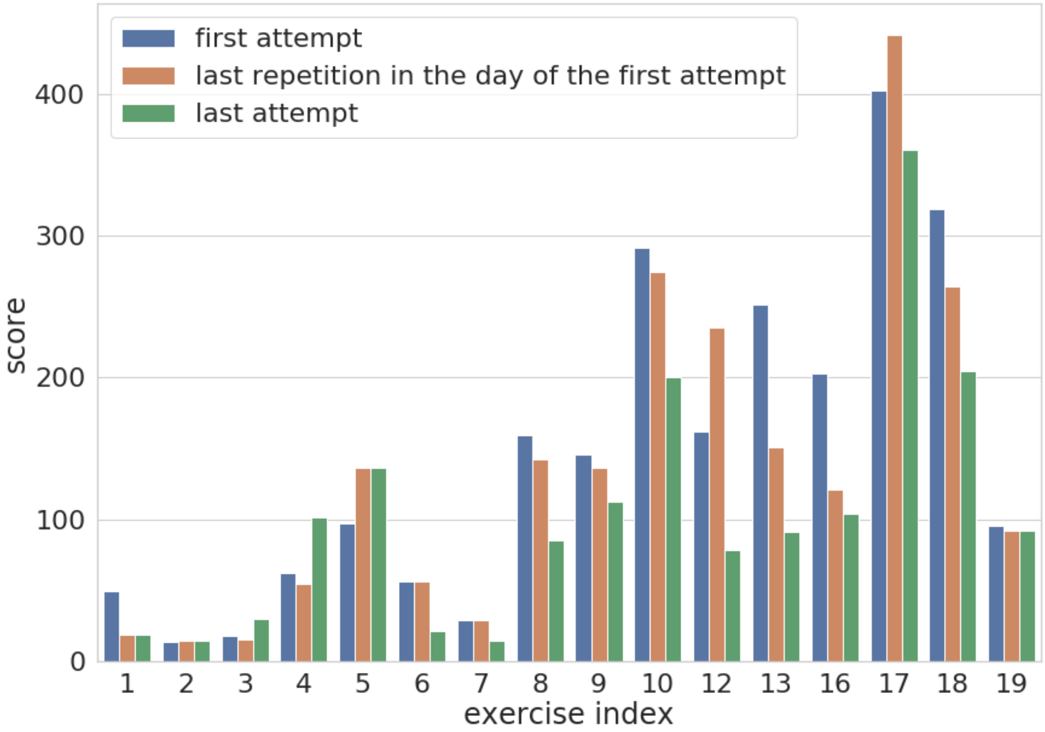


Figure 3.12.: HEiKA Experiment: Average deviation (score) for all participants and exercises. The results show measurement during the first attempt (second repetition), the last repetition on the day when exercise is introduced, and the last repetition (on the same day or the last day). The exercises are indexed according to table 3.1. Author: Xingbo Wang [172, 154].

shows the same data for the first and the last exercise attempts in boxplot[5] form to emphasize the dispersion of measurements between participants. Finally, figure 3.14 compares closely different attempts of the same tasks. The significant differences between the first attempt and the attempt on the last day are highlighted. The exact significance results from the ANOVA analysis are given in table 3.2.

Figure 3.12 indicates that the deviation correlates with the training paths' lengths, showing minimal results on the first day compared to the following days. The first three exercises were probably simple for users since there is no considerable difference between them. On the first day, the first two exercises with *artificial forces*, i.e., exercise 4 and 5 with one stronger *artificial force*, shown in figures 3.10d and 3.10e, show larger

[5] Appendix section C.1 provides a detailed explanation of the boxplot representation.

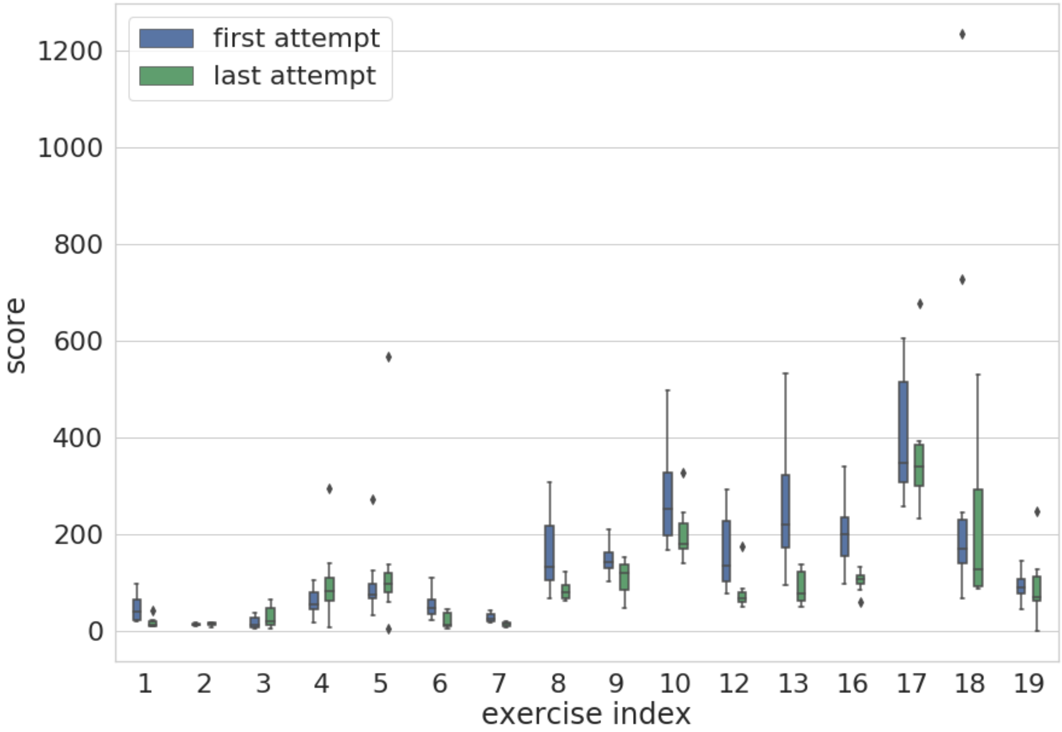


Figure 3.13.: HEiKA Experiment: Average deviation (score) for all participants and exercises. The results show measurement during the first attempt (second repetition) and the last attempt (on the same day or the last day) in boxplot form. The boxplots are defined as explained in section C.1. The exercises are indexed according to table 3.1. Author: Xingbo Wang [172, 154].

deviations than those without *artificial forces*. Also, the exercises with two forces on the first day (figure 3.10f, on the first day, i.e., the last two tasks, show a slightly larger deviation than the exercises without forces, but much smaller deviation than the exercises with one *artificial force*. This confirms the hypothesis that weaker forces result in smaller deviations and suggest that they were possibly too weak in the case of two forces. Still, it shows that a task's physical difficulty can be adjusted with the strength of the *artificial forces*. On the second day, tasks 8, 9, and 10 show the expected deviation changes between multiple repetitions: the decreasing deviation over different attempts. The absolute values cannot be directly compared since each of those exercises uses a different reference path. Task 11 is not shown since it was intended as a "warm-up" for the *inverted controls* concept. The same applies to exercise 14 (3.3), which did not have a reference path. This was avoided for safety reasons since the users would be in a narrow space between RoboTrainer and a wall. The complete path tasks 12 and 13 were the longest. Still, they show medium deviation. This is probably due to the higher velocity with which users could navigate RoboTrainer during exercise since these task have long straight sections and curves with a larger radius. This leads to lower calculated deviation, as shown in equation (3.2). It also seems that the *artificial force* concept did not significantly influence the deviation. The reversed tasks on the fourth day have much more significant deviations than the exercises without *inverted controls* on the same reference paths, i.e., tasks 10 and 17, 9 and 18. Task 19 measured how well the participants can rotate the robot around its center, so these results show the RoboTrainer's center's deviation from its initial position.

Figure 3.13 shows the participants' mean deviation values' dispersion during the first and last exercise attempts. For almost all exercises, the deviation's dispersion is smaller for the last attempt than for the first one. This is shown by smaller boxes which indicate that 50 % of the participants have closer mean values, and by the closer whiskers to it, representing the dispersion of minimum and maximum values. It is interesting to observe that high outliers are present almost only in the tasks with *artificial forces* and *inverted controls*. Comparing the tasks with the *artificial forces* on the first day, users get used to those over time. This can be observed in task 7 with two *artificial forces* and smaller values' dispersion compared to exercise 1. Also, a clear difference between tasks with *artificial forces* 4 - 5 and 6 - 7 can be observed concerning the value dispersion and the number of outliers. The small dispersion of users' values in exercises 6 and 7 suggest that the *artificial forces* were too weak. In the representation in figure 3.13, the influence of the artificial forces concept is more visible, e.g., by comparing tasks 2 and 6 and 12 and 13. The users have more dispersed deviations in tasks with *artificial forces*. The same applies to the exercises with *inverted controls*, e.g., 9 and 18 and 10 and 17.

Comparing the deviation in different repetitions, i.e., attempts, in figure 3.14, the users showed significant improvement in precision of the control of RoboTrainer in all tasks, except 4.2/5.8 (17) and 4.3/5.9 (18) (see table 3.2). Compared to the first attempt, the tasks' average deviation is worse during the last attempt on the first day for tasks 1.5/5.1 (7) and 4.2/5.8 (17). For the second case, i.e., task 4.2/5.8 (17), this can be explained by participants' fatigue since they had to use the *inverted controls*-a concept during the whole session. Other tasks show expected improvement with increasing precision, i.e., decreasing the deviation (score) with each repetition. This representation shows the influence of *inverted controls* on the control precision, especially emphasizing the difference between users, resulting in more considerable variance. Also, this variance is the reason why the two exercises with *inverted controls* do not significantly differ between the first and last attempts.

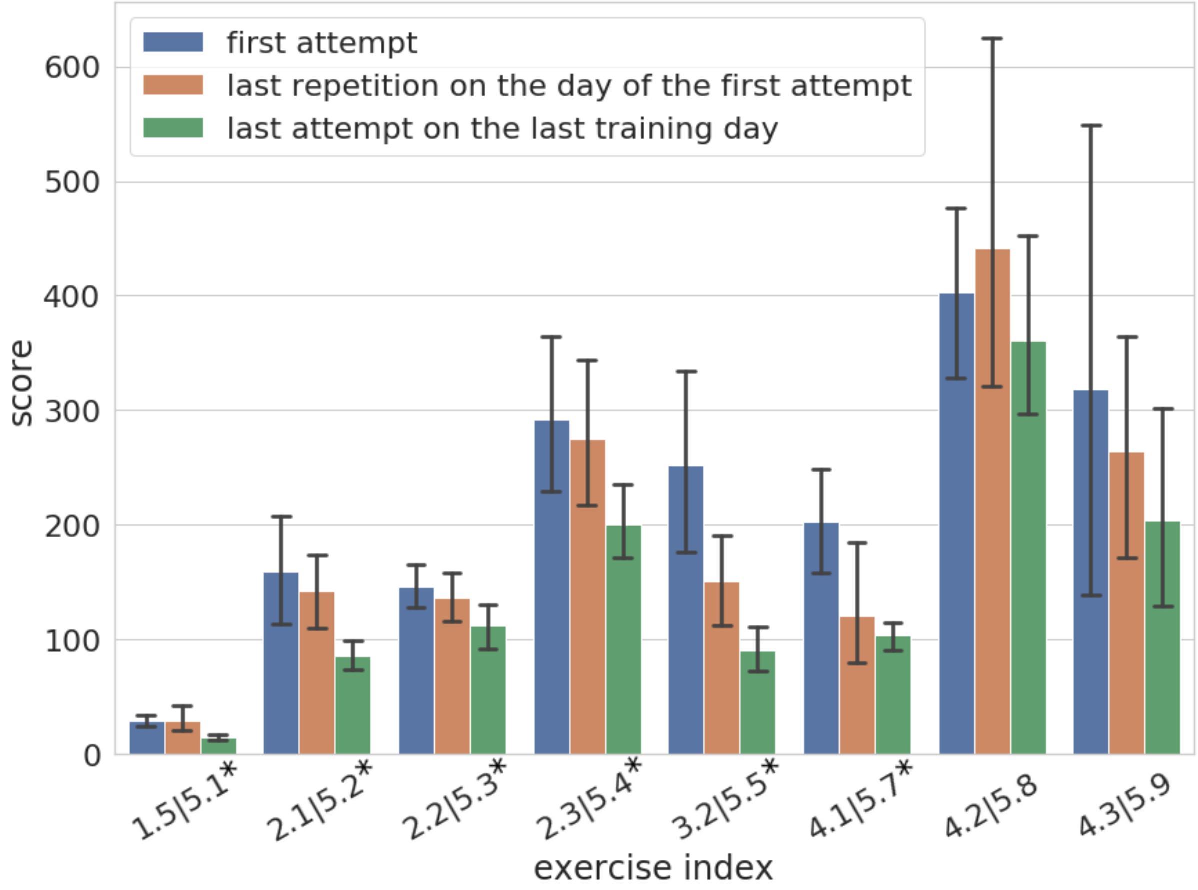


Figure 3.14.: HEiKA Experiment: Average deviation (score) for all participants and different task attempts for the task with an attempt on the last day. Significant values are marked with "*" (see table 3.2). Author: Xingbo Wang [172, 154].

Table 3.2.: HEiKA Experiment: p^*- and F★-values for ANOVA test for differences between the first attempt on the first day and the last attempt on the last day for error score and time measurements. Significant values are highlighted in italic. Source: [154]

Tasks	Score*	Score★	Time*	Time★
1.5/5.1	*0.0003*	19.8130	0.1639	2.1062
2.1/5.2	*0.0098*	8.34833	0.0626	3.9386
2.2/5.3	*0.0362*	5.1264	*0.0004*	18.3855
2.3/5.4	*0.0442*	4.6796	*0.0011*	14.9105
3.2/5.5	*0.0016*	13.7431	*0.0283*	5.6904
4.1/5.7	*0.0009*	15.5662	*0.0115*	7.9172
4.2/5.8	0.4672	0.5518	*0.0421*	4.7871
4.3/5.9	0.3799	0.8103	0.3033	1.1228

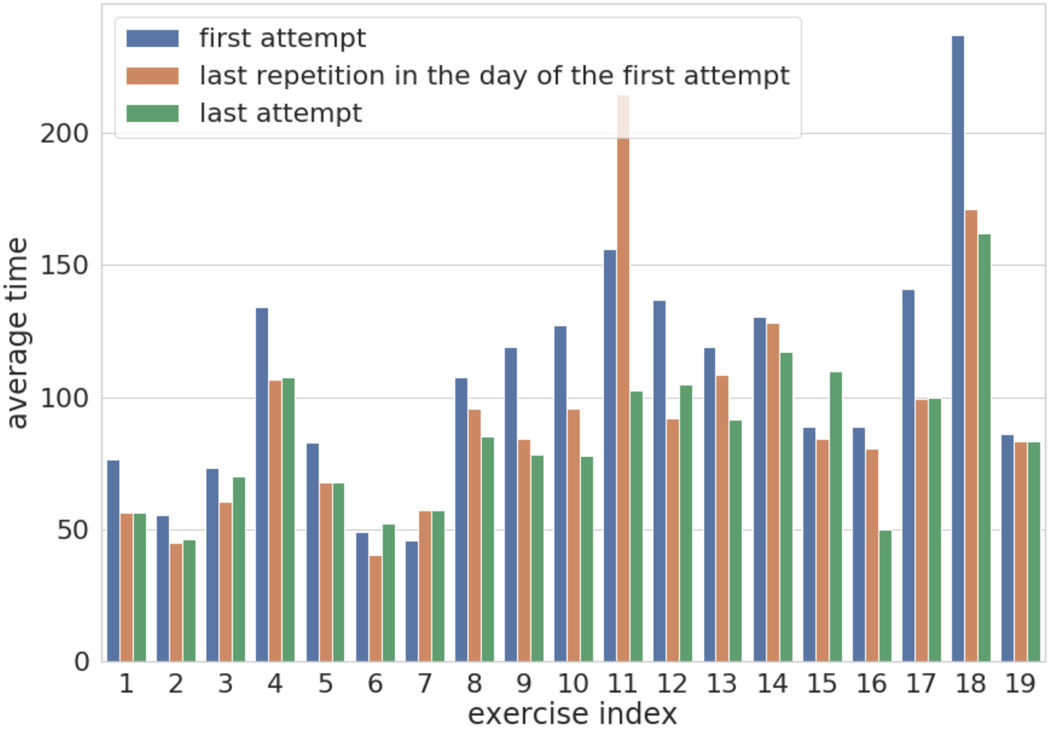


Figure 3.15.: HEiKA Experiment: Average time in seconds for all participants and exercises. The results show measurement during the first attempt (second repetition), the last attempt of the first day when exercise is introduced, and the last attempt (on the same day or the last day). The exercises are indexed according to table 3.1. Author: Xingbo Wang [172, 154].

Time Performance during the Training

The data on participants' time performance in seconds is also shown in three different forms in figures 3.15 to 3.17. Figures 3.15 and 3.16 show the average time measurements for all participants and exercises comparing different attempts as average values and in boxplot form, respectively. Figure 3.17 compares the significance of the first and last attempts repeated on the last day.

There is a time improvement throughout the training for almost all exercises as shown in figure 3.15. On the first day, exercises 1 to 7 show the increased average time for the first tasks with one *artificial force*, concretely tasks 4 and 5. The average time for tasks 6 and 7 is comparable to the average time without artificial forces (task 1). A similar observation is also done for average deviation measurements (see the previous section).

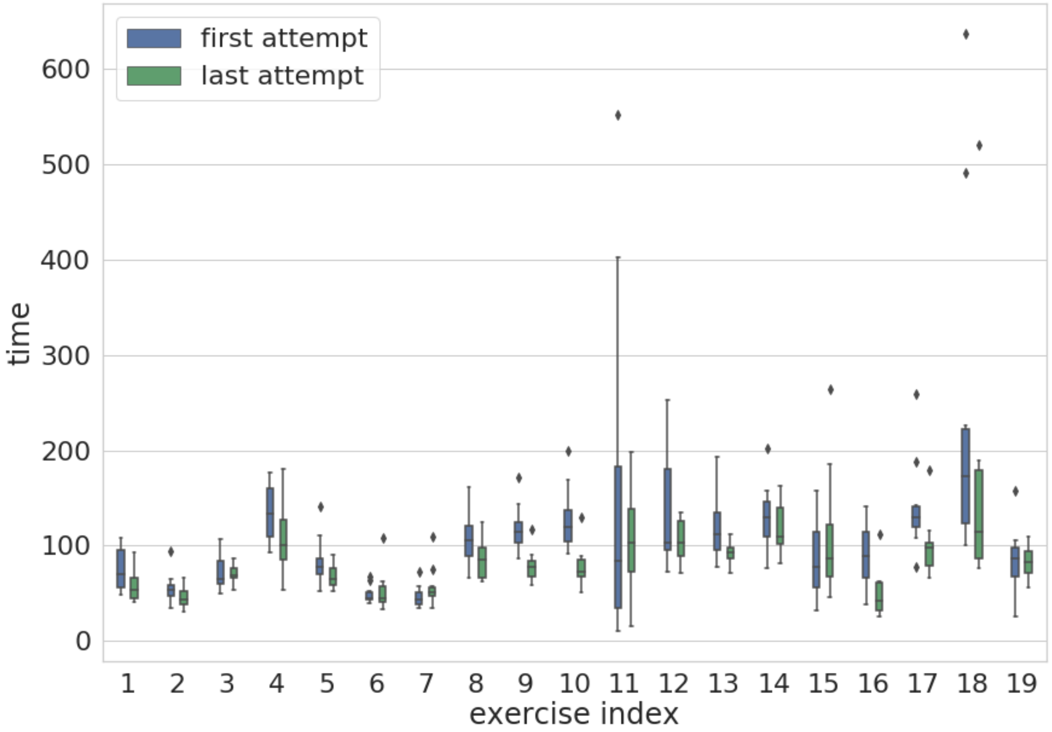


Figure 3.16.: HEiKA Experiment: Average time in seconds for all participants and exercises. The results show measurement during the first attempt (second repetition) and the last repetition (on the same day or the last day) in boxplot form. The boxplots are defined as explained in section C.1. The exercises are indexed according to table 3.1. Author: Xingbo Wang [172, 154].

The second day's tasks required more time than those from the first training day. This confirms that the proposed order of tasks has increasing complexity. For exercise 11, the first exercise with the *inverted controls*, participants needed more time to solve them than for all other exercises. The only exception is task 18. The exceptionally high average time needed for task 11 for the last attempt on the task's first day is especially interesting. The participants' fatigue could have caused this since that particular exercise was the last one of the second day. The differences in average time on day three, between exercises 12 and 15, can be explained by different reference paths. The exercises on the full path, 12 and 13, do not significantly differ. The exercises with the *inverted controls* concept during day four show a large increase in time needed to finalize them, especially when comparing tasks 18 and 9. The last exercise, 19, shows the same average time for almost all repetitions. This is probably because all the measurements are done on the same day and that the participants' skills settled at their level.

The boxplot representation of the average time measurements shows similar behavior as for the deviation. The tasks with the *inverted controls* concept show higher outliers compared to the individual plots and they have a larger dispersion of measurements. Comparing the tasks with the *artificial force* concept during the first day, i.e., tasks 4 and 5, show increased average time and data dispersion. The first exercise with the *inverted controls* concept stands out from all other data with a considerable dispersion of its whiskers. Interestingly, task 13 with the *artificial force* concept shows better results than its counterpart without forces, task 12. The only explanation for this would be the familiarity with the training path and the high-velocity users can achieve on it – resulting in the weaker influence of the *artificial force* concept. Comparing tasks 10 and 17 does not clearly show the *inverted controls* concept's influence, but exercises 9 and 18 do. On the other hand, tasks 17 and 18 have the clearest outliers in both attempts. This indicates that the *inverted controls* concept does not equally influence all the participants.

The time performance of different attempts of the same exercise is compared in figure 3.17. Five of nine compared tasks showed significant improvement in participants' time performance throughout the training, i.e., between the first and last exercise attempts. In general, the differences in exercise average times between the exercises with and without high-level concepts, i.e., *artificial force* and *inverted controls*, are not as evident as for the deviation measurements. The exception is task 18 (4.3/5.9), which has longer average times than its counterpart, exercise 9 (2.2/5.3).

User-Experience Assessment

The users' experience is assessed using questionnaires after each task using a Likert-type scale with five response categories. Table 3.3 shows the summary of user's answers for all tasks as median and range of the answers. For clarity, figure 3.18 provides plots of average values of the users' responses. The same is done for the three questions asked at the end of each day, presented in table 3.4 and figure 3.19.

The participants rated most of the exercises as medium complex and estimated their performance also to medium. Only for task 3.4 (15), one participant rated the performance as unsatisfactory. The reason for this is that the *inverted controls* concept was confusing for this participant. Otherwise, the participants rated their performance systematically as slightly worse than the tasks' complexity. The only exceptions are exercises 4.3 (18) and 5.10. On the last day of the training, the participants tended to perceive the repeated first four tasks as simpler and their performance as better. This is the case for average (figure 3.18), as well as for the median and range values (table 3.3). The users rated the tasks with the *inverted controls* concept, i.e., 3.4, 4.x, and 5.9, as more complex. For the tasks with the *artificial forces*, there is no clear tendency in the users' answers. The new exercise at the end of the training was evaluated as one of the easiest tasks and the users were satisfied with their performance.

Table 3.3.: Assessment of users' experience for all exercises. Colors in the table mark the assessment of the same tasks on different days. Source: [154]

		Task Complexity		**User Performance**	
ID	index	Median	Range	Median	Range
1.1	1	2	1 – 3	2.5	2 – 3
1.2	2	2	1 – 4	3	2 – 4
1.3	3	2.5	1 – 3	3	2 – 3
1.4-1	4	2	1 – 4	3	2 – 4
1.4-2	5	2.5	1 – 3	3	2 – 3
1.4-3	6	3	1 – 3	3	2 – 4
1.5	7	2	1 – 3	2.5	2 – 3
2.1	8	3	1 – 4	3	2 – 5
2.2	9	2	1 – 3	3	2 – 3
2.3	10	3	1 – 3	3	2 – 4
2.4	11	3	1 – 3	3	2 – 4
3.1	12	2	1 – 3	3	2 – 4
3.2	13	2	1 – 4	2	1 – 4
3.3	14	2	2 – 4	3	2 – 4
3.4	15	4	2 – 5	3.5	2 – 6
4.1	16	3	2 – 5	3	2 – 5
4.2	17	3	1 – 5	2	2 – 4
4.3	18	3	2 – 5	3	2 – 5
4.4	19	3	2 – 5	3	2 – 5
5.1		2	1 – 2	2	1 – 3
5.2		2	1 – 2	2	1 – 3
5.3		2	1 – 2	2	1 – 2
5.4		1.5	1 – 3	2	1 – 3
5.5		1.5	1 – 2	2	1 – 3
5.6		2	1 – 3	2.5	1 – 4
5.7		2.5	1 – 3	2	2 – 4
5.8		2	1 – 3	2.5	1 – 3
5.9		3	1 – 5	4	2 – 5
5.10		2	1 – 3	2	1 – 3

Scale: 1 - very easy, very good; 5 - very complex, very bad;
6 - not solvable, unsatisfactory

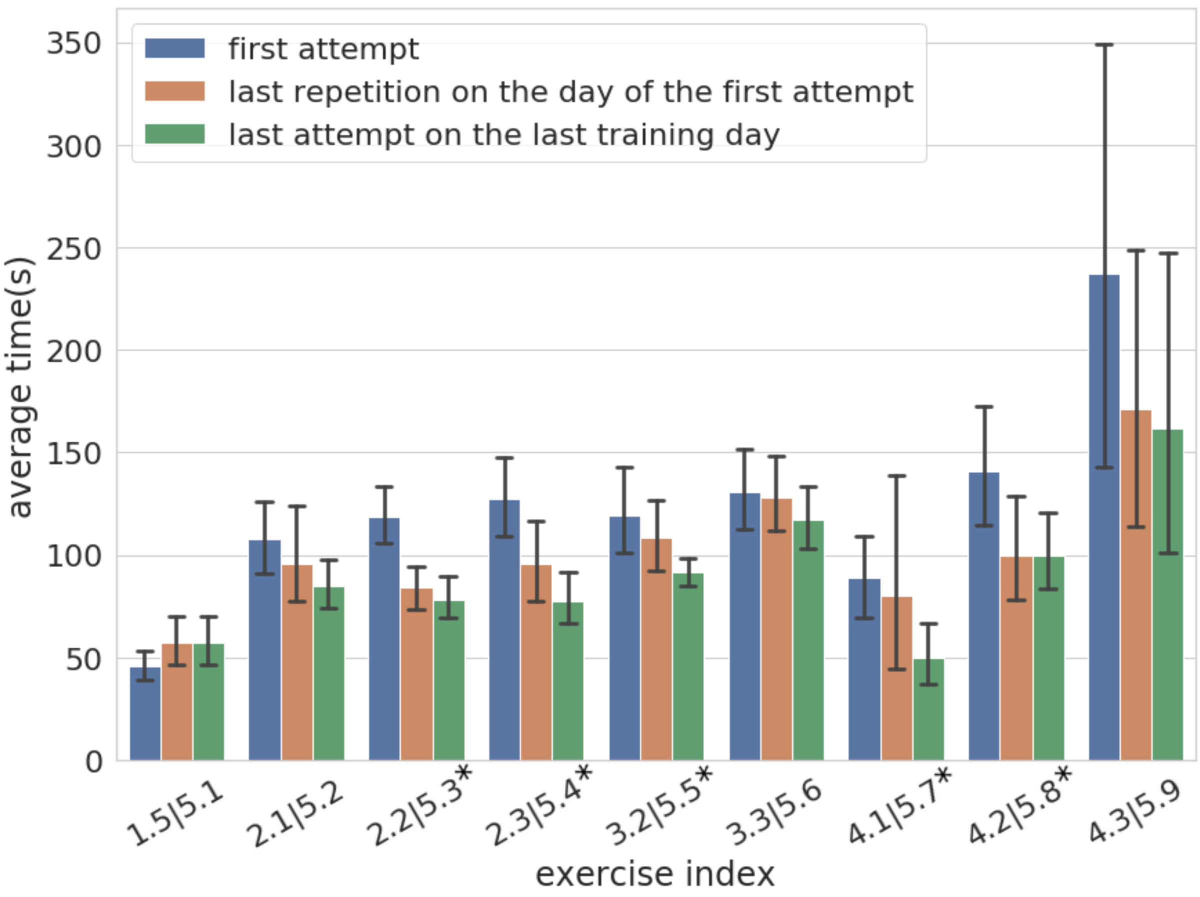


Figure 3.17.: HEiKA Experiment: Average time in seconds for all participants and different task attempts for the task with repetition on the last day. Significant values are marked with "*" table 3.2. Author: Xingbo Wang [172, 154].

At the end of each training day, the participant made an overall rating of the handling and safety of and accustomization to RoboTrainer (see section 3.3.1). The first four days show the users' tendency to experience the device's handling with increasing complexity (figure 3.19). On the fifth day, they evaluated the handling as easy. This confirms the hypothesis about the proposed task complexity and shows that users get used to the device during the training week. The users felt safe during the whole training with a slight reduction on the fourth day. That day, the users handled the device with *inverted controls*, which probably caused them to feel less safe. The participants stated that they were accustomed to the device relatively fast on all days, with a smaller range shift toward worse rating at days three and four. Looking at the average values in figure 3.19, the best rating was given on the last day of the training session. This probably lies in the fact that all the tasks the users did were already known to them. No participant evaluated the handling of the device as unsafe, impossible or unintuitive, i.e., no participant answered any of the three questions with a grade six.

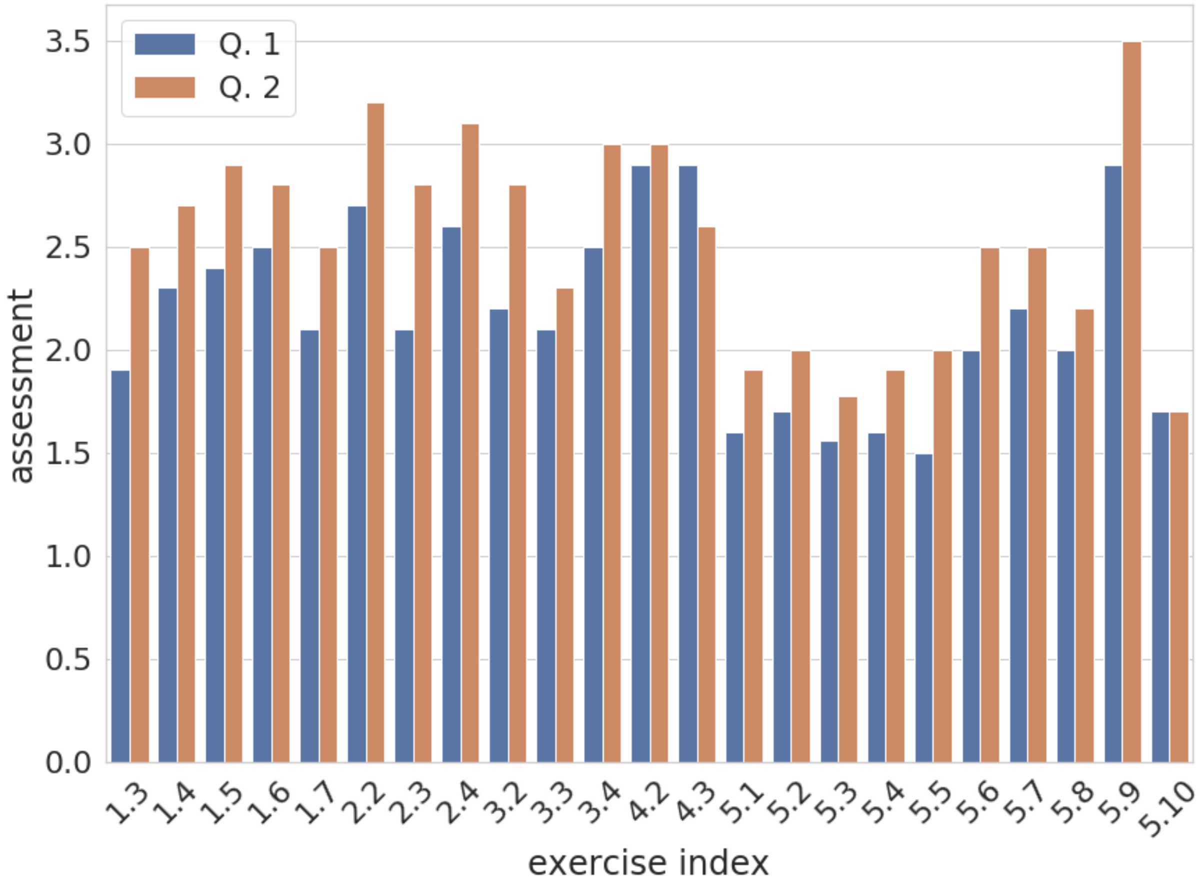


Figure 3.18.: Average values of users' answers regarding the tasks' complexity (blue bars) and self-assessment (brown bars). The rating scale is the following: 1 - very easy, very good up to 5 - very complex, very bad, and 6 - not solvable, not at all. Author: Xingbo Wang [172, 154].

Table 3.4.: Results of the user experience questionnaire each day. Source: [154].

	Device Handling		**Device Safety**		**Accustom to Device**	
Days	Median	Range	Median	Range	Median	Range
1	2	$1-3$	2	$1-3$	2	$1-3$
2	2.5	$1-3$	2	$1-3$	2	$2-3$
3	3	$2-3$	2	$2-3$	2	$2-4$
4	3	$2-5$	2.5	$2-4$	2	$2-4$
5	2	$1-3$	2	$1-3$	2	$1-3$

Scale: 1 - very easy, very good, very fast;
5 - very complex, very bad, very slow;
6 - not solvable, extremely unsafe, not at all

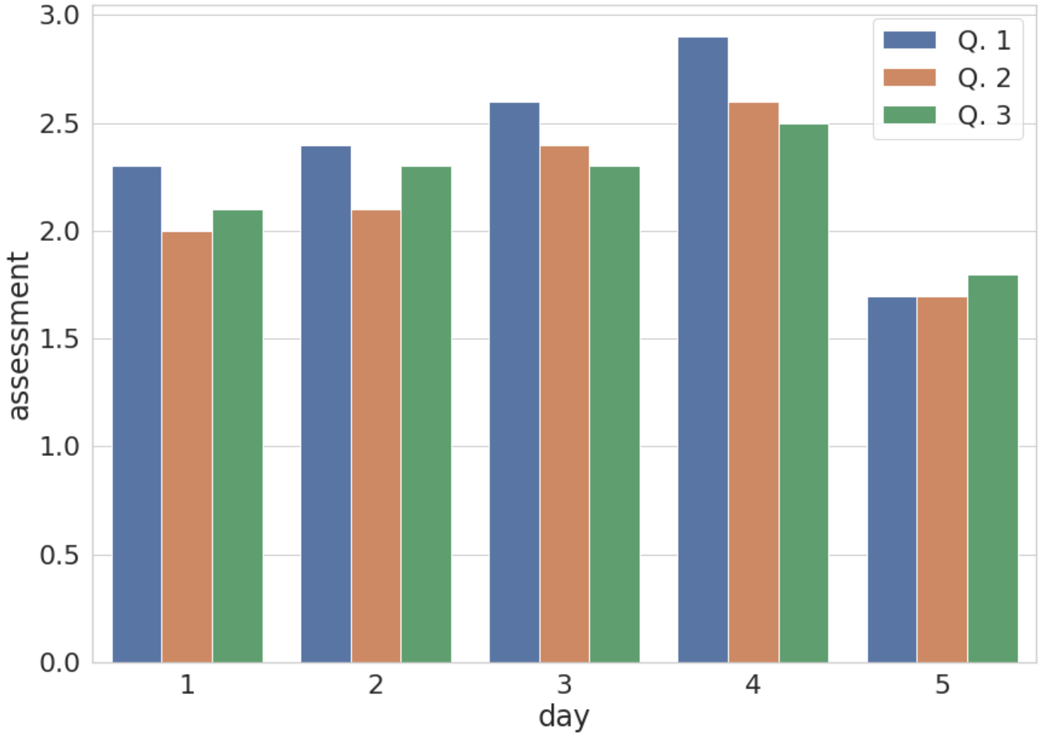


Figure 3.19.: Average score of assessment of three questions asked at the end of each day. The bars are sorted as the above-listed questions (see section 3.3.1). The scale is following 1 - very easy, very safe, very fast, 5 - very complex, very unsafe, very slow, and 6 - not solvable, extremely unsafe, not at all. Author: Xingbo Wang [172, 154].

Each day, the participants had the opportunity to comment on the training exercises and the interaction with RoboTrainer. Some of them experienced the training as easy and some of them compared it with weight training. Some participants had issues with narrow passages such as doors because the system would stop if they tried to "push" it towards an obstacle. All participants mentioned they needed to adapt to the control of the device with reversed sideways controls.

High-Level Control Concepts

The *inverted controls* concept showed expected behavior during the training. This means that such tasks resulted in worse user performance and users' were slower, i.e., they needed more time. The same result was confirmed in the user experience assessment,

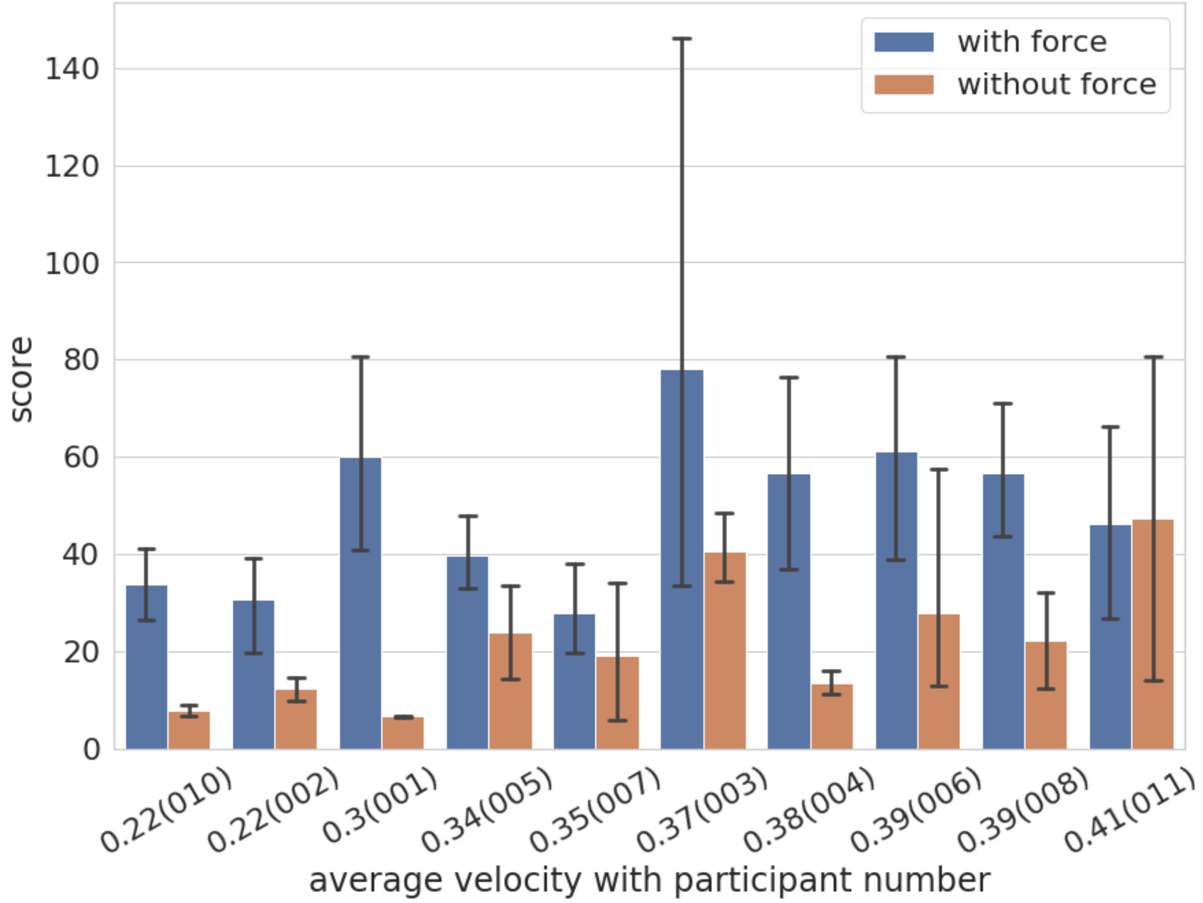


Figure 3.20.: Deviation comparison for the path with and without *artificial forces* concept. The data shown are for the path “line *c1*” sorted considering the participants’ average speed. Author: Xingbo Wang [172, 154].

where the tasks on day four regarding robot handling were rated as more complex. This concept also tended to influence users’ safety feeling to the worst value on day four.

The *artificial force* was also effective in providing unexpected RoboTrainer’s behavior. To examine the influence of the concept of the *artificial forces* more closely, two tasks along the same path are compared in figure 3.20. The figure compares the deviations of exercises 2 and 4 for each participant separately. The data are sorted with increasing average speed to evaluate its influence on the RoboTrainer’s deviation. From figure 3.20, no clear conclusion can be established between the data sets. Nevertheless, the statistical analysis shows that *artificial forces* significantly (ANOVA: $p = 0017$) increase the error score, i.e., RoboTrainer’s deviation. There is no explanation for why participant 011’s deviations with and without *artificial forces* are almost identical. The participants evaluated the tasks with *artificial forces* with only a slight tendency to be more complicated. Therefore, no clear outcome is identifiable.

During exercise 13, a deficiency regarding the current integration of the concept of the *artificial forces* into RoboTrainer's controller is highlighted. A force opposite to the movement was too strong for one user who, therefore, could not pass it. When the user released the RoboTrainer, the robot started strolling towards the person, which presented a safety risk. This was caused by the implementation of *artificial forces*, which can override the user's input and change the RoboTrainer's movement unexpectedly. Therefore, if the *artificial forces*' strength is not adjusted to users' strength, such situations can happen. These issues should be investigated and dealt with in the future[6].

3.4. Conclusions

This chapter presents the concept of training with a Smart Walker-like device. The training aims to provide motor activation for the elderly and persons with mild cognitive impairment (MCI) in order to improve their physical and mental state. During the training, the users interact with a mobile robotic device which provides physical support on the one side and challenges on the other side. The users have to navigate, i.e., guide the device along predefined paths marked on the ground of the training environment. For this to be achieved, a mobile robot *rob@work* is extended with a force-torque sensor (FTS) and handles as a user interface. This prototype, called RoboTrainer Prototype, is controlled using admittance dynamics of the first order, transforming users' input forces into its movement. The main admittance controller is extended by the high-level control concepts called *artificial forces* and *inverted controls*, to realize versatile and challenging training. The *artificial forces* concept enables invisible disturbances in the training environment which users need to neutralize by reacting with a counterforce. The *inverted controls* concept makes the control of the RoboTrainer unintuitive by reversing the left and right control directions. This concept stimulates additional cognitive load to control the RoboTrainer precisely.

Ten participants with mild cognitive impairment evaluated the device's functionality and the influence of the training in a pilot study. It is shown that the training with the RoboTrainer is feasible and the participants felt safe during the interaction. The high-level control concepts are suitable to adjust the complexity of the training. The *artificial forces* concept's complexity can be adjusted with their strength and orientation on a predefined path. The *inverted controls* concept does not provide any adjustments in terms of intermediate steps, but starting with simpler tasks, made the training more challenging. The objective data confirm these hypotheses, i.e., the precision of following the reference path, users' time performance, and the participants' subjective assessment.

[6] Chapter 6 provides concepts to handle such situations and to neutralize RoboTrainer movement it those would result in a dangerous robot's movement.

These results confirm the expectations that robot-based training for the target group is feasible and enhances future research confidence. Still, based on the observations during the pilot study, there are several possible improvements regarding the RoboTrainer and the training. Those are extensively discussed in the following section.

3.5. Lessons Learned

Observations and users' comments from the pilot study serve as valuable input regarding potential improvements of the RoboTrainer Prototype for future development. In the following, those experiences and improvements are discussed.

Footprint's size and geometry The RoboTrainer Prototype's base has a rectangular footprint and provides sufficient space for user's feet during interaction. The device's handles are outside the footprint which offers only limited physical support for a user since the base could tilt backwards if their whole weight is shifted on RoboTrainer's handles. This situation happened at least once during the study. The RoboTrainer Prototype's base is oriented with the short side toward the user to enable passing through the doors. Therefore, the length of the device's base, i.e., its influence on the maneuverability of RoboTrainer Prototype, was a limiting factor in some scenarios.

Mechanical construction and base stiffness The *rob@work* base was developed as an omnidirectional platform for logistic tasks inside a warehouse or production area. Therefore, it is not constructed to be influenced by external forces as in this thesis. Still, the mechanical construction of the base endured the whole pilot study. Only one issue occurred with the fastening of a wheel. The constructors of the *rob@work* knew this issue and, with their help, it was solved fast. Nevertheless, this issue should be considered for future work, and the affected parts should be constructed differently. Besides that, RoboTrainer Prototype has slight backlash in the construction of drive-steer modules so that a user can feel it on the handles when the platform is not moving. This did not present an issue in the pilot study, but this backlash should be reduced to provide better physical support to a user.

Attachment of the handles Since the handles are added additionally to the *rob@work* platform, their stiffness was insufficient for this use case. The main issue was the torsion of the handles when a user was applying high moments. This issue could be solved by a different geometry of the handle's support construction. This would especially be evident when doing more agile training with younger persons.

Safety components RoboTrainer Prototype has a laser-scanner with configurable safety fields around the robot to avoid collisions with the user and the environment. It also has emergency stop switches on the front- and backside. Nevertheless, the safety concept should be more advanced in agile training when interaction forces are up to 200 N. This can be achieved with additional emergency stop switches – easily accessible by the user and a wireless emergency-stop button – controlled by the training supervisor. Also, more performant safety sensors and a safety Programmable logic controller (PLC) is needed.

Controller The use of admittance control in the pilot study was a reasonable choice since it provided well-known dynamics of RoboTrainer Prototype and predictable influence of its parameters. The controller's drawbacks were fixed and predefined device dynamics for all users. The first issue caused visibly different behavior and performance of users based on their physical strength and fitness. The second issue, caused by higher mass and damping in the admittance controller, resulted in RoboTrainer's slower dynamics and longer stopping distances. If the parameters were chosen to support better behavior at high velocities, RoboTrainer Prototype would be too sensitive when standing and also at the beginning of a movement.

Control concepts The high-level control concepts, i.e., control action, influenced the participant's performance and effort as expected during the pilot study. Regarding the *inverted controls* concept, there are no acute improvements that need to be done in the future. The pilot showed that the *artificial forces*' wrong configuration and strength could overload a user's input and potentially cause dangerous situations. This was described in section 3.4. Due to the approach, the user's input was changed before the admittance controller, by the *artificial forces* concept. In this case, it is impossible to separate the user's input and the concept's influence from RoboTrainer's velocity. So, it is not possible to limit its influence if it endangers a user. Therefore different architectures for integrating the high-level control concepts in the control loop should be investigated in the future.

Controller's reference frame Conventional wheeled walkers or Smart Walkers (SWs) with differential drive have their control frame's origin based in the middle between two differentials, i.e., rear and wheels. Omnidirectional SWs can place those in an arbitrary position in the environment. The controller's origin position influences the walker's behavior in curves, as described at the end of section 2.3.5. In the pilot study, the *rob@work* platform's default control reference frame is placed in the device's geometrical center. This, sometimes, resulted in participant's clumsy turns with the RoboTrainer Prototype. Changing the control reference frame's position, i.e., the center of rotation, could, on one side, provide better controllability and, on the other side, an additional building block for training.

4 Design of a Device for Active Training

This chapter provides ideas, requirements, and mechatronic design of the *RoboTrainer v2* device developed in this thesis. The new device addresses the RoboTrainer Prototype's issues, and its implementation adheres to norms and directives wherever possible. Special care is invested in the safety concept during hardware design to provide conditions for an evaluation with inexperienced participants without any background in robotics. The initial mechanical design was done by Mayer [116] in his Bachelor's Thesis. A brief description of the overall mechatronic design is published in Stogl, Hein, and Mende [153].

The design process of RoboTrainer v2 began with a description of its use, a definition of its scope, and a characterization of involved people in the training scenario. This analysis, presented in section 4.1, is a crucial phase in RoboTrainer v2's risk and safety evaluation. From there, the technical requirements on mechatronic design for the RoboTrainer v2 are defined (section 4.2). This process started with scoping and analysis of the norms and directives for conventional walkers and assistive robots. These requirements, discussed in section 4.2.1 and section 4.2.2, provide information on binding design directives and risk mitigation strategies using safety devices. Section 4.2.3 gives a list of functional requirements to realize the training. The functional requirements are based on the experience with the RoboTrainer Prototype and the state-of-the-art overview from chapter 2. section 4.3 describes the mechatronic concept of the RoboTrainer v2 and its realization, whereas the safety concepts are discussed in section 4.4. Finally, a short overview and conclusion of this chapter is given in section 4.5. A brief discussion on possible further safety measures can be found in section 4.4.3.

Throughout this chapter, both names "RoboTrainer v2" and "RoboTrainer" describe the version of the device developed for this thesis. When referring to RoboTrainer Prototype, i.e., the device version used as proof of concept in chapter 3, “RoboTrainer Prototype” is used.

4.1. Use of the RoboTrainer

The RoboTrainer v2 is a device for strength and coordination training with healthy persons. Figure 4.1 shows RoboTrainer v2 with its user. A user interacts with RoboTrainer by applying force directly to its handles. This type of interaction is called physical human-robot interaction [61]. The handles are mechanically fixed to a force-torque sensor (see figure 4.7), which measures the user’s inputs causing the device’s movement. A user’s task is to navigate RoboTrainer along predefined paths marked on the floor of the training area. Depending on the task and user’s performance, the RoboTrainer can support or disturb a user during a task using spatial control actions (SCAs) (c.f. chapter 5). RoboTrainer v2 is used in an indoor environment with at least 4.5 m x 8 m of free space. As a research device, any use of RoboTrainer has to be supervised by an experienced engineer and one additional person who coordinates the training. The device is used multiple times by one person in training, but only once per day in sessions of the duration of up to two hours.

There are three different roles of persons needed when using the RoboTrainer: the user, the training supervisor, and the technical supervisor. At least one person from each role should be present during the training and the number of observers should be kept as low as possible. If there are any observers in the training environment, they have to stand in an area not used for the training and follow instructions from the supervisors. In the following, each type of involved roles is described.

Users of the RoboTrainer In the scope of this work, RoboTrainer is used by healthy adults in good physical shape for motor and force training. In the future, the device aims to be used by elderly persons and persons with cognitive impairments, providing novel cognitive therapies addressing motor interaction. Therefore, the device should be able to provide physical support for its users. Moreover, RoboTrainer’s appearance should reassemble a known device for elderly persons to lower the acceptance barrier. The most obvious device is a walker.

Training Supervisor A training supervisor is a person who understands the high-level functionality of the RoboTrainer and has a more profound understanding of the training. This person is instructed by technical supervisors before the training and has prepared the training scenario together with them. A training supervisor guides users

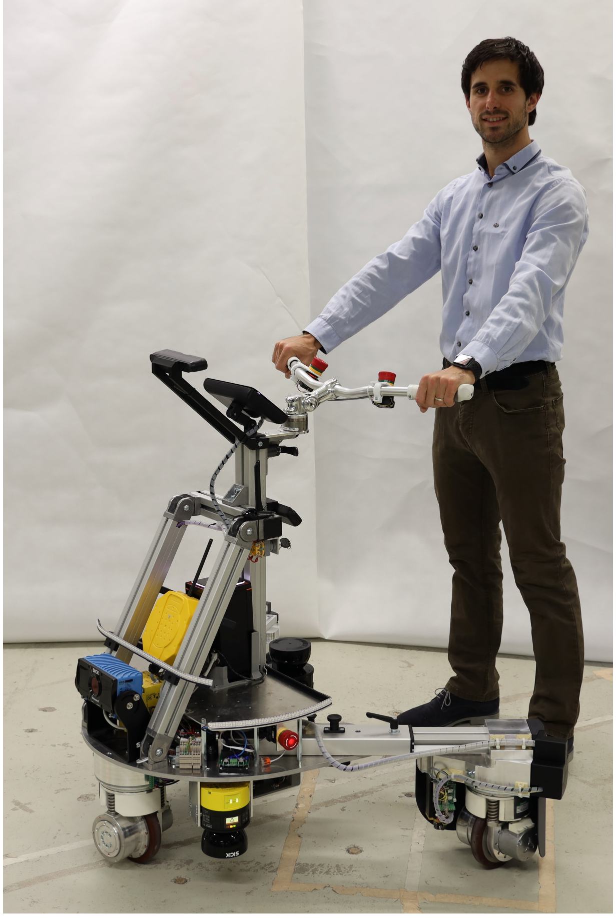


Figure 4.1.: RoboTrainer v2 and its user on the test parkour at the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics. The markings on the floor represent paths and RoboTrainer's positions which a user has to follow.

through the training and observes their interaction with the RoboTrainer. As an additional risk mitigation strategy, a training supervisor should have the possibility to interrupt the training if a user is overwhelmed with a task or the RoboTrainer endangers a user. A possibility to achieve this could be the use of a wireless emergency stop switch.

Technical Supervisor A technical supervisor has in-depth technical knowledge about RoboTrainer v2, and is able to use its configuration software and adapt its internal parameters. During the training, a technical supervisor observes the internal states of the RoboTrainer on a Control PC. The role is to observe the functional status of the device and reconfigure its software and hardware.

4.2. Technical Requirements for RoboTrainer v2

The long-term goal of research with RoboTrainer v2 is to provide a robotic training device for elderly persons with mild cognitive impairment (MCI) to slow down the progression of their disease. Ideally, training functionalities would be integrated into commonly used devices, e.g., wheeled walkers. According to Martins et al. [110], the resulting device is a Smart Walker (SW). Smart Walkers are currently not regulated by any specific norm. Therefore, norms for classical walkers and general norms for machines and robots' safety were consulted during the design process. Especially, specific norms for the safety of personal care robots provide valuable information on the required performance of safety measures. The two most important norms for the design of the RoboTrainer are *ISO 11199-2:2005 Walking aid manipulated by both arms - Requirements and test methods - Part 2: Rollators* [79]; and *ISO 13482:2014 Robots and robotic devices - Safety requirements for personal care robots* [81]. The latter references and concertizes ISO 10218-1:2011 [78], ISO 13849-1:2015 [82], IEC 60204-1 [73], ISO 12100:2010 [80], ISO/TR 14121-2:2012 [83], and ISO/TS 15066:2017 [84] regarding general information for risk assessment, and collaborative and safe operation of mobile robots.

4.2.1. Requirements from Walkers-Related Norms

The design of the RoboTrainer v2 is motivated by the appearance of conventional wheeled walkers to increase its acceptance by elderly persons. Orienting the footprint and size of RoboTrainer to a conventional walker also has other benefits, like the device's usability in an indoor environment and the ergonomic height of the handles for elderly persons. The most relevant norm for conventional walkers is ISO 11199-2:2005 [79], which provides exact test methods a walker has to pass to come out on the market. The following paragraphs discuss relevant parameters from this norm.

ISO 11199-2:2005 defines the smallest wheel diameter of 75 mm for indoor and 180 mm for outdoor use of a walker. Since RoboTrainer v2 should have at least three wheels to enable movement without the need for a user to balance it out, the ISO 11199-2:2005 prescribes to have running brakes. Nevertheless, this criterion only makes sense for a conventional, passive walker. A Smart Walker should have brakes integrated with the primary wheel controller and they would resemble to the form of switches to enable the device's movements. Another possibility is to use sensors that observe the user's posture for enabling or breaking a SW. The latter approach is currently not feasible from the safety perspective. ISO 11199-2:2005 also prescribes the width of the handgrips between 20 mm and 50 mm. One of the goals with RoboTrainer v2 is to provide mechanical adaption to the user's needs. If such an adaption exists, ISO 11199-2:2005 prescribes easy and secure fixing of parts when in use. Although not prescribed with ISO 11199-2:2005, a mechanical adjustment should be possible without tools.

One of the most critical design decisions for any walker and SW is its footprint, size, weight, and placing of the components. The walkers' function strongly influences those parameters. For any conventional walker, this is a typical engineering issue: on one side, a large footprint and weight provide better physical support for a user, however, on the other side, users should be able to move indoors in cluttered environments and be able to transport a walker easily. Most of the classical walkers investigated for this thesis have width up to 75 cm, length up to 80 cm, and weight up to 10 kg. These values, except the weight[1], should be used as a guideline when designing RoboTrainer v2. In any case, the maximal width of a SW should be less than 90 cm, which is the standard width of entrances in barrier-free public and living buildings. Those values are specified by *DIN 18040-1:2010 Construction of accessible buildings - Design principles - Part 1: Publicly accessible buildings* [36], and *DIN 18040-2:2011 Construction of accessible buildings - Design principles - Part 2: Dwellings* [37]. The handle height, i.e., mainly the maximal height, of most classical walkers is between 80 cm and 105 cm and it is adaptable within the range of 10 cm to 15 cm.

Although a wheeled walker's size and footprint are not standardized, ISO 11199-2:2005 [79] specifies minimal tilt angles for a walker before it may turn over. These are 3.5° for a sideward, 7° for the backward, and 15° for the forward tilt to the horizontal plane. These values indirectly influence the size, footprint, and component-placing of a SW.

Based on the above-presented investigation of standards and research regarding the size of classical and Smart Walkers, following requirements for RoboTrainer v2 are defined:

RW1 RoboTrainer should pass through a 90 cm wide door.

RW2 The hand grips' width should be between 20 and 50 mm.

RW3 All mechanical adjustments on RoboTrainer should be doable without any additional tools.

RW4 The handle height should be at least 80 cm and adjustable for at least 10 cm.

RW5 RoboTrainer's handles are placed near the base footprint's geometry center so that a user can not flip it over unintentionally.

RW6 The wheels of the RoboTrainer should be more than 75 mm in diameter.

4.2.2. Requirements from Norms for Machines and Robots

The fundamental for all machines and devices developed and used in the European Union (EU) is the directive 2006/42/EC, called *Machinery Directive* [38]. The directive provides a set of essential health and safety requirements that every device on the market

[1] It is expected that RoboTrainer v2 will be much heavier than a conventional walker, just by considering active drives, batteries, and control PC.

and every device used in a workplace has to comply with. The "CE" marking[2] on a product or device outlines compliance with the *Machinery Directive*. Therefore, all off-the-shelf components used in RoboTrainer v2 have to have "CE" marking and the device as a whole has to provide safety functionalities like emergency stop and (re-)start interlock. The Machinery Directive affects a broad range of devices. Therefore, it does not name specific standards and norms but expects producers and operators to implement the relevant ones. In the remainder of this subsection, the norm on safety requirements for personal care and assistant robots are discussed, as well as the relevant inherited standards.

ISO 13482:2014 classifies mobile assistant robots into two types depending on their size, weight, speed, and manipulation possibilities. A robot is considered "small" if, when falling or tipping over, it cannot collide with the user's upper body. A robot is "light" if its mass is so small that injuries for its users are improbable and a user can lift the robot if trapped. The speed of a robot is "slow" if it is lower than the average walking speed of the user group determined during the risk assessment. RoboTrainer v2 is classified as a mobile assistant robot of type two. RoboTrainer v2 is not "small" since its handles could collide with the upper body of an adult person; it is not "light" because, in case of a collision, injuries are probable and a person cannot lift it if trapped; and it is not "slow" since its maximal speed (1.595 m/s) is higher than the average speed of adults between 20 and 40 years (1.26 m/s to 1.34 m/s) [138].

For the mobile assistant robot type 1.2, ISO 13849:2015 [81] defines minimal safety Performance Level (PL) *d* in section 6.1.3, table 1. The ISO 13482:2014 further describes possible safety functions and conditions for their fulfillment. The relevant functions are emergency halt, safety halt, and safety-related detection of the robot's environment. These halt functions have to comply with IEC 60204-1 [81]. ISO 10218-1:2011 and ISO/TS 15066:2017 provide general information and clarification on safety measures from ISO 13482:2014. Furthermore, ISO 12100:2011 provides requirements for risk assessment and risk mitigation during the design of the RoboTrainer v2.

The safety-related detection of the environment, i.e., safe distance to the obstacles, is a common approach in mobile robotics. EN ISO 13849:2015 names two possibilities to implement this safety function: stopping a device at a safe distance to the environment and safe speed limiting depending on the distance from obstacles. The speed reduction approach needs a direct connection between the motors and the safety CPU. This connection often is not a trivial task, since it needs cabling changes at the drive level. The second issue with this method is that it interferes with the admittance controller of

[2] The short official description about the meaning of "CE" marking can be found on the official website of the European Commission (`ec.europa.eu`) under "Internal Market, Industry, Entrepreneurship and SMEs", "Single market and standards", "CE marking" (direct URL: `https://ec.europa.eu/growth/single-market/ce-marking_en`). (accessed: Oct. 23, 2020)

the device. In this case, RoboTrainer's velocity would become unpredictable from the controller's perspective, leading to unreliable RoboTrainer v2's behavior in the training context.

ISO 13482:2014 allows implementing the safety distance approach with or without direct contact with a robot's environment. An implementation that uses contact, e.g., bumpers or safety edges, leads to a larger footprint. To avoid RoboTrainer's footprint enlargement, RoboTrainer v2 uses safe laser range finders (LRFs), i.e., distance sensors, for contact-less environment detection. Another advantage of using LRFs is that they provide data about the training environment. These data are helpful for 2D mapping and localization. The localization is relevant for the training investigated by this thesis in order to determine users' task performance and the device's footprint. They should enable movement between the rooms, like it is possible with a conventional walker. When choosing hardware for the contact-less environment detection, devices must implement norm IEC 61496-3:2018[3] [75] because it has to be possible to detect persons in the RoboTrainer vicinity, e.g., supervisors of the training.

Another suitable protection possibility from ISO 13482:2014 is safety-related force monitoring, realized using a safety force-torque sensor (FTS) as an input interface. As shown in the pilot study results in section 3.3, the interaction forces vary enormously between users and depend on their physical condition. Therefore, using this approach would need individual per-user parameters, which must be done manually. Another uncertainty about using this method is that the measured forces are primarily user-caused and could have very high, short peaks when sudden direction changes occur. Based on these assumptions and the fact that there are not many safety force-torque sensors on the market, this option is not further considered.

Based on the above-discussed standards, the following safety requirements for RoboTrainer v2 are defined.

RS1 The RoboTrainer's components and the RoboTrainer as a whole have to comply with the directive 2006/42/EG ("Machinery Directive").

RS2 The safety components of the RoboTrainer have at least Performance Level d (ISO 13849:2015).

RS3 RoboTrainer has to implement an emergency halt function described in ISO 13482:2014, section 6.2.2.2, for emergencies.

RS4 RoboTrainer has to implement a safety halt function described in ISO 13482:2014, section 6.2.2.3, as a risk mitigation strategy.

RS5 RoboTrainer has to implement safety-related environment detection for stopping it based on the distance of an obstacle in the environment.

[3] When choosing safety components for RoboTrainer v2, IEC 61495-1:2012 [74] was valid.

RS6 The safety laser range finder (LRF) should output raw data for mapping and localization purposes.

RS7 The safety components should not enlarge the footprint of the RoboTrainer.

4.2.3. Functional Requirements for Mechatronic Design

The functional requirements for RoboTrainer v2 are defined based on the experience with RoboTrainer Prototype. The background of the requirements related to the pilot study is discussed in section 3.5. Other requirements are use-case specific to provide the RoboTrainer's functionalities beyond the current state of technology and science.

R1 Mechanical structure of the RoboTrainer should be as simple as possible, modular, and built from off-the-shelf components.

R2 RoboTrainer's mechanical structure has possibilities to attach and detach additional sensors.

R3 RoboTrainer's base has omnidirectional kinematics.

R4 The mechanical construction has to be stable in a steady state. Therefore, high stiffness between handles and wheels, with minimal backlash when influenced by external forces, is needed.

R5 Footprint of the RoboTrainer should be adjustable to provide a smaller and larger support area depending on the scenario.

R6 User interface is implemented by measuring forces.

R7 Input device and mechanical construction have to support a constant load of 50 kg and a peak load of 250 kg.

R8 Autonomy, i.e., time running on the batteries, has to be at least 90 min.

4.3. Mechatronic Design

The CAD-Model of the RoboTrainer v2 is shown in figure 4.2. The mechanical structure was inspired by a conventional walker to create a sense of known device for its users. The RoboTrainer v2's final mechanical design is a compromise between its functionality, off-the-shelf components, and components used in RoboTrainer Prototype. This section explains the design decisions and puts them in relation to the requirements from section 4.2.

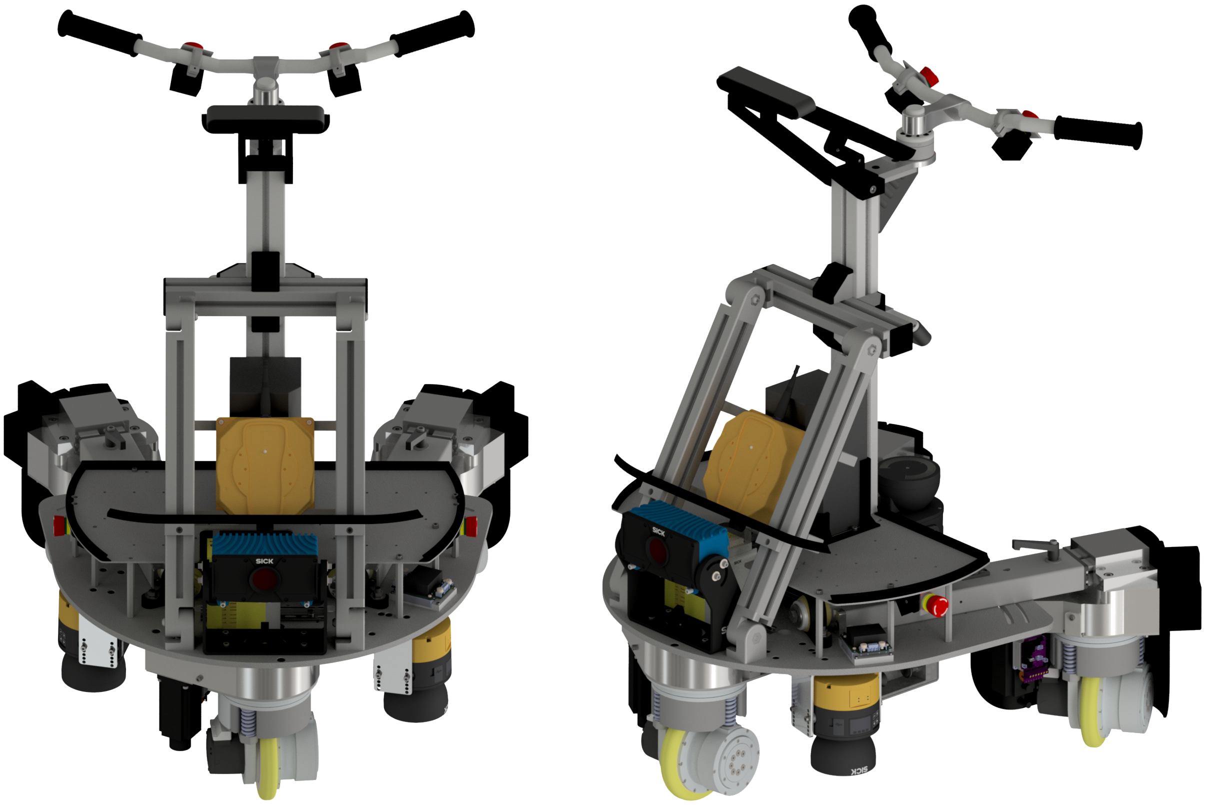


Figure 4.2.: RoboTrainer v2 CAD model (front view - left; side view - right).

Compared to RoboTrainer Prototype, this device has three active wheels and a somewhat triangular footprint. The decision for three wheels is to reduce the amount of hardware and the RoboTrainer's price but still keep movement flexibility and physical stability as high as possible. The wheels are active drive-steer modules developed for the *Care-o-bot*® service robot by Fraunhofer-Institut für Produktionstechnik und Automatisierung (Fraunhofer IPA). These wheels provide the base's omnidirectional movement (requirement - R3) and high stiffness since there are no passive parts on the wheels (R4). The chosen drive-steer modules are mechanically completely renewed compared to the modules used in RoboTrainer Prototype, but still use the same drivers and software stack. A module weights 8.0 kg, has a payload capacity up to 75 kg[4] (R7), can achieve a maximal linear velocity of 1.595 m/s, maximal torque of 16.8 N m, and has a wheel diameter of 160 mm (RW6). The handles' support construction is designed to eliminate stiffness issues known from RoboTrainer Prototype and support the handle's height adaption. For this, a larger vertical strut profile (40 mm × 60 mm compared to 20 mm × 20 mm) is used and two diagonal profiles (40 mm × 40 mm) are mounted off-center. These increase handle stiffness to torsion when a user is applying torque (R4). The handles on RoboTrainer

[4] To avoid the wheel's rubber deformation, the recommended payload for continuous operation is 50 kg.

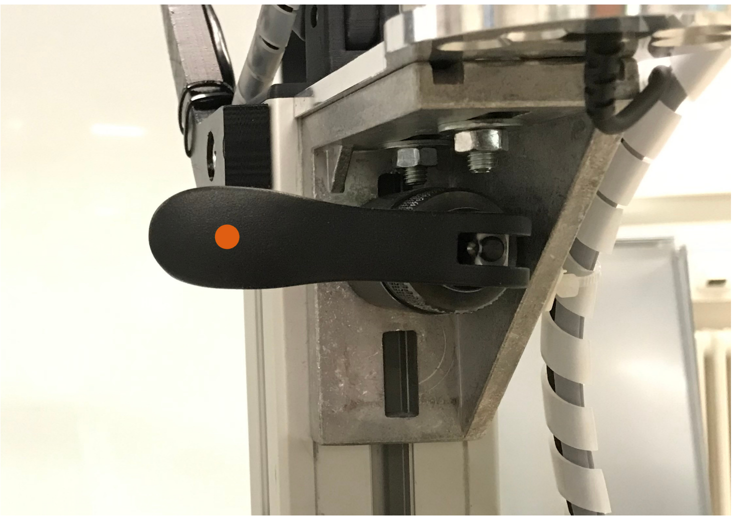


Figure 4.3.: RoboTrainer v2 toolless handle adaption real-world realization.

v2 are mounted on a bracket and fastened using quick-release skewers (RW3) to provide handles height adaption (RW4). Figure 4.3 shows the realization of the handle adjustment on the real robot. The handle height is adjustable between 90 cm and 105 cm from the floor. The ATI Mini58 force-torque sensor (FTS) mechanically couples the adjustment bracket and the handle to measure the user's input (R6). The FTS model is the same as in RoboTrainer Prototype. It is chosen for its small dimensions and high overload values[5] (R7). RoboTrainer v2 has the same bike handles as RoboTrainer Prototype, providing a known interface for a user. The principal horizontal rod of the handles has a 22 mm diameter, providing a standardized interface for many different grips on the market (RW2).

The rear wheels of the RoboTrainer v2 are placed outside the base's footprint so that the handles are placed closer to its geometrical center than was the case with Robo-Trainer Prototype. This placement results in higher stability against tilting and tilting-over (RW5) and provides better physical support for a user. One of the unique features of the RoboTrainer v2 is the possibility to change its footprint (R5) in two degrees of freedom. The combination of extreme values, i.e., closed–open and short–long configurations, are shown in figure 4.4. In the scope of the training, the technical supervisor does the reconfiguration manually, taking care of sufficient mechanical stability and robustness of the RoboTrainer v2 after reconfiguration. The joints are adjustable without the need for tools (RW3). The concept is depicted in figure 4.5. The RoboTrainer's footprint adjustment permits mechanical adaption to individual user. Therefore, large footprints (Figures 4.4b and 4.4d) are used if more support is needed and if a user needs an agile device, compact footprints (Figures 4.4a and 4.4c) are used. The smallest footprint

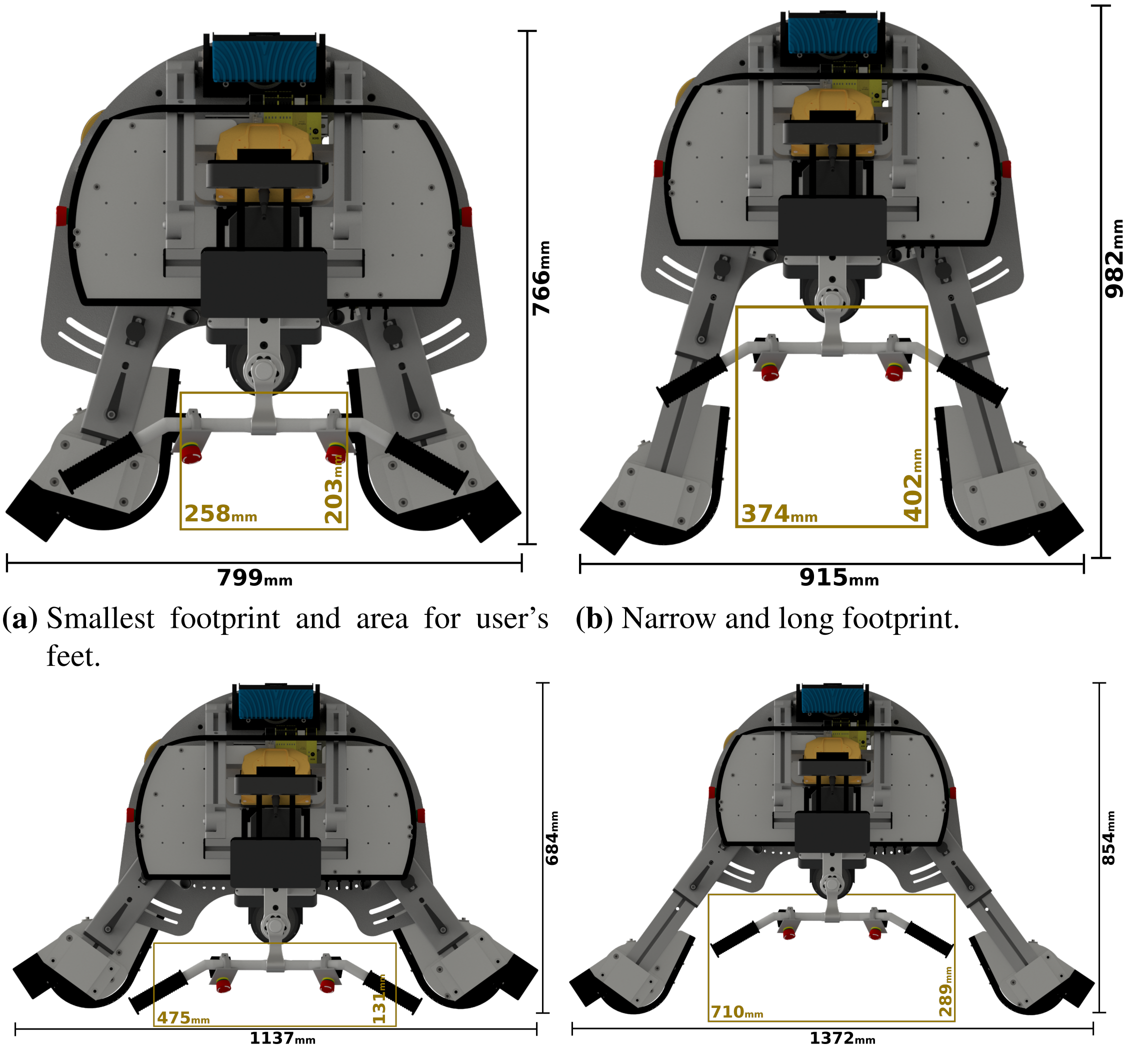
[5] The exact values are provided in appendix section B.3.

(figure 4.4a) enables RoboTrainer v2 to pass through the door, according to DIN 18040-1:2010 [36] and DIN 108040-2:2011 [37] (RW1). The same footprint is also used for the RoboTrainer v2's transport. The concept and implementation of the mechanism for the footprint change are investigated in simulation using Finite element method (FEM) analysis in the Bachelor's thesis of Mayer [116]. This analysis showed that, during regular operation, the wheels' fixtures could take a load of 1000 N with safety factor $S = 5$. The final concept, depicted in figure 4.5, emerged throughout discussions with Mr. Mayer and enables only one person to change the device's footprint, i.e., the technical supervisor.

Regarding human-machine interfaces (HMIs), the RoboTrainer v2 has multiple interfaces to gather information about its user and to present information to them. The input interfaces, i.e., sensors, are the FTS, laser-scanners, and two *Asus Xtion Pro live* RGBD cameras. The laser scanners detect the positions of the user's legs. One of the cameras is mounted on the handle's vertical rod to observe users' feet positions and one on top of the RoboTrainer to observe the user's upper body. The HMIs toward the user are a display mounted on top of the main vertical rod and a RGB LED stripe with individually controllable LEDs on the upper plate of the base. Both can provide feedback for a user, depending on the scenario. More details about those devices are given at the end of this section.

Considering the functional requirements defined in section 4.2.3, the following development and integration considerations are made. Wherever possible, the mechanical structure is built, from off-the-shelf components (R1), e.g., wheels, handles, and profiles for fixing handles, rear wheels and sensors. Aluminum profiles, which form the primary structure of the RoboTrainer v2, enable RoboTrainer's modular design (R2). As standard machining elements, aluminum profiles also enable very flexible and robust assemblies (R7). Custom parts for the RoboTrainer v2 are: the base, upper plate, adapters to connect off-the-shelf components, and some of the covers. The custom RoboTrainer's parts are highlighted with colors in figure 4.6. The base and upper plate as two central components are shown in blue. The green parts are critical for RoboTrainer's construction and therefore realized from aluminum (R7). The magenta-colored parts are 3D printed holders and covers for sensors and electronics.

One of the size-limiting factors of rob@work3 robots, i.e., RoboTrainer Prototype, is their battery package (cf. figure 3.2). Those robots use lead-fleece batteries that are simple to use since they do not need additional electronics to monitor their cells. However, the main disadvantages of those batteries are a rigid cell geometry and low energy density compared to their volume and weight. Therefore, they were often a limiting factor in robot design but, nowadays, are still used in industrial mobile platforms, where not size and weight but the unit price is the most relevant factor. The design flexibility of RoboTrainer v2 is ensured by using lithium-ion polymer (LiPo) batteries, which is the battery type with the highest energy density. LiPo batteries have to be managed carefully, especially regarding physical damage and minimal cell voltage. Therefore, they



(a) Smallest footprint and area for user's feet.

(b) Narrow and long footprint.

(c) Wide and short footprint.

(d) Wide and long footprint.

Figure 4.4.: Extreme positions of the RoboTrainer v2's rear wheels configurations. The black numbers on the right and at the bottom edge of each figure show length and width of its footprint, respectively. The golden square with two numbers in the lower-left and lower-right corner shows the width and length of the area for the user's feet.

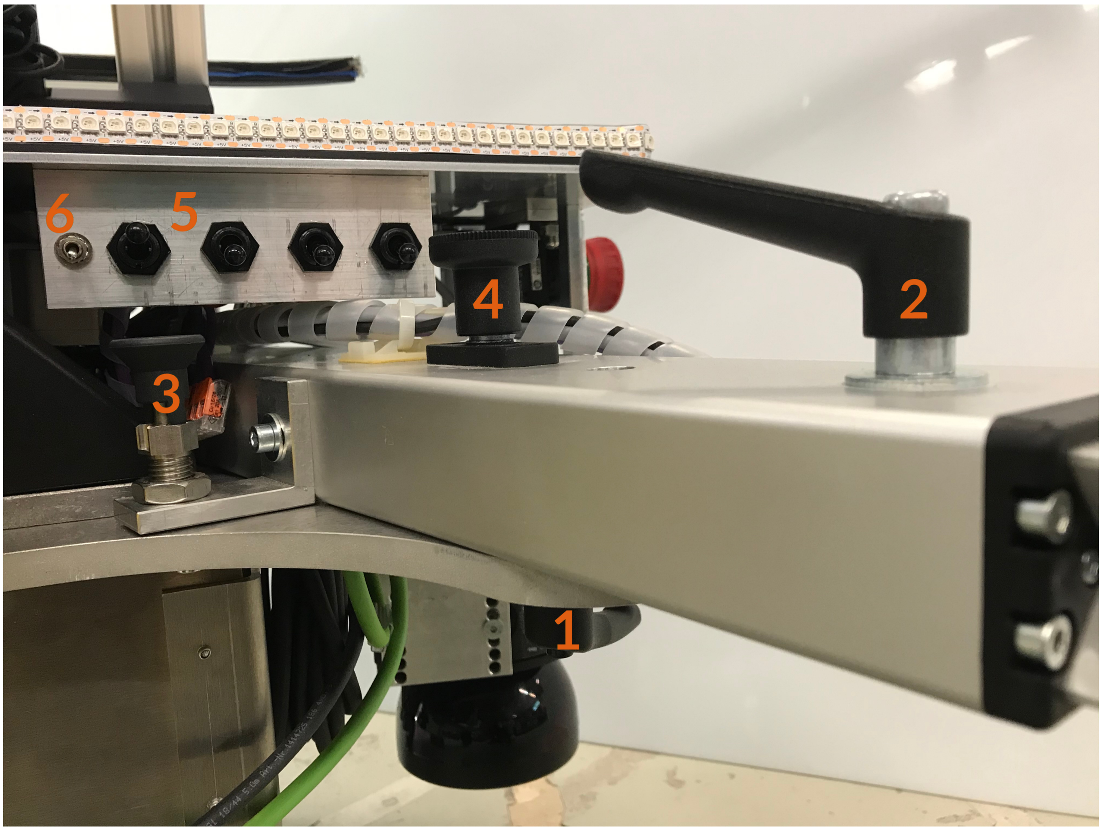


Figure 4.5.: RoboTrainer v2's toolless concept for reconfiguration of the rear wheels and switches to change safety laser scanners' configuration to correspond to the RoboTrainer v2 footprint. The components in the figure are the following: 1 - quick-release skewer for fixing the angle of a rear-wheel; 2 - quick release skewer for fixing the longitudinal position of the rear wheel; 3. indexing plunger for angle (5 positions); 4 - indexing plunger for the longitudinal position (4 positions); 5 - switches for choosing the active observation field of safety laser scanners; and 6 - switch for turning on the laser diode markers.

need additional monitoring electronics for battery cells during the charging process and during use. The battery package of RoboTrainer v2 is placed under the baseplate inside a 3D-printed fixture (shown in magenta in figure 4.6 right) and a custom-made aluminum container (shown in green). The fixture enables simple removal of the battery package from RoboTrainer for charging[6] and ensures secure attachment to the RoboTrainer when in use. The package is made from two individual *SLS APL 21000mAh 6S1P 22,2V 15C+/30C* batteries. Each battery has six LiPo-cells, a nominal voltage of 22 V and a capacity of 21 A h. Compared to the battery package of RoboTrainer Prototype with

[6] Charging is done outside for RoboTrainer v2 for safety reasons since LiPo batteries must be charged on a fire-resistant underground.

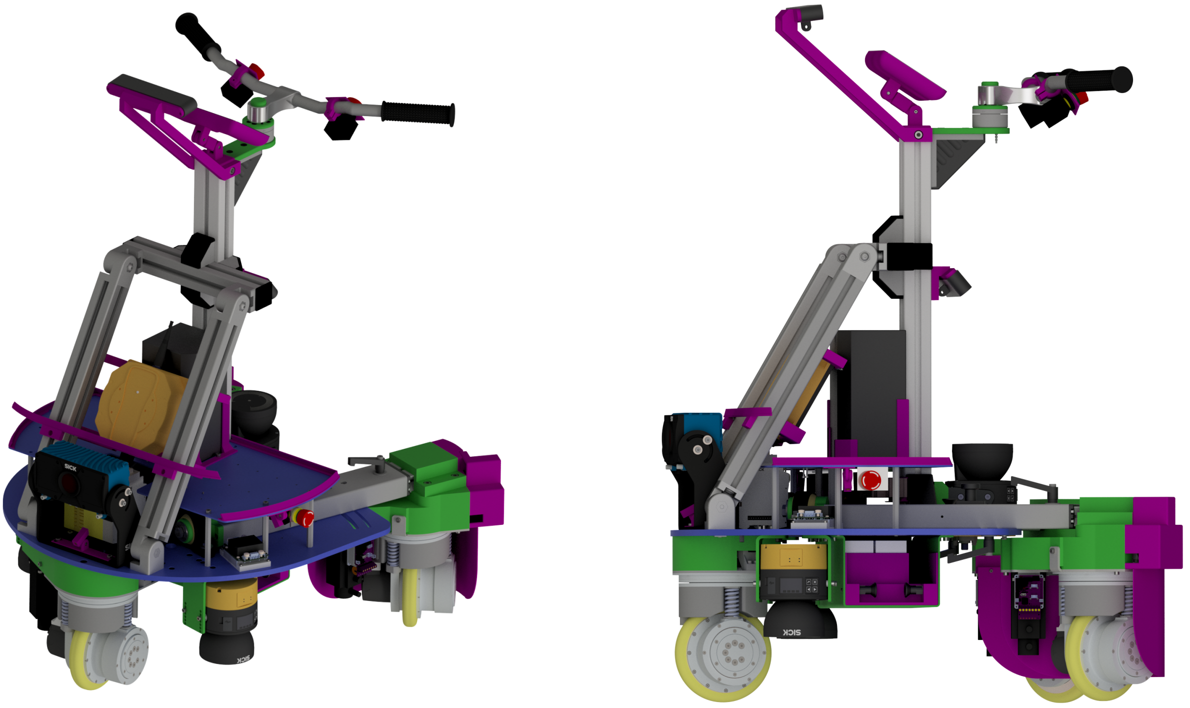


Figure 4.6.: RoboTrainer v2's custom parts in its CAD model. The base and upper plate are shown in blue. Green color marks aluminum parts where stability and robustness are essential, e.g., the base and adapters to fix the wheels. Magenta marks 3D printed parts serving as holders and coverings.

four 12 V batteries with 21 A h each, RoboTrainer v2's battery package has less capacity. Nevertheless, the RoboTrainer v2 has only three wheels, and the goal autonomy time is approximately two hours (R8), compared to five to six hours of autonomy for RoboTrainer Prototype. The two hours is the appropriate time to prepare the RoboTrainer for a one-hour training session and carry it out. There has to be some free time between training sessions to move recorded interaction data from the RoboTrainer's control PC. This time can be used for charging the batteries.

The battery package of the RoboTrainer v2 provides a voltage between 50 V when the battery is full and 42 V when the battery package needs recharging. The battery voltage directly is used only to power the wheels, i.e., the motor controllers. The used motor controllers are *Gold whistle* from *Elmo Motion Control Ltd*. For other electronic components, there are four DC-DC converters from *Traco Electronic AG* integrated into RoboTrainer v2 to provide an adequate power supply for different components. All four converters are used in CMF packaging, i.e., terminal block form with filters for electromagnetic compatibility to EU directives and norms. The following converters are used: (1) 24 V 200 W DC-DC converter with voltage trim-down resistor to 19.6 V for PC supply; (2) 24 V 100 W DC-DC converter for the FTS sensor, SICK Visionary-T camera and safety components; (3) 24 V 100 W DC-DC converter as an intermediate step for (4) 5 V 100 W DC-DC converter for LED-Stripe and laser diode markers.

Figure 4.7 shows RoboTrainer v2 marking all electronic components unrelated to the device's safety. The electronic components marked with green numbers are the following:

Force-torque sensor (1): ATI Industrial Automation Mini58 FTS used as the user's input device. This FTS is chosen for its compact design and high robustness. The exact technical details are given in appendix section B.3.

PC (2): Gigabyte Brix GB-BNi7HG6-1060, with Intel® Core™ i7-7700HQ processor with four cores at a frequency of 2.8/3.8 GHz and NVIDIA GeForce® GTX 1060 mobile graphic card. Additionally, the RAM was upgraded to 32 GB, and a CAN interface PCIe M.2 card from *PEAK-System Technik GmbH*[7] with two channels is added.

3D Environment Sensor (3): SICK Visionary-T indoor 3D camera is used to detect "3D" obstacles, e.g., tables where the laser scanners can only see their legs. The Visionary-T Camera is not a safety-rated sensor but still provides functionality to define protected areas in its field of view (FOW). If an obstacle comes into a specified protection area, an output signal is switched. This feature enables a connection to the safety CPU to provide an additional risk mitigation measure. It is necessary to emphasize that this measure cannot be used in the safety risk assessment.

Camera for 3D upper body detection (4): ASUS Xtion PRO live RGBD sensor is mounted on a custom holder to observe the user's upper body. This sensor was chosen because of its low noise and the ability to provide data for objects at 60 cm distance, which were the main issues when testing other depth sensors, e.g., Intel Realsense.

Camera 3D feet and shin detection (5): Second ASUS Xtion PRO live RGBD sensor is mounted on the central vertical rod with view toward the floor behind the RoboTrainer v2 to track the user's feet position in 3D.

Display for a user (6): the 10-inch display is mounted on the top of the handle's central rod. The technical supervisor uses this display for direct access to RoboTrainer's PC. Furthermore, the display is used to present interaction data to the user during training, e.g., the direction of the user's input forces.

RGB LED Stripe (7): an LED stripe with individually controllable RGB modules is installed on the upper plate around the RoboTrainer v2. The stripe is used as HMI to communicate RoboTrainer's internal state.

[7] The CAN-USB interfaces from the same producer are used in RoboTrainer Prototype.

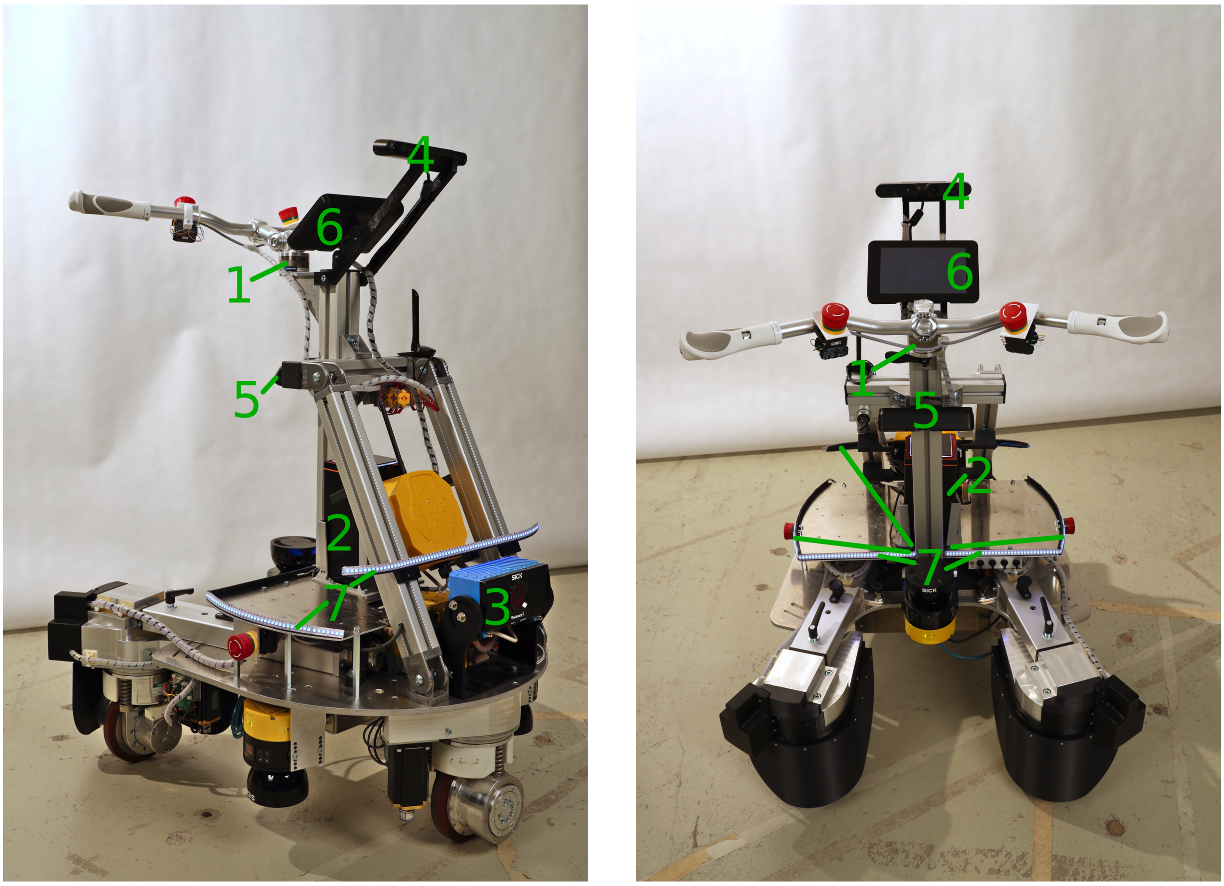


Figure 4.7.: RoboTrainer v2 with marked non-safety components – side view (left) and back view (right). 1 - force-torque sensor; 2 - PC; 3 - 3D environment camera; 4 - 3D camera for upper body detection; 5 - 3D camera for feet and shin detection; 6 - Display toward the user; 7 - LED stripe around RoboTrainer.

4.4. Safety Concept for RoboTrainer v2

The safety considerations of machines and robots often are avoided in scientific research because of the following reasons:

1. time-consuming risk assessment;
2. expensive safety hardware and complicated implementation of safety norms;
3. results of the risk mitigation could collide with the intended functionality; and
4. lack of experience and knowledge of robotic researchers in the area of machinery safety.

The resulting lack of safety explains why so many Smart Walkers (SWs) were never tested with aimed users or, at least, independent persons, but only evaluated by their creators and fellow researchers. The safety concept for RoboTrainer v2 is an essential aspect of its design to enable its evaluation in a pilot study. This section describes the design process and decisions for safety measures and outlines its implementation. Section 4.4.3 discusses the disadvantages of the developed safety concept and possible improvements, providing precious input for the reader.

4.4.1. Concept

Before manufacturing the parts for RoboTrainer v2, a risk analysis based on ISO 12100:2010 [80] and ISO/TR 14121-2:2012 [83] is conducted in two sessions in a group of six persons from IAR-IPR's scientific staff. One of those persons already had industrial experience regarding risk assessment. First, the usual risks like mechanically-caused scratches or bruises, or risks caused by electrical components, are gathered. Subsequently, the specific scenario-related risks are collected. The risks are classified into three categories regarding RoboTrainer's operation modes: (i) the general operation, i.e., risks regarding malfunction of the control, mechanical injuries, electronic components, and radiation from RoboTrainer's sensors; (ii) maintenance mode, i.e., risks concerning technical supervisor (e.g., by reconfiguration of the RoboTrainer v2's footprint) and (iii) training mode for healthy adult persons, i.e., the risks concerning the training.

The risks for users of RoboTrainer have their origin in its functionality and the training design. Those risks emerge in the following hazardous situations:

H1 RoboTrainer is faster than its users and pulls them behind itself;

H2 RoboTrainer collides with its users or a third person, after which the affected person is injured or falls.

H3 User's feet collide with parts of the RoboTrainer, e.g., collision with the wheels when moving side-wards.

H4 RoboTrainer drives over a user's or third person's body parts, e.g., feet, hand, or a finger.

H5 A user or a third person gets trapped between RoboTrainer and a fixed object, e.g., a wall or a pillar.

H6 The RoboTrainer's wheels could injure feet of its users and also third persons when rotating fast.

H7 The RoboTrainer tilts over and injures its user or a third person.

The risk mitigation is conducted according to ISO 12100:2010 [80]. This means that, for each risk, first the construction measures are to be concerned, then measures that require safety functions of control devices, and finally, organizational measures. The Performance Level (PL) for safety functions is determined according to ISO 13849-1:2015 [82].

The last two hazardous situations, H6 and H7, could and therefore had to be solved using construction measures. A solution to mitigate H6 could be wheels' protection to prevent contact between a person and rotating wheel parts when RoboTrainer is running. H7 could be mitigated by the well-aimed placing of components on RoboTrainer's base to set its center of mass as low as possible.

The risk situations H1 to H5 can be further classified into those that endanger the RoboTrainer's user and those that endanger third persons. To protect a mobile robot from a collision with persons in its vicinity and the environment, safety-related environment detection followed by a safety halt is usually the used concept. The two possibilities, speed reduction in the vicinity of other objects and strict safety distance that stops the device immediately, and their advantages and disadvantages are already discussed in section 4.2.2. RoboTrainer v2 uses safety distances (RS5) to avoid collision with a third person or the environment because of its more straightforward realization. Nevertheless, this method and speed reduction are not suitable to protect the user because of its vicinity to RoboTrainer. Further, the distance between RoboTrainer and its user changes during interaction by a few tens of centimeters. Therefore, it is impossible to specify the user's safety distance and so, other strategies to protect the user have to be used. Those strategies involve active handling to stop the RoboTrainer when user feels uncomfortable or overwhelmed, and additionally, a training supervisor should be able to stop the RoboTrainer from a distance. Emergency stop switches should be easily accessible by the user and permanently installed on the device in the first case. A second solution is the use of a wireless emergency stop switch.

To get a value for the safety distance around RoboTrainer v2, an experiment was conducted to measure the stopping distance from RoboTrainer's maximal speed of 1.595 m/s when an emergency halt was activated. The distance, including the reaction time of safety components, was approximately 30 cm. The reason for only approximate measurements is the lack of exact measuring instrumentation and environment. The main issue is to correctly determine RoboTrainer's position and synchronizing such a system with an emergency halt switch. Nevertheless, the measured distance is sufficient since there are also delays in the safety equipment itself. Therefore, an additional safety distance has to be added to this measurement.

In the following subsection, the concrete realization of here discussed concepts is given.

4.4.2. Realization

This subsection describes the realization of the presented safety concept. In the next paragraph, construction safety measures are shortly presented. Before describing the implemented safety functionalities in detail, an overview of utilized safety hardware is given. The safety functionalities and components are used according to the risk assessment and definitions from ISO 13849-1:2015 [82].

Hazardous situation H6 is mitigated using 3D-printed protections on the rear wheels toward the user (see black parts on the rear wheels in figure 4.8). To protect persons in the vicinity of RoboTrainer, a safety functionality that observes the area around the device is used. This functionality is described later. Hazardous situation H7 is mitigated with the same measures to realize requirement RW5, i.e., placing heavier components as low as possible to bring the RoboTrainer v2's center of mass down.

Safety Hardware

Figure 4.8 shows the side and back-view of the RoboTrainer v2 with numbered safety components. The core of the safety hardware is *Flexi Soft Safe EFI-pro System*[8] from *SICK AG*. The system constitutes from a safety CPU, two general-purpose input/output (GPIO) modules, and a safety gateway for connecting safety laser scanners on the one side and the main RoboTrainer's PC on the other side. The GPIO-modules are used for connecting permanently installed emergency stop switches, wireless receiver for emergency stop buttons, and Safe Torque Off (STO) inputs of the motor controllers. The safety CPU, GPIO-modules, gateway, and motor controllers have Performance Level PLe. Also, in all combinations of safety-signal paths, the safety performance is kept at PLe.

RoboTrainer v2 has three *microScan3 Pro - EFI-Pro*[8] safety laser-scanners, also from *SICK AG*. Two of them are mounted on the left and right sides of the RoboTrainer v2 and the third on the back, directed toward the user (see figure 4.8). The laser scanners have a view angle of 270°. Therefore, it is sufficient for many robots to have two of them. This is not the case with RoboTrainer v2 because of its possibility to change the footprint. Because of that, the wheels are placed outside the main plate (see figure 4.4), which results in "shadows" for laser scanners' safety fields on the left and right sides. The shadows significantly limit the view on the user's shin, resulting in a need for the third laser scanner oriented toward the user. Figure 4.9 depicts the laser scanners' safety fields and the resulting "shadows". The laser scanners are placed inside RoboTrainer v2's footprint (requirement RS7). Based on the RoboTrainer's stopping distance

[8] Exact hardware types are provided in section B.1.

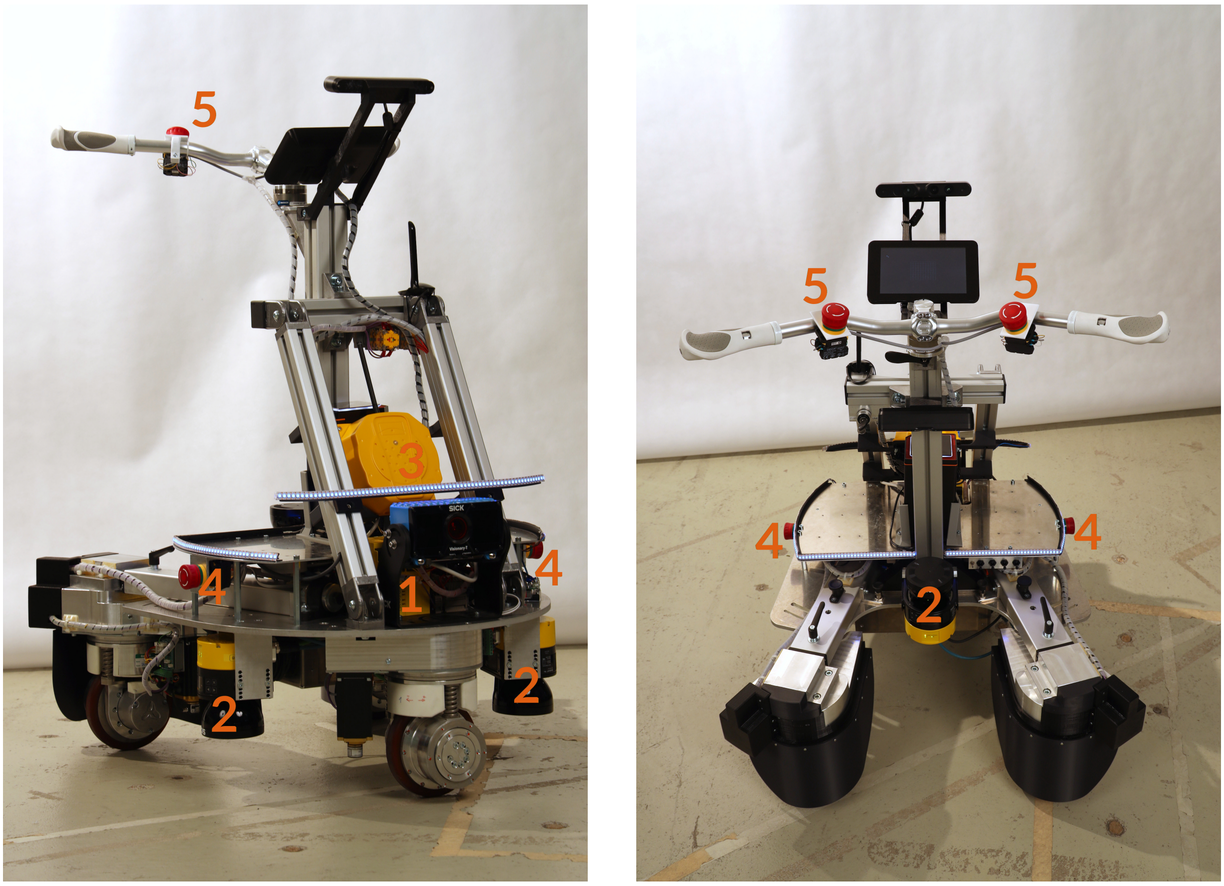
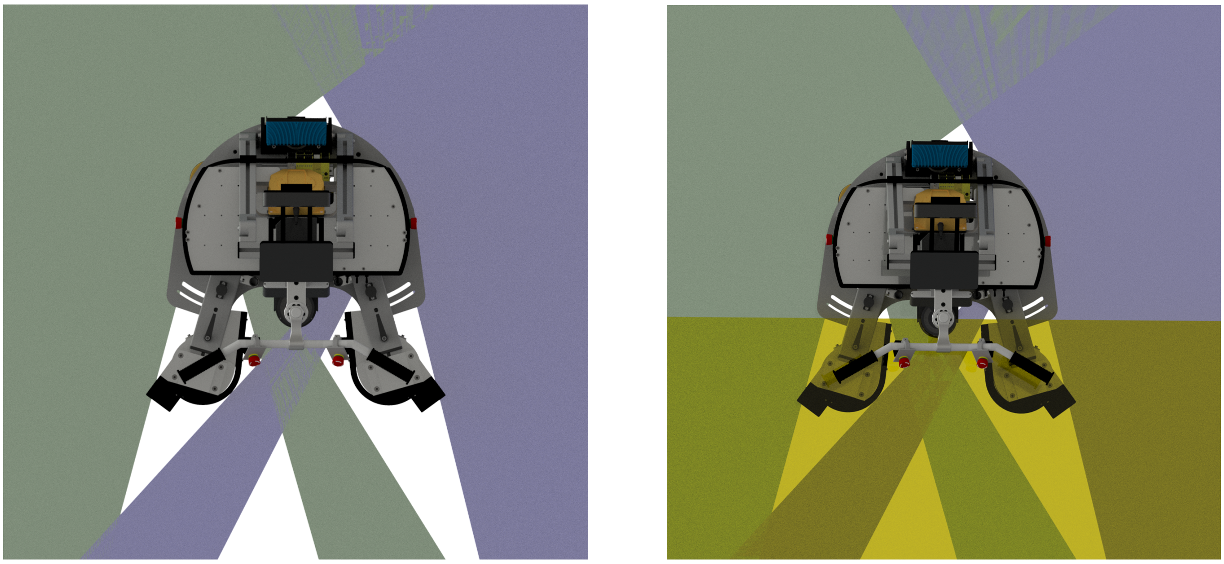


Figure 4.8.: RoboTrainer v2 with marked safety components – side view (left) and back view (right). 1 – SICK Safe EFI-pro System; 2 – SICK Safety laser scanners; 3 – Tyro Gemini 1S – wireless emergency switches button receiver; 4 – Emergency stop switches on the base; 5 – Emergency stop switches on the RoboTrainer‘s handle (for the user).

(30 cm), the laser scanner's response time (60 ms), output activation time (35 ms), the communication delay to safety CPU's (30 ms), and RoboTrainer v2's maximal speed of 1.595 m/s, the safety protection distance according to ISO 12100:2010 [80] can be calculated: $0.3\,\mathrm{m} + 0.125\,\mathrm{s} \cdot 1.595\,\mathrm{m/s} = 0.499\,375\,\mathrm{m} \approx 0.5\,\mathrm{m}$. The safety fields begin at a distance of 60 cm from the RoboTrainer v2 to compensate potential measurement errors of stopping distance and other potential delays of safety equipment, e.g., switching of safety GPIOs. The safety laser scanner *microScan3 Pro* also can stream its raw data (RS6), i.e., distance measurements to the obstacles needed for mapping and localization algorithms. The communication between laser scanners and safety CPU is implemented using the EFI-pro bus with the CIP Safety protocol. The laser scanner configuration is done using *SICK Safety Designer* software. The safety performance level of laser scan-



(a) Left and right laser scanner.

(b) All three laser scanners.

Figure 4.9.: Visualization of the laser scanner's safety fields of RoboTrainer v2 in the closed-short position of rear wheels. Left scanner - dark-green; right scanner - violet; rear scanner - yellow.

ners is PLd and it is not reduced when adding them into the safety-signal path to activate the STO function of the *ELMO Gold Whistle*.

The wireless emergency stop switch from *Tyro Products B.V.*[8] enables the training supervisor to stop RoboTrainer v2 if the user is overwhelmed during the interaction. The wireless emergency switch's essential properties are the minimal reaction time of 0.5 s and the safety Performance Level, PLc. This does not comply with requirement RS2. Nevertheless, the wireless emergency stop enables reaction from a distance in a risky situation or in a case of unexpected device behavior.

Finally, there are four permanently installed emergency stop switches on the RoboTrainer v2. Two of them are on the side of the device, thus accessible for persons around the RoboTrainer. The other two switches are positioned on the RoboTrainer's handle to be easily accessible for its user. All four switches are illuminated and show the safety status of the RoboTrainer as follows: *Illumination off* – at least one of the safety components, i.e., an emergency stop switch, wireless emergency stop or a laser scanner, has a safety condition triggered and the drive-steer modules, i.e., its motors, are not energized; *Illumination on-off at the frequency of 1 Hz* – all safety components are in normal state, i.e., no safety condition triggered, motors are not energized and user confirmation of safe state with power key is needed; *Illumination on* – all safety components are in normal state and the motors are energized.

All named safety components and aforementioned non-safety-related components comply with the *Machinery Directive* (EU's directive 2006/42/EC) and have "CE" markings. The only exception is the LED-control board, designed and realized at the IAR-IPR by Michael Mende.

Safety Functionalities

Based on the above-presented norms and requirements, safety concept and hardware, there are four safety-related functionalities implemented into the RoboTrainer v2:

(i) *(Re-)Start Interlock* (germ. (Wieder-)Anlaufsperre);

(ii) *Emergency stop* (germ. Not-Aus-Schutzeinrichtung);

(iii) *Safety Stop and Risk Mitigation – safety-related environment detection* (germ. Sicherheitsstop und Risikominderung – sicherheitsbezogene Umgebungserkennung) and

(iv) *Wireless Emergency Stop* (germ. Drahtlose Not-Aus Vorrichtung).

The (re-)start interlock function avoids unintended movement of the RoboTrainer v2 or its parts when starting the device or after using emergency stop switches. The function implements additional confirmation before enabling the motors' power (RS1, RS3). The confirmation is done by the technical supervisor using the power key. The interlock is implemented using the "Restart" logic block in the safety CPU. The wiring is designed in dual-channel architecture with the safety performance PLe. The interlock function is the basis for all other safety functions using the same safety paths.

The emergency stop function uses permanently installed emergency stop switches to turn off the motor's power, according to ISO 13482:2014 [81] (requirement RS3). The user activates the switches during training in a dangerous situation and the technical supervisor is using them when reconfiguring RoboTrainer's footprint. The primary purpose of these switches is to bring RoboTrainer to a safe state. The wiring for this functionality is designed in dual-channel architecture with PLe of involved safety devices (RS2). After activating this functionality, the (re-)start interlock is also activated.

The safety stop is based on a safety-related environment detection and uses three laser scanners permanently mounted on the RoboTrainer (RS5). This safety functionality does not activate (re-)start interlock, which means that the motor power is automatically reactivated when obstacles are removed from the safety field of all laser scanners (RS4). The laser scanners limit the safety performance of this functionality to PLd. Since RoboTrainer v2 can change its footprint and the laser scanners are placed inside (RS7), multiple pre-configured safety fields need to be adjusted to the RoboTrainer's current footprint. This adjustment can be performed using four black switches on the back of the RoboTrainer v2 shown in figure 4.5. The configuration of the laser scanners and their fields is done using *SICK Safety Designer* software.

The final safety functionality of RoboTrainer v2 is a wireless emergency stop. The training supervisor holds the wireless emergency switch during the whole training. After activation of this functionality, the safety interlock is also activated. The wiring is done in dual-channel architecture resulting in PLc, where the performance limiting factor is wireless communication.

4.4.3. Discussion on Further Safety Measures

As a research platform, RoboTrainer v2's safety hardware and software are designed with only few restrictions. Therefore, there are still some open safety questions for the unsupervised use of RoboTrainer. This section shows different options for extending the safety features and discusses consequences on the use and control of RoboTrainer.

The behavior of RoboTrainer v2 is currently independent of the fact if the user is standing behind it and holding its handles or not. Such functionality would be a substantial improvement from a safety perspective since movement of the RoboTrainer would only be allowed when the user is holding the device properly with both hands. This would enable some high-level functionalities, such as automatic drift compensation of force-torque sensors to increase robustness of the control. To address this issue, the integration of three-position enabling switches was considered during design process of RoboTrainer v2. Those switches are colloquially known as "dead man" [9] and are industry standard for protecting manual robot control tools, e.g., teach pendants and remote controllers.

With the three-position enabling switches, robot's movement would only be possible if held in the middle position and would react if the operator releases it, i.e., relaxes its grip or stiffs the grip and presses the switch. During the pilot studies and the tests throughout development, it was observed that users either like to use a firm grip around handles or, mostly after some experience with the RoboTrainer, hold the handles relatively relaxed. Using three-position enabling switches would probably draw the user's attention from the actual training and make it more complicated when using control actions. Holding the switch in the middle position could introduce a new risk or even make it impossible to conduct training with challenging control actions, i.e., repeatedly activation of the switch. Using two-position enabling switches would solve the issue when the user stiffs his hands firmly, but it would still need an additional user's attention. Another solution is to use capacitance sensors in handles as, for example, used in the drive-wheel of a car. However, there are no off-the-shelf safety capacitance sensors on the market that would be easy to integrate into the RoboTrainer's handle.

[9] A good explanation of this term is provided on Wikipedia: `https://en.wikipedia.org/wiki/Dead_man%27s_switch`

The safe detection of the user's hands would also have a significant impact on other RoboTrainer's components, especially on the control of the base. The software for controlling the wheels would need to be adapted to support the initialization, i.e., zeroing the wheels (steering motor) of the RoboTrainer without communication to the driving motor. Therefore, the software would need to support two use-cases, an *autonomous* mode where a user should not be near RoboTrainer and an *interactive* mode where the user's presence and direct contact to RoboTrainer's handles are expected. Finally, the cabling of drive-steer wheels has to be changed, as well as the safety logic, to control the Safe Torque Off (STO) function for a drive and a steering motor separately.

For unsupervised training, the information about contact between the user and the RoboTrainer is not sufficient, but the information, if a user is in the vicinity of it, is essential. To do this safely, the contour detection functionality of safety laser scanners has to be further investigated. Using this functionality would make it possible to detect a person's legs and avoid any automatic movement if a person is too close to RoboTrainer.

When preparing the RoboTrainer v2 for the evaluation, the safety laser scanner configuration for different kinematics was complicated and very time-consuming to implement. The main issues were shadows from the drive-steer modules and the position of the rear laser scanner. As a result, the left and right laser scanner, placed under RoboTrainer v2's base, can not observe the area between the rear wheels, i.e., the user's legs, in all configurations. On the other hand, the area between the rear wheels for the user's legs is significantly narrowed by the rear laser scanner's safety fields. The main reason for this is its position in the middle of the footprint. A solution for this issue could be detecting contact between the user and the rear wheel protections, i.e., 3D printed black parts on the rear wheels in figure 4.8. Nevertheless, the off-the-shelf safety edges and bumpers were not suitable because they would limit the area for the user's feet and their integration would be very complex to enable different kinematic configurations. A custom solution in form of sensitive skin would be needed to achieve this.

4.5. Conclusion on the Device Design

This chapter discusses the design of a novel research device for active training – called RoboTrainer v2. First, the use of the device and the roles of the persons involved in the training are explained. Since the device is inspired by a wheeled walker for elderly persons but has properties of a mobile robot, the devices' technical requirements are investigated from the booth's perspectives. This analysis outlines the relevant standards and discusses their influence on RoboTrainer v2's design. The two main parts of the chapter are mechatronic design and safety implementation.

The RoboTrainer v2 is developed as a research tool for a controlled environment to investigate how a robotic system can support and challenge its user during motor training. It is designed as a unique device with this functionality in mind. Therefore, the RoboTrainer v2, due to its complexity of use and price, is not suitable for broader production and use. Nevertheless, during the development and work on the RoboTrainer v2, potential improvements are gathered to address those issues. Some of those are:

- Use of ultrasound sensors for safety protection of its surrounding in combination with smaller and cheaper non-safety laser scanners;
- Integration of sensors to detect if the user is holding the handles;
- Addition of sensors for automatic detection of RoboTrainer's kinematic configuration;
- Indication of battery level on the body of the RoboTrainer and automatic shutdown if battery level is low;
- Addition of diffusion on the LED stripes to improve their visibility from the user's perspective.

There are, of course, many other possibilities to optimize the RoboTrainer v2. In its current form, it enables many novel experiments where participants are involved, especially in human-robot interaction, biomechanics, and the research on influence of motor activation to an individual's cognition.

5 Control of Devices for Active Training

The state-of-the-art overview (section 2.3.4) presents various approaches for controlling Smart Walkers (SWs). Generally, the control method's choice depends on used hardware, i.e., sensors and walker's kinematics. As the next-generation RoboTrainer, *RoboTrainer v2* has very similar hardware and software to *RoboTrainer Prototype*. RoboTrainer v2 also has an omnidirectional mobile base and it uses a force-torque sensor (FTS) as input device. In these cases, the walker's movement is usually controlled by the admittance equation (equation (2.12)). Such a first-order system, i.e., PT1 element, is a natural choice since the main goal is to convert the user's input forces into the walker's velocity. The admittance control also has a few other advantages. First, it provides intuitive interaction toward a user, for example, with a shopping cart; second, its dynamics can be adjusted using two intuitive parameters, i.e., mass and damping.

The main difference between controllers for RoboTrainer Prototype and RoboTrainer v2 is that the first uses feed-forward admittance control, i.e., open-loop, to generate the velocity reference for its mobile base, the second one uses feedback, i.e., closed-loop approach. The control approach of the RoboTrainer v2 is admittance control with a model using pole placement.

This chapter presents the control architecture of RoboTrainer v2 and discusses different aspects of it. In section 5.1, an overview of the controller architecture and the control strategy is given. Moreover, transformations equations and the exact control values for the RoboTrainer v2 are given to better understand the thesis' concepts. Section 5.2 describes the interaction between user and RoboTrainer, considering its influence on the walker's controller, user-walker-system properties and stability. Section 5.3 starts with the state-of-the-art method developed by Chuy, Hirata, and Kosuge [27] to avoid oscillations during the interaction. In the first part, an extension of the user-walker interaction model for control is presented. The coupling factor introduced in [27] is defined more precisely and additional inertia to model the user's reaction performance is introduced. The section further discusses the implications of the proposed model regarding the user-walker system's stability. The second part presents the application and extension of the [27]'s approach for the RoboTrainer (section 5.3.2). Finally, the stabilization approach is optimized in terms of performance and provides a refined implementation – initially done during the Bachelor's Thesis of Muth [124].

The next contribution is the investigation of velocity-dependent adaptions of the admittance equation's parameters in section 5.4. The main reason for this is reducing the user's effort to move with a walker on constant velocity. Yu, Spenko, and Dubowsky [185] were the first who investigated this and made the proposal to reduce the controller's damping with increase of the walker's velocity. A second approach, initially developed during the Master's Thesis of Zumkeller [189], adjusts the user's input force ahead of the controller and, therefore, influences the mass and the damping of the admittance equation in the same proportion. In section 5.5, two strategies for individualizing the admittance equation for a specific user are presented. The first strategy measures the user's interaction force with the RoboTrainer and adjusts the controller's mass and damping. The second strategy aims to adjust the velocity-dependent adaption margins based on the user's distance to a Smart Walker. The approaches mentioned above and the short results are partially published in [157]. The chapter finalizes in section 5.6 with the details about implementing the presented control approach using ros_control framework [24].

5.1. Control Architecture and Strategy

This section gives an overview of the control architecture of RoboTrainer v2 and explains its differences from the controller used in the prototype presented in section 3.2.2. Figure 5.1 depicts the controller's architecture. The RoboTrainer is controlled in Cartesian space, leveraging the same direct kinematics (DK) and inverse kinematics (IK) libraries used in RoboTrainer Prototype. The wheel controller adjusts the wheels' steer angle using an impedance-position controller without influencing the driving dynamics of the RoboTrainer. The RoboTrainer's motor controllers, i.e., *ELMO Gold Whistle* boards (see section 4.3), together with their physical properties, determine the device's natural dynamics. Those controllers are parameterized to attain the best performance by using the manufacturer's auto-identification procedure. From the control perspective of this thesis, the mentioned components, depicted in black in figure 5.1, are considered RoboTrainer's “hardware”.

The extended admittance controller from RoboTrainer Prototype and the initially implemented control actions (CAs) are depicted in green in figure 5.1. For RoboTrainer v2, the CAs are implemented with their admittance elements (see section 6.1), influencing the

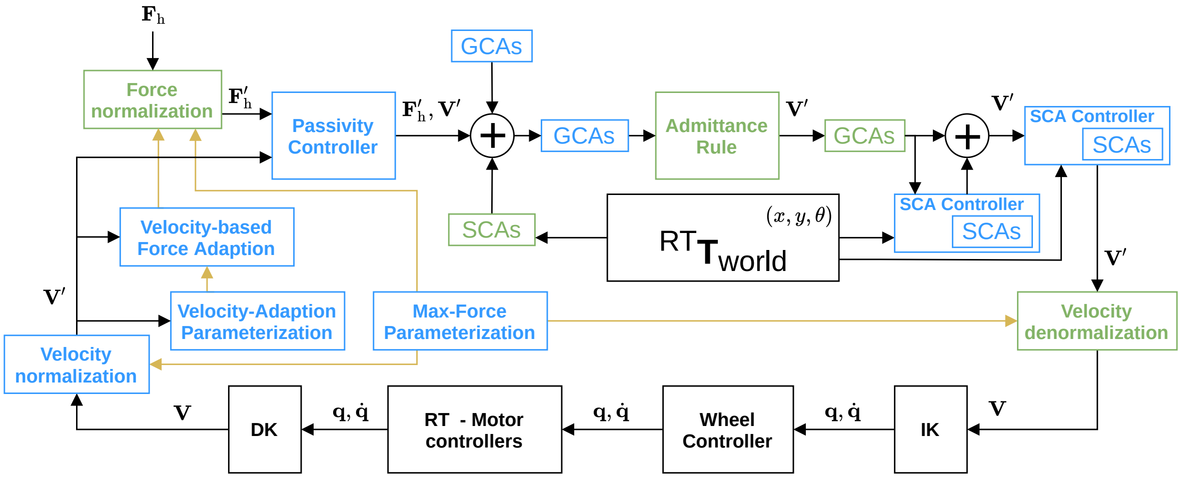


Figure 5.1.: Overview of the RoboTrainer v2 control architecture. The blocks marked in black represent the hardware, low-level controllers and localization module. The green color marks the extended admittance controller from RoboTrainer Prototype (see section 3.2.2), and the blue color marks the functionalities added for the RoboTrainer v2. Yellow arrows represent data flow relevant to the parameterization and adaption components. The acronyms *SCAs* and *GCAs* depict the influence of the spatial control actions (SCAs) and global control actions (GCAs) on the controller's data flows. Those are described in detail in chapter 6.

main admittance controller's output velocity. This separation is done in order to reduce the influence of control actions on the internal states of the main admittance controller and achieve better control over the behavior of the RoboTrainer. Furthermore, each CA is encapsulated into a specialized passivity controller (SCA Controller) to increase user safety (see section 6.2). The details on the control actions are provided in chapter 6.

The new functionalities, i.e., per-user individual parameters, velocity-based adaption, and passive behavior, modify the user's input and therefore are placed ahead of the extended admittance controller. These components are shown in blue (figure 5.1). The application of individual parameters and velocity-based adaption adjusts the upper limit of the input force and output velocity. Therefore, these components influence the controller parameters indirectly. This approach guarantees the correct application of maximal values and a more straightforward implementation. The parameterization and adaptation data-flows are marked with yellow arrows (figure 5.1). The remaining sections of this chapter provide the concepts and implementation details about the components mentioned above.

5.1.1. Details on the Implementation of the Control Strategy for RoboTrainer

This section describes the coherence between the admittance equation's constants and normalization constants, i.e., upper limits, of the controller's input force and output velocity. The provided equations enable a better understanding of the concepts and discussions in this chapter and their comparison to state-of-the-art. As for the RoboTrainer Prototype, the admittance controller's inputs and outputs are bounded in the range $[-1, 1]$ to provide the portability and hardware-agnostic parameter tuning for the desired dynamics. Since these limits indirectly influence the physical admittance parameters, i.e., mass in kg and damping in N/(m/s), transformations between implementation and physical parameters are needed. The physical properties and default controller parameters for the RoboTrainer v2 are given in table 5.1.

Table 5.1.: Physical properties and default controller parameters of the RoboTrainer v2. The angular velocity of the RoboTrainer depends on the actual setup of the wheels (see section 4.3, figure 4.4). Therefore, the velocity around the z-axis is given as the maximal radial velocity. The gain K and time constant T are defined as in equation (2.17).

	Parameter	Value
	Weight M_{a}	84.5 kg
Physical Specifications Maximal Force and Torque Limits $\boldsymbol{F}$	F_{x}	±100 N
	F_{y}	±100 N
	T_{z}	±30 N m
Maximal Velocity Limits $\boldsymbol{V}$	V_{x}	±1.6 m/s
	V_{y}	±1.2 m/s
	V_{z}	±1.6 m/s
Default Gain $\boldsymbol{K}$	K_{x}	1.3
	K_{y}	1.3
	K_{z}	1.0
Default Time Constant $\boldsymbol{T}$	T_{x}	0.5
	T_{y}	0.5
	T_{z}	0.3

The RoboTrainer's controller uses equation (5.1) as the central admittance equation[1]:

$$\boldsymbol{V}(k) = \boldsymbol{F}(k-1) * \boldsymbol{K}\left(1 - e^{\frac{-1}{\boldsymbol{T}r}}\right) + \boldsymbol{V}(k-1) * e^{\frac{-1}{\boldsymbol{T}r}} \tag{5.1}$$

[1] The equations in this and the following sections use the operator "$*$" and fraction with matrices for element-wise calculation explained in section 1.4.

where

$$\boldsymbol{K} = \frac{1}{\boldsymbol{D}} \tag{5.2a}$$

$$\boldsymbol{T} = \frac{\boldsymbol{M}}{\boldsymbol{D}} \tag{5.2b}$$

$\boldsymbol{D}$ and $\boldsymbol{M}$ are equivalents of the constants for the time-continuous admittance equation (2.12). The constant $r = \frac{1}{\Delta t}$ is the controller's update rate.

Transformations to determine the controller's physical properties, i.e., mass and damping, can be determined using the time-continuous transfer function (equation (2.16)). The normalization factor for the damping constant $\boldsymbol{D}_0$ can be calculated using the steady-state velocity equation (2.23):

$$\boldsymbol{D}_0 = \frac{\boldsymbol{F}_0}{\boldsymbol{V}_0} \tag{5.3}$$

where $\boldsymbol{F}_0$ and $\boldsymbol{V}_0$ represent the upper limits for the user's input force and device's output velocity. Using the controller's default values from table 5.1, equation (5.2a), and equation (5.3), the damping factor for each degree of freedom (DOF), i.e., linear movement along the x and the y-axis, and the rotation around the z-axis, can be calculated (equation (5.4)).

$$\frac{\boldsymbol{D}}{\boldsymbol{D}_0} = \frac{1}{\boldsymbol{K}} \Rightarrow \boldsymbol{D} = \frac{\boldsymbol{D}_0}{\boldsymbol{K}} = \frac{\boldsymbol{F}_0}{\boldsymbol{V}_0} * \frac{1}{\boldsymbol{K}} = \begin{bmatrix} D_\mathrm{x} \\ D_\mathrm{y} \\ D_\mathrm{z} \end{bmatrix} = \begin{bmatrix} 62.5\,\mathrm{N/(m\,s)} \\ 62.5\,\mathrm{N/(m\,s)} \\ 25\,\mathrm{N/(m\,s)} \end{bmatrix} \tag{5.4}$$

Using these results and equation (5.2b), the controller's default mass constant $\boldsymbol{M}$ can be calculated (equation (5.5)).

$$\boldsymbol{M} = \boldsymbol{D} \cdot \boldsymbol{T} = \frac{\boldsymbol{F}_0}{\boldsymbol{V}_0} * \frac{\boldsymbol{T}}{\boldsymbol{K}} \begin{bmatrix} M_\mathrm{x} \\ M_\mathrm{y} \\ M_\mathrm{z} \end{bmatrix} = \begin{bmatrix} 31.25\,\mathrm{kg} \\ 31.25\,\mathrm{kg} \\ 12.5\,\mathrm{kg} \end{bmatrix} \tag{5.5}$$

Figure 5.2 depicts the local and controller coordinate system. The coordinate system is right-handed.

5.2. Description of the User-RoboTrainer Interaction for Control

Observing the interaction between the user and a Smart Walker, only the controller's overall dynamics is interesting. A schematic representation of the user-walker interaction

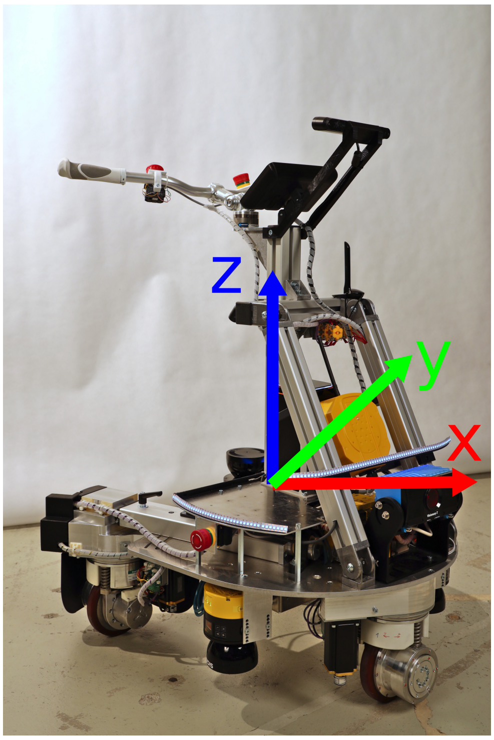


Figure 5.2.: Right-handed, controller's coordinate system of the RoboTrainer v2. From the user's perspective, the x-axis shows to the front, the y-axis to the left, and the z-axis up.

is shown in figure 5.3. In the scheme, the extended admittance controller represents the whole upper part of figure 5.1, $\boldsymbol{F}_{\mathrm{h}}$ is equivalent to $\boldsymbol{F}_{\mathrm{in}}$ and the low-level controller and the hardware of the RoboTrainer are presented as one block. Holding the handles of the RoboTrainer, a user experiences its acceleration and force generated by its movement. Depending on the concrete interaction situation, the control loop in figure 5.3 can be seen as feed-forward, i.e., open, or feedback, i.e., closed. These situations are described in the remainder of this section.

During the intended use of a Smart Walker, users decouple by their body measured input force F_{h} from the influence of acceleration a_{out} and force F_{out} resulting from the walker's movement (cf. figure 5.3). Therefore, this use-case can be described with the following feed-forward control loop:

$$\boldsymbol{v}_{\mathrm{RT}} = \boldsymbol{G}_{\mathrm{RT}} * \boldsymbol{G}_{\mathrm{controller}} * \boldsymbol{G}_{\mathrm{FTS}} * \boldsymbol{F}_{\mathrm{h}} \tag{5.6}$$

where $\boldsymbol{v}_{\mathrm{RT}} = \begin{bmatrix} v_{\mathrm{x}} & v_{\mathrm{y}} & \omega_{\mathrm{z}} \end{bmatrix}^{\mathsf{T}}$ is the resulting three-dimensional velocity of a smart walker. The $\boldsymbol{F}_{\mathrm{h}} = \begin{bmatrix} F_{\mathrm{x}} & F_{\mathrm{y}} & F_{\mathrm{z}} & M_{\mathrm{x}} & M_{\mathrm{y}} & M_{\mathrm{z}} \end{bmatrix}^{\mathsf{T}}$ denotes 6D input forces and moments from the user measured by the FTS with transfer function $\boldsymbol{G}_{\mathrm{FTS}} = \mathrm{diag}\begin{pmatrix} f_{\mathrm{F_x}} & f_{\mathrm{F_y}} & f_{\mathrm{F_z}} & f_{\mathrm{M_x}} & f_{\mathrm{M_y}} & f_{\mathrm{M_z}} \end{pmatrix}$, where f marks a transfer function for each axis. The measured force is used as input into the device's controller with transfer function $\boldsymbol{G}_{\mathrm{controller}} = \mathrm{diag}\begin{pmatrix} g_{\mathrm{CTRL_x}} & g_{\mathrm{CTRL_y}} & 0 & 0 & 0 & g_{\mathrm{CTRL_z}} \end{pmatrix}$, where

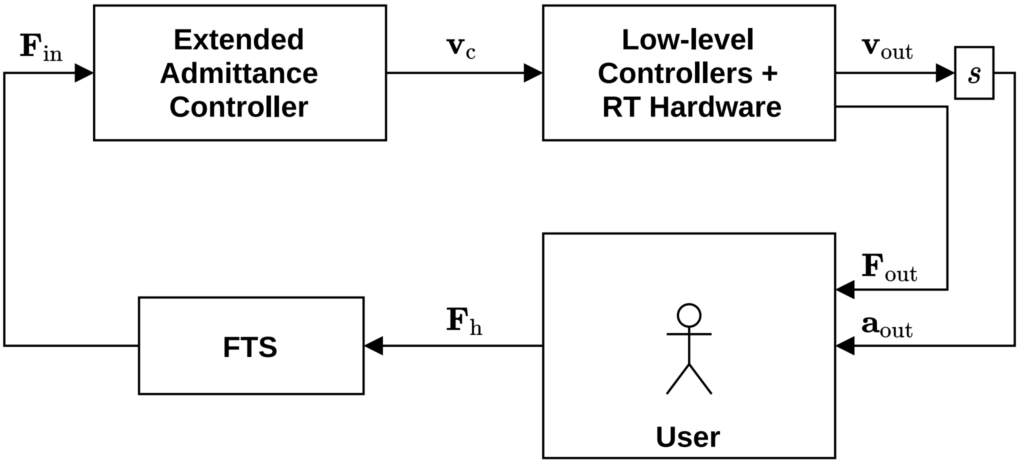

$\mathbf{F}_{\mathrm{in}}$
Extended Admittance Controller
$\mathbf{v}_{\mathrm{c}}$
Low-level Controllers + RT Hardware
$\mathbf{v}_{\mathrm{out}}$
s
$\mathbf{F}_{\mathrm{out}}$
$\mathbf{a}_{\mathrm{out}}$
$\mathbf{F}_{\mathrm{h}}$
FTS
User

Figure 5.3.: The generalized control loop of the user interaction with a Smart Walker (SW), i.e., RoboTrainer. When the device is used as expected, a user decouples the RoboTrainer's output acceleration $\boldsymbol{a}_{\mathrm{out}}$ and force $\mathbf{F}_{\mathrm{out}}$ caused by its movement from its input force $\mathbf{F}_{\mathrm{h}}$. The RoboTrainer's output force experienced by the user is due to the acceleration difference between the device and the user's hands.

g represents a transfer function for the respective degree of freedom. The transfer function $\boldsymbol{G}_{\mathrm{RT}} = \mathrm{diag}(\, g_{\mathrm{RT_x}} \; g_{\mathrm{RT_y}} \; g_{\mathrm{RT_z}} \,)$ includes the inverse-kinematics, low-level control, and the device itself.

The RoboTrainer's controller $\boldsymbol{G}_{\mathrm{controller}}$ considers each degree of freedom independently. This is the usual control approach when using an FTS as the input device of a SW, because it enables easy use with a predictable outcome. Still, challenging interaction strategies are feasible by transforming the user's input force or controller's output velocity.

Initially, the controller dynamics used for RoboTrainer v2 is the same as that of RoboTrainer Prototype with experimentally determined values. The goal is to adjust the controller's parameters to provide suitable dynamics, i.e., movement agility, for the user. If the parameters are unsuitable, the walker feels heavy or too light and oscillates in the user's hands. Yu, Spenko, and Dubowsky [185] gave a detailed investigation of such effects (see chapter 2). The default values of the used admittance controller are given in table 5.1.

The feed-forward control from equation (5.6) can be further simplified by approximating the transfer functions of the FTS with a static gain $\boldsymbol{G}_{\mathrm{FTS}} = \boldsymbol{G}_{\mathrm{RT}} = 1$. The reasoning for approximating the FTS transfer function is that its measurement frequency of 800 Hz is much higher than the control frequency of 50 Hz. Therefore, the force filters' dynamics, i.e., mean and low-pass filter, is insignificant compared to the admittance controller dynamics. Figure 5.4 confirms this assumption by comparing the input and filtered force's dynamics with the velocity generated by the admittance controller and the measured RoboTrainer's velocity.

Since the hardware's velocity does not track the velocity generated by the admittance controller, the dynamics of the RoboTrainer's hardware, i.e., inverse kinematics and motor controllers, is investigated more closely. An experiment using the reference velocity, shown in figure 5.5, is conducted to identify hardware dynamics. The reference velocity provokes step responses of RoboTrainer changing the reference by 10% above and below the velocity of 0.5 m/s. This enables an identification of transfer function without bringing the motors in saturation that happens when RoboTrainer's velocity changes from zero, e.g., hardware response around time 1 s. Therefore, for the identification, data between the time 2 s and 6 s were used. The measured response includes the inverse kinematics, low-level controller, CAN communication delays, internal dynamics of the motor controllers, e.g., acceleration ramps, and dynamics defined by RoboTrainer v2's physical properties. Since the measured response has overshot, the identified system is expected to be at least second-order. Using *MATLAB*'s [115] *control-toolbox*, the measured response is identified by testing continuous-time transfer functions up to the tenth order with a different number of zeros. The identified fourth-order transfer function has three minimal-phase zeros and a fit of 89.78% (best fit of all tested systems). The identified system has two real $p_1 = -1.20$ and $p_2 = -2.18$ and two complex conjugate roots $p_2 = -5.01 \pm 16.38i$. Figure 5.6 shows verification of the identified transfer function with the controller's velocity from figure 5.4.

Also, an identification based on data from figure 5.4 was done. This time, a first-order transfer function was used based on the expected dynamics from the data. The estimated system has the pole $p = -6.29$ without a zero and a fit of 88.61%. Its response to the user input is shown in figure 5.7. The latter estimation provides relevant information about RoboTrainer v2's dynamics during actual usage. Comparing the poles of the default admittance controller $p_\mathrm{c} = -2$ (see section 5.1.1) and the estimated one $p_\mathrm{e} = -6.29$, it is concluded that the admittance controller dynamics is the most relevant for RoboTrainer v2's movement. This difference in dynamic is also visible in figure 5.4, from which it can be assumed that the admittance controller has the slowest dynamics. Therefore, in the remainder of this chapter, it is assumed that $\boldsymbol{G}_\mathrm{RT} = 1$, to keep the discussion as straightforward as possible.

The control loop in figure 5.3 is closed by the user detecting acceleration and force resulting from the RoboTrainer's movement. The acceleration is caused by the RoboTrainer itself and the force due to the acceleration's difference between RoboTrainer and the user's hands. Although those two states are not separable by physics laws, depending on the use of the RoboTrainer, their significance varies in different analyses. The acceleration-feedback is more relevant when the device is used in *compliance* mode as a Smart Walker. In this mode, users adjust their input force depending on the movement, i.e., the acceleration of the device. The force-feedback gains relevance by activating con-

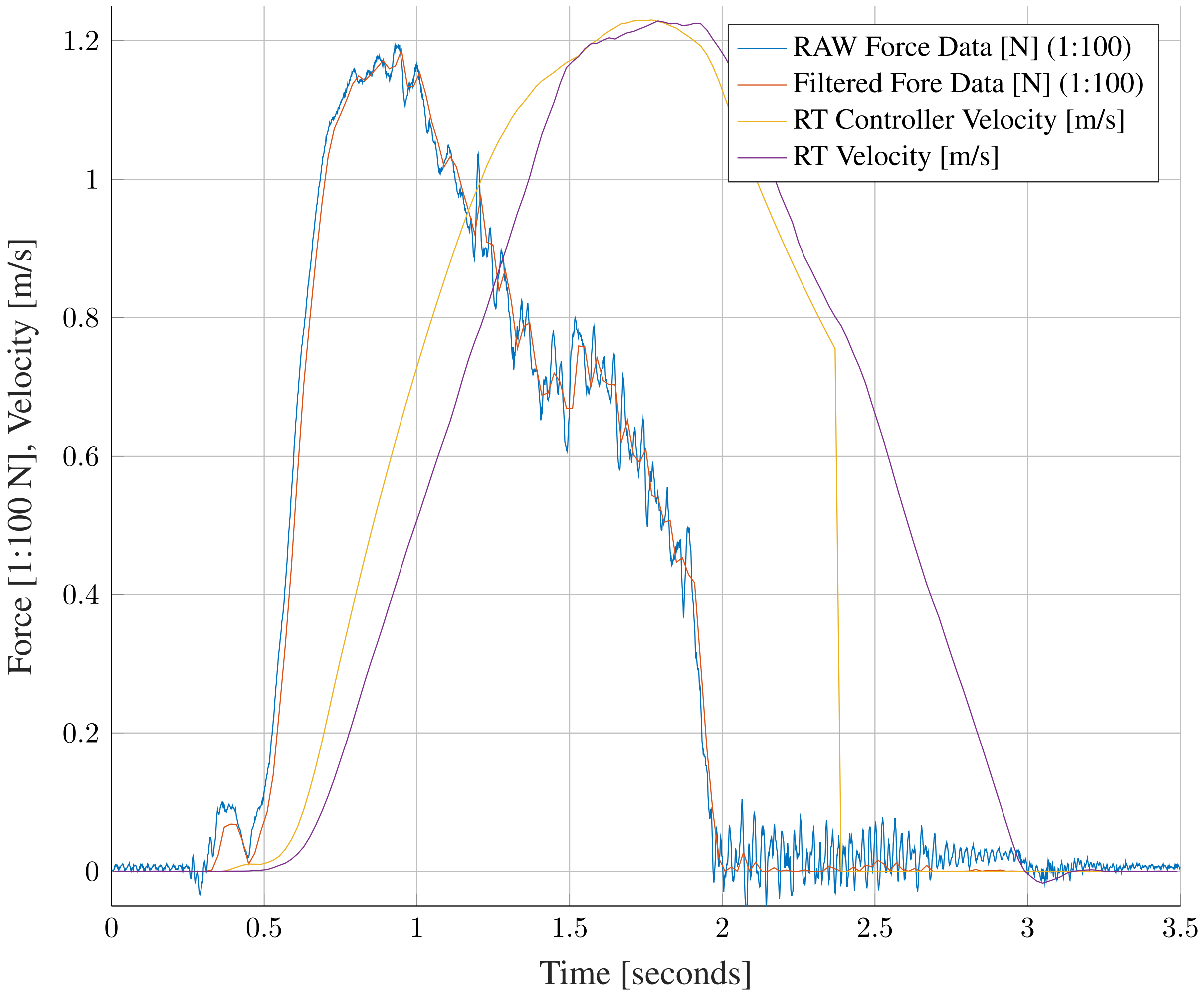


Figure 5.4.: The dynamic responses of the RoboTrainer v2's force filters, admittance controller, and the measured RoboTrainer's velocity by the motor controllers. The dynamics of the force filters are insignificant compared to the admittance controller, i.e., the filtered force on the controller's input (red) is almost the same as the raw force input (blue). The admittance controller's output velocity (yellow) determines the dynamics of the hardware's velocity (violet). After a sudden release of the handles (time: 2.4 s), the RoboTrainer v2 stops with a predefined ramp in the motor controllers.

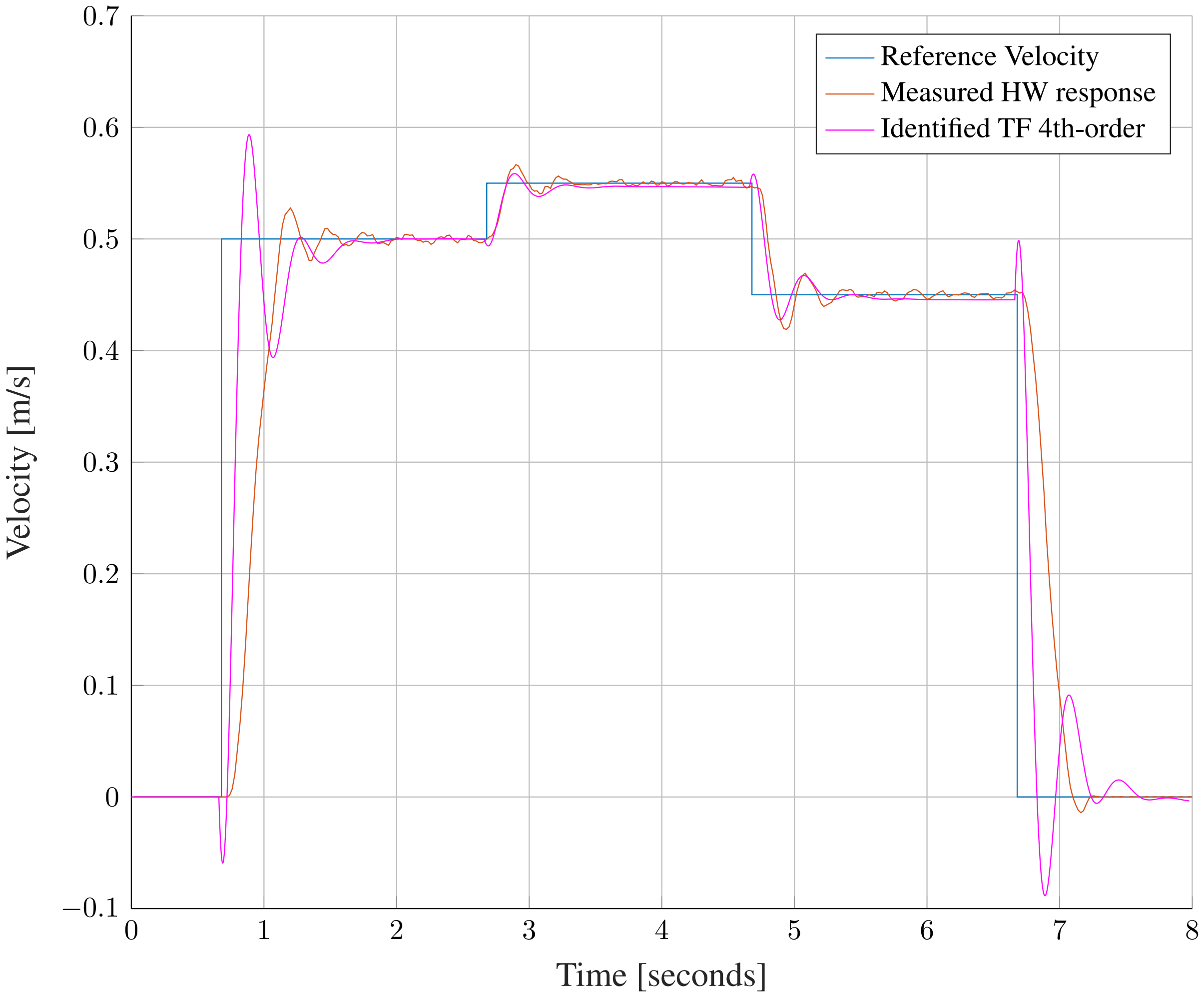


Figure 5.5.: RoboTrainer v2's hardware dynamic response to the reference velocity of 1 m/s. The transfer functions 4th order get a fit of 89.78%. The blue line represents the reference velocity, the red line the measured device's velocity, and the magenta line the response of the identified transfer function.

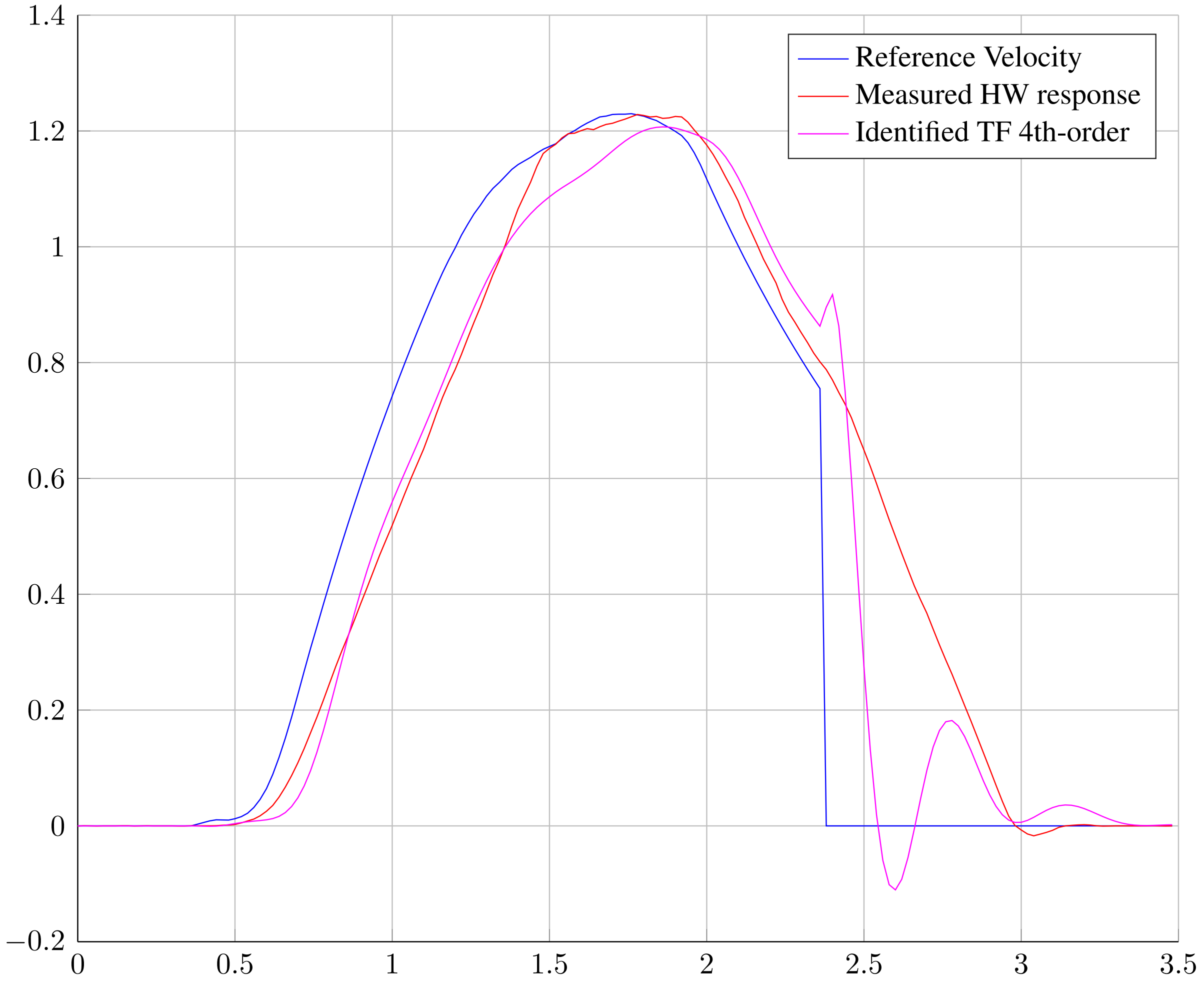


Figure 5.6.: RoboTrainer v2's hardware dynamic response to the RT controller velocity from figure 5.4. The blue line represents the reference velocity, the red line the measured device's velocity, and the magenta line the response of the identified 4th-order transfer function.

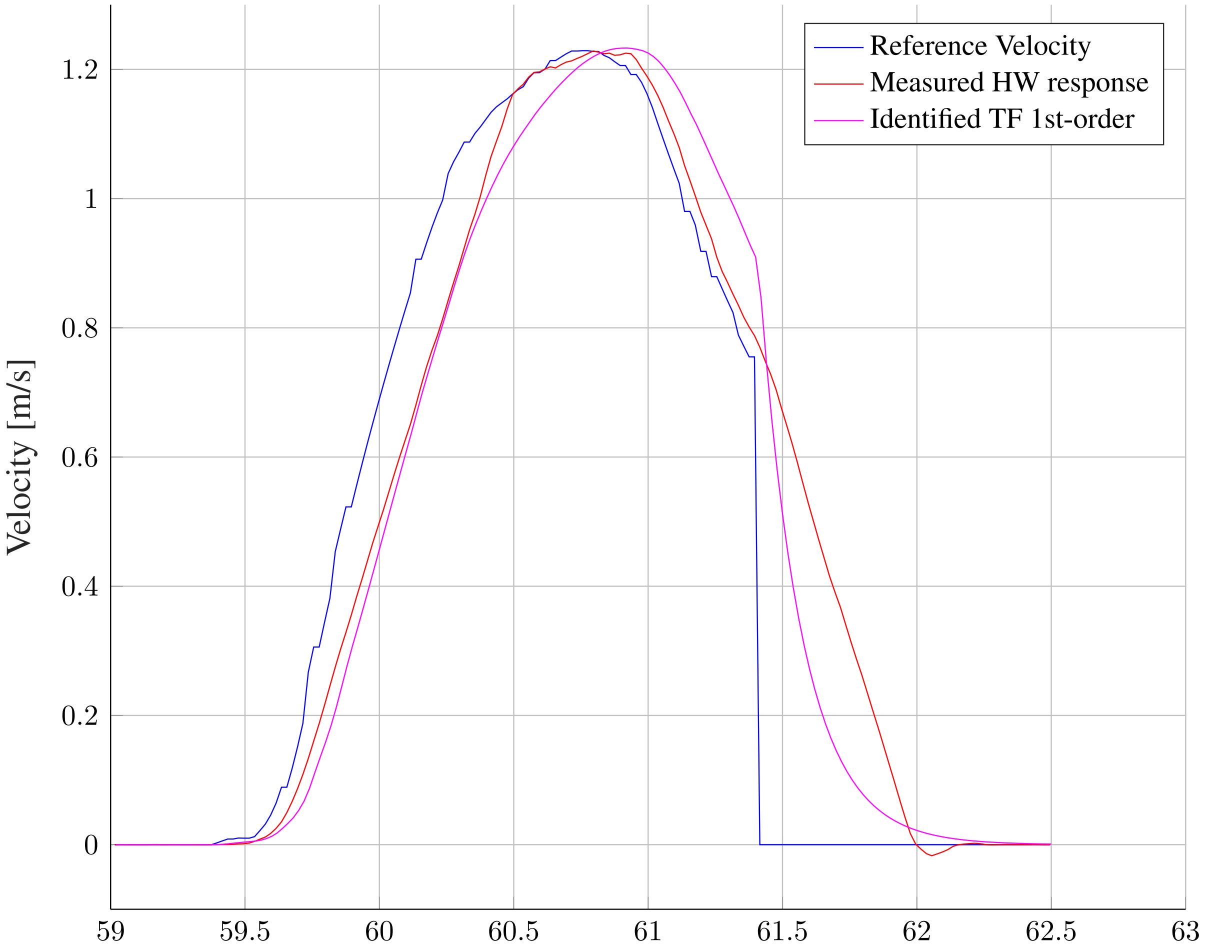


Figure 5.7.: RoboTrainer v2's hardware dynamic response to the RT controller velocity from figure 5.4. The first-order system's estimation is sufficient to approximate the hardware's response to interaction data. The blue line represents the reference velocity, the red line the measured device's velocity, and the magenta line the response of the identified transfer function.

trol actions that users have to feel and overcome. The force exchange between a user and the RoboTrainer also significantly influences the user's safety, where RoboTrainer's output velocity and force influence could endanger a user. An example of such a situation and a solution to it is given in the following section 5.3.

5.3. Stability of the User-Walker System in Active Training

This section investigates the stability of the user-Smart Walker system. First, the reasons for unstable and oscillatory behavior are described in section 5.3.1. Furthermore, section 5.3.2 discusses possible approaches for detecting and stabilizing the interaction in those situations.

5.3.1. Reasons for Unstable Behavior of the User-Walker System

When users interact with RoboTrainer for the first time, they tend to use powerful forces and stiff their arms and hands. This is especially observable at low velocities when, for example, the RoboTrainer needs to be stopped or positioned precisely. The users try to stop the RoboTrainer by applying force and stiffing their arms, which is a natural way to stop a passive object with wheels, i.e., a shopping cart. On the other hand, the active devices operate somewhat differently because their sensors are rigidly coupled to the handles. This leads to user's unintentionally generated input force, resulting in the device's movement according to the admittance equation (equation (2.12)). This behavior decreases the stability of the user-RoboTrainer system and may cause oscillations of the device. Those oscillations endanger a user and may lead to breakage of the device's parts.

As mentioned in the previous section 5.2, the user-walker system's oscillations also occur if the admittance controller has too fast dynamics, i.e., low mass (cf. equation (2.17)). So, if the device's inertia is too low, it cannot absorb the user's body oscillations caused by walking and the device starts to “bounce” in the user's hands.

For the situations detailed in the preceding two paragraphs, the relevant interaction parameter is the stiffness of the user-walker system. In the first case, the user increases the hands' stiffness so that the force due to the acceleration difference (see figure 5.9) can not be compensated by the admittance controller anymore. In the second case, the disturbance force is caused by the user's upper-body oscillations when walking.

In the physical human-robot interaction (pHRI) with a Smart Walker, a user can be considered the ultimate controller. Therefore, the most obvious solution for the user to keep the user-walker system stable [169, p.198] would be the adaption of its arms' stiffness. A user would need to increase their compliance as the robots do when interacting with a stiff environment [169, p.196]. Nevertheless, Smart Walkers' goal is to support elderly persons and possibly those with limited motor and cognitive capabilities. Therefore, individual adjustment of RoboTrainer's dynamic parameters is the only solution. The work of Chuy, Hirata, and Kosuge [27], and Yu, Spenko, and Dubowsky [185], as well as the practical experience with RoboTrainer Prototype show that the physical coupling and stiffness of the user-walker system is a relevant issue in this type of interaction.

Meer and Rock [118] investigated the parameters' influence on stability in the interaction between a passive environment and an impedance-controlled robot. Translated to the user-walker interaction, a walker is equivalent to the passive environment and a user is equivalent to the robot and controller. Chuy, Hirata, and Kosuge [27] followed the same logic investigating the case where users stiff their hands when interacting with a Smart Walker (SW) (see figure 2.13a). Their work concludes that the user-walker system is stable if

$$M_{\mathrm{d}} > \frac{hM_{\mathrm{a}}}{h+1} \tag{5.7}$$

M_{d} is the desired mass of the admittance equation, M_{a} actual mass of the device, and h is the coupling factor between force resulting from the device's actuators and disturbance superposed to the intended user's force. Section 2.3.4 provides more details about the approach from Chuy, Hirata, and Kosuge [27].

The analysis from [27] does not explain the nature of the coupling factor h. For example, a specific definition of the value range is missing, making the theoretical analysis unclear. Analyzing the influence of the coupling factor h during interaction with the RoboTrainer, it can be concluded that the disturbance transfer function $H(s) = h$ is caused merely by force resulting from acceleration discrepancy between the RoboTrainer and the user's hands:

$$\boldsymbol{F}_{\mathrm{dis}} = \boldsymbol{M}_{\mathrm{a}} * (\boldsymbol{a}_{\mathrm{out}} - \boldsymbol{a}_{\mathrm{h}}) \tag{5.8}$$

where $\boldsymbol{M}_{\mathrm{a}}$ is the actual mass of the walker, $\boldsymbol{a}_{\mathrm{out}}$ the walker's acceleration, and $\boldsymbol{a}_{\mathrm{h}}$ the acceleration of the user's hands[2]. In the worst case, a user does not move its hands so that the disturbance force has maximal value $\boldsymbol{F}_{\mathrm{dis}} = \boldsymbol{M}_{\mathrm{a}}\boldsymbol{a}_{\mathrm{out}}$, i.e., the full force from the walker's actuators superposes the user's intention force. This situation occurs when the user holds its arms and hands completely stiff, i.e., a user is not compliant and the coupling factor $\boldsymbol{h} = \overrightarrow{\mathbf{1}}_3$[3]. The opposite case is when the user ideally follows the movement of the RoboTrainer, so $\boldsymbol{a}_{\mathrm{out}} = \boldsymbol{a}_{\mathrm{h}} \rightarrow \boldsymbol{F}_{\mathrm{dis}} = \overrightarrow{\mathbf{0}}_3$. In other words, a user is fully compliant,

[2] Equation (5.8) uses the operator "$*$" for element-wise matrix calculation explained in section 1.4.

[3] $\overrightarrow{\mathbf{1}}_3$ represents a vector filled with ones and size three.

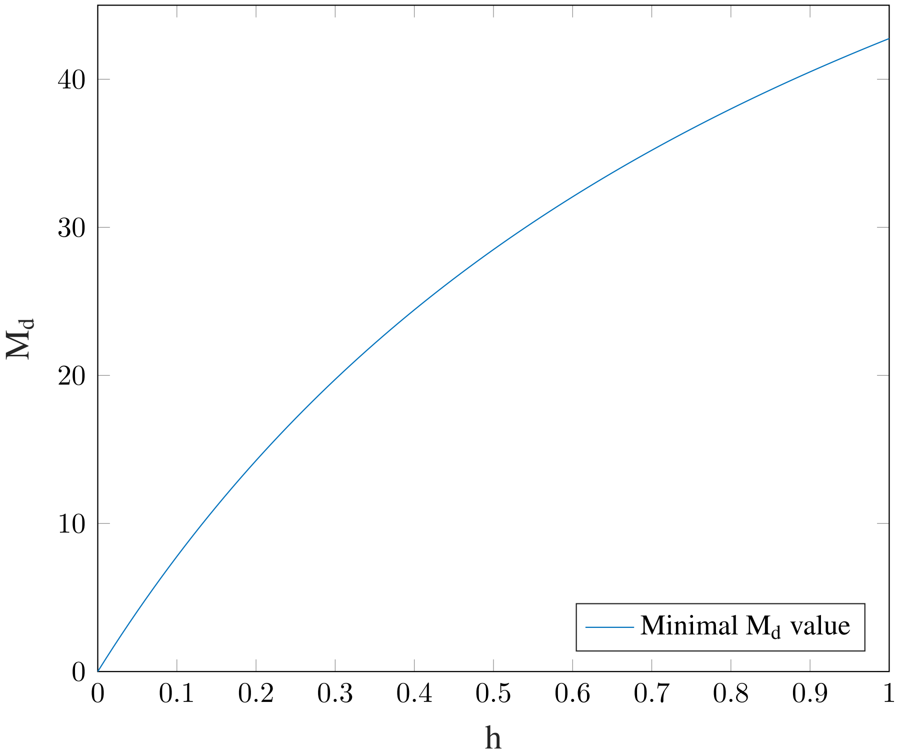


Figure 5.8.: Minimal values for the controller's desired mass M_d depending on the coupling factor h for the RoboTrainer ($M_\mathrm{a} = 84.5\,\mathrm{kg}$).

which is comparable to a walker's free movement depending only on the dynamics of the admittance equation. Therefore, the coupling factor $h \in [0, 1]$ is defined as the inverse of the user's compliance.

By applying the previous definition to equation (5.7), the controller's desired mass should be larger than $M_\mathrm{d}(h) \in (0, \frac{M_\mathrm{a}}{2})$ to have stable behavior in all situations. From the practical observations, Chuy, Hirata, and Kosuge [27] report that oscillations occur when $M_\mathrm{d} < \frac{M_\mathrm{a}}{3}$. This discrepancy is probably caused by the simplified model used in the theoretical analysis. Physical limits of the used sensors and actuators could reduce the whole system's sensitivity and positively impact the user-walker system's stability, as shown by Meer and Rock [118].

Figure 5.8 shows the relation between the coupling factor h and desired mass M_d from the admittance element (in equation (2.17) marked as M) to satisfy the criteria from equation (5.7). This relation is essential to consider when designing a controller for a Smart Walker since some degree of coupling is always present, i.e., h is always > 0.

The second issue, where the oscillations occur if the admittance controller's desired mass is small, can be partially explained with the previous observations, i.e., with the equation (5.7). The observation is based on the experiences of RoboTrainer Prototype's eval-

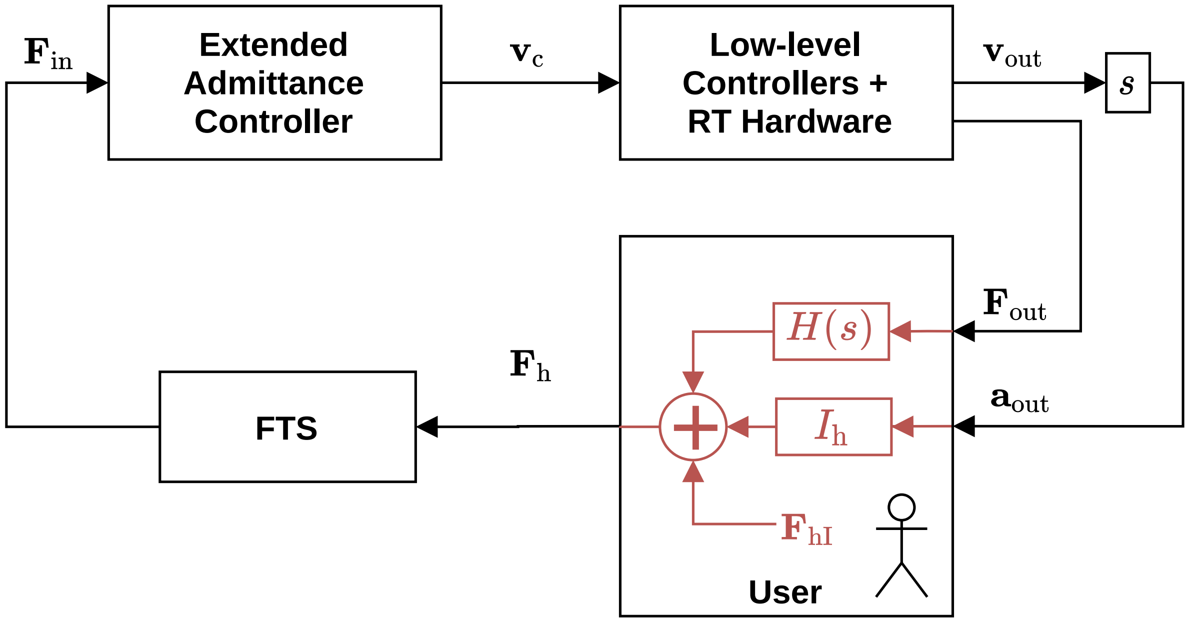

$\mathbf{F}_{\text{in}}$
Extended Admittance Controller
$\mathbf{v}_{\text{c}}$
Low-level Controllers + RT Hardware
$\mathbf{v}_{\text{out}}$
s
$\mathbf{F}_{\text{out}}$
$H(s)$
$\mathbf{F}_{\text{h}}$
$\mathbf{a}_{\text{out}}$
FTS
I_{h}
$\mathbf{F}_{\text{hI}}$
User

Figure 5.9.: The extended model of the user-walker system used for the stability analysis. Besides the coupling transfer function $H(s) = \boldsymbol{h}$, an inertia $\boldsymbol{I}_{\text{h}}$ that models the user's reaction-performance to the walker's acceleration is added. The inertia is influenced by the walker's acceleration $\boldsymbol{a}_{\text{out}}$, and it is superposed to the user's intended force $\boldsymbol{F}_{\text{hI}}$. The disturbance is equivalent to the actuator's internal inertia commonly used when modeling control systems.

uations with healthy young adults. This group of users is rather relaxed when using RoboTrainer. Therefore, the coupling factor h is expected to be small. Still, some users had issues with instabilities. Therefore, in this research, the control model shown in Figure 2.13a from [27] is extended by additional disturbance superposed on the user's intentional force caused by their reaction performance. Figure 5.9 shows this extended model, where, besides the coupling transfer function $\boldsymbol{H}(s) = \boldsymbol{h}$, an additional disturbance resulting from "users' inertia" $\boldsymbol{I}_{\text{h}}$, is added. The extended disturbance is modeled as inertia equivalent to the actuator's internal inertia commonly used when analyzing the robotic control systems, as shown by Meer and Rock [118]. It describes the user's reaction performance to the SW's movement and expresses their perceptional and motor skills. Compared to the coupling factor h, which is impossible to measure without external sensors, e.g., electromyographic measurements of muscle activity, the introduced inertia is measurable using the user's input force and the walker's acceleration. A concrete method for this estimation is out of the scope of this thesis and, therefore, part of the future work.

Considering this inertia, the user's input written for all six dimensions is

$$\boldsymbol{F}_{\text{h}}(t) = \boldsymbol{F}_{\text{hI}}(t) + \boldsymbol{I}_{\text{h}}\ddot{\boldsymbol{x}}_{\text{h}}(t) \tag{5.9}$$

where $\boldsymbol{F}_{\mathrm{h}}$ is the user's input measured by the force-torque sensor, $\boldsymbol{F}_{\mathrm{hI}}$ is the intended input force of the user, $\boldsymbol{I}_{\mathrm{h}}$ abstract inertia modeling the user's reaction performance, and $\ddot{\boldsymbol{x}}(t)$ is the acceleration or user's reaction to RoboTrainer's movement. Applying equation (5.9) to one-dimensional equation (2.2) and equation (2.3) results in equation (5.10) for the apparent dynamics and equation (5.11) for the actual SW's dynamics.

$$M_{\mathrm{d}}\ddot{x}(t) + D_{\mathrm{d}}\dot{x}(t) = F_{\mathrm{hI}}(t) + I_{\mathrm{h}}\ddot{x}_{\mathrm{h}}(t) \tag{5.10}$$

$$M_{\mathrm{a}}\ddot{x}(t) = F_{\mathrm{hI}}(t) + I_{\mathrm{h}}\ddot{x}_{\mathrm{h}}(t) + F_{\mathrm{acc}}(t). \tag{5.11}$$

During the interaction, acceleration $\ddot{\boldsymbol{x}}_{\mathrm{h}}$ should be equal to the acceleration of a SW $\ddot{x}$ since the user tries to achieve $\boldsymbol{F}_{\mathrm{h}}(t) \approx \boldsymbol{F}_{\mathrm{hI}}$. Using this assumption, the desired acceleration of a SW becomes

$$a_{\mathrm{d}}(s) = \frac{F_{\mathrm{hI}}(s)}{M_{\mathrm{d}} - I_{\mathrm{h}} + \frac{D_d}{s}} \tag{5.12}$$

and the actual acceleration is

$$a_{\mathrm{a}}(s) = \frac{F_{\mathrm{hI}}(s) + F_{\mathrm{acc}}(s)}{M_{\mathrm{a}} - I_{\mathrm{h}}} \tag{5.13}$$

Substituting $M'_{\mathrm{d}} = M_{\mathrm{d}} - I_{\mathrm{h}}$ and $M'_{\mathrm{a}} = M_{\mathrm{a}} - I_{\mathrm{h}}$ in equations (5.12) and (5.13) and doing the same analysis regarding disturbance force, as in [27] (details in section 2.3.4), results in modified user-walker system stability criteria:

$$M_{\mathrm{d}} > \frac{hM_{\mathrm{a}} + I_{\mathrm{h}}}{h+1} \tag{5.14}$$

This increases the necessary value of the admittance controller's inertia M_{d}, i.e., the curve in figure 5.8 shifts in the positive direction along the y-axis. Therefore, for the same admittance parameters, different users will experience the stability of the system differently.

The inertia caused by the user's performance is generally unknown. For the later analysis, it is estimated to be the tenth of the RoboTrainer's mass. The model with additional inertia of the user's performance is more sensitive to the coupling factor's negative influence on the user-walker system stability. Therefore, the user's fitness significantly influences the interaction's stability with a smart walker.

5.3.2. Stabilization of the User-Walker System

The admittance equation (equation (2.2)) used to generate desired dynamics for a Smart Walker is stable for any physically feasible set of parameters, i.e., as long as desired mass and damping have positive values. The previous section 5.3.1 shows that the user-walker system becomes unstable when positive feedback of force and acceleration caused by the walker's movement occurs on the controller's input. This feedback is caused by the coupling stiffness between the user and a SW and the inertia of the user's reaction to the device's acceleration. The simplest solution to this feedback is to increase the admittance equation's desired inertia $\boldsymbol{M}_{\mathrm{d}}$ to satisfy the inequality in equation (5.14). Nevertheless, this solution decreases the agility of a SW and increases the physical burden on users.

Chuy, Hirata, and Kosuge [27] propose adding proportional gain in front of the admittance controller to damp the input measured by the force-torque sensor. The gain $\boldsymbol{K}(s) = \boldsymbol{k}$ modifies the stability criteria from equation (5.7) to equation (5.15).

$$M_{\mathrm{d}} > \frac{khM_{\mathrm{a}}}{1+kh} \tag{5.15}$$

Chuy, Hirata, and Kosuge [27] report that the user-walker system is stable for $k = 0.5$, i.e., $M_{\mathrm{d}} = \frac{M_{\mathrm{a}}}{4}$. Equation (5.15) is also satisfied when observing the worse case where the coupling factor $h = 1$ and the stabilization gain $k = 0.5$.

Since the approach from [27] is straightforward, its use is investigated with the RoboTrainer. Extending the stability criteria from this thesis (equation (5.14)), it becomes:

$$M_{\mathrm{d}} > \frac{khM_{\mathrm{a}} + I_{\mathrm{h}}}{1+kh} \tag{5.16}$$

Deducting from equation (5.16), the stabilization gain k has to satisfy the criteria in equation (5.17). The exact values of the constants $h \in [0, 1]$ and I_{h} are generally unknown. For the analysis, the largest coupling factor, $h = 1$, is used and the inertia caused by the user's performance is estimated to be a tenth of the RoboTrainer's actual mass.

$$k < \frac{M_{\mathrm{d}} - I_{\mathrm{h}}}{h \cdot (M_{\mathrm{a}} - M_{\mathrm{d}})} \tag{5.17}$$

Detection of non-Passive Behavior

Based on equation (5.17), the user-walker system's stable state can be ensured in every situation, even if the user has completely stiff arms when interacting with the RoboTrainer (as in figure 5.10). Nevertheless, using those parameters, a challenging training is not possible because of the limited device's agility. A solution to this is a situational adaption of the stability gain k depending on the user-walker system's stability. Chuy, Hirata, and Kosuge [27] propose an energy observer to achieve this. The energy observer calculates the interaction energy between the user and a walker $E_{\text{system}} = \int_0^t \boldsymbol{F}_{\mathbf{h}} \cdot \dot{\boldsymbol{x}} dt$ (equation (2.7)) and evaluates the energy sign. This energy calculation is called "Chuy's energy" in the remainder of the chapter. If the energy is negative, the walkers' output

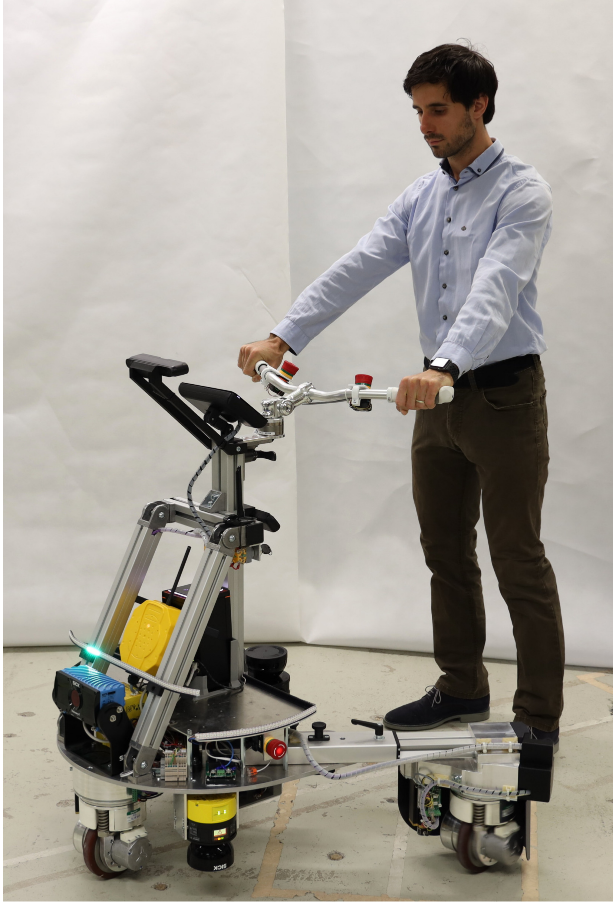


Figure 5.10.: A user tries to control RoboTrainer v2 with stiff arms. Photographer: Michael Mende, June 4, 2020

velocity and the user's input force have different directions, indicating the user-walker system's oscillations. More details are provided in section 2.3.4 (page 38).

Implementing the approach form [27] on the RoboTrainer, negative energy flow could not be detected. There are higher interaction forces with RoboTrainer and faster movement than the SW from [27], resulting in high energy values over time. In the experiment done with the RoboTrainer Prototype, Chuy's energy does not fall off when oscillations occur but increases slowly (see figure 5.11). The relationship between the user's input force and the walker's velocity shows a phase shift of $\frac{1}{4}$ of the period between those values. This means that for only $\frac{1}{4}$ of the time, the interaction power $P = \boldsymbol{F}_{\mathrm{h}} \cdot \dot{\boldsymbol{x}}$ is negative. Therefore, its magnitude has to be high to result with negative energy.

Based on these experimental observations, the energy calculation for the RoboTrainer is limited in time using a sliding-window approach with a predefined size. The implemented discrete integral for energy calculation is shown in equation (5.18)[4]. $\boldsymbol{E}_{\mathrm{n}}$ represents the energy for the last N control steps at step n, $\boldsymbol{F}_{h,i}$ is the user's input force, $\dot{\boldsymbol{X}}_i$ walkers velocity at the i-th step, and the Δt is the control period.

$$\boldsymbol{E}_{\mathrm{n}} = \sum_{i=n-N}^{n} \boldsymbol{F}_{h,i} * \dot{\boldsymbol{X}}_i \Delta t \tag{5.18}$$

Figure 5.11 shows the energy calculation results using the sliding-windows approach and compares it to Chuy's energy. The sliding-window reduces the calculated energy very fast on the first occurrence of negative power and then oscillates around zero as long as input force oscillations are present. A positive side effect of this behavior is also damping of RoboTrainer's response to any oscillations induced by the user, which protects the hardware of the RoboTrainer. Chuy's energy does not fall under zero during the use with RoboTrainer and therefore, the original approach form [27] is not suitable for use with RoboTrainer.

During the Bachelor's Thesis of Muth [124], further extension of the energy calculation (equation (5.18)) was investigated. An example is a weighted sum of the interaction power, where the most recent measurements have a higher weight. Nevertheless, no significant improvements were achieved. Therefore, it is decided to use the calculation from equation (5.18) because of its implementation simplicity and efficiency. This means that it is unnecessary to sum up all measurements inside the sliding window on each control step but only deduct the oldest and add the most recent one, which results in only two mathematical operations.

The stabilization gain and the energy monitor are implemented as part of the "Passivity Controller", as shown in figure 5.1.

Adaption of the Stabilization Gain

The last two sections discuss methods for detection of instabilities of the user-walker system and stabilization approach using a stabilization gain. The goal is to detect the negative interaction energy between a user and a Smart Walker and to reduce the stabilization gain from $k = 1$ to $k < 0.5$. Chuy, Hirata, and Kosuge [27] do not provide exact methods to reduce stabilization gain. Therefore, it is assumed that, when the negative interaction energy happens, the stabilization gain is set immediately to $k = 0.5$ to constrain the oscillations' time expansion. When the interaction energy is positive again, the stabilization gain is changed to $k = 1$ again.

[4] Equation (5.18) uses the operator "$*$" for element-wise matrix calculation explained in section 1.4.

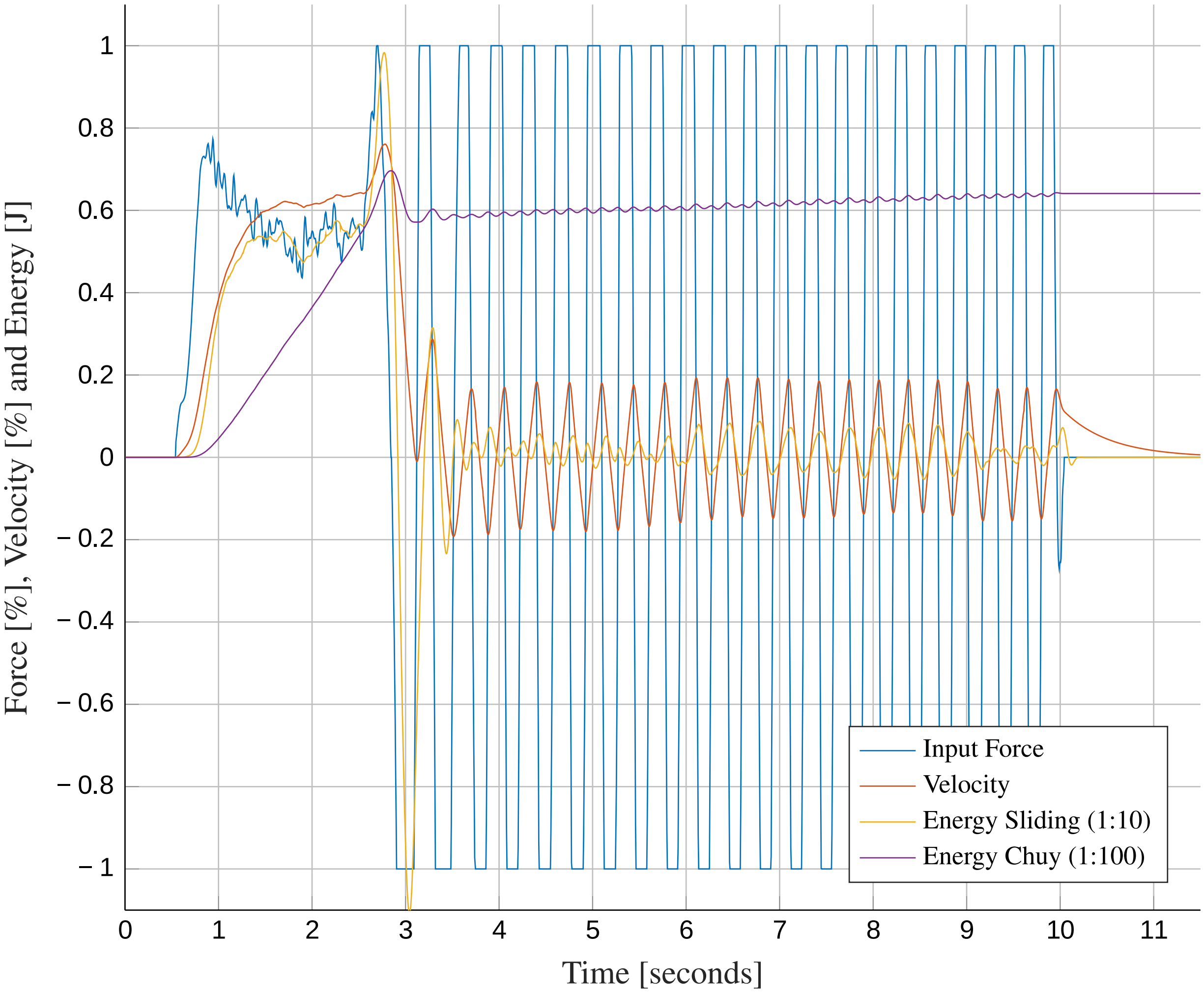


Figure 5.11.: Comparison of the energy calculation with approach from Chuy, Hirata, and Kosuge [27] ("Energy Chuy") and extension from this work ("Energy Sliding" when oscillations occur (at 3 s). The phase shift of approximately $\frac{1}{4}$ of the oscillation period between the input force and output velocity of the RoboTrainer produces a negative interaction power when the energy flow is from the RoboTrainer toward the user. Source: Stogl et al. [157].

The stabilization gain k, in case of oscillations between a user and RoboTrainer, is defined according to equation (5.17). The coupling constant is assumed to have maximal value, $h = 1$, and the user's reaction inertia is estimated to $I_{\mathrm{h}} = \frac{M_{\mathrm{a}}}{10}$. The factor I_{h} is chosen to have a minor influence. In the future, methods to estimate the user's performance inertia I_{d} will be investigated.

Disjointed, i.e., abrupt change of the stabilization gain from $k = 1$ to $k = \frac{M_{\mathrm{d}} - I_{\mathrm{h}}}{M_{\mathrm{a}} - M_{\mathrm{d}}}$ causes uncomfortable jerk on the RoboTrainer. Therefore, a linear change of the value over time of 0.5 s is implemented. Figure 5.12 shows the change of the stabilization factor during interaction with the RoboTrainer as the violet line[5]. For each degree of freedom, the stabilization is implemented separately.

Influence of the Stabilization Approach on Training Performance

Testing the RoboTrainer's stabilization approach in training scenarios, a dynamic decline was detected if sudden direction change occurred. This decline is caused by negative energy that occurs rapidly if direction is suddenly changed, e.g., from forward to backward or left to right. The reason for this is the sensitivity of the sliding-integral approach from equation (5.18).

To better understand this issue and oscillation damping behavior, data analysis for the following scenario is done:

(I) the user moves RoboTrainer approximately two meters forwards;

(II) user suddenly changes direction and moves back to the start;

(III) the user moves forward again and stiffs his arms;

(IV) the user waits until the oscillatory behavior is damped and then removes his hands from RoboTrainer's handles.

The first part (steps one and two) investigates the energy limiter's influence on sudden direction changes. In the second part (steps three and four), RoboTrainer's reaction to a non-passive behavior is evaluated. Figure 5.12 shows the results of this test.

The issue with performance decline at direction change is conquered by developing a method for detecting a user's intention to change RoboTrainer's movement direction. The method detects if there are only one or multiple direction changes in a specified time interval. During the first change, even the energy becomes negative, the stabilization gain is not adapted, but the controller observes the user's further reactions. If another direction change happens or the interaction energy (equation (5.18)) is still negative after a

[5] In the test, the stabilization gain is changed between $k = 1$ and $k = 0.02$.

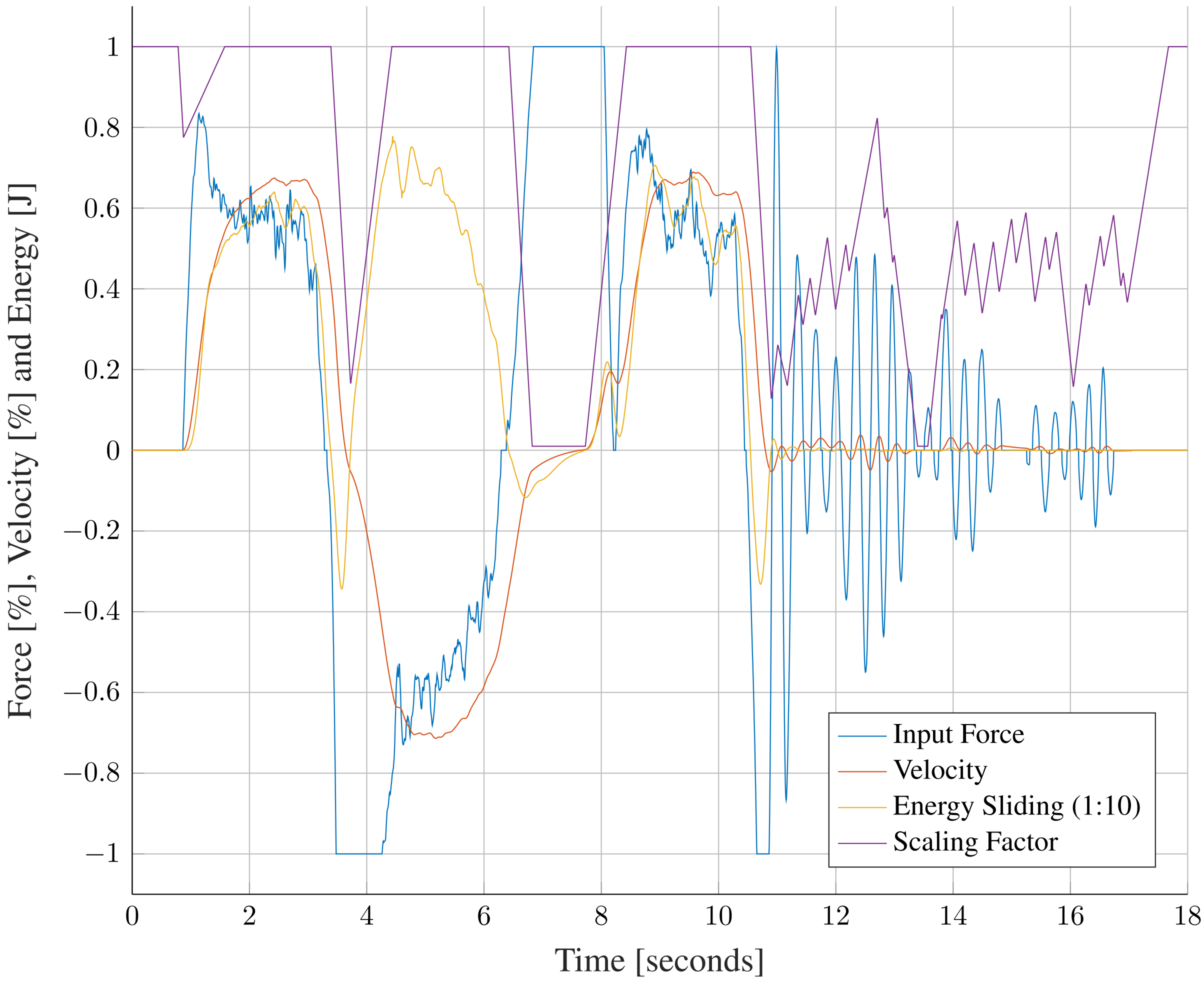


Figure 5.12.: Influence of the sliding integral on a sudden change of RoboTrainer's direction. The user starts moving forward (at 1 s) and suddenly changes RoboTrainer's direction (at 4 s), reducing scaling factor. In the test, the stabilization gain is changed between $k = 1$ and $k = 0.02$. At 6 s, the intended direction is changed again, which causes another reduction of the scaling factor. Since the user wants to change RoboTrainer's direction, they use strong force opposite to the device's movement. At that moment, the dynamics of the RoboTrainer is limited by reducing the stabilization gain k, and it is slowed down (the red line at 7 s changes very slow). At 10 s, user provokes the oscillatory behavior of RoboTrainer, which leads to the reduction of scaling factor and stabilization of RoboTrainer, i.e., output velocity gets very small.

predefined time interval, the stabilization gain is adapted. Assuming the second direction change is not detected, the scaling factor is reset to its maximal value $k = 1$. Figure 5.13 shows the same scenario when detecting the user's intention to change moving direction.

5.4. Velocity-Dependent Adaption

Passive walkers are constructed to have minimal friction between the chassis and the wheels. Therefore, if modeled by using the admittance equation (equation (2.1)), the damping constant would be negligible. Combined with very light construction, the conventional wheeled walkers are easy to manipulate, and users can utilize them for hours before they get tired. Conversely, a conventional walker's effortless movement also causes falls in situations where a walker "runs away" from its user [104].

The passive Smart Walkers, with embedded sensors and electronics to provide cognitive assistance and steering support, are more massive than conventional walkers and, therefore, a burden for a user during extended use (see chapter 2). Adding motors to a SW makes it possible to compensate the active walker's weight and provide a similar experience as with a conventional walker. Nevertheless, an active Smart Walker provides possibilities to overcome the shortcomings of the conventional walkers by, e.g., adjusting the damping and mass factor to avoid unintentional movement on slopes or run-away situations. A more significant damping factor of the admittance equation increases the user's overall effort to move a SW. This effort is determined merely by the damping constant of the admittance controller (equation (2.23) [185]). A larger damping constant is desirable at the walker's lower velocities to increase the walker's manipulability and reduce the oscillations. On the other hand, when the user moves at a constant speed, a large damping factor D reduces the user's force input F_{h} and, therefore, the amount of energy E_{h} towards a SW (equation (5.19)).

$$E_{\mathrm{h}} = \int_0^t \frac{F_{\mathrm{h}}^2}{D} dt \tag{5.19}$$

Therefore, Yu, Spenko, and Dubowsky [185] proposed a linear damping-adaption model (equation (2.8)) where damping is reduced from the maximal value when a walker is standing still to minimal value at maximal velocity.

Experimenting with the linear damping adaption presented in [185], the RoboTrainer was more comfortable to navigate at higher velocities. Nevertheless, the stopping way was prolonged if the user releases the device during the interaction, due to the low damping. This behavior is expected because of the controller's time-constant increase with damping reduction at higher velocities (reminder equation (2.17): $T_{\mathrm{admittance}} = \frac{M_{\mathrm{d}}}{D_{\mathrm{d}}}$). Looking at the analysis of SW's behavior in respect to the admittance controller parameters as

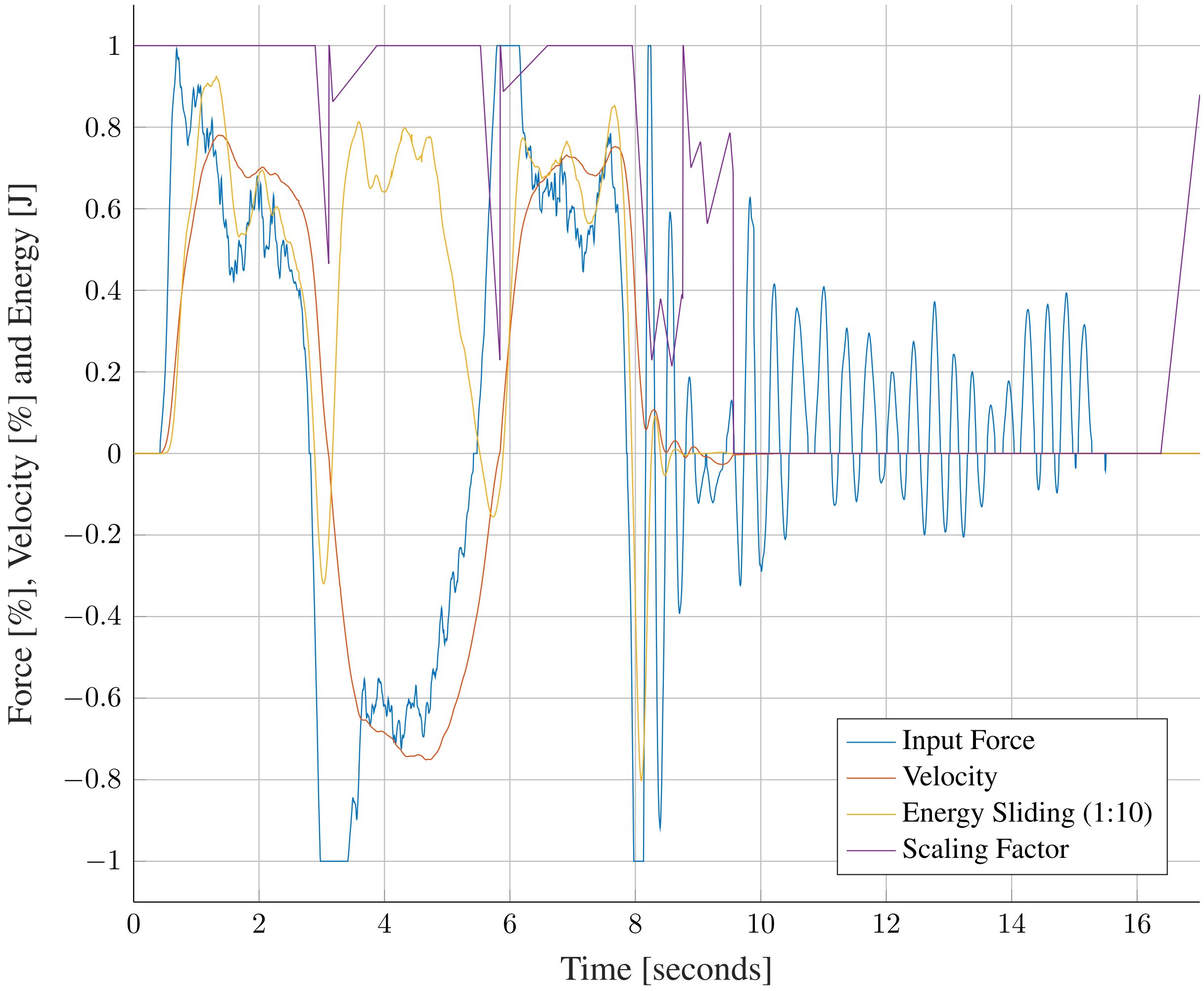


Figure 5.13.: The adapted method for scaling factor with the detection of the user's intention to keep the RoboTrainer's dynamics when the user suddenly changes direction. When a direction change happens (approx. 3 s and 6 s), the scaling factor is reduced quickly but corrected to its maximal value since direction change is detected. When oscillatory behavior happens (at 8 s), the algorithm reduces the scaling factor as intended. This method additionally reduces the scaling factor to zero (at 10 s) when oscillatory behavior exists longer than a second to protect RoboTrainer's hardware from mechanical stress. In the test, the stabilization gain is changed between $k = 1$ and $k = 0.02$.

analyzed by Yu, Spenko, and Dubowsky [185] in figure 2.14, it seems that changing both damping and mass in the same ratio would make SW “lighter” while keeping the dynamic profile the same. Other advantages of this approach are: (I) the desired adaption can be achieved by only manipulating the user’s input force; and (II) the influence of control actions (CAs) is changed accordingly, without additional synchronization of main controller’s and CAs’ internal models.

The approach in this thesis adapts the maximal force limit $\boldsymbol{F}_{\max}$ from equations (5.4) and (5.5)) depending on the RoboTrainer’s normalized velocity $\mathrm{v}_{\%}(t)$. Concretely, it adapts the user’s maximal force $\mathbf{F}_{U_{\max}}$ between two configurable extremes, $\mathrm{f}_0 \in [0.35, 1.0]$ for standstill and $\mathbf{f}_{v_{\max}} \in [1.0, 2.0]$ for maximal velocity. This adaption equation is:

$$\mathbf{F}^{*}_{\max}(t) = \mathbf{F}_{H_{\max}} * (\mathbf{f}_0 + (\mathbf{f}_{v_{\max}} - \mathbf{f}_0) * \mathbf{F}_{\%}(\mathbf{v}(t))) \tag{5.20}$$

where $\mathbf{F}_{\%}(\mathbf{v}(t))$ is an arbitrary scaling function[6]. This thesis proposes a non-linear function as scaling function (equation (5.21)). The parameters r and s represent the curvature and steepness, respectively.

$$\mathbf{F}_{\%}(\mathbf{v}(t)) = \frac{1.0 - (\mathbf{v}_{\%}(t) \cdot r + 1.0)^{-s}}{1.0 - (r + 1.0)^{-s}} \tag{5.21}$$

Figure 5.14 shows the non-linear function from equation (5.21) for parameters $r \in [0.1, 1.0]$ and $s = 9.0$. This scale function is chosen because it provides higher granularity, i.e., steeper adaption, at lower velocities. This provides better maneuverability in curves and around obstacles. Furthermore, at higher velocities, the function causes less adaption. For the remainder of this work, $s = 9$ and $r = 0.55$ are chosen experimentally for the final evaluation.

In general, the scaling parameters should be chosen using the following rules: (I) at lower velocities $f_0 \geq 1.0$ stabilizes the SW, i.e., the input force is scaled down; and (II) at higher velocities $f_{v_{\max}} \leq 1.0$ lowers the user’s effort to keep their pace effortlessly. The default parameters $f_0 = 1.7$ and $f_{v_{\max}} = 0.5$ are chosen to decrease the RoboTrainer’s sensitivity by seventy percent and decrease the needed force at the maximal velocity by a half.

During the implementation, some standard functions are investigated as the scaling function, e.g., hyperbolic tangent (tanh) function. Nevertheless, they were missing possibilities for fine-tuning.

Figures 5.15 to 5.17 show the RoboTrainer’s velocities when using controllers with static parameters, damping adaption from Yu, Spenko, and Dubowsky [185], and the non-linear adaption from this thesis. The presented data were gathered in an experiment, where the user was walking straight ahead for approximately 5 s. The figures show the user’s input

[6] Equation (5.20) uses the operator “$*$” for element-wise matrix calculation explained in section 1.4.

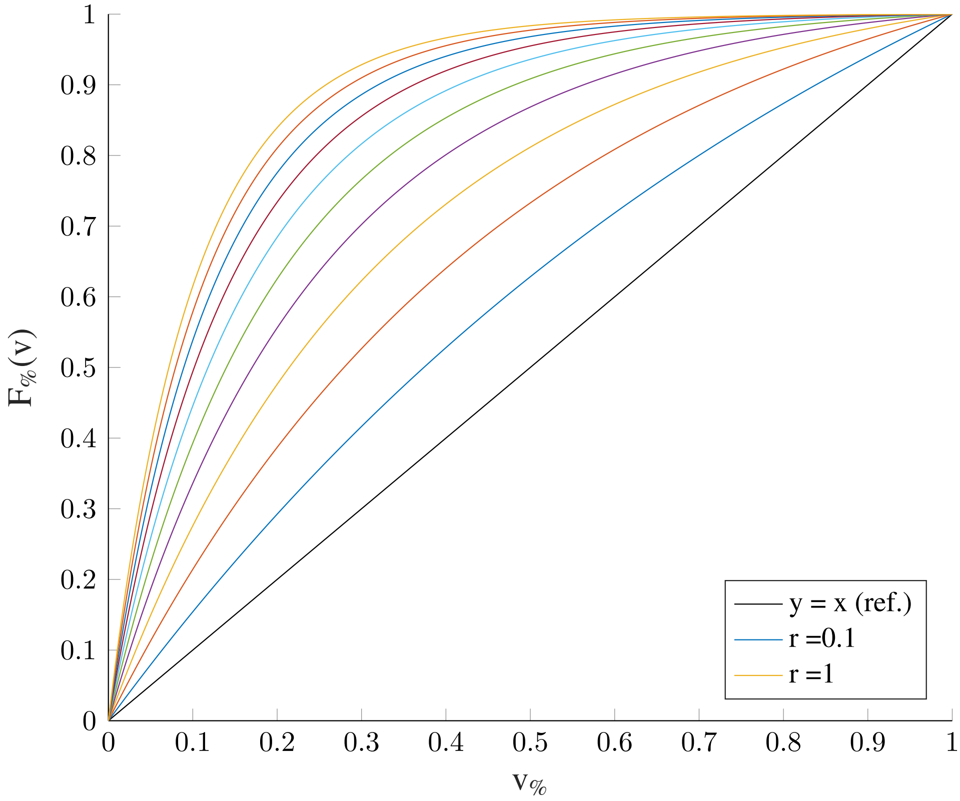


Figure 5.14.: Non-linear scaling of user's input force in respect to velocity (5.21), with $r \in [0.1, 1.0]$ (step 0.1), $s = 9.0$.

force $\boldsymbol{F}_{\mathrm{h}}$, normalized and adapted input force, and RoboTrainer's velocity. The normalized force is not adapted in the first two controllers and corresponds to scaled raw input force. The average interaction force $\overline{F}_{\mathrm{h}}$ and RoboTrainer's velocity $\overline{V}_{\mathrm{RT}}$ are calculated and shown above the figure for all three controllers. The controller with static parameters has the highest average force and lowest velocity (figure 5.15). Therefore, to move the RoboTrainer with those fixed parameters, maximal effort is needed. The controller with the damping adaption provides a higher average velocity for a lower average force (figure 5.16). The damping is adapted to $80\,\%$ of its initial value at maximal velocity. The responses in figures 5.15 and 5.16 are very similar because the controllers' dynamics do not change significantly. In the case of non-linear adaption, the average force is the lowest with the highest average velocity. The default parameters were used with the values mentioned above. Figure 5.17 also shows the drawback of this approach very well, that is, a SW's sensitivity on high-frequency oscillations. This is caused by the "amplification" of the input force when adapted using equation (5.21).

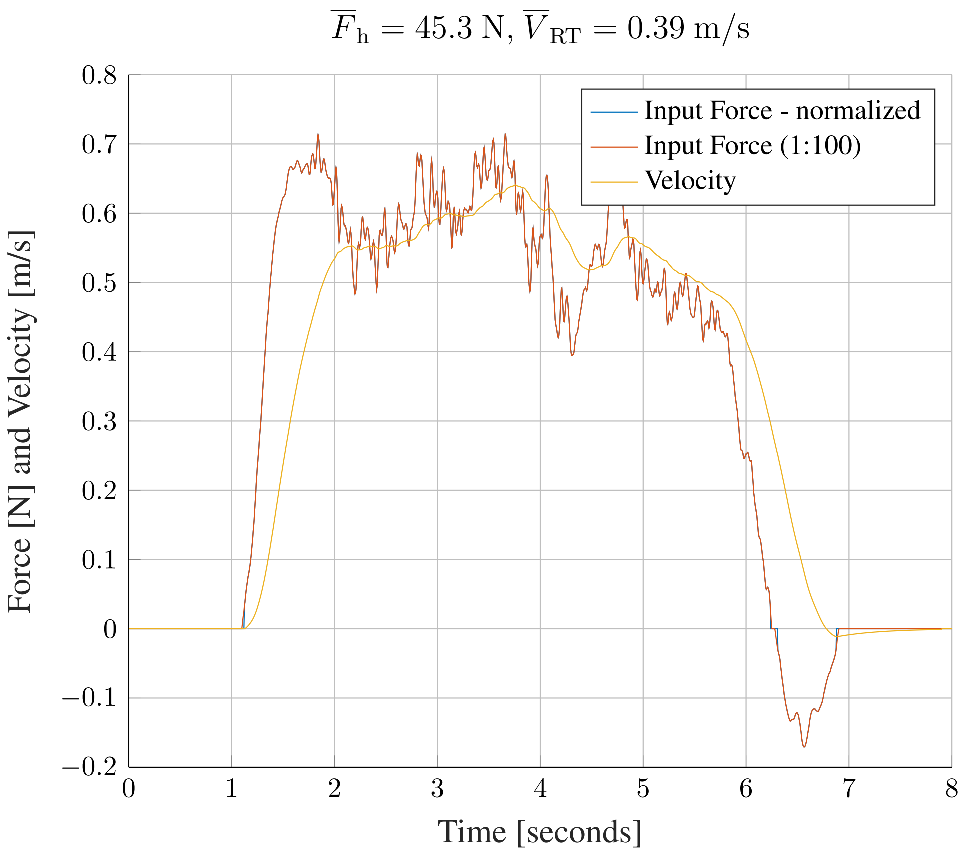


Figure 5.15.: RoboTrainer's velocity response using the controller with static parameters.

5.5. Per-User Specific Control

During the evaluation of the RoboTrainer Prototype, different users had significant differences in the resulting interaction with the RoboTrainer Prototype (cf. section 3.3). The main reasons, probably, are the variance in participants' strength and fitness. Therefore, this thesis investigates possibilities for the user-specific control parameters of the RoboTrainer v2. The strategies are implemented as part of the Master's Thesis of Zumkeller [189].

Investigated literature in chapter 2 provides only device-specific parameterization and calibration methods for admittance-controlled Smart Walkers. This section discusses the two most appropriate methods. Martins et al. [110] present a calibration strategy for a novel handle interface in which small movements, i.e., shifts, determine the Smart Walker's velocity. Their work also proposes a calibration, i.e., parameterization sequence, to determine appropriate user input gains. Nevertheless, their approach is a walker- and input-device-specific one, with the primary purpose to calibrate those de-

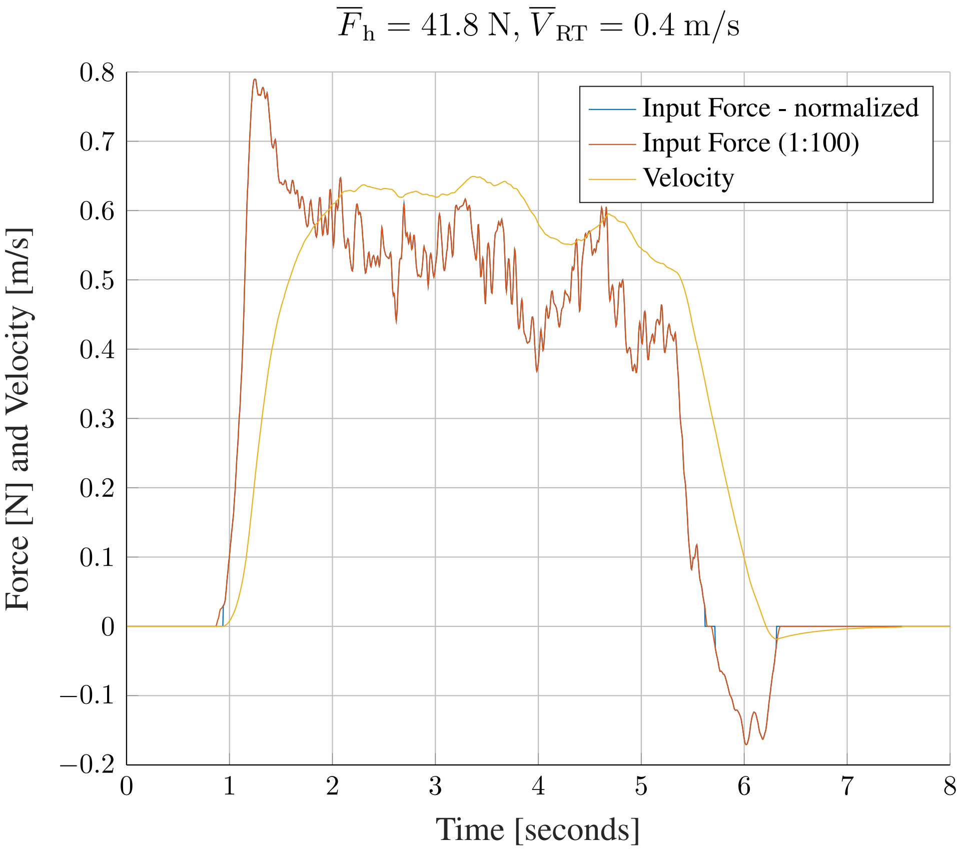


Figure 5.16.: RoboTrainer's velocity response using the controller with velocity-based damping adaption by Yu, Spenko, and Dubowsky [185]. The damping was maximally reduced by 20 %.

vices to each other and not to determine individual user's parameters. The calibration from Martins et al. [110] is done offline, and their SW can only rotate and move forward, whereas RoboTrainer has an omnidirectional base. The second approach from literature is presented by Chuy, Hirata, and Kosuge [30], where the curve-performance of a SW is individually adapted to a user. The authors propose a shift of SW's center of rotation (CoR) in order to compensate user's asymmetry in forces of the left and the right arm. The used SW has forearm support where the user's weight and balance on its left and right leg are correlated to the user's input more strongly than this is the case for RoboTrainer. Chuy, Hirata, and Kosuge [30] use a specific reference path in the SW's environment to measure a force disbalance in the user's input to follow the predefined curvature as precisely as possible.

Based on the resources from the literature and the experience from the evaluation study with RoboTrainer Prototype, the following requirements for the parameterization process are defined:

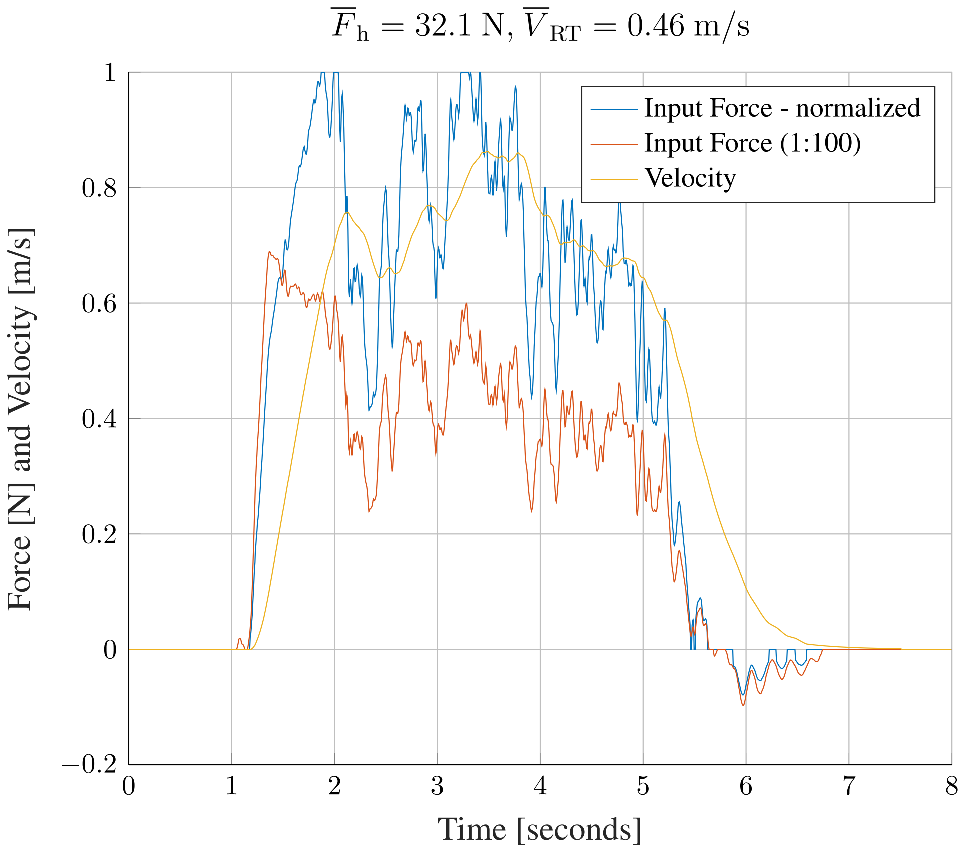


Figure 5.17.: RoboTrainer's velocity response using the controller with non-linear adaption from equation (5.21). Default parameters were used.

RP1 The process(es) has to be simple enough for users without any experience with SWs;

RP2 The process should consider all degrees of freedom (DOF) of a SW;

RP3 The differences, if any, between left and right linear movement and left and right rotation have to be considered;

RP4 If possible, both parameters of admittance control, i.e., mass and damping, have to be included in the parameterization process;

RP5 The process has to calculate the user's control parameters online and apply them automatically;

RP6 The parameterization process must be independent of a specific environment and executable without any SW's environment preparation.

The most relevant parameters for the interaction with a RoboTrainer-like SW are the admittance equation's mass and damping parameters (equation (2.1)). As discussed in section 5.4, steady-state velocity is defined by the input force and the damping factor (see equation (2.23)). Therefore, the steady-state behavior can be adjusted by determining the user's interaction forces and adapting them to achieve the desired behavior. If the user's input force is modified and not directly the damping parameters, the controller's mass is also adjusted indirectly (cf. section 5.1.1). Therefore, adjusting the output-velocity limits determines the gain and the time constant of the admittance equation (equations (5.2a), (5.2b), (5.4) and (5.5)). Therefore, it is sufficient to determine the user's interaction forces in the parameterization process and adjust the RoboTrainer's velocity limits to achieve the desired dynamics. In this work, the user's maximal force is determined and used as the scaling factor for user's input and controller's velocity output.

Another factor of acceptance for a SW is velocity-dependent load reduction experienced by a user when walking (see section 5.4). The second parameterization method provides a process to determine the upper and lower limits for the proposed non-linear adaption suggested in section 5.4. The method uses an optimization of the user's shin distance from the SW.

The evaluation of both parameterization strategies can be found in section 7.1.

5.5.1. Parameterization Strategy for Measuring of Maximal User's Interaction Forces

This section presents the method for measuring the user's maximal interaction force with a SW. The primary purpose is to achieve similar SW's behavior for users with different physical strengths, i.e., different interaction forces. The method identifies the user's natural interaction force, here called the base force, and adapts the system's maximal velocity so that the dynamic parameters remain the same for all users. The admittance equation's mass and damping are adapted according to equations (5.4) and (5.5).

The parameterization strategy uses a concept of virtual spring against which the user pushes the Smart Walker. The user provides a force or a torque in a specific direction to the SW's sensor. The device reacts with a movement according to the default admittance controller with fixed parameters. The controller's parameters are arbitrary but usually with a slower dynamic, i.e., larger mass parameters, to ensure slower movements and increase the feeling of safety. According to distance from the start position d and constant k, the opposing force F is calculated using the spring equation (5.22) [2, p 508].

$$F_{\text{spring}} = k \cdot d \tag{5.22}$$

The resulting spring force is subtracted from the user's input force while users are encouraged to push the robot as strong as they feel comfortable. The user pushes the robot until their input force and the opposing spring force equalize, and a SW stops moving.

The user has to keep this position for a few seconds, during which the average input force is calculated and stored as its maximal interaction force. After a user releases the SW's handles and has sufficient distance, the device moves autonomously to its start position. The process is repeated for each degree of freedom, for positive and negative directions separately. The only exception is the backward direction, where users would have to pull the robot toward themselves as strong as possible. This situation increases the safety risk for the user and it is, therefore, avoided. During the whole process, the device's movement is limited to one axis. During the parameterization procedure, the RoboTrainer shows its state using the LED stripe (section 4.3). Each parameterization phase is coded with different colors and directions in which LEDs are lightning.

Afterwards, the measured maximal input forces are used as new upper limits for the input force discussed in section 5.1.1. The approach's primary effect is that users with different interaction forces, i.e., physical strength, can achieve their natural walking speed without additional effort.

The proposed parameterization process merely requires the user's force measurements and the SW's odometry without adapting the SW's environment. Therefore, this method can be implemented on any Smart Walkers using a force-torque sensor (FTS) as input device.

5.5.2. Parameterization Strategy for Measuring Velocity-Dependant-Adaption Parameters

The personalized admittance equation using the maximal interaction force, still needs a significant effort to push a SW with constant velocity. Therefore, the RoboTrainer uses the non-linear adaption method presented in section 5.4. The current section describes the method for identification of individual scaling parameters f_0 and $f_{v_{\mathrm{max}}}$.

During this parameterization, a SW can move only forward and backward, i.e., along the longitudinal axis from the user's perspective. A user walks a distance of more than 5 m forward and backward twice. Both times, the average distance between the user's shin and the SW is measured. The distance is calculated using laser scanner data and the algorithm implemented during the Bachelor's Thesis of Azanov [11]. The first run uses the admittance controller with the individual user's force limits and fixed parameters. This distance is considered a "natural" distance for the user since users tend to adapt their distance and their velocity towards SW's distance and velocity. This distance is the baseline for the second repetition. The boundaries f_0 and $f_{v_{\mathrm{max}}}$ are adjusted during the second repetition, so user moves faster but still keeps its distance close to the baseline from the first repetition. Specifically, the user's relative input force $\mathbf{F}_{\%} = \frac{F_{\mathrm{u}}}{F_{\mathrm{max}}}$ and the quadratic difference between current $dist(t)$ and baseline distance $dist_{\mathrm{base}}$:

$$\delta_{\mathrm{d}}(t) = \|dist(t) - dist_{\mathrm{base}}\|_2 \qquad (5.23)$$

are used to tune parameters using the following rules:

$$f_{\mathrm{m}} = \begin{cases} f_{\mathrm{m}} + \delta_{\mathrm{d}}(t) \cdot (1 - f_{\mathrm{m}}) & \text{if } \mathbf{F}_{\%} < 1 \text{ and } \delta_{\mathrm{d}}(t) < 0 \\ f_{\mathrm{m}} - \delta_{\mathrm{d}}(t) \cdot (1 - f_{\mathrm{m}}) & \text{if } \mathbf{F}_{\%} < 1 \text{ and } \delta_{\mathrm{d}}(t) > 0 \end{cases} \tag{5.24}$$

$$f_{0} = \begin{cases} f_{0} + \delta_{\mathrm{d}}(t) \cdot (1 - f_{0}) & \text{if } \mathbf{F}_{\%} > 1 \text{ and } \delta_{\mathrm{d}}(t) > 0 \\ f_{0} - \delta_{\mathrm{d}}(t) \cdot (1 - f_{0}) & \text{if } \mathbf{F}_{\%} > 1 \text{ and } \delta_{\mathrm{d}}(t) < 0 \end{cases} \tag{5.25}$$

These tuning rules provide more restrictive behavior for a larger distance and more agile behavior for a smaller distance.

5.6. Implementation of the Control Approach using ros_control Framework

RoboTrainer v2's software builds upon the Robot Operating System (ROS) framework [133]. As de-facto standard software for robotic research, ROS enables easy use of the state-of-the-art algorithms for different modules needed to run a robot. In 2017 Chitta et al. [24] presented the ros_control as part of ROS designed for control tasks. The main difference to the rest of the framework is a monolithic node that uses shared-memory access to data between its sub-components. Each sub-component, except the representation of the robot's hardware, is a plugin that can be dynamically loaded and unloaded during the system's run-time. This plugin infrastructure enables the exchange of controllers without stopping the connection to the robot's hardware. On the other hand, the monolithic design with shared-memory access enables the control loop's real-time performance, keeping a possibility to integrate the robot's control with the high-level algorithms available in ROS, e.g., localization and navigation.

Leveraging ROS in the booth versions of the RoboTrainer, RoboTrainer Prototype and RoboTrainer v2, enabled the use of the same software stack. Therefore, parallel to the mechanical design of RoboTrainer v2 (see chapter 4), also software development has started. Figure 5.18 gives an overview of RoboTrainer v2's controller's implementation, highlighting internal classes. As the first step toward the final architecture, the node-based control of the RoboTrainer Prototype is re-implemented using ros_control framework resulting in the base controller (on the right in figure 5.18). The spatial control action (SCA) Virtual Force is initially integrated into the controller. During the Bachelor's thesis of Wern [178], the SCAs are realized as dynamically loadable filters for the base controller. This approach is currently used, and it is detailed in section 6.3. The base controller is implemented as a child class of the wheel controller, which provides cartesian interfaces, integrates direct and inverse kinematics and the impedance controller for

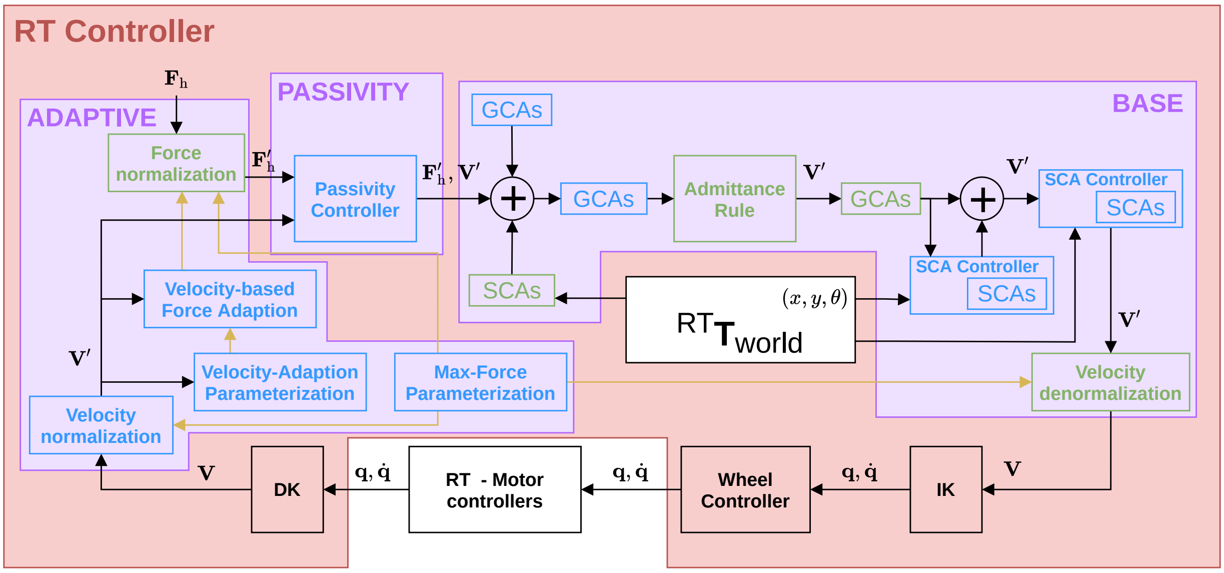


Figure 5.18.: RoboTrainer v2 control architecture – internal classes. The RoboTrainer's controller as a monolithic control block is bounded by the red line, including all compile-time dependencies. The controllers' internal classes, developed for this thesis, are highlighted with violet rectangles.

orienting the wheels. For brevity, details on the functionality of the wheel controller and its interplay with the geometry module are not given here but can be found in the project's open-access repository[7].

The main challenge was the extension of the base controller with passive behavior, i.e., stabilization controller and adaptive controller providing parameterization and velocity-based adaption strategies. The classical approach is cascade control [99, p. 646, Fig.26.24], as proposed in section 5.1. Nevertheless, the ros_control framework does not support the chaining of controllers at the moment. Therefore, the passivity and adaptive controller are implemented as further specializations of the base controller's class, using the internal flags to synchronize the communication to the hardware.

Figure 5.19 provides information about ROS packages used to realize the control of the RoboTrainer v2's hardware. The controller's output is sent to the *ELMO* motor controller using the implementation of *CANopen* profile 402[8] for ROS in `ros_canopen` metapackage[9]. The *ATI Mini 45* force-torque sensor is also connected to the RoboTrainer's PC using CAN interfaces. Nevertheless, the FTS uses a proprietary protocol imple-

[7] `https://github.com/ipa320/cob_control - cob_omni_drive_controller` package

[8] CAN in Automation – CANopen standard: `https://www.can-cia.org/canopen`

[9] `https://github.com/ros-industrial/ros_canopen`

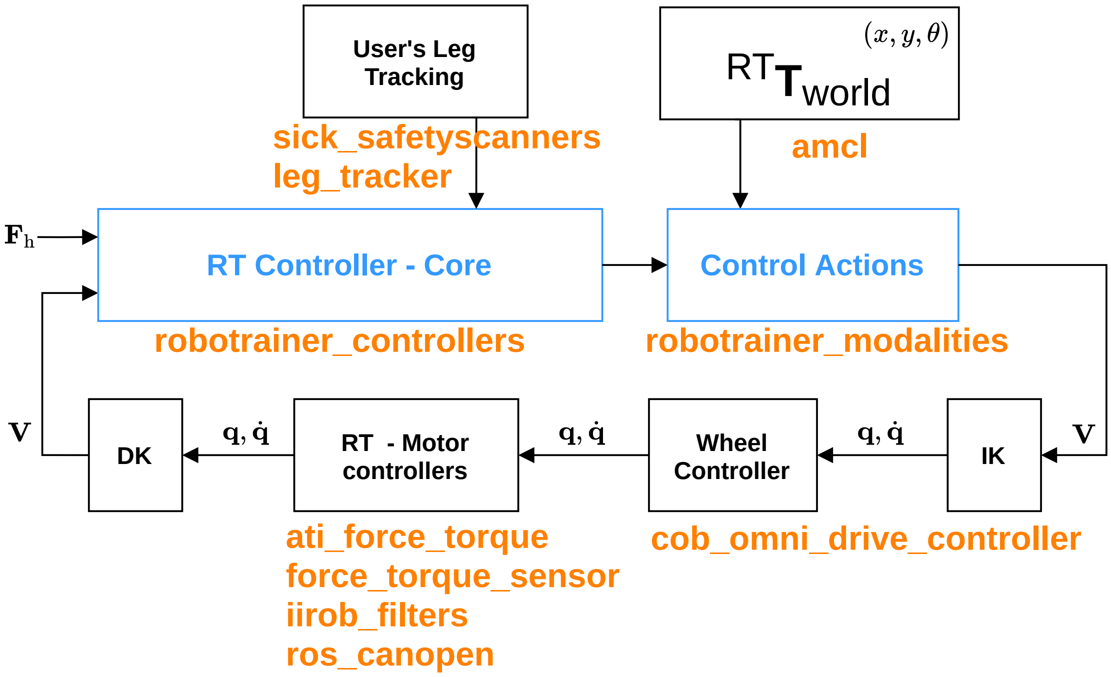


Figure 5.19.: RoboTrainer v2 control architecture – used ROS packages. The RoboTrainer's controller is shown in simplified form, splitting extended admittance controller and control actions. Under each block, in orange, ROS-package names that implement the functionality are listed. Details about listed ROS packages are given in the above text.

mented in the ati_force_torque ROS package[10]. For the integration of the FTS in ros_control, a sensor abstraction layer from package force_torque_sensor[11] is used. The force_torque_sensor uses the iirob_filters[12] package for filtering capabilities. The public packages for connecting an FTS to ROS result from this thesis.

The user's distance from the RoboTrainer used for the parameterization is determined from the laser-scanner data using the leg_tracker ROS package implemented during the Bachelor's Thesis of Azanov [11]. The package is hardware agnostic and works with both versions of the RoboTrainer using two different types of laser scanners. The RoboTrainer v2 uses sick_safetyscanners package[13] to receive the data from

[10] https://github.com/KITrobotics/ati_force_torque

[11] https://github.com/KITrobotics/force_torque_sensor

[12] https://github.com/KITrobotics/iirob_filters

[13] https://github.com/SICKAG/sick_safetyscanners

SICK microScan Pro 3 laser scanners. `acml` package from the ROS's navigation stack[14] is used for localization with data from laser scanners.

5.7. Conclusion on the RoboTrainer's Control

This chapter presents and discusses the controller design for RoboTrainer v2. After presenting the overall control architecture and implementation details specific to this work, a detailed description of user-RoboTrainer interaction is given. The section analyses the dynamics of RoboTrainer v2 and formulates the fundamental user-walker interaction loop. The chapter continues with the interaction's stability analysis providing reasons for the unstable user-walker system's behavior and extending a state-of-the-art approach to RoboTrainer. The second part of the chapter focuses on individual adaption strategies of RoboTrainer's control to enable pleasant interaction for users with diverse interaction approaches and physical conditions. The chapter finalizes the concrete implementation of the control approach using the Robot Operating System (ROS) framework.

The work presented in this chapter provides concrete guidelines to achieve stable and passive behavior of an active walker towards its users and, at the same time, individual adaptability. Therefore, the RoboTrainer's controller provides extensive configuration possibilities that could provide more individualized and user-centric interaction. After finalizing the RoboTrainer v2, interacting with it directly and observing others using it, some ideas worth investigating were gathered. Some ideas regarding control are:

- The introduced user performance's dynamics, which is still unknown in the scope of this work, could be estimated by tracking user's movement using motion capture systems;
- The approach to ensure user-walker system's stability could only change the desired mass of the admittance controller instead of damping the user's input;
- After the maximal user force parameterization, the maximal velocity could be adapted to achieve individual RoboTrainer dynamics;
- Other velocity-based parameter adaptions are imaginable, e.g., non-linear damping adaption.

To conclude, what is presented here, is: RoboTrainer v2's controller provides many possibilities to achieve individual interaction with a Smart Walker, especially in cases where interaction with higher forces is expected and the agile performance of the device is needed. Some benefits of the strategies presented here are evaluated in the studies with healthy participants, presented in chapter 7.

[14] `https://github.com/ros-planning/navigation`

6 Control Actions as Modifiers of Smart Walkers' Behavior

Smart Walkers (SWs) aim to provide additional functionalities compared to conventional wheeled walkers, especially sensorial and cognitive assistance. Typical use-cases of those functionalities are obstacle detection and avoidance [174, 185, 29, 136]. Some authors also propose using SWs for rehabilitation purposes [159, 114], but without details on how to realize them. Martins et al. [112] show that it is possible to realize a three-week-long Ataxia patient therapy using a SW. In chapter 3, this thesis presents a concept for motor activation of elderly persons with mild cognitive impairment (MCI) using a SW prototype. Motivated by the results and the acceptance of mentioned devices, as well as the work of Schwenk et al. [141] which utilizes sensory feedback to realize novel motor training for persons with MCI, this section investigates technical concepts to achieve an engaging and challenging motor training with a SW.

The training concept uses global and spatially-limited behavior modifiers as the building blocks, called *control actions (CAs)*, or more specifically, *global control actions (GCAs)* and *spatial control actions (SCAs)*. During training, the primary user task is to guide, i.e., navigate, the RoboTrainer along predefined paths marked on the floor along which control actions (CAs) are placed. The CAs are using a virtual force field (VFF) approach to influence the SW's behavior. This is usually done by modifying the user's input before the SW's dynamic controller calculates the desired velocity [174, 185]. Unlike those methods, this section shows an approach where the main controller's dynamics and the dynamics of CAs are separated. Moreover, it presents methods to assure user's safety and SW's passive behavior, even under the influence of control actions. Compared to the work of Chuy, Hirata, and Kosuge [29], the approach here makes a situational distinction regarding the SW's position, velocity, and configuration of control actions. Furthermore, data structures for storing training configurations and an easy-to-use graphical user interface (GUI) for creating training scenarios with the spatial control actions are presented.

The chapter is structured as follows. The first section 6.1 presents the concept, and all implemented control actions (CAs). Some issues and their solutions concerning all SCAs are shown and discussed on the *SCA–Force Area* example. Section 6.2 discusses the safety and passivity functionalities implemented on top of the CAs. Evaluation results for those functionalities are shown in the example of the *SCA–Force Area*. Finally, the graphical user interface based on the *ROS*'s 3D visualization tool, *rviz*, is shortly presented, showing the implementation and data structures for configuring the control actions.

The methods from this chapter are published as conference papers Stogl et al. [155] and Stogl et al. [157] and partially implemented during the Bachelor's Thesis of Groten [59], Wern [178], and Muth [124].

6.1. Concept of Control Actions

The basis for the training with RoboTrainer is a path marked on the floor of the training environment. This is shown in the figures from the evaluation in chapter 7, e.g., figure 7.5). The users are supposed to navigate RoboTrainer along the marked paths with as slight deviation as possible. The deviation is measured based on the RoboTrainer's localization system and the virtual representation of the paths are recorded before the training (an example environment is given in figure 7.6). To adjust the training's complexity, two types of control actions (CAs) are realized: *global control actions (GCAs)* – affecting the RoboTrainer's behavior in the whole training environment and *spatial control actions (SCAs)* – affecting the RoboTrainer in a bounded area.

Summarized, eight control actions with the following functionality are created:

GCA–Counterforce specifies the static force a person needs to move RoboTrainer.

GCA–Center of Rotation changes the RoboTrainer's center of rotation (CoR) from its origin to the specified point on the ground plane.

GCA–Inverted Controls inverts the RoboTrainer's left-right translation or left-right rotation.

SCA–Counterforce is the same as *GCA–Counterforce* but bounded inside a specific area.

SCA–Area enables an arbitrary RoboTrainer's behavior-change inside its influence area.

SCA–Force Area provides a spatially limited disturbance force in an arbitrary direction. This SCA further develops the concept of the *artificial forces* used with the RoboTrainer Prototype (see chapter 3).

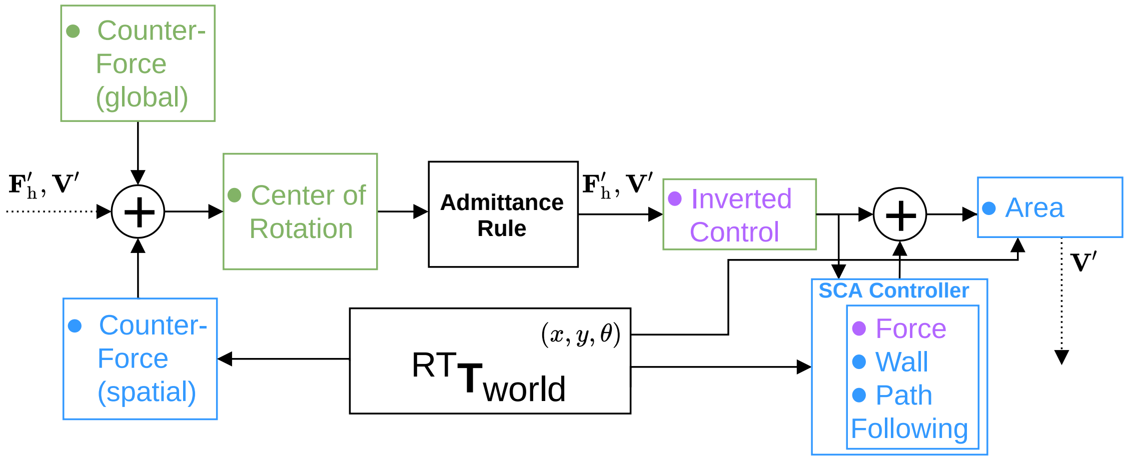


Figure 6.1.: Overview and placement of the control actions (CAs) in the RoboTrainer's controller. The black blocks represent the main admittance rule defining RoboTrainer's dynamics and the localization module providing data for spatial control actions (SCAs). The SCAs are marked in blue and global control actions (GCAs) in green. The CAs implemented for the RoboTrainer Prototype and integrated into the new library have their names highlighted in violet. Some CAs alter the controller's internal states directly, e.g., *GCA–Inverted Controls*, and some of them are superposed to those, e.g., CA–Counterforce. Exact details on how individual control actions alter the controller's internal states are given in sections 6.1.1 and 6.1.2.

SCA–Path Following helps users to follow the predefined path.

SCA–Wall defines a repulsive invisible wall in the environment.

Figure 6.1 shows control actions inside RoboTrainer's controller, detailed in chapter 5. In contrast to RoboTrainer Prototype where the user's input force is modified with the influence of the *artificial forces*-concept, the *SCA–Force Area* influences the main admittance rule's output velocity. Some CAs alter the controller's internal states directly, e.g., *GCA–Inverted Controls*, and some of them are superposed to those, e.g., CA–Counterforce. Exact details of how individual control actions alter the controller's internal states are given in the following sections. Moreover, the SCAs are wrapped into an *SCA Controller*, which provides passivity and its safety functionalities are described in section 6.2.

6.1.1. Global Control Actions

The global control actions (GCAs) are a sort of a parameter-extension of the main admittance rule. They are applied before and after the rule, according to figure 6.1. All GCAs are passive by design. This means that they do not cause RoboTrainer's movements without explicit user's intention.

GCA–Counterforce

The *GCA–Counterforce* is created for force-training scenarios with RoboTrainer. It provides a constant effort by defining a counterforce that users have to overcome to move the RoboTrainer. A counterforce is defined in RoboTrainer's coordinate system as a vector in the XY-plane. For example, suppose the GCA–Counterforce with force 20 N is defined in the negative X direction (RoboTrainer's local coordinate system). In that case, users need to push with force >20 N to move the RoboTrainer. The same is applicable for the torques, i.e., rotation, around the Z-axis. The CA–Counterforce F_{CF} modifies the user's force $F_{\mathrm{h}i}$ at time-step k for each degree of freedom i as follows:

$$F_{\mathrm{out}i}(k) = \begin{cases} sign(F_{\mathrm{h}i}(k)) \left(\|F_{\mathrm{h}i}(k)\|_2 - \|F_{\mathrm{CF}i}\|_2\right) & \text{if } F_{\mathrm{h}i}(k) > F_{\mathrm{CF}i}, \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

The $sign()$ represents the sign function, i.e., returns -1 for negative or 1 for positive argument.

GCA–Center of Rotation

The *GCA–Center of Rotation* modifies the RoboTrainer's apparent center of rotation. If no transformation of user input force is done, the rt's controller assumes that the forces and torques are applied to the base coordinate system's origin. For RoboTrainer Prototype, this is in the geometrical center of its *rob@work* base and for RoboTrainer v2 it's on the corner of the base plate just under the force-torque sensor (FTS) (see beneath the number two in figure 4.8 right). In the RoboTrainer's XY plane, the center of rotation is moved parallel to the ground. Therefore, it does only influence torques not user's input forces. This approach is proposed by Chuy, Hirata, and Kosuge [28] (see section 2.3.5). The *GCA–Center of Rotation* at position X_{cor} and Y_{cor} modifies users input torque T_{h} at time-step k as follows:

$$T_{\mathrm{out}}(k) = F_{\mathrm{hx}}(k) \cdot Y_{\mathrm{cor}} - F_{\mathrm{hy}}(k) \cdot X_{\mathrm{cor}} + T_{\mathrm{h}}(k) \tag{6.2}$$

The F_{hx} and F_{hy} are components of the user's interaction force in the X- and Y-Axis directions.

GCA–Inverted Controls

The *GCA–Inverted Controls* modifies the left-right translations and rotation for inversion of both. As a result, the RoboTrainer will move to the left when the user's input force is given to the right and vice versa. This GCA is applied after the admittance rule, and it changes the sign of the calculated velocity. Additionally, the GCA can be configured to reduce RoboTrainer's velocity if inverted controls are used. The *GCA–Inverted Controls* is applied on the Y component of the linear velocity V_{y} and the rotational velocity ω at time-step k using the following equation:

$$V_{\text{out}}(k) = min(-1 \cdot V_{\text{y}}(k), -1 \cdot sign(V_{\text{y}}(k)) \cdot V_{\text{inverted\_max}_y}) \tag{6.3}$$

$$\omega_{\text{out}}(k) = min(-1 \cdot \omega(k), -1 \cdot sign(\omega(k)) \cdot \omega_{\text{inverted\_max}}) \tag{6.4}$$

The function min yields the smaller of two numbers. The function $sign$ yields the sign of a number, i.e., -1 for negative or 1 for a positive number.

6.1.2. Spatial Control Actions

The spatial control actions (SCAs) are filters modifying the main admittance controller's output velocity. The only exception is SCA–Counterforce with the same function as its global version but additionally influencing the user's input force in a limited area. The SCAs are placed along the training path and they rely on the localization functionality of the RoboTrainer. An artistic overview of the SCAs is shown in figure 6.2.

The spatial control actions can be separated into two types when regarding their influence on RoboTrainer's output velocity. The first type are "modification SCAs", which directly change the rt's velocity and they are passive by their nature (cf. explanation for GCAs in section 6.1.1). *SCA–Area* (cf. figure 6.1) is in this group. The second type are "superposition SCAs", whose influence is superposed to the RoboTrainer's velocity calculated by the main admittance controller. The superposition SCAs modify the RoboTrainer's velocity according to equation (6.5), where $\mathbf{V}_a(k)$ is the calculated velocity of the action a from the set of active SCAs A.

$$\mathbf{V}_{\text{out}}(k) = \mathbf{V}_{\text{main}}(k) + \sum_{a}^{A} \mathbf{V}_a(k) \tag{6.5}$$

The superposition SCAs are *SCA–Force Area*, *SCA–Wall*, and *SCA–Path Following*. All three of them use the virtual force field (VFF) principle with internal admittance dynamics (equation (2.12)) to calculate the velocities from virtual force fields.

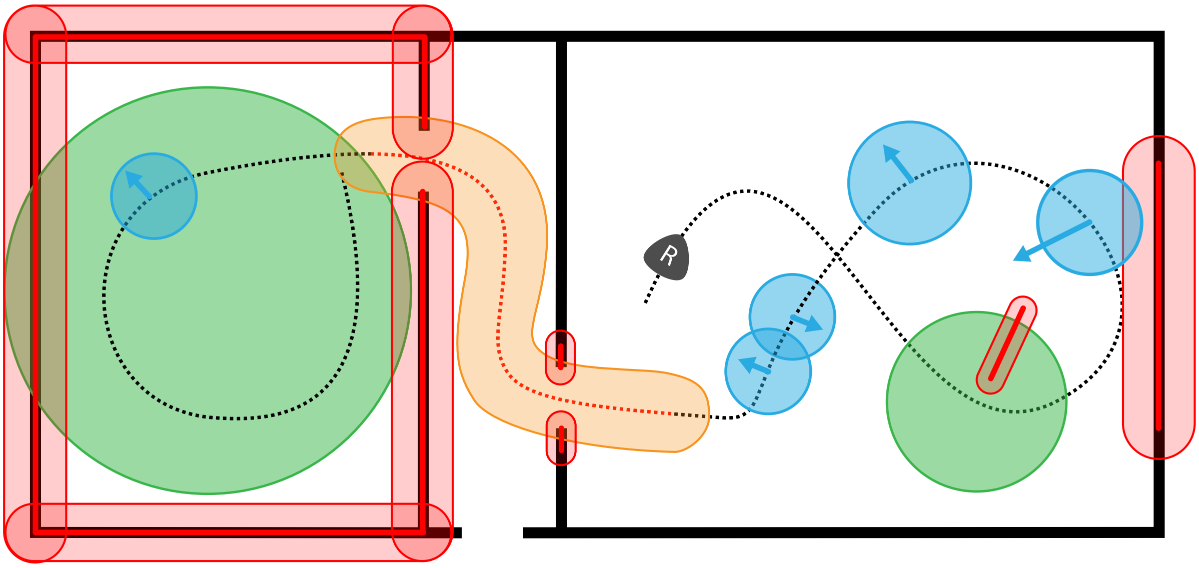


Figure 6.2.: Artistic representation of a scenario with the RoboTrainer using spatial control actions (SCAs). The dashed black line represents the scenario's base path, the triangle with the letter "R" the RoboTrainer. The solid black lines are the borders of the training environment, e.g., walls. The SCAs are marked with colors. Blue circles with an arrow depict *SCAs–Force Area*, green circles represent *SCAs–Area*, red lines depict *SCAs–Wall*, and the orange area *SCA–Path Tracking*. Author: Peter Wern [155, 178].

SCA–Area

The *SCA–Area* modifies the RoboTrainer's behavior, i.e., velocity, in an arbitrary manner. Some examples of this manipulation are presented in figure 6.3. The manipulation executed with this SCA is two-sided. On the one hand, unexpected, spatially limited behaviors can make the training more complicated. On the other side, one can limit some RoboTrainer's functionalities to increase safety for the user, e.g., the RoboTrainer's maximal velocity is limited in the room on the left in the artistic scenario (see figure 6.2). Multiple manipulations can be defined within a single area and the areas of different *SCA–Area*-instances can overlap. However, since this can cause a race condition in the calculation, only the SCA with its center closest to the RoboTrainer is active. Further, some manipulations, like the inversion of the movement, may lead to a "bouncing robot" under certain conditions. This should be considered when planning the specific training scenario. Besides the manipulations presented in figure 6.3, there is more still imaginable but not implemented yet.

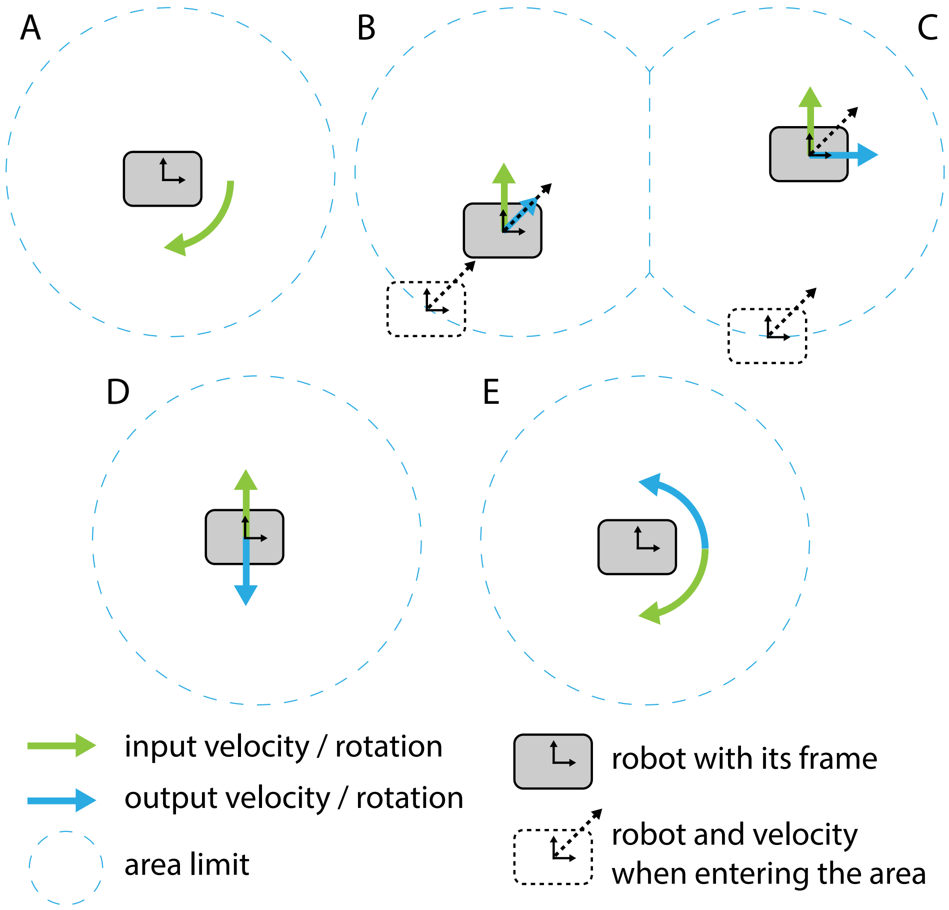


Figure 6.3.: Concept of the *SCA–Area* with different resulting behaviors: (A) RoboTrainer's rotation is blocked, (B) movement is fixed to the driving direction at the entrance of the area, (C) movements along an axis in the plane (X, Y) are switched, (D) movements along an axis are inverted, (E) the rotation is inverted.

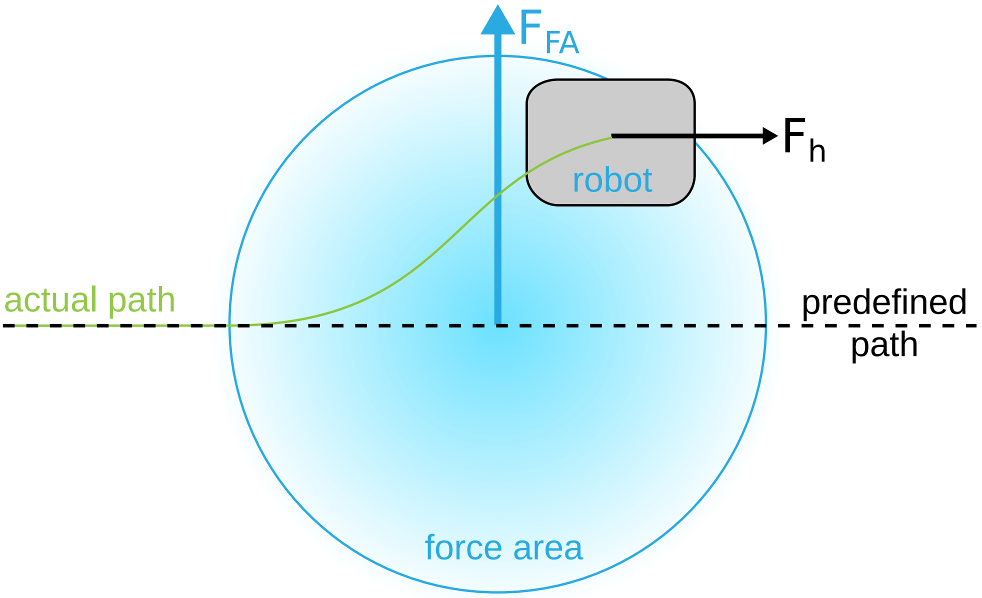


Figure 6.4.: Idea of the *SCA–Force Area*. A user navigates the RoboTrainer along the predefined path with force F_{h}, where the *SCA–Force Area* pulls RoboTrainer with the force F_{FA}. The blue circle marks the border of the virtual force.

SCA–Force Area

An *SCA–Force Area* is defined in the same manner as discussed for the concept of *artificial forces* presented in chapter 3, i.e., by a force vector and its influence radius. An *SCA–Force Area* affects the RoboTrainer when inside its influence radius. Its properties are defined in the environment's coordinate system on the ground plane. When influenced by a Force Area, RoboTrainer feels like on a slope drifting in the force's direction. It is expected from the users to compensate this drift and to try to keep the RoboTrainer on the predefined path. An *SCA–Force Area* is usually defined directly on a training path, therefore, strictly connected to it. Figure 6.4 presents the idea of this SCA graphically.

To avoid jerk caused by a disturbance force's "jump" when entering an influence area, the strength of a force inside an area changes based on a profile. The implemented profiles are trapezoidal, i.e., the one profile used with RoboTrainer Prototype, the exponential profile and the Gaussian. Those have the minimum value on the border of the influence radius and maximal value in the middle of it. Since there were no significant differences between the three proposed profiles, the trapezoidal profile is assumed in the following

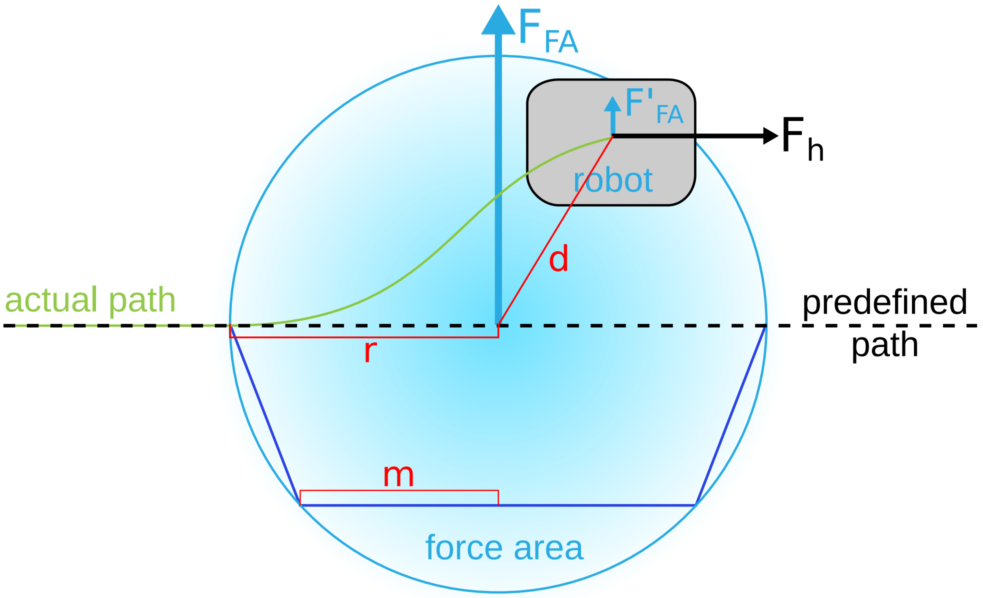


Figure 6.5.: Force change profile inside an *SCA–Force Area*. A user navigates the RoboTrainer along the predefined path with force F_h, where the *SCA–Force Area* pulls RoboTrainer with the force F_FA. The blue circle marks the border of the virtual force. The blue area in the middle shows the changing strength of the disturbance force inside the area. The darker blue line shows the trapezoidal profile along one DOF. The red markings depict the trapezoidal function parameters (equation (6.6)). The small blue arrow on the robot depicts the force area's current value at the specific position F'_FA.

considerations. In this case, the virtual force field-function inside the influence radius is:

$$\mathbf{F}_\mathrm{out} = \mathbf{F}_\mathrm{FA} \cdot \begin{cases} 0 & \text{if } d > r \\ \frac{1}{m-r} \cdot (d - r) & \text{if } m < d < r \\ 1 & \text{if } d < m \end{cases} \tag{6.6}$$

r is the radius of the force area, $2 \cdot r$ is the length of the trapezoid's longer base, m is the length of the trapezoid's shorter base, and d is the distance from RoboTrainer to the center of the force area. Figure 6.5 depicts those values in an example.

If RoboTrainer is inside an area, disturbance force is calculated and transformed to the device's control-coordinate system. The resulting velocity is then calculated, using the internal admittance model. The analysis of data from the pilot study with RoboTrainer Prototype (section 3.3) indicates that for users who move faster, the effect of *SCA–Force Area* is smaller [10, 154]. The reason for this is admittance dynamics used to generate disturbance velocity[1]. Therefore, it would be sensible to compensate those velocity effects to have the same deviation regardless of the RoboTrainer's velocity [178, 155]. The resulting disturbance velocity is calculated using admittance dynamics' response to the input force from equation (2.18)[2]:

$$\mathbf{v}_{\mathrm{FA}}(t) = \frac{1}{\mathbf{D}_{\mathrm{FA}}} * \left(1 - e^{-\frac{t}{\frac{\mathbf{M}_{\mathrm{FA}}}{\mathbf{D}_{\mathrm{FA}}}}}\right) * \mathbf{f}_{\mathrm{FA}}(t) = \mathbf{K}_{\mathrm{FA}} * \left(1 - e^{-\frac{t}{\mathbf{T}_{\mathrm{FA}}}}\right) * \mathbf{f}_{\mathrm{FA}}(t) \tag{6.7}$$

The deviation d, for a constant force $\mathbf{f}(t) = \mathbf{F}_{\mathrm{FA}}$, is then the result of the integration of equation (6.7) over time:

$$d = \int_0^{\hat{t}} \mathbf{v}_{\mathrm{FA}}(t) = \mathbf{K}_{\mathrm{FA}} * \left(\hat{t} + \mathbf{T}_{\mathrm{FA}} e^{-\frac{\hat{t}}{\mathbf{T}_{\mathrm{FA}}}} - \mathbf{T}_{\mathrm{FA}}\right) * \mathbf{F}_{\mathrm{FA}} \tag{6.8}$$

The goal is to achieve deviation d_{v} for velocity $v \in (0, V_{\max}], \mathbb{R}$ to be the same as the deviation d_{norm} for norm velocity $\mathbf{v}_{\mathrm{norm}} = \lambda v$, i.e., $d_{\mathrm{v}} = d_{\mathrm{norm}}$. Moving at velocity v, the user needs time $t = \frac{2r}{\mathbf{v}}$ to traverse the *SCA–Force Area* with radius r.

As a result, the compensation of the disturbance force $\mathbf{F}_{\mathrm{FA}}$ concerning norm velocity $\mathbf{v}_{\mathrm{norm}}$ and the RoboTrainer's actual velocity $\mathbf{v}_{\mathrm{RT}}$ inside the *SCA–Force Area* with radius r, is calculated by

$$\mathbf{F}'_{\mathrm{FA}} = \mathbf{F}_{\mathrm{FA}} * \frac{\frac{2r}{\mathbf{T}_{\mathrm{FA}} \cdot \|\mathbf{v}_{\mathrm{norm}}\|_2} + e^{-\frac{2r}{\mathbf{T}_{\mathrm{FA}} * \|\mathbf{v}_{\mathrm{norm}}\|_2}} - 1}{\frac{2r}{\mathbf{T}_{\mathrm{FA}} * \|\mathbf{v}_{\mathrm{RT}}\|_2} + e^{-\frac{2r}{\mathbf{T}_{\mathrm{FA}} * \|\mathbf{v}_{\mathrm{RT}}\|_2}} - 1} \tag{6.9}$$

The simulation of disturbance force compensation for $v_{\mathrm{norm}} = 0.3$ and $v_{\mathrm{RT}} = 0.5$ is shown in figure 6.6. The goal is to have the same-sized areas under the green and solid blue lines, which is not the case when looking at figure 6.6. The problem lies in calculating the disturbance velocity from the *SCA–Force Area*'s force. This calculation does not only modify the amplitude but also when disturbance's peaks occur (cf. two dashed lines in figure 6.6).

Figure 6.7 shows the dynamics of a disturbance with a trapezoidal force profile. The figure underlines the time delay of the disturbance's velocity to its position. This time delay reflects itself in spatial uncertainty in RoboTrainer's scenarios.

Therefore, an alternative calculation of the disturbance velocity is proposed. Based on the steady-state equivalency of the admittance model (equation (2.23)), it is sensible to

[1] In the pilot study with RoboTrainer Prototype, the disturbance velocity is generated using the main admittance controller (see section 3.2.2).

[2] The following equations use the operator "$*$" and fraction with matrices for element-wise calculation explained in section 1.4.

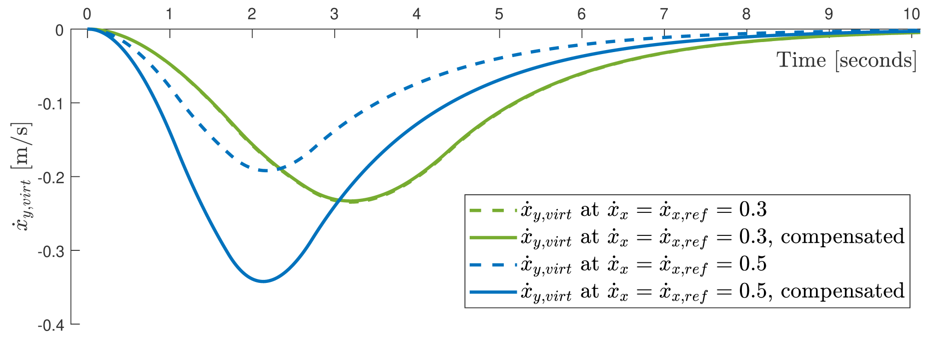


Figure 6.6.: Evaluation of velocity compensation in the *SCA–Force Area*. The figure compares disturbance velocity with (solid line) and without (dashed line) compensation. The reference velocity $v_{\text{norm}} = 0.3$ tests are marked with the green line, and the tests on higher velocity $v_{\text{RT}} = v_{\text{virt}} = 0.5$ with the blue line. Author: Peter Wern [178, 155].

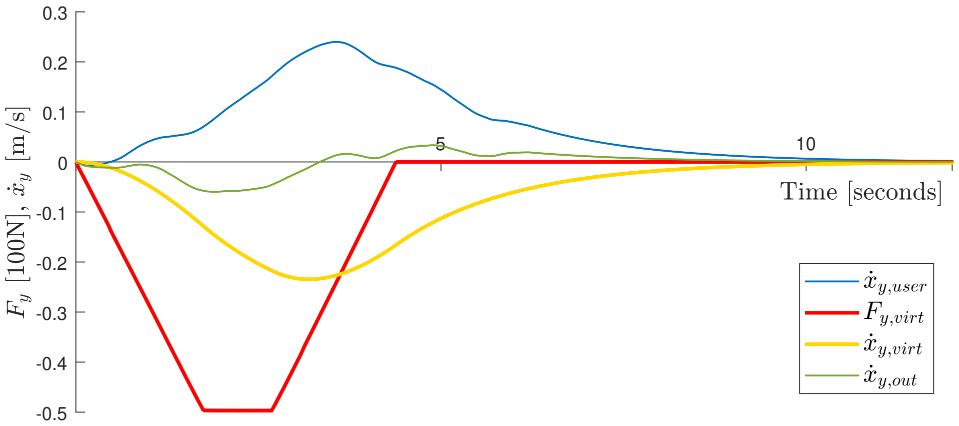


Figure 6.7.: Effect of the SCA–Virtual Force on users' velocity. The red line is the amplitude of the Virtual Force's trapezoidal disturbance. The yellow line marks disturbance velocity calculated using admittance dynamics. The blue line is the velocity resulting from the main admittance controller, i.e., users' input, and the green line is the resulting velocity after the superposition. Author: Peter Wern [178, 155].

use a scaling factor to calculate a disturbance velocity from the corresponding force using equation (6.10)[3].

$$\mathbf{v}_{\mathrm{FA}}(t) = \frac{\mathbf{f}_{\mathrm{FA}}(t)}{\mathbf{D}_{\mathrm{FA}}} = \mathbf{K}_{\mathrm{FA}} * \mathbf{f}_{\mathrm{FA}}(t) \tag{6.10}$$

The drawback of this approach is that the user can experience some jerk since the disturbance velocity is no longer smoothed by admittance dynamics. This can be solved using disturbance force profiles with smoother transitions, e.g., Gaussian or exponential function. Non-dynamic calculation of the velocity also provides transparent RoboTrainer's behavior, regardless of the user's speed and with clear spatial borders of the disturbance. Therefore, this approach is used in the user-evaluation of RoboTrainer v2 at German Sport University Cologne (DSHS).

The scaling factor approach still has the issue that users who move faster will experience the disturbance for a shorter time. Using the same compensation method as above, i.e., assuming the constant disturbance force $\mathbf{f}(t) = \mathbf{F}_{\mathrm{FA}}$, the deviation, then, is calculated with:

$$d = \int_0^{\hat{t}} \mathbf{v}_{\mathrm{FA}}(t) = \left\| \mathbf{K}_{\mathrm{FA}} * \mathbf{f}_{\mathrm{FA}} \cdot \hat{t} \right\|_2 \tag{6.11}$$

So, the disturbance force-compensation factor can be calculated using:

$$\mathbf{F}'_{\mathrm{FA}} = \mathbf{F}_{\mathrm{FA}} * \frac{\frac{2r}{\|\mathbf{v}_{\mathrm{norm}}\|_2}}{\frac{2r}{\|\mathbf{v}_{\mathrm{RT}}\|_2}} = \mathbf{F}_{\mathrm{FA}} * \frac{\|\mathbf{v}_{\mathrm{RT}}\|_2}{\|\mathbf{v}_{\mathrm{norm}}\|_2} \tag{6.12}$$

SCA–Path Following

The *SCA–Path Following* helps users to keep the RoboTrainer on the predefined path. The SCA is defined along a specific training path and, therefore, strictly related to it. The Path Tracking generates a virtual force field with a gradient orthogonal to the path, with the force's minimum on the path itself. The gradient profile is arbitrary, e.g., linear trapezoidal, quadratic, or Gaussian. The gradient is limited with a maximal deviation allowed for users to distance themselves from the path. This functionality creates a virtual corridor that users cannot leave since *SCA–Path Following* at the corridor's border overrides the user's input in order to keep them in that area. Figure 6.8 presents the concept of this SCA graphically. The velocity of this SCA is also calculated using admittance dynamics.

Figure 6.9 shows a real-world example of SCA–Path Tracking and its functionality. The maximum deviation is set to 0.5 m. The maximum value of the virtual force is set to 100 N.

[3] The following equations use the operator "$*$" and fraction with matrices for element-wise calculation explained in section 1.4.

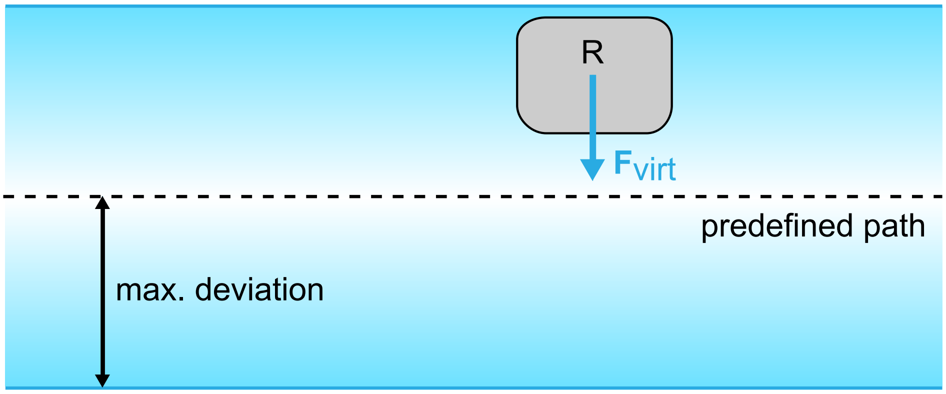


Figure 6.8.: Idea of the *SCA–Path Following*. If a user moves the RoboTrainer (“R”) away from the path, a virtual force $\mathbf{F}_{\text{virt}}$ is applied orthogonal to and facing the path. The maximal deviation parameter defines a corridor users are allowed to deviate from the path. The saturation of the blue background represents the strength of the virtual force. Author: Peter Wern [155].

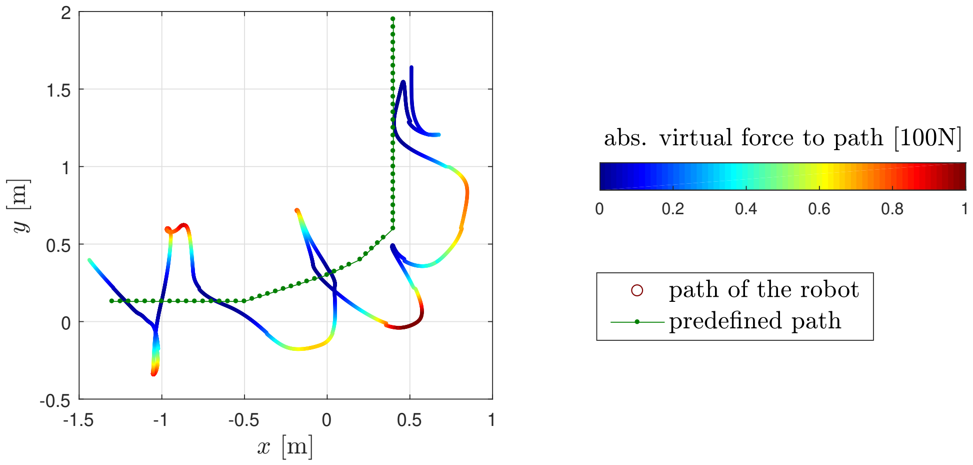


Figure 6.9.: Real-world example of the SCA–Path Tracking. The reference path is shown in green, and the path of the RoboTrainer is a multi-color line. The color on the line indicates the virtual force’s intensity to keep the robot on the reference path. The maximum deviation was set to 0.5 m. Therefore, RoboTrainer cannot go further away from the path (e.g., coordinates x=0.5, y=0). Author: Peter Wern [178, 155].

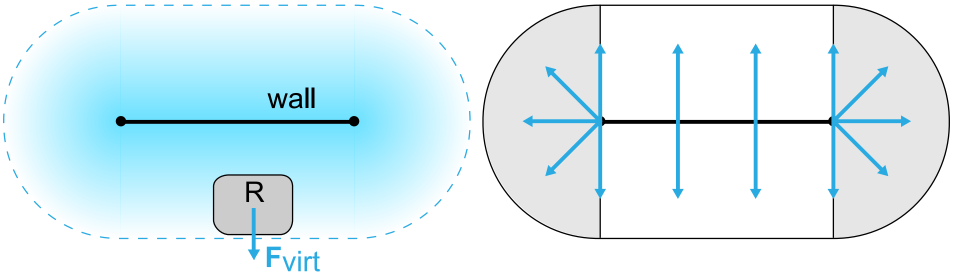


Figure 6.10.: Idea of the *SCA–Wall*. Left: If the RoboTrainer ("R") is located within the influence area (dashed line) of the Wall, a repulsive virtual force, depending on the distance from the wall, is generated. The saturation of the blue background represents the strength of the virtual force. Right: The *SCA–Wall* has repulsive force in all directions from the wall. In gray areas, the RoboTrainer is repelled by the wall's endpoint and in the white area from the wall itself. Author: Peter Wern [178, 155].

SCA–Wall

The *SCA–Wall* defines a non-passable obstacle for RoboTrainer, invisible in the physical environment. A wall generates a repulsive virtual force field around itself, up to a predefined distance, i.e., influence area. This force is pushing the RoboTrainer away in the direction orthogonal to the wall. The gradient direction is determined depending on RoboTrainer's position to the wall and its endpoints. The gradient from the endpoints is not orthogonal to the wall, but it is the RoboTrainer's approaching direction (figure 6.10 right). Figure 6.10 left depicts the idea behind this control action. The gradient profile is arbitrary, as for the *SCA–Path Following*, and the velocity is also calculated using admittance dynamics. The *SCA–Wall* represents a soft barrier used to block dangerous areas, e.g., stairways, and virtually limits the training space, or challenges the user by placing invisible obstacles into a training environment.

6.2. Passivity and Safety for Control Actions

This section provides concepts for monitoring and limiting the control actions to guarantee passive behavior and user safety. The concept and the results were published in the conference paper Stogl et al. [157].

The superposition CAs may lead to non-passive RoboTrainer's behavior since they are added directly to the main controller's output. This means that RoboTrainer moves, although there is no clear intention to do that from the user. This is a problem for all superposition-control actions. Still, for the *CA–Counterforce* it's simple to integrate the protection against this directly (see equation (6.1)). Therefore, the most critical control actions regarding RoboTrainer's passivity are the *SCA–Force Area*, *SCA–Path Following*, and *SCA–Wall*.

The control actions generally only consider their positions in the training environment and the training path, without broader knowledge about the users' performance and fitness. Moreover, the control actions are often reused between the individual exercises to reduce training's configuration time. This may lead to an inappropriate configuration of their parameters for individual users and potentially to dangerous situations.

For better understanding, such dangerous situations are explained exemplary on the *SCA–Force Area*. This SCA is defined on a predefined path, with a force vector, i.e., direction and amplitude and an influence radius (see section 6.1.2). It is intended to disturb a user during training by pushing the RoboTrainer in a specified direction. The users need to "feel" this disturbance and try to keep RoboTrainer on the predefined path.

If not appropriately configured, a *SCA–Force Area* may exceed the user's input force and overwhelm them. In such a situation, users can react in two ways: (i) release RoboTrainer's handles or (ii) try to overcome the disturbance force with their strength. Using spatial control actions' implementations proposed in the previous section (section 6.1.2), both cases lead to dangerous situations. In the first case, RoboTrainer will start moving by itself in the disturbance force's direction. In the second case, RoboTrainer pushes or pulls its users outside of the influence radius. Therefore, two concepts for avoiding such situations are developed. To realize them, affected SCAs are wrapped into *SCA Controllers* (see figure 6.1) to additionally process the information about the user's input force.

Passivity for Control Actions

This concept is inspired by the passivity concept for a controller presented in section 5.3. This means, RoboTrainer should not move, even under the influence of control actions if there is no user's intention. To keep the system passive, the velocity resulting from SCA should not exceed the speed resulting from user's input provided by the main admittance controller. This means that individual SCA may not exceed the main controller's velocity. To avoid limitation of the training challenge, this rule is only applied if the user's input is lower than some predefined minimal force $\|F_h\|_2 < F_{min}$, which is defined considering the user's fitness and performance. So, if the passivity functionality is active, the superposition-spatial control actions influence the RoboTrainer's velocity by using the equation (6.13)[4]. The additional scaling factor limits the SCA's velocity if it is larger

[4] Equation (6.13) uses the operator "$*$" for element-wise matrix calculation explained in section 1.4.

than the user-induced speed.

$$\mathbf{V}_{\text{out}}(k) = \mathbf{V}_{\text{main}}(k) + \sum_{a}^{A} \mathbf{V}_a(k) * f(\mathbf{V}_{\text{main}}(k), \mathbf{V}_a(k)) \tag{6.13}$$

where

$$f(\mathbf{V}_{\text{main}}(k), \mathbf{V}_a(k)) = \begin{cases} 1 & \text{if } \|F_h\|_2 \geq F_{min} \\ 1 & \text{if } \left\|\mathbf{V}_{\text{main}}(k)\right\|_2 \geq \left\|\mathbf{V}_a(k)\right\|_2 \text{ and } \|F_h\|_2 < F_{min} \\ \frac{\left\|\mathbf{V}_{\text{main}}(k)\right\|_2}{\left\|\mathbf{V}_a(k)\right\|_2} & \text{if } \left\|\mathbf{V}_{\text{main}}(k)\right\|_2 < \left\|\mathbf{V}_a(k)\right\|_2 \text{ and } \|F_h\|_2 < F_{min} \end{cases} \tag{6.14}$$

The concept is evaluated by using *SCA–Force Area* which configuration is oriented towards the user and stronger than their maximal input force. A potentially dangerous situation without the passivity concept for spatial control actions is shown in figure 6.11. The figure shows only one degree of freedom for a clear representation, i.e., longitudinal, x-Axis of RoboTrainer. The figure depicts the situation where the user is overwhelmed by the disturbance force and decides to remove the hands from RoboTrainer's handlebars (at 5 s). At that moment, RoboTrainer starts to move towards the user, i.e., it continues its movements according to the velocity generated by the virtual force field (at 6 s) until the influence of the field disappears (at 8 s).

Using the passivity concept for SCAs, RoboTrainer stops when the user releases the device's handles (figure 6.12 at 5 s to 7 s). The RoboTrainer only starts to move again when the user's intention is detected by the velocity generated mainly by the main controller. The disturbance velocity is damped using equation (6.13) because the user's input force is less than $F_{\%\text{min}} = 0.5$. The effect of the passivity concept is also the same for *SCA–Path Following* and *SCA–Wall*.

Safety for Control Actions

The safety concept for control actions is designed to protect the user from non-intentional RoboTrainer's movement in their direction. Generally, it is allowed for SCAs to "override" velocity generated by the user's input, but not when this velocity results in RoboTrainer's movement toward the user. Suppose the user is using stronger forces than the predefined minimal interaction force $\|F_h\|_2 \geq F_{min}$. In that case, the resulting velocity of a strong SCA's force will be limited if this would result in the backward movement of the RoboTrainer. For this functionality, a protection angle of 30° behind the RoboTrainer is defined. Then, the influence of the spatial control action is limited using equation (6.13), where $f(\mathbf{V}_{\text{main}}(k), \mathbf{V}_a(k)) = \frac{\left\|\mathbf{V}_{\text{main}}(k)\right\|_2}{\left\|\mathbf{V}_a(k)\right\|_2}$.

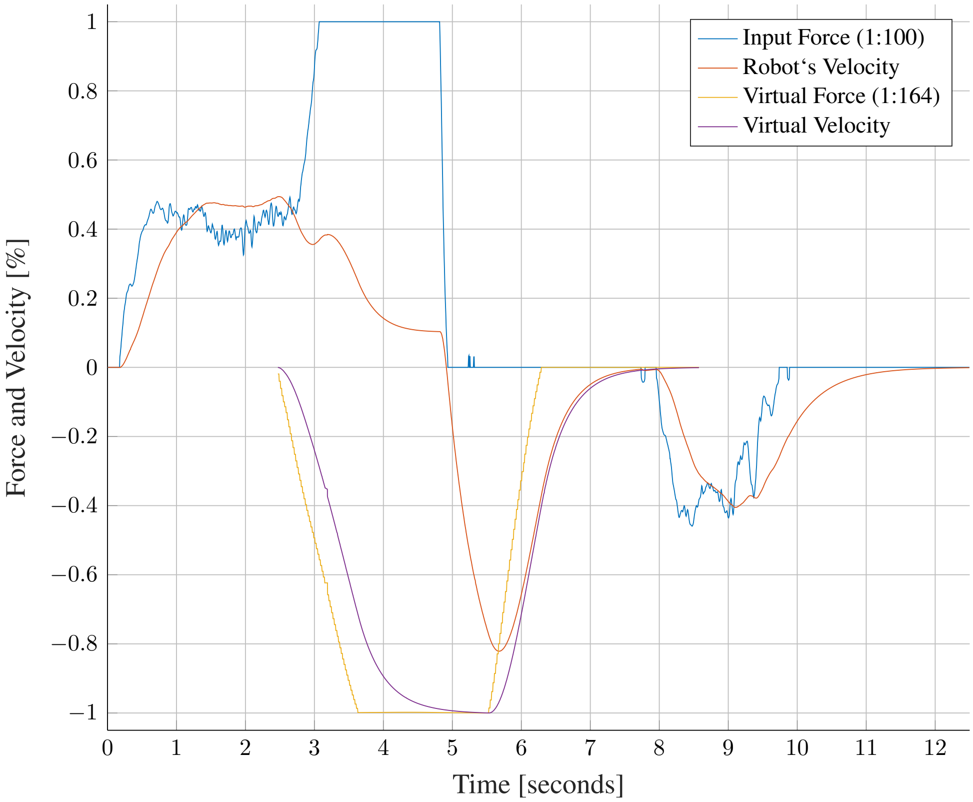


Figure 6.11.: A dangerous situation without the passivity concept for SCAs. The user is overstrained by the disturbance force and lets RoboTrainer go (at 5 s). RoboTrainer moves towards the user (6 s) until the disturbance force's influence disappears (at 8 s). After that, the user moves RoboTrainer again.

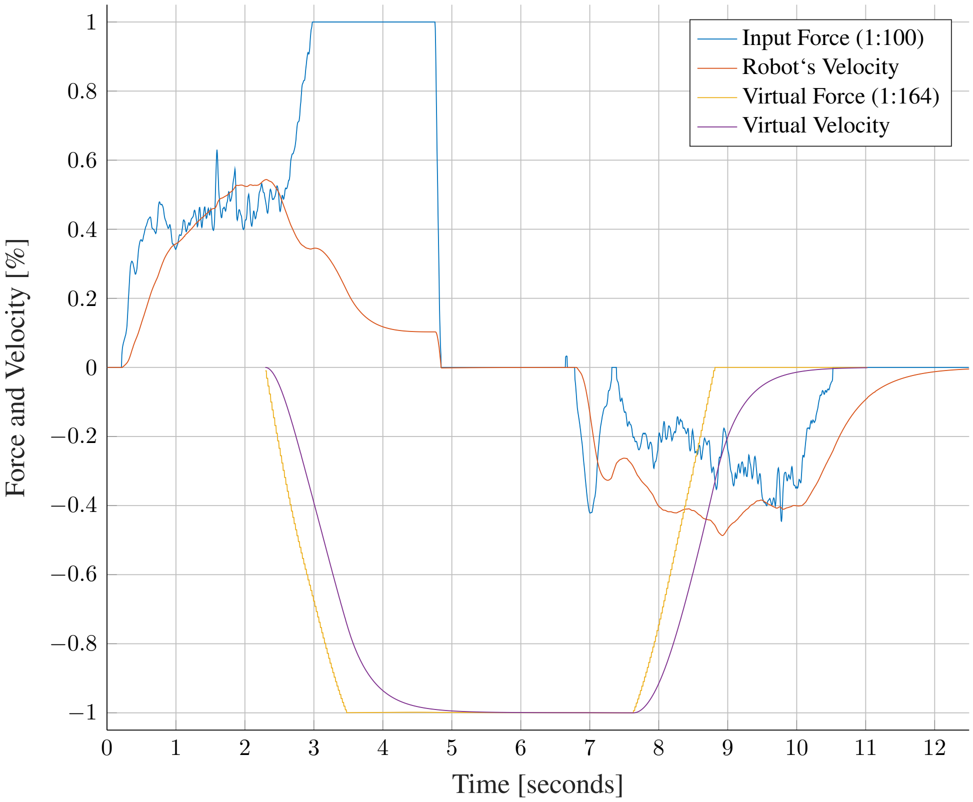


Figure 6.12.: A dangerous situation with the passivity concept for SCAs. When the user removes hands from the handles (at 5 s), RoboTrainer stops. RoboTrainer does not move (5 s to 7 s), even in a disturbance force field, until the user's intention is detected (6.3 s to 10.2 s). The influence of the disturbance velocity is damped because of low user's forces ($F_\text{h} < F_{\%\text{min}} = 0.5$).

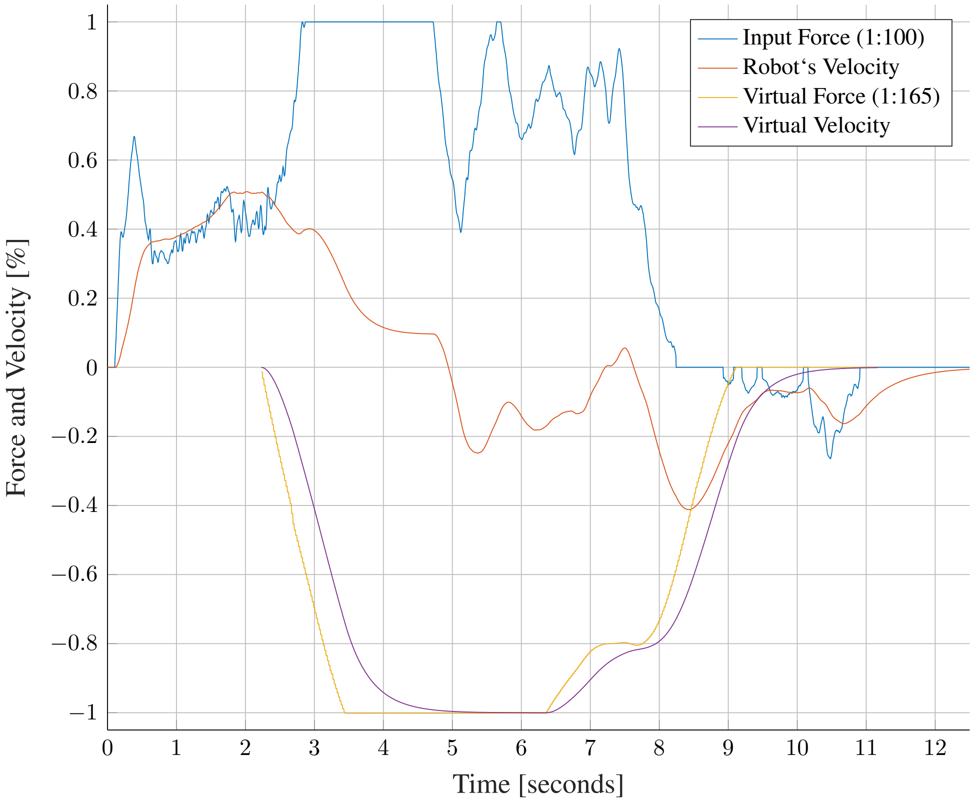


Figure 6.13.: A dangerous situation without the safety concept for SCAs. The user tries to overcome a disturbance force which is too strong. Without any limitations, the user is pushed back from the disturbance force field (negative robot velocity at 5 s to 7 s) against its intention (user’s input force is positive).

The safety concept is evaluated using *SCA–Force Area* configured to be oriented towards the user and stronger than their maximal input force. A potentially dangerous situation without the safety concept for spatial control actions is shown in figure 6.13. The figure shows only one degree of freedom, i.e., longitudinal, x-Axis of RoboTrainer for clear representation. The figure depicts the situation where the user tries to overcome a too strong disturbance force. Without the safety concept, RoboTrainer starts to push the user backward out of the *SCA–Force Area*’s area without their intention. The input force’s positive direction and negative RoboTrainer’s velocity indicate this (figure 6.13 5 s to 7 s).

When using the safety concept, in the same situation, the RoboTrainer will not move backward (figure 6.14 - 4 s to 9 s) until there is an explicit user’s intention for it (10 s to 14 s). After 10 s, the passivity concept damps the disturbance velocity according to equation (6.13).

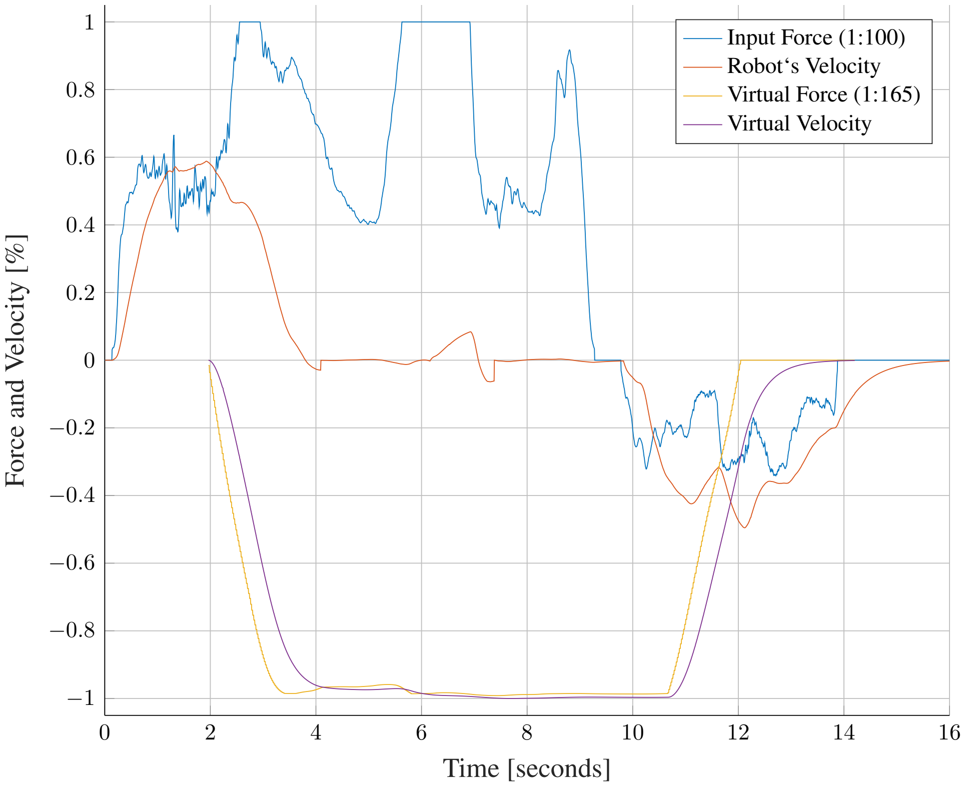


Figure 6.14.: A dangerous situation with the safety concept for SCAs. Even if the disturbance force is stronger than the user's input force, RoboTrainer does not move towards the user against its intention (4 s to 9 s). The robot's velocity is stopped at zero and becomes negative only when the user intends to move RoboTrainer towards himself (10 s to 14 s).

Table 6.1.: Parameters definition for global control actions.

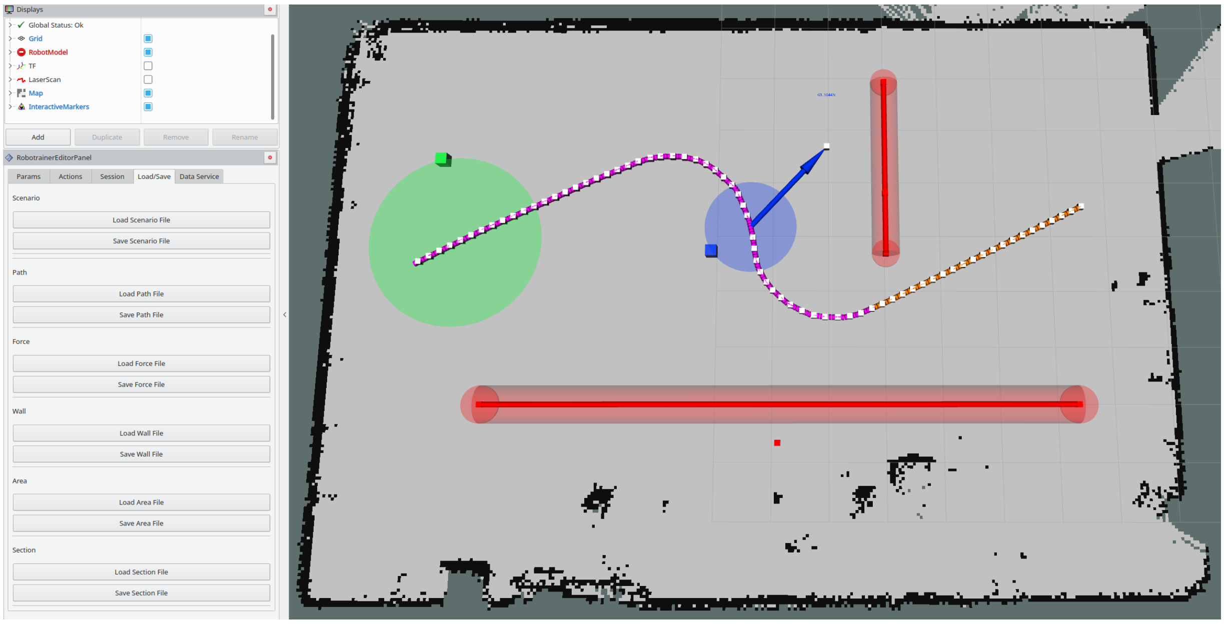
Parameter Name	Description
	GCA–Counterforce
enabled	Flag to enable and disable the GCA
counterforce_x	Linear force against the user (Value range: 0.0 N to 70.0 N)
counterforce_y	Linear force in left/right direction (positive to the left) (Value range: −25.0 N to 25.0 N)
countertorque_z	Torque for left/right rotation (positive in the clockwise direction) (Value range: −8.0 N m to 8.0 N m)
	GCA–Center of Rotation
adapt_cor	Flag to enable and disable the GCA
cor_x	Distance from the rt's kinematics center along the x-axis (positive values away from the user) (Value range: −1 m to 2 m)
cor_y	Distance from the rt's kinematics center along the y-axis (positive values to the left) (Value range: −1 m to 1 m)
	GCA–Inverted Controls
y_reversed	Flag to enable and disable the inversion of left/right linear movements
rot_reversed	Flag to enable and disable the inversion of left/right rotation

6.3. Implementation of the spatial control actions

The control actions (CAs) are implemented in RoboTrainer's controller in two ways, as shown in figures 5.1 and 6.1. The global control actions (GCAs) are unique and integrated directly into RoboTrainer's base controller (cf. figure 5.18). Table 6.1 shows their parameters defined in the controller's namespace. Note that for brevity reasons, only the most important parameters are listed. Other parameters, like the definition of CA's dynamic behavior, are not shown. The parameters for all control actions are managed using *dynamic_reconfigure*[5] ROS package providing run-time configuration. This is useful when switching between the tasks for user studies, testing, and fine-tuning purposes.

The spatial control actions (SCAs) are based on the ROS-filters, which can be dynamically loaded as needed into RoboTrainer's controller. This approach reduces the RoboTrainer's controller's run-time footprint, as only used SCAs in the current scenario are loaded. The only exception is SCA–Counterforce integrated into the main controller because of code-reuse from its global version. The implemented SCAs-classes manage all defined actions of the same type. The parameters for the definition of an SCA's in-

[5] *Dynamic_reconfigure*-package: `https://wiki.ros.org/dynamic_reconfigure`.



Displays

- Global Status: Ok
- Grid ☑
- RobotModel ☑
- TF ☐
- LaserScan ☐
- Map ☑
- InteractiveMarkers ☑

Figure 6.15.: *RoboTrainerEditor* – Configuration GUI for SCAs based on ROS-*rviz* 3D-visualization tool. The predefined path for the training's task is shown in violet. The orange part of it marks *SCA–Path Following*. The red, half-transparent cylinders represent the *SCA–Wall*, the blue arrows on a blue circle the *SCA–Force Area*, and a green circle *SCA–Area*. On the left, the control panel of the *RoboTrainerEditor* is shown.

.

stance are presented in table 6.2. The SCAs are stored in the configuration file of a task. An example file with path[6] and associated spatial control actions definition is shown in listing 6.1.

Since the SCAs are placed freely into the training environment considering the training path, their direct configuration in a configuration file is very cumbersome. Therefore, a graphical user interface (GUI) for their configuration is developed. The GUI is realized as a plugin for ROS's 3D visualization tool *rviz*. The GUI provides management of the training scenarios, like recording training paths from a real environment, defining SCAs, and storing, loading, and activating RoboTrainer's scenarios. The GUI is shown in figure 6.15, representing the example scenario from listing 6.1. Groten [59] developed the initial version of the GUI during his Bachelor's Thesis.

[6] For brevity, only, first and last points of the path are shown.

Table 6.2.: Parameters definition for spatial control actions.

Parameter Name	**Description**
	SCA–Counterforce
counter-force_area_scaledown_dist	Distance factor from the CA's center to reduce the force linearly, i.e., adaption of the shorter base of the trapeze (see explanation in section 6.1.2 (Value range: 0.0 % to 1.0 %)
area_counter_force_x	Linear force against the user (Value range: 0.0 N to 70.0 N)
area_counter_force_y	Linear force in left/right direction (positive to the left) (Value range: −25.0 N to 25.0 N)
area_counter_torque_z	Torque for left/right rotation (positive in clockwise direction) (Value range: −8.0 N m to 8.0 N m)
	position is defined by the *SCA–Area*
	SCA–Area
area	(x, y, z) coordinates of the Area's center.
area_functions	list of functions in the area: keep_direction, keep_rotation, invert_direction, invert_y, invert_rotation, double_speed, half_speed, apply_counterforce
margin	(x, y, z) coordinates of a point on the outer border
	SCA–Force Area
area	(x, y, z) coordinates of the RoboTrainer's center.
arrow	force_distance_function: function for force profile, e.g., trapezoidal, Gaussian, exponential (x, y, z) disturbance force vector, defined from the "area"-point
margin	(x, y, z) coordinates of a point on the outer border
	SCA–Path Following
start	label of the start point on the path
end	label of the end point on the path
force_distance_function	function for force profile, e.g., trapezoidal, Gaussian, exponential
max_deviation	influence distance of the SCA and maximal radius RoboTrainer can distance itself from the path
	SCA–Wall
L	(x, y, z) coordinates of the left edge of the wall
R	(x, y, z) coordinates of the right edge of the wall
area	influence distance of the wall
force_distance_function	function for force profile, e.g., trapezoidal, Gaussian, exponential

Listing 6.1 Example Scenario Description with spatial control actions (SCAs)

```
area:
  config:
    area_names: [area_0]
  data:
    area_0:
      area: {x: 9.719657897949219, y: 0.09479331970214844, z:
          -9.5367431640625e-07}
      area_functions: [keep_rotation]
      margin: {x: 10.037101745605469, y: -1.42730712890625, z
          : -9.5367431640625e-07}
force:
  config:
    force_names: [force_36]
    newton_per_meter: 30.0
  data:
    force_36:
      area: {x: 4.4069564794915115, y: -0.17966789915493103,
          z: 0.0}
      arrow: {force_distance_function: trapezoidal, x:
          -41.26884834592698, y: -45.06252917220266,
        z: 5.7220458984375e-05}
      margin: {x: 5.106315612792969, y: 0.2706996500492096, z
          : 1.9073486328125e-06}
section:
  config:
    section_names: [section_0]
  data:
    section_0: {end: point84, force_distance_function:
        trapezoidal, max_deviation: 0.3,
      start: point1}
wall:
  config:
    wall_names: [wall_0, wall_1]
  data:
    wall_0:
      L: {x: 9.093323707580566, y: 3.0119736194610596, z:
          -9.5367431640625e-07}
      R: {x: -1.3671283721923828, y: 2.9980289936065674, z:
          0.0}
      area:
        area: 0.6630122824922358
        cube: {x: 3.897209644317627, y: 3.66713547706604, z:
            -4.76837158203125e-07}
      force_distance_function: trapezoidal
    wall_1:
      L: {x: 1.9714610576629639, y: 0.3105745315551758, z: -1.9073486328125e-06}
      R: {x: 1.9761264324188232, y: -2.924464464187622, z: 0.0}
      area:
        area: 0.5
        cube: {x: 1.9737937450408936, y: -0.8069449663162231, z: -9.5367431640625e-07}
      force_distance_function: trapezoidal
path:
  path_name: ’’
  pivot_points: []
  point0: {x: -1.56125023851361, y: -0.5538629079224573, z: 0.0}
  ...
  ...
  point281: {x: 10.478712778186829, y: 0.5432576833420972, z: 0.0}
   points: [point0, ..., point281]
scenario: sca_parameters_example
scenario_id: scenario_id202010121935
```

6.4. Conclusions on Control Actions for Training

This chapter gives an overview of the *control actions* concept. The CAs are fundamental building blocks of the training with RoboTrainer. Besides the training path, they provide essential support and challenges for users with various physical and cognitive fitness. There are two types of control actions: *global control actions (GCAs)* and *spatial control actions (SCAs)*. The global control actions change the RoboTrainer's behavior within the whole training environment regardless of the path setup. The spatial control actions modify the device's behavior in a specified area and they are usually directly associated with a training path.

The control actions are based on the virtual force field concept, creating a potential field in the virtual representation of the training's environment. The only exceptions are *GCA–Inverted Controls* and *SCA–Area*, which modify the RoboTrainer's behavior based on a custom function. The main difference to the approaches presented in the state-of-the-art (chapter 2) is that CAs have their dynamics independent from the main controller's dynamics. This approach provides simpler decoupling between user- and device-caused velocities and enables situational adaption of their ratios.

The control actions modify RoboTrainer's behavior in two different means, i.e., by directly changing the user's input of the main controller's output or by superpositioning those values. The superpositioning control actions may induce dangerous situations for the user by overriding their intention. Therefore, two concepts, passivity and safety, are developed and discussed in this chapter. The evaluation results show the desired effects of these two concepts. Additionally, a concept for adapting the CAs' disturbances with respect to the RoboTrainer's actual velocity is presented to achieve more comparable training effects for different users. The approach tries to equalize disturbance forces' effects concerning the RoboTrainer's deviation. Unfortunately, it does not compensate the effects entirely. Therefore, a new promising approach could be the equalization of the disturbance energy for different velocities. This approach should be investigated in the future.

The chapter finalizes with the implementation details about control actions. The relevant configuration parameters, structures for storing data, and graphical user interface for configuration of training scenarios, called *RoboTrainerEditor*, are presented. The Editor and user-readable configuration were beneficial before and during the evaluations with the RoboTrainer.

7 User Evaluations

In the long term, RoboTrainer v2 should enable versatile physical and challenging cognitive training for adult persons. During training, the user navigates the RoboTrainer using force input along a predefined parkour, on which RoboTrainer changes its behavior as defined by control actions. Therefore, the interaction with the user is the most significant success and acceptance factor. Thus, the theoretical and experimental analysis provided in the last chapters is just proof of technical functionality and not the device's influence on users and their acceptance.

This chapter presents the results of user evaluations regarding RoboTrainer's features. The first study with a focus on adaptive and individualized control is presented in section 7.1. The study investigates the intuitiveness and convenience of the parameterization processes and their influence on the RoboTrainer's controllability and users' performance. The evaluation group consisted of 22 healthy participants between 20 and 40 years old, recruited at the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics (IAR-IPR). The users were required to fill out a questionnaire after each task to get feedback on investigated functionalities. The study showed that the implemented adaption and individualization of the RoboTrainer's controller presented in section 5.5 is well accepted by users making the interaction more acceptable. Also, the users felt safe during the whole time when interacting with the RoboTrainer and they preferred rather complex tasks.

The second study was done in cooperation with the *German Sport University Cologne*. It focused on the overall evaluation of RoboTrainer v2 and the influence of spatial control action on cognitive and motor levels. From a technical perspective, the newly introduced non-linear adaption controller in section 5.4 is compared to the linear-damping adaption controller from Yu, Spenko, and Dubowsky [185]. Besides the assessment using questionnaires after each task, interaction data were recorded and analyzed. The results showed RoboTrainer v2's suitability for the proposed training and discovered unexpected effects of its wheel-setup on user-walker interaction. The participants found the RoboTrainer's control intuitive and always felt very safe.

The following sections' structure follows the recommendations of Ko, LaToza, and Burnett [88] on the design of controlled experiments for software engineering tools. The sections begin with presenting evaluation variables, followed by the demographic and experience details about the evaluation group. After the description of the assignments and tasks, details about the used measurement technique are provided. The disclosure and discussion of the results are given together. Each section closes with a conclusion on findings in the corresponding study.

7.1. Evaluation of Adaptive and Individualized Control

This evaluation was conducted with 22 participants. The participants used RoboTrainer Prototype with the default and individualized parameters and answered a set of questions on the Likert-type scale. Besides this subjective feedback, interaction force, RoboTrainer's velocity, and deviation from the predefined paths were recorded. The study provided information regarding default control parameters for the RoboTrainer in future studies. The results from this section are partially published in Stogl et al. [157].

7.1.1. Evaluation Setup

Variables

The study's primary goal is to evaluate the feasibility and the effects of the parameterization strategies presented in section 5.5. The study should help to understand how individual controller parameters influence the user's interaction and performance. Therefore, the following variables are investigated by using subjective measurements, i.e., questionaries:

(1) intuitiveness and convenience of the parameterization processes;

(2) influence of personalized maximal force on the controllability and performance;

(3) influence of non-linear adaptive control versus control with fixed parameters on controllability and performance;

(4) influence of the spatial control action (SCA) and global control action (GCA) on the effort needed for task execution; and

(5) subjective influence of the position center of rotation (CoR) on control of RoboTrainer.

The hypotheses for the variables were the following:

(I) the parameterization processes are intuitive and easy to perform by the participants without any prior knowledge;

(II) when using individual control parameters, the participants experience the control of the RoboTrainer as simpler than with the default parameters;

(III) the participants find the tasks and parameterization as interesting and they feel safe the whole time;

(IV) adding GCA and SCA makes the tasks more challenging; and

(V) moving the CoR further from the user, i.e., towards the front of the RoboTrainer, like on a wheelbarrow, provides more comfortable handling.

Group

The evaluation group consisted of 22 participants (3 females) between 20 and 40 years of age. The participants were recruited at the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics (IAR-IPR) with a call-for-participation on the institute's mailing list. Therefore, the participants were university students (13) and researchers (9) with computer science, electrical, or mechanical engineering background. The only inclusion criteria for the participants was non-existent known issues or diseases of the musculoskeletal apparatus. To estimate the experience of the participants, they were asked the following two questions ahead of evaluation with RoboTrainer:

(1) "How much previous knowledge and experience do you have in the use of conventional walkers?"; and

(2) "How much previous knowledge and experience do you have in the use of RoboTrainer?".

The answers were recorded using a Likert-type scale with five response categories ranging from "1 - no experience" to "5 - very much experience". Regarding the experience with a conventional walker, 1 participant reported to be very experienced, 1 participant intermediate, and the others said to have little (3) or no experience (17). Regarding interaction with RoboTrainer, 13 participants reported no experience, 3 participants little, 1 participant intermediate experience, and 5 participants assessed themselves as experienced (3) or very experienced (2). In the latter group are myself, the authors of the paper Stogl et al. [157], and students who worked with RoboTrainer in the past. Tables 7.1 and 7.2 present an overview of demographic data and experience self-assessment in a condensed form.

Table 7.1.: Demographics of participants ($n = 22$) in the evaluation of individual control parameters and non-linear adaptive control.

Gender		Age			Occupation		Experience* (median, range)	
Female	Male	20-25	25-30	30-35	Student	Researcher	conventional Walker	Robotrainer
3	19	8	12	2	13	9	1 (1-5)	1 (1-5)

* for detailed distribution, see table 7.2

Table 7.2.: Experience of participants ($n = 22$) in use of a conventional walker and RoboTrainer.

Conventional Walker					RoboTrainer				
1	2	3	4	5	1	2	3	4	5
17	3	1	0	1	13	3	1	3	2

* the answers were recorded on a Likert-type scale with five categories: “1 - no experience” to “5 - very much experience”

Assignments and Tasks

After gathering the demographic data, the participants were introduced to the RoboTrainer and its use. The complete details about this are given in Appendix section D. The task was to navigate the RoboTrainer Prototype (figure 3.2) along the predefined paths shown in figure 7.1. For the evaluation, five segments shown in figure 7.1 were defined. The users navigated RoboTrainer forward on all segments except on segment three. Table 7.3 presents the evaluation tasks in the execution order with the corresponding description and RoboTrainer’s setup. Throughout this section, the tasks are marked with suffix “T” and parameterization steps with suffix “P” ahead of their IDs (cf., table 7.3 and following tables). The evaluation was done on two consecutive days and it took approximately 30 min per participant.

Measurements

The subjective evaluation was done using questionnaires after each task, after parameterization, and at the end of the evaluation session. Objective measurements gathered with RoboTrainer are: the user’s interaction force, RoboTrainer’s velocity, and deviation from the predefined paths.

After each exercise and parameterization step, the participants answered following two questions:

- “How complex was the task for you to solve?” *(Complexity)*[1];

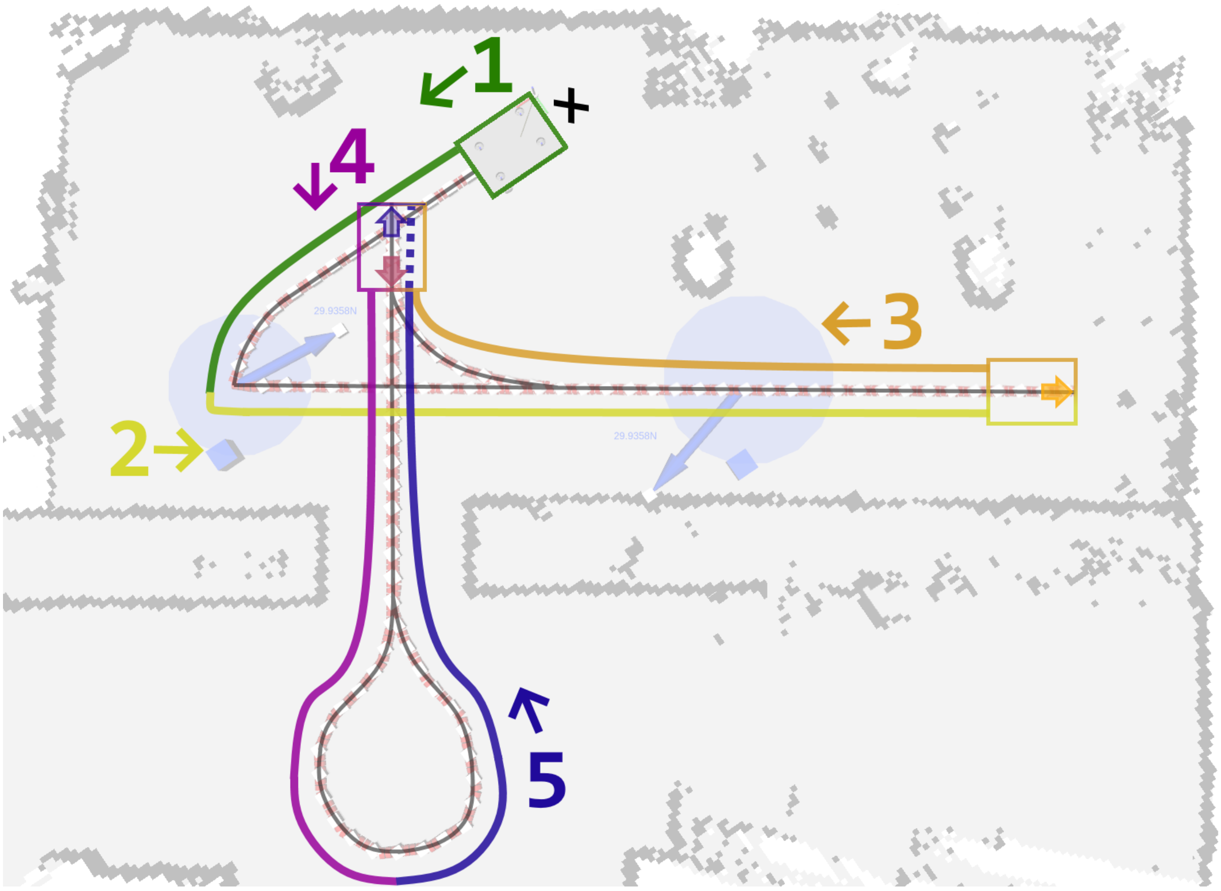


Figure 7.1.: Parkours used in the evaluation of adaptive and individualized control.

- "How well did you solve the task?" *(Self-Assessment)*[1].

The participants' answers were recorded using a Likert-type scale with five response categories ranging from "1 - very complex" to "5 - very simple" and "1 - very bad" to "5 - very good", respectively.

Besides the questions, the participants rated their level of agreement with the following statements:

1. "The task was very interesting." *(Interest)*[1];
2. "I had to make significant effort to solve the task." *(Effort)*[1];
3. "I felt safe executing the task." *(Safety)*[1].

The answers were, again, recorded on a Likert-type scale with the five categories: (1) strongly disagree, (2) disagree, (3) neutral, (4) agree, and (5) strongly agree. Those three statements are omitted for the parameterization P1 since there was nothing to "solve" in terms of a task.

To reduce the influence of growing experience in interaction with RoboTrainer, the order of tasks T2 and T3, and T4 and T5 is blindly randomized over participants, i.e., some did T2 before T3, and some vice versa.

The effects of the changed center of rotation (CoR) were measured after a participant executed two consecutive repetitions of parkour "4-5", where a 180° turn along a predefined radius is done. The first repetition was with the CoR in the middle of the RoboTrainer and the second one with moved CoR away from the user and more to the front of the RoboTrainer Prototype. After the second repetition, the participants were asked the following question:

1. "How complex was the control of RoboTrainer?" *(Complexity)*[1]

and if they agree with the following statements:

1. "The RoboTrainer was easier to steer."
2. "I felt safe during the task." *(Safety)*[1].

The answers were recorded on a Likert-type scale with values, as explained in the paragraphs above. For details on the RoboTrainer's configuration during this task (T8), see table 7.3.

After finalizing all tasks, the users were asked to agree on ten statements from table 7.9 to evaluate the training and RoboTrainer in general. The results are recorded on a Likert-type scale with five categories: (1) strongly disagrees, (2) disagree, (3) neutral, (4) agree, and (5) strongly agrees.

7.1.2. Results and Discussion

The complete results of the evaluation are presented in tables and figures hereunder. Tables 7.4 to 7.9 show the participants' questionnaire answers and figures 7.2 to 7.4 show data measured with RoboTrainer. In both cases, the results are presented in two forms. The results from questionnaires are shown as a number of answers in each category on a Likert-type scale, e.g., table 7.4, and, additionally, as median, minimal, and maximal values of those answers, e.g., table 7.5. Those two representations provide a different level of detail and insight. The first representation shows how data are scattered on the scale and the second provides statistical information. The same approach is made for some data presented in figures, where a scattered graph, e.g., figure 7.3, and a boxplot representation, e.g., figure 7.2, are combined. The samples in the boxplots are represented

[1] A term in parentheses is a short reference used in the evaluation tables (table 7.4 to table 7.9).

as follows: 50 % of the data are inside the box, whiskers are 1.5 times the inner-quartile range or indicate minimum/maximum values, and outliers are represented with a red cross. The red line and the area inside the blue rectangle represent the mean and standard deviation of measurements.

The participants' answers for evaluating the parameterization processes (table 7.4) and the tasks (table 7.6) are compared using the statistical test *Mann-Whitney U-test* and the *T-Test*. The reason for this is that there are small samples in which false positives are possible because only one significance test is used. The tests assessed the two-tailed hypothesis with a significance level of $p < 0.05$.

The first task (T1) is done as a warm-up task to allow users to "feel" the RoboTrainer's control. For many users, this task was the first contact with RoboTrainer ever. Therefore, it provides information about the users' first impressions and could be considered the baseline for evaluating further tasks. No user found the first task complex, felt unsafe, or thought their performance was very bad. Nevertheless, one participant, a first-time user, found the task "not at all" interesting. Otherwise, there is no significant difference between the responses of first-time users and participant's , who already used RoboTrainer before this evaluation.

The participants rated the complexity of parameterization processes "simple" (P1) and "very simple" (P2) (tables 7.4 and 7.5). None of the participants found the parameterizations "very complex", and only one participant rated P1 as "complex". The *Self-Assessment* results are similar. The participants rated P2 simpler than P1, probably because of the P2's similarity to a normal task and growing participants' experience during the evaluation. The users found P2 medium interesting and reported that it was rather easy to do it and that they felt safe during the process. The participants' individual forces during P1 were $F_x = 118,51 \pm 50,00$ N, $F_y = 85,76 \pm 27,36$ N, and $T_z = 37,22 \pm 13,19$ N m. The results are shown graphically in the boxplot[1] form in figure 7.2. Since there is a large spread, especially in the force values, figure 7.3 shows all the measurements. Looking at the data, especially for linear forward force (x-axis), it can be seen that the participants grouped into two clusters, around 80 N and 170 N. The reason for this effect is unknown. There is no obvious correlation between the experience and maximal force, but there is a positive correlation between forward and sideward forces $C_{F_x,F_y} = 0.71$. For the torque value, there is no correlation to neither of the forces. The torque values are between 20 N m to 40 N m, except for the values for the last two participants, which are around 70 N m. The reason for these measurements is also unknown, especially since the force values were not significantly different from other participants (figure 7.3). Figure 7.4 shows the average distance between the participants and RoboTrainer for forward $d_{fw} = 89.06 \pm 5.87$ cm and backward $d_{bw} = 88.88 \pm 4.29$ cm walking during the parameterization P2. Except for the few outliers, the participants' distances were within the

[1] Appendix section C.1 provides a detailed explanation of the boxplot representation.

15 cm range, without a significant difference between forward and backward movement. These results regarding forces and torques during parameterization provide insight into the interaction between the users and RoboTrainer. Therefore, those values provide a good estimate for the default controller's parameters.

Comparing the effects of the maximal force parameterization, i.e., `T2` and `T3`, there is a slight tendency for `T3` to be easier in terms of *Complexity* and *Effort* for the participants (cf. tables 7.6 and 7.7). The participants estimated their performance significantly better ($t(21) = -2,15, p = .019$) when using individual maximal force (`T3` - $M = 4.23$, $SD = 0.69$) compared to baseline parameters (`T4` - $M = 3,63$, $SD = 1,09$), when tested with *T-Test*. On the other hand, *Mann-Whitney U-test* shows no significance (Mann–Whitney $U = 167, n_{1,2} = 22, p = .08$, two-tailed). This tendency to significant difference is also confirmed when asking the participants to compare the repetitions directly. To the question "Which repetition was easier for you to finish?" nineteen (19) participants answered `T3`, and three (3) from 22 answered `T2`.

The influence of individual velocity adaption shows insignificant effects on the user's answers. Both tasks `T4` and `T5`, are rated almost equally across all categories (tables 7.6 and 7.7). During those tasks, the *Virtual Force* spatial control action (SCA) was activated. Two virtual forces are added to the path, represented by a blue circle and an arrow in figure 7.1. To measure subjective effects on a user, we asked the participants how many forces they "felt" on the path. The participants could choose a number of forces between 0 to 3, and in both cases, the answer was 1 (median). This could be because the participants did not feel the virtual force in the curve since its strength is reduced because of a low velocity. To directly compare the controller with fixed parameters and non-linear adaption, the participants were again asked to choose the repetition where the control felt better. 20 from 22 participants chose the repetition with non-linear velocity adaption, and 2 of them chose a controller with fixed parameters.

Tasks `T6` and `T7` were both done using baseline parameters, with a purpose to investigate the influence of sideward movements and inverted controls on the user's performance. It is expected that those types of tasks are experienced as more complex and that the results will have the same tendency as those from the pilot study with persons with mild cognitive impairment (MCI) (section 3.3). Table 7.8 shows the *T-test* and *Mann-Whitney U-test* results of significant differences in user evaluation between `T2` and `T6`, and `T2` and `T7`, respectively. The results are shown in the APA Style [8, 60]. The hypothesis for the variables are expected to change from baseline as follows:

(1) *Complexity* raises (value falls).

(2) The user's *self-assessment* falls because of the more complex task (value falls).

(3) The second task is more *interesting* (value raises).

(4) The participant should provide more *effort* (value raises).

(5) They should feel less *safe* because the control is changed (value falls).

The participants confirmed that T6 and T7 are more *complex* than T2. The difference between T2, i.e., baseline, and T6 is significant for the *T-test*, but not for the *U Test* and therefore this result only represents a tendency. For T7, the significance is clear, i.e., the task with inverted controls is experienced as more complex. The users evaluated their performance as worse for task T7 compared to baseline, whereas task T6 only shows the tendency without significant difference. The participants found T6 significantly less *interesting* and T7 tends to be more interesting than baseline T2. This could be because only one change, i.e., sideways control, was not challenging enough to engage the participants after gathering some experience with the RoboTrainer. The users tend to use more effort to complete T6 and significantly more effort to complete T7. Regarding safety, there is almost no measurable difference between T2 and T6, whereas the participants felt significantly less safe when controlling the RoboTrainer with inverted controls in T7.

The steering of the RoboTrainer with the changed center of rotation (CoR) was easy and the participants preferred it compared to CoR in the center of the device (T8). The participants also felt rather safe during this task (table 7.6).

Observing the results for all tasks from table 7.6, the users were increasingly used to RoboTrainer during the evaluation, confirming the answers regarding parkour's *complexity* and participants' *effort*. Also, the participants tend to be more confident in their performance and tend to feel *safer*. Overall, the participants felt safe during all exercises, with only one participant reporting to feel unsafe in one exercise (T2 – table 7.6). There is a tendency for participants to feel less safe in more complex exercises (e.g., T7). The participants' *interest* fell when a task was repeated on the same parkour without significant changes in RoboTrainer's control, cf. tasks T2 to T5. Therefore, more extended training with healthy adult users between 20 to 40 years should be versatile.

Table 7.9 shows participants' answers to final statements given immediately after finishing all the tasks with RoboTrainer. The users rated the use of RoboTrainer as interesting and stated that they could use it without any problems. Two users stated they felt unsafe when using RoboTrainer, but there is a strong tendency towards a "safe" feeling when using it (statement nr. 10), and no participant believed that they would get hurt when using RoboTrainer (statement nr. 5). Therefore, statement nr. 2 probably does not provide the representative result as its cumbersome definition with double negation could confuse some participants. As expected, the participants rated the RoboTrainer Prototype's design as moderately suitable for this application. Only one participant reported that additional help was needed during the training while most of the participants did not need any help and they had sufficient time to get used to RoboTrainer. Also, the majority of the participants (17) found the change of the tasks as appropriate. Some tasks' complexity is rated as average, but many users (7) experienced them as relatively simple.

Table 7.3.: Ordered list of tasks and parameterization processes used in the evaluation with 22 participants. The tasks have prefix T and parameterization prefix P. During the evaluation, tasks T2 and T3, as well as T4 and T5, were done in randomized order between the users to reduce the impact of the experience on the results. The columns describe the configuration of RoboTrainer's controller for each task.

ID	Name	Parkour	# repetitions	virtual forces	max-force parameters	velocity adaption	CoR shift
T1	Line	"1"	2	no	default	none	default
P1	User's maximal force parameterization						
T2	Parcours baseline	"1-2-3"	1	no	default	none	default
T3	Parcours max-force	"1-2-3"	1	no	user	none	default
P2	Adaptive velocity parameterization						
T4	Parcours fixed	"1-2-3"	1	yes	user	none	default
T5	Parcours non-linear	"1-2-3"	1	yes	user	non-linear	default
T6	Sidewards	"1"	2	no	default	none	default
T7	Sidewards Inverted Y	"1-2-3"	2	no	default	none	default
T8	CoG default/front	"4-5"	1	no	default	none	front

Table 7.4.: User's answers regarding the parameterization processes for each evaluation category. The numbers represent the absolute number of answers for each category.

Parameterization	Complexity†					Self-Assessment†					Interest*					Effort*					Safety*				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
P1 - Maximal-Force	0	1	5	9	7	0	4	3	6	9			—					—					—		
P2 - non-linear Velocity	0	0	2	7	13	0	1	3	3	15	0	4	8	6	4	10	8	2	1	1	0	0	1	3	18

† Likert-type scale with five response categories ranging from "1 - very complex" to "5 - very simple" and "1 - very bad" to "5 - very good", respectively.

* Likert-type scale with five categories: (1) strongly disagrees, (2) disagree, (3) neutral, (4) agree, (5) strongly agree.

Table 7.5.: Median and range of user's answers regarding the parameterization processes for each evaluation category.

Parameterization	**Complexity**[†]		**Self-Assessment**[†]		**Interest**[*]		**Effort**[*]		**Safety**[*]	
	Median	Range	Median	Range	Median	Range	Median	Range	Median	Range
P1 - Maximal-Force	4	2 – 5	4	2 – 5	—		—		—	
P2 - non-linear Velocity	5	3 – 5	5	2 – 5	3	2 – 5	2	1 – 5	5	3 – 5

[†] Likert-type scale with five response categories ranging from "1 - very complex" to "5 - very simple" and "1 - very bad" to "5 - very good", respectively.

[*] Likert-type scale with five categories: (1) strongly disagrees, (2) disagree, (3) neutral, (4) agree, (5) strongly agree.

Table 7.6.: User's answers regarding the tasks for each evaluation category. The numbers represent the absolute number of answers.

ID	**Name**	**Complexity**[†]					**Self-Assessment**[†]					**Interest**[*]					**Effort**[*]					**Safety**[*]				
		1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
T1	Line	0	1	4	6	11	0	1	8	7	6	1	5	6	5	5	9	7	5	1	0	0	3	4	5	10
T1	Line (1st time users)	0	0	3	5	5	0	1	6	3	3	1	1	4	4	3	3	5	4	1	0	0	2	2	3	6
T2	Parcours baseline	0	3	6	8	5	1	2	6	8	5	0	0	2	6	14	3	4	8	4	3	1	1	0	5	15
T3	Parcours max-force	0	0	5	9	8	0	0	3	11	8	0	0	2	7	13	6	4	9	1	2	0	0	1	5	16
T4	Parcours fixed	0	1	4	9	8	0	0	5	11	6	0	2	3	11	6	6	7	5	3	1	0	0	1	5	16
T5	Parcours non-linear	0	0	3	8	11	0	0	4	8	10	0	2	4	8	8	5	11	2	3	1	0	0	1	3	18
T6	Sidewards	0	6	7	7	2	0	4	9	5	4	0	2	3	10	7	2	4	9	4	3	0	1	3	5	13
T7	Inverted Y	4	7	8	3	0	1	6	9	5	1	0	0	1	4	17	0	3	6	7	6	0	5	3	5	9
												RoboTrainer is easier to steer.														
T8	CoG front	0	0	2	9	11						2	3	3	5	9						0	0	2	3	17

[†] Likert-type scale with five response categories ranging from "1 - very complex" to "5 - very simple" and "1 - very bad" to "5 - very good", respectively.

[*] Likert-type scale with five categories: (1) strongly disagrees, (2) disagree, (3) neutral, (4) agree, (5) strongly agree.

Table 7.7.: Median and range of user's answers regarding the tasks for each evaluation category.

ID	Name	Complexity[†]		Self-Assessment[†]		Interest[*]		Effort[*]		Safety[*]	
		Median	Range	Median	Range	Median	Range	Median	Range	Median	Range
T1	Line	4,5	2 – 5	4	2 – 5	3	1 – 5	2	1 – 4	4	2 – 5
T1	Line (1st time users)	4	3 – 5	3	2 – 5	4	1 – 5	2	1 – 4	4	2 – 5
T2	Parcours baseline	4	2 – 5	4	1 – 5	5	3 – 5	3	1 – 5	5	1 – 5
T3	Parcours max-force	4	3 – 5	4	3 – 5	5	3 – 5	3	1 – 5	5	3 – 5
T4	Parcours fixed	4	2 – 5	4	3 – 5	4	2 – 5	2	1 – 5	5	3 – 5
T5	Parcours non-linear	4,5	3 – 5	4	3 – 5	4	2 – 5	2	1 – 5	5	3 – 5
T6	Sidewards	3	2 – 5	3	2 – 5	4	2 – 5	3	1 – 5	5	2 – 5
T7	Inverted Y	2,5	1 – 4	3	1 – 5	5	3 – 5	4	2 – 5	4	2 – 5
						RoboTrainer is easier to steer.					
T8	CoG front	4,5	3 – 5			4	1 – 5			5	3 – 5

[†] Likert-type scale with five response categories ranging from "1 - very complex" to "5 - very simple" and "1 - very bad" to "5 - very good", respectively.

[*] Likert-type scale with five categories: (1) strongly disagrees, (2) disagree, (3) neutral, (4) agree, (5) strongly agree.

Table 7.8.: Significant differences ($p^* < .05$) in the participant's answers between baseline task `T2` and tasks `T6` and `T7`. The measured statistics are ***T-test*** and ***Mann-Whitney U-test***. Only significant results are shown. In a case where the test results differ, the p values are shown in a standard font.

ID	Complexity		Self-Assessment		Interest		Effort		Safety	
	T-test	U-test	T-test	U-test	T-test	U-test	T-test	U-test	T-test	U-test
`T2`	$M = 3.82$ $SD = 0.8$	–	$M = 3.86$ $SD = 0.94$	–	$M = 4.59$ $SD = 0.59$	–	$M = 2.95$ $SD = 1.13$	–	$M = 4.55$ $SD = 0.74$	–
`T6`	$M = 3.23$ $SD = 1.0$	$U = 158.5$ $n_{1,2} = 22$	$M = 3.43$ $SD = 1.03$	–	$M = 4.05$ $SD = 0.92$	$U = 151.5$ $n_{1,2} = 22$	$M = 3.10$ $SD = 1.18$	–	$M = 4.43$ $SD = 0.87$	–
	$T(21) =$ 2.21 p* = .0329	p = .512	–	–	$T(21) =$ 2.52 $p^* = .0155$	$p^* = .035$	–	–	–	–
`T7`	$M = 2.55$ $SD = 1.1$	$U = 73$ $n_{1,2} = 22$	$M = 2.95$ $SD = 0.97$	$U = 123$ $n_{1,2} = 22$	$M = 4.81$ $SD = 0.40$	–	$M = 3.76$ $SD = 1.04$	$U = 152.5$ $n_{1,2} = 22$	$M = 3.85$ $SD = 1.24$	$U = 163$ $n_{1,2} = 22$
	$T(21) =$ $5.12,$ $p^* < .001$	$p^* < .001$	$T(21) =$ 3.19 $p^* = .003$	$p^* = .005$	–	–	$T(21) =$ -2.36 $p^* = .023$	$p^* = .036$	$T(21) =$ 2.39 p* = .021	p = .066

Table 7.9.: User's answers to general questions asked at the end of the evaluation session. The numbers represent the absolute number of answers and median values.

Nr.	Question	Grade					Median
		1	2	3	4	5	
1.	It was interesting to use RoboTrainer.	0	0	2	4	16	5
2.	I could use RoboTrainer without any problems.	0	0	5	6	11	4,5
3.	I didn't felt unsafe when interacting with RoboTrainer.	2	4	3	6	7	4
4.	Design of RoboTrainer is optimal for me.	0	5	9	5	3	3
5.	During the training I was scared to get hurt.	17	3	2	0	0	1
6.	I needed additional help during the Training.	14	4	3	0	1	1
7.	I didn't have enough time to get used to RoboTrainer.	12	7	3	0	0	1
8.	The training was very fast. The task were switched too often.	17	4	0	1	0	1
9.	Some tasks were hard to solve.	7	3	4	6	2	3
10.	I felt safe when using RoboTrainer.	0	0	2	7	13	5

*the answers were recorded on a Likert-type scale with five categories:

(1) strongly disagrees, (2) disagree, (3) neutral, (4) agree, (5) strongly agrees.

7.1.3. Findings from the Evaluation

This section summarizes the results from the evaluation of adaptive and individualized control and provides a short outlook.

The participants well accepted the proposed parameterization processes and the use of automatically determined individual parameters was preferred by them (19 of 22). The users' forces are grouped around two values, 80 N and 170 N, with the most users around the lower value. Therefore, a choice of 100 N as the default maximal users' force was confirmed. There is no significant change between participants regarding distance to RoboTrainer. So it seems that the device's mechanical construction mainly influences the distance between the user and the device. The influence of individual velocity adaption shows insignificant differences, nevertheless, the participants preferred (20 of 22) individual velocity adaption limits when asked directly. Similarly, the participants preferred (13 of 22) the shift of the center of rotation (CoR) to the edge of RoboTrainer, i.e., further away from them than the CoR in the middle of the device.

The spatial control actions (SCAs) show that participants prefer more challenging tasks after some interaction with RoboTrainer. The participants also reported feeling safe during tasks even though the device's control changed unintuitively. The users value the challenging tasks more than the impact of counter-intuitive control on their safety feeling.

Overall, the training with RoboTrainer Prototype was rated as interesting, and all users could finish it without any problems. The participants would prefer even more challenging tasks and a different design of the device. The latter issue is addressed with the design of RoboTrainer v2.

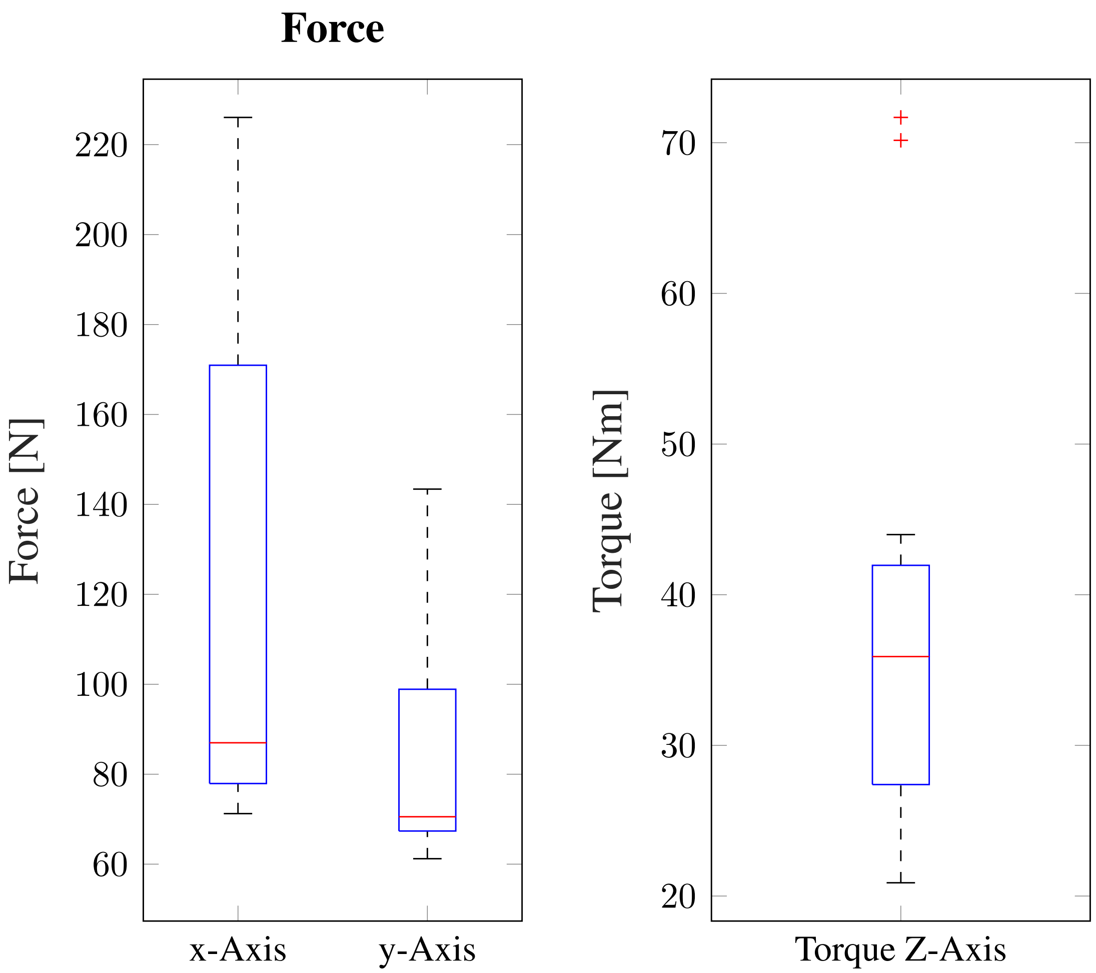


Figure 7.2.: Measured user's force and torque during parameterization of "maximal user's force" (P1). The force values (left graph) tend to be higher than the average. There are two groups of values around 90 N and 180 N. The dispersion of torque data is relatively uniform, except for two outliers around 70 N m. For details, see figure 7.3.

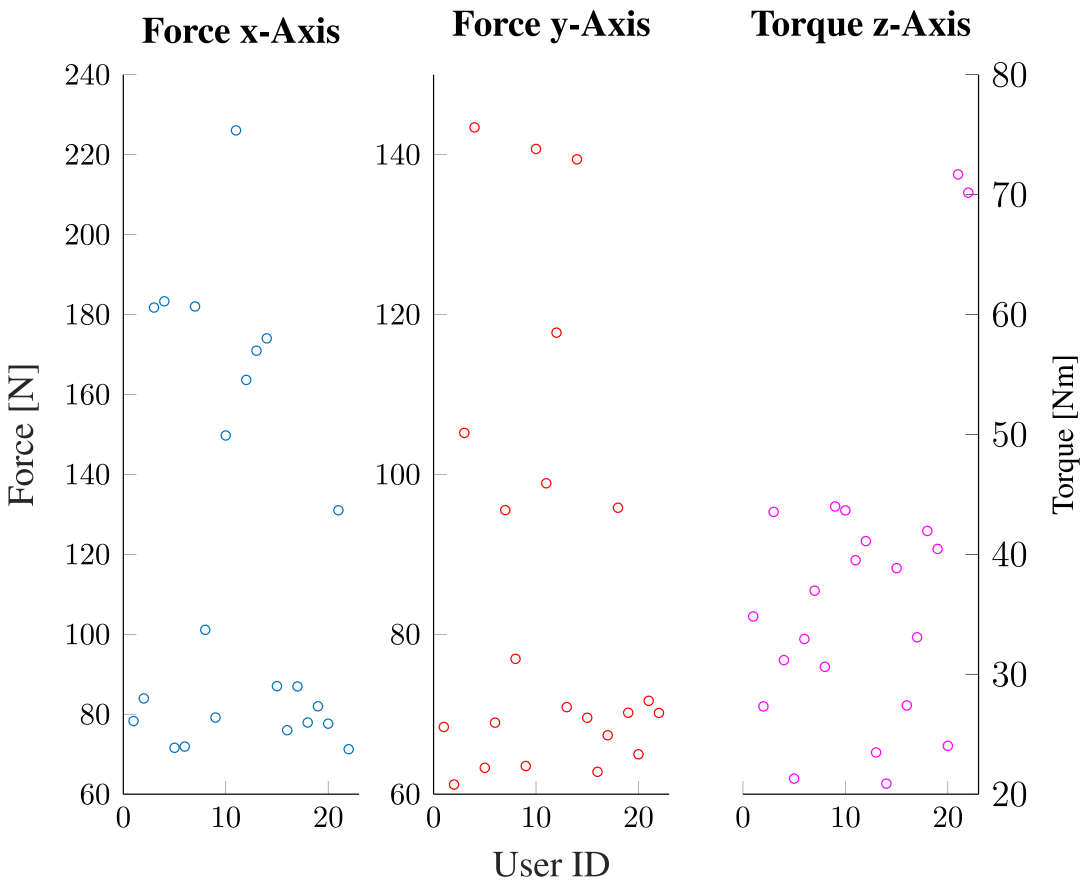


Figure 7.3.: Measured user's force and torque during parameterization of "maximal user's force" (P1). The data show a grouping that causes a very high variance of x-Axis force in figure 7.2.

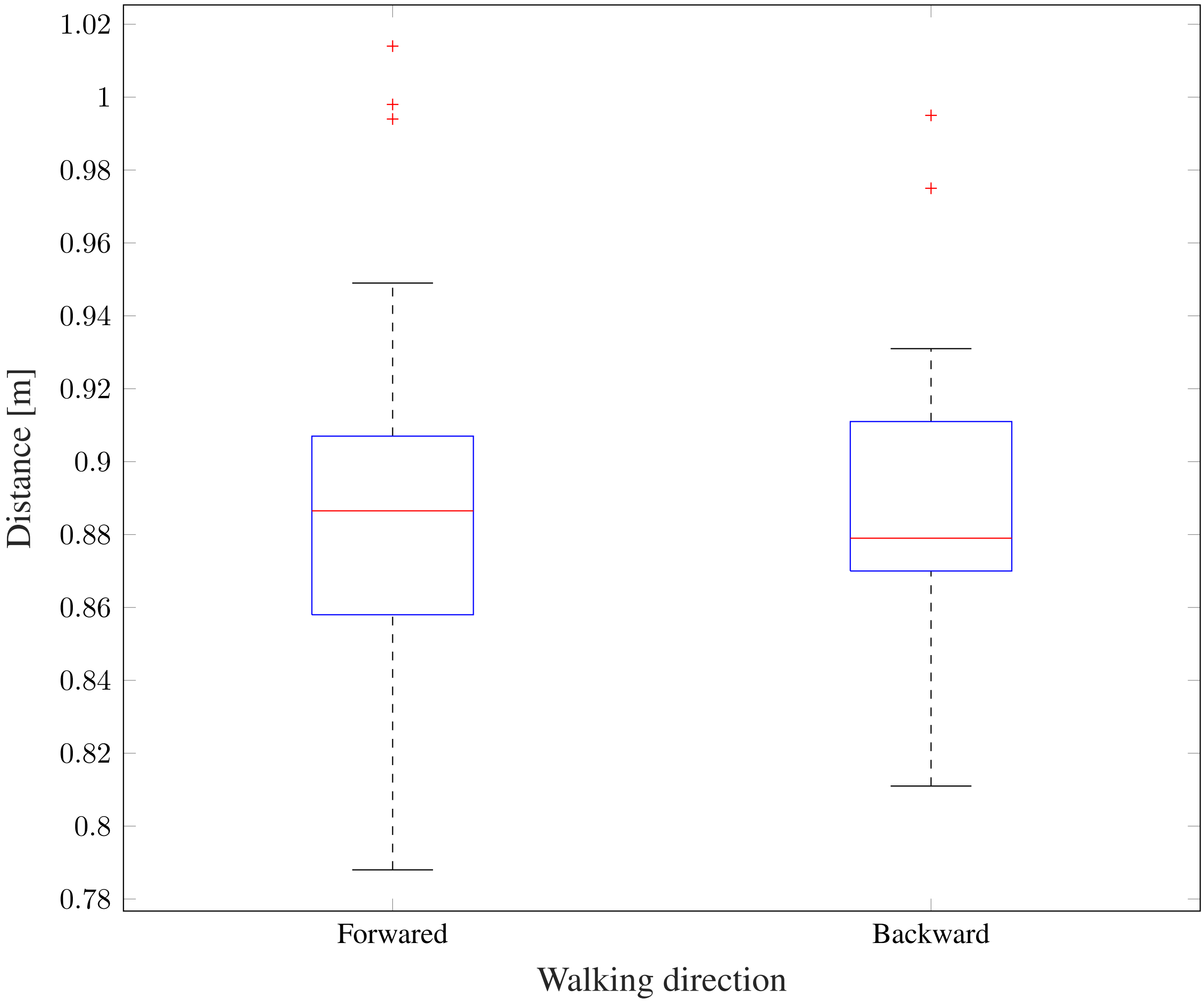


Figure 7.4.: Parameterization of velocity adoption: Distance between the users' legs and RoboTrainer.

7.2. Evaluation of the RoboTrainer v2

This section presents the evaluation of the RoboTrainer v2 in a study with 13 participants. The study was conducted in the scope of the project *Learning Robotic-Assistance Systems for Neuro-Muscular Training (RoSylerNT)* at German Sport University Cologne (DSHS). The goal was to investigate how users' interaction forces change by increasing interaction experience and how RoboTrainer and spatial control action (SCA) influence the users' movement and brain activity. Regarding RoboTrainer v2's design, the participants gave feedback on two extreme device footprint positions. From the device's control perspective, the state-of-the-art velocity-adaption algorithm from Yu, Spenko, and Dubowsky [185] (equation (2.8)) and non-linear adaption from this thesis are compared. This section assesses data gathered by the RoboTrainer and users' questionnaires.

7.2.1. Evaluation Setup

Variables

The study's primary goal is to evaluate users' interaction with the novel device, RoboTrainer v2, and compare the velocity-based adaption to the state-of-the-art approach. The study should provide insights into how persons with non-technical backgrounds perceive and interact with the device. The following variables were investigated using questionnaires and interaction data:

(1) subjective intuitiveness and safety of the interaction;

(2) changes of interaction forces by increasing interaction-experience;

(3) influence of the global control actions (GCAs) and spatial control actions (SCAs) on user and controllability;

(4) influence of different adaptive control strategy on users' experience and performance;

(5) influence of RoboTrainer v2 wheel setup on users.

The hypothesis for the variables are the following:

(I) the users find overall interaction with the RoboTrainer v2 intuitive and feel safe during the whole time;

(II) the users adapt to the RoboTrainer's controller and their interaction forces converge over time;

(III) the GCAs' and SCAs' influence is evident in subjective and objective results;

Table 7.10.: Demographics of participants ($n = 13$) in the evaluation of RoboTrainer v2.

Gender		Age			Occupation	
Female	Male	Mean	min	max	Student or Researcher	External
8	5	29.08 ± 5.574	20	38	11	2

Table 7.11.: Experience of participants ($n = 13$) in the evaluation of RoboTrainer v2.

Conventional Walker					RoboTrainer				
1	2	3	4	5	1	2	3	4	5
9	2	2	0	0	12	1	0	0	0

* the answers were recorded on a Likert-type scale with five categories: "1 - no experience" to "5 - very much experience"

(IV) the non-linear adaptive control strategy gives better results compared to linear velocity adaption from Yu, Spenko, and Dubowsky [185];

(V) the users feel more comfortable using opened-wheels setup.

Group

The evaluation group consisted of 13 participants between 20 and 38 years of age. The participants were recruited by the German Sport University Cologne (DSHS). 11 participants were students or researchers in the field of sport's sciences and two were external participants. The only inclusion criteria were non-existent issues or diseases of the musculoskeletal apparatus and overall physical fitness. To estimate the participants' experience, they were asked about their experience with a conventional walker ahead of the evaluation. The answers were recorded using a Likert-type scale with five response categories ranging from "1 - no experience" to "5 - very much experience".

Tables 7.10 and 7.11 give an overview of participants' demographic data and experience self-assessment.

Assignments and Tasks

Before the experiments with RoboTrainer, the participants were given general instructions about interaction with the device. The instructions were exactly the same as in the previous evaluation (section 7.1). The task was to navigate the RoboTrainer v2 (figure 4.1) along the marked paths shown in figure 7.5. Figure 7.6 shows the virtual representation of the training environment with the training paths and spatial control actions.

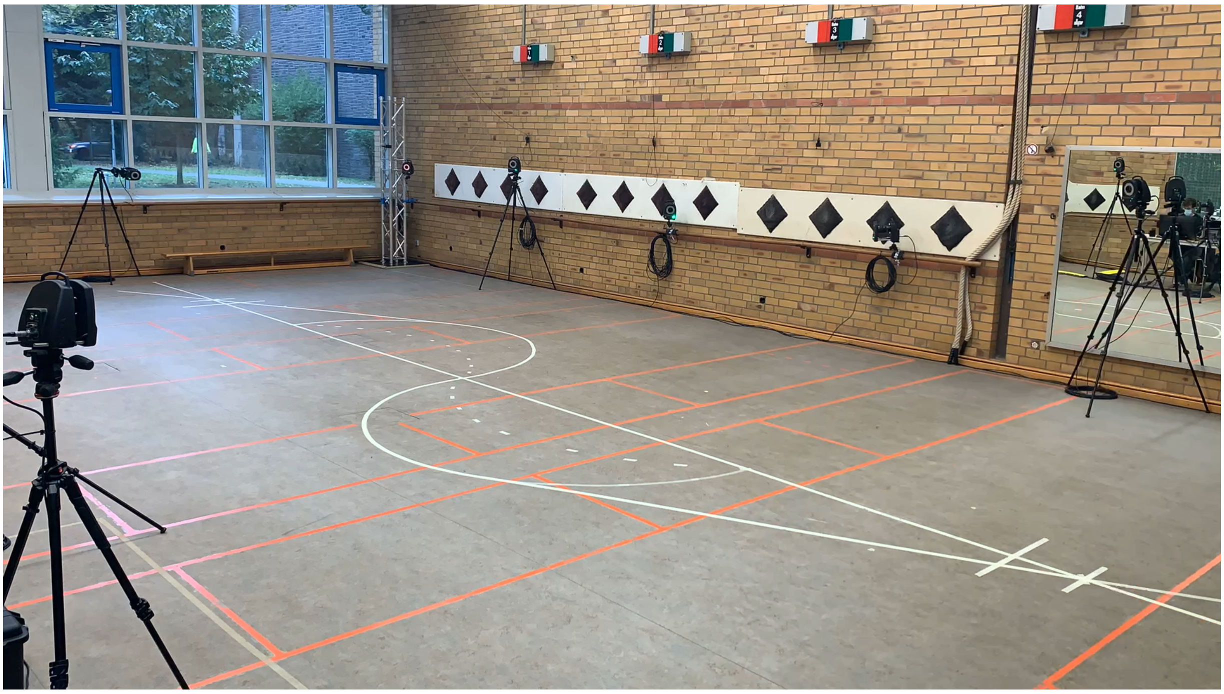


Figure 7.5.: Evaluation environment at German Sport University Cologne in the project Learning Robotic-Assistance Systems for Neuro-Muscular Training. The two paths, "line" and "curve", are marked with white tape. In the right lower corner, cross lines mark the starting position for the line. Author: Björn Braunstein, October 16, 2020

Table 7.12 shows the list of all the tasks and parameterization processes in their execution order. For each task, the RoboTrainer's setup is given. The tasks had a different amount of repetition depending on the measurements conducted during each. Task `T2` was repeated 30 times with eight *SCA–Force Area* configurations shown in figure 7.6. The configurations were set in randomized order so that participants could not anticipate when the disturbance happens. The disturbances had the same strength and direction, i.e., always to the left from the moving direction, but their placement on the path varied. Throughout this section, the tasks are marked with suffix "`T`" and parameterization steps with suffix "`P`" ahead of their IDs (cf., table 7.12 and the following tables). The evaluation was done on four consecutive days, October 13.-16. 2020. The experiment with RoboTrainer took between 60 and 90 minutes per participant, and the whole experiment with preparation between 120 and 180 minutes.

Before the tasks, the participants were instructed to follow the predefined lines on the floor with the RoboTrainer's front laser-marker. For the parameterization steps, they were instructed to push the RoboTrainer v2 until they feel confident and hold it in this position.

Table 7.12.: Ordered list of tasks and parameterization processes used in the RoboTrainer v2's evaluation with 13 participants. The tasks have prefix `T` and parameterization prefix `P`. During the evaluation, tasks `T4-DA` and `T4-NL` were randomized between the users to reduce the impact of the experience on the results. The columns describe the RoboTrainer's configuration for each task. The Violet tasks are done by the last three participants to investigate the influence of the control actions' parameters on users' movement and brain activity. Task `T2-2X` is done instead of `T2`, and `T3-C2` is done additionally to the other tasks.

ID	Name	Path	# repetitions	controller★	FA◊	IC♦	HW♠
`P1`	User's maximal force parameterization						5-1
`T1`	Line baseline	Line	6	fixed	no	no	5-1
`T2`	Line with force	Line	30	fixed	yes	no	5-1
`T2-2X`	Line with force (2x strength)	Line	30	fixed	yes	no	5-1
`P2`	User's maximal force parameterization						5-1
`T4-DA`	Curve – algorithm 1	Curve	6	damping linear	no	no	5-1
`T4-NL`	Curve – algorithm 2	Curve	6	non-linear	no	no	5-1
`T3-C`	Curve – inverted	Curve	4	fixed	no	yes	5-1
`T3-C2`	Curve – inverted (2x max. vel.)	Curve	4	fixed	no	yes	5-1
`P3`	User's maximal force parameterization						5-1
`T5`	Curve narrow wheels	Curve	2	fixed	no	no	1-4

★ *fixed* – admittance rule with fixed parameters;

★ *non-linear* – velocity-based non-linear parameter adaption from this thesis;

★ *damping linear* – velocity-based linear damping adaption from Yu, Spenko, and Dubowsky [185]

◊ *SCA–Force Area*

♦ *GCA–Inverted Controls*

♠ Setup of the RoboTrainer's footprint: *5-1* – "open-short" setup (figure 4.4c); *1-4* – "closed-long" setup (figure 4.4b)

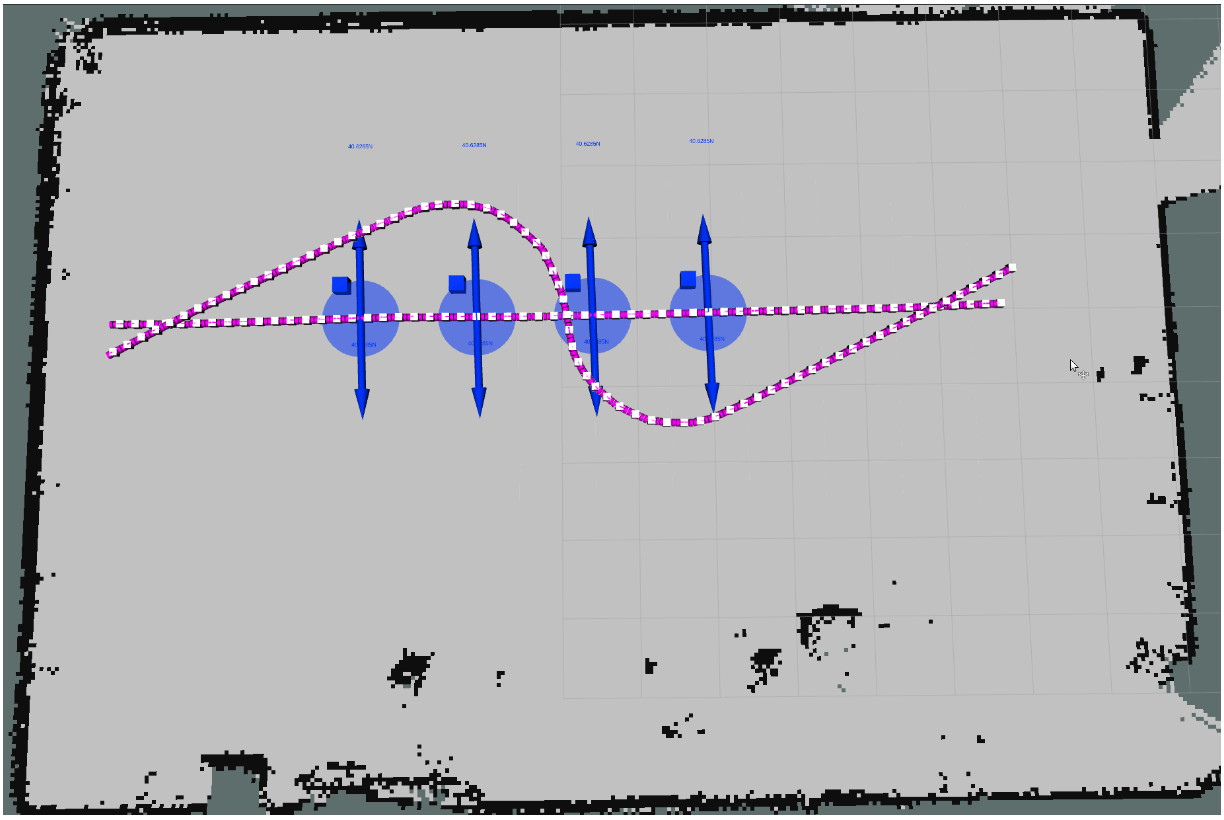


Figure 7.6.: Virtual representation of the Evaluation environment at German Sport University Cologne in the project Learning Robotic-Assistance Systems for Neuro-Muscular Training. Two paths, “line” and “curve”, and used disturbance forces are shown. During each trial with the *SCA-Force Area*, one of the forces was active, always showing to the left from the participants’ perspective.

Measurements

For the objective evaluation of users’ interaction with the RoboTrainer and their performance, the following data were recorded during the study:

1. average input (interaction) force;
2. parameterization forces;
3. average RoboTrainer’s velocity;
4. average deviation from the predefined path;
5. average trial duration;
6. average user’s distance to the RoboTrainer; and

7. average disturbance force of the SCAs.

The subjective evaluation is done using questionnaires after each parameterization and task. The participants had to answer a set of questions and state the level of agreement with predefined statements. Table 7.13 lists all the questions and maps them to corresponding tasks and parameterizations. A one-word reference is defined in the evaluation tables in the next section for each question and statement. The answers are recorded using a Likert-type scale with five response categories.

After task T2, three additional questions were asked to get feedback about the *SCA–Force Area*. The questions and possible answers were the following:

1. "The forces were too strong." ("1 - strongly disagrees" to "5 - strongly agree");
2. "How many forces did You feel?" (multiple choice 0 to 5);
3. "In which direction did You feel the forces?" ("left", "right", "forward", "backward").

The sequence of tasks T4-DA and T4-NL were chosen randomly over participants so that seven (7) of them had "linear damping"-algorithm as the first one. Directly after the tasks, only three questions were asked, indicating the influence of different adaption algorithms. The remaining questions were about both tasks and they are marked with T4 in table 7.13. To compare two adaption algorithms, the following question immediately after task T4-NL was asked:

> "Did you experience the difference between the algorithms? With which one You had better control?"
>
> - "The control felt the same."
> - "The first algorithm was better."
> - "The second algorithm was better."

7.2.2. Results and Discussion

The results' presentation of RoboTrainer v2's evaluation study is structured with respect to variables and hypotheses defined above. The questionnaire results are shown in tables 7.21 and 7.22. The data recorded with the RoboTrainer v2 are shown in figures 7.7 to 7.13. The data are represented using boxplots in standard configuration. For a detailed explanation, see appendix section C.1. When showing a comparison of different repetitions, i.e., trials, of a task, two or three samples are chosen depending on the overall number of repetitions (cf. table 7.12). Three samples are used if there were more than six (6) repetitions and otherwise two.

Table 7.13.: List of questions and statements used in the questionnaires for the RoboTrainer v2 evaluation. For each question, a one-word reference is defined and used in the following evaluation. The third column gives the meaning of lowest and highest Liker-type scale categories. The final column provides task IDs where the questions or statements are used. Task $A4$ indicates questions and statements asked after tasks with both algorithms.

Question	Reference	Categories	Task ID
	Parameterization Assessment		
"How intuitive the RoboTrainer's control felt?"	Intuitiveness	"1 - not operable" to "5 - very intuitive"	P1, P2, P3
"How complex was the parameterization process for you?"	Complexity	"1 - very complex" to "5 - very simple"	P1, P2, P3
"How intuitive was the parameterization process in whole?"		"1 - very unintuitive" to "5 - very intuitive"	P1,
"I felt safe executing the task."	Safety	"1 - strongly disagrees" to "5 - strongly agree"	P1, P2, P3
	Task Assessment		
"How complex was the task for you to solve?"	Complexity	"1 - very complex" to "5 - very simple"	T1, T2, T4, T3-C, T5
"How well did you solve the task?"	Self-Assessment	"1 - very bad" to "5 - very good"	T1, T2, T4, T3-C, T5
"How intuitive was the RoboTrainer's controls felt?"	Intuitiveness	"1 - not usable" to "5 - very intuitive"	T1, T2, T4-DA, T4-NL, T3-C, T5
"How did you experience the speed of RoboTrainer?"	Speed	"1 - too slow" to "5 - too fast"	T1,T4-DA, T4-NL, T3-C, T5
"How did you experience the weight of RoboTrainer?'	Weight	"1 - too heavy" to "5 - too light"	T1, T4-DA, T4-NL, T3-C, T5
"The task was very interesting."	Interest	"1 - strongly disagrees" to "5 - strongly agree"	T1, T2, T4, T3-C, T5
"I had to make significant effort to solve the task."	Effort	"1 - strongly disagrees" to "5 - strongly agree"	T1, T2, T3-C, T5
"I felt safe executing the task."	Safety	"1 - strongly disagrees" "5 - strongly agree"	T1, T2, T4-DA, T4-NL, T3-C, T5
"I had to take care not to collide with my legs against the RoboTrainer."	Collision	"1 - strongly disagrees" to "5 - strongly agree"	T1, T4, T3-C, T5

Intuitiveness and Safety of Interaction

The participants described the interaction with the RoboTrainer as intuitive for all tasks (cf. tables 7.21 and 7.22). The exceptions are tasks `T1` and `T3-C`, where the device's control is rated as medium intuitive. One participant rated the interaction in the first task, `T1`, as "very unintuitive". Such a result is plausible considering the first contact with a Smart Walker at all. task `T3-C` is supposed to be less intuitive than other tasks since *GCA-Inverted Controls* is utilized. Observing the absolute number of answers in each category for the first three tasks, `P1`, `T1`, and `T2`, there is a tendency to experience the RoboTrainer's behavior as more intuitive as training advances. This is an expected outcome because users are gaining more experience with RoboTrainer v2.

All the participants felt safe during the whole interaction with RoboTrainer v2. There were only individual answers stating medium "safety" experience. From a technical perspective, the study allowed optimizing the safety configuration. This was especially the case for the laser scanners' safety fields (cf. figure 4.9). On the first evaluation day, one participant often activated the emergency stop. This participant triggered a safety field under the RoboTrainer on the backside. The reason is feet-swing during walking, where for a very short period, the toes are located under the device. This depends on the users' stride. After this observation, the fields were adapted for the other participants to enable smooth evaluation. In the future, it is worth considering to also optimize the safety fields around the device. They are currently configured based on conservative calculation, being rather too large. This configuration did not disturb the data acquisition when participants were following the paths. However, from time to time, participants were stopped when turning the robot at the end of the paths. The reason was the narrow space between the paths' end and camera tripods. This should be taken into account when designing future training environments.

Changes of the Interaction Force During the Training

The users' interaction forces were measured during parameterization processes and all tasks. The parameterization measurements represent a static situation because users and RoboTrainer were not moving within the training area. Those measurements give insight about how users' approach to interact with RoboTrainer has changed. On the other side, the measurements during the tasks expose users' interaction forces in dynamic training situations.

Figure 7.8 shows the maximal force and torque measurements from the parameterization tasks: `P1`, `P2`, and `P3`. The measured average force values are expected, showing that users spend the strongest force in the forward direction and weaker forces sidewards. The torque values cannot be directly compared to the linear forces since they depend

Table 7.14.: Significant differences ($p^* < .05$) in the measured interaction force during parameterization. The significance is measured using *Mann-Whitney U-test* since the measurements do not have a normal distribution. Comparison is shown column-wise.

Task ID	***X***	***Rot_left***
P1	$M = 75.64$; $SD = 32.13$	$M = 24.14$; $SD = 9.39$
P2	$M = 99.90$; $SD = 32.28$ $U = 32$; $n_1 = 13$; $n_2 = 12$; $p^* = .0133$	$M = 34$; $SD = 10.20$ $U = 30$; $n_1 = 13$; $n_2 = 12$; $p^* = .0098$

Table 7.15.: Significant differences ($p^* < .05$) in the measured average input force for the tasks. The measured statistics are *T-test* and *Mann-Whitney U-test* depending on the variables' normal distribution. T3-C does not have normal distribution. Comparison is shown row-wise.

T1	**T2**	**T3-C**
$M = 52.64$; $SD = 12.51$	$M = 61.11$; $SD = 14.06$	$M = 61.04$; $SD = 6.99$
	$T(59) = -2.44$; $p^* = .0176$	$U = 181$; $n_1 = 34$; $n_2 = 20$; $p^* = .0045$

on the handlebar's geometry. The left-right forces, *Y_left* and *Y_right*, and torques for the rotation, *Rot_left* and *Rot_right*, have approximately the same values. The average interaction values are similar to those from the previous evaluation study (cf. figure 7.2). The values stay almost the same over multiple repetitions of the parameterization (cf. figure 7.8a). Significant changes are only seen between tasks P1 and P2 for the *X* and *Rot_left* directions (table 7.14). The data does not confirm the hypothesis that the interaction force of the participants converges during the interaction. This would lead to variances, i.e., whiskers in the boxplots, becoming smaller with each repetition to confirm this (cf. figure 7.8a).

The users' average interaction force, shown in figure 7.7, shows a perpetual increase between different repetitions of the first two tasks, T1 and T2. Nevertheless, those changes only show a tendency since the changes are not significant. The other tasks have approximately the same mean values between the first and the last repetitions (cf. figure 7.7a). Figure 7.7b shows that participants used stronger input forces in the follow-up tasks with the standard controller (cf. table 7.15. The tasks with the controller's adaption, T4-DA and T4-NL, show significantly[2] lower values than all other tasks. This is expected behavior and the reasons will be explained later when comparing the adaptive strategies.

[2] For brevity, the exact values are not shown because this fact is not essential for further discussion.

Table 7.16.: Significant differences ($p^* < .05$) in interaction force for the tasks with *SCA-Force Area* (`T2` and `T2-2X`) of multiple variables compared to the ground-truth (`T1`). The measured statistics are *T-test* and *Mann-Whitney U-test* depending on the variables' normal distribution. Comparison is shown column-wise.

Task ID	Input Force	Velocity	Time
`T1`	$M = 52.64;$ $SD = 12.51$	$M = 0.95; SD = 0.19$	$M = 9.41; SD = 2.86$
`T2`	$M = 61.11;$ $SD = 14.06$	$M = 1.07; SD = 0.18$	$M = 7.94; SD = 1.92$
	$T(59) = -2.44;$ $p^* = .0176$	$T(59) = -2.56;$ $p^* = .0130$	$U = 605; n_1 = 34;$ $n_2 = 27; p^* = .0346$
`T2-2X`	$M = 70.10;$ $SD = 7.21$	$M = 1.18; SD = 0.09$	$M = 7.02; SD = 0.95$
	$T(41) = -3.92;$ $p^* = .0003$	$T(41) = -3.37;$ $p^* = .0017$	$U = 250; n_1 = 34;$ $n_2 = 9; p^* = .0040$

Influence of Global and Spatial Control Actions

The *SCA-Force Area* was utilized in tasks `T2` and `T2-2X`. The difference was that in the first case, a force of 20 N is used, and in the latter 40 N. The *GCA-Inverted Controls* was evaluated in tasks `T3-C` and `T3-C2`, with the difference that the latter case had two times faster dynamics, i.e., the maximal velocity limit for left and right movement was two times higher and set to 1.0 m/s. For details about the influence of those maximal values on RoboTrainer's controller, refer to section 5.1.1.

SCA-Force Area

To evaluate the influence of the *SCA-Force Area*, tasks `T2` and `T2-2X` are compared to the ground-truth task `T1` and to each other. The comparison is made for all the variables shown in figures 7.7 to 7.13. The results for task `T2-2X` should be taken "with a grain of salt" because of a small sample of three participants.

The participants used stronger force in tasks where *SCA-Force Area* was present (cf. table 7.16) than in tasks without it, e.g., `T1`. There is only a tendency for stronger interaction (cf. figure 7.7) when using stronger virtual force. The same is true for the average participants' velocity, which positively correlates with the disturbance force's strength. The average velocity is not significantly different between tasks `T2` and `T2-2X`. Interestingly, the average deviation (figure 7.10) from the "straight line"-path does not show

Table 7.17.: Significant differences ($p^* < .05$) between the repetitions for average deviation measurements. The measured statistics are *T-test* and *Mann-Whitney U-test* depending on the variables' normal distribution. Comparison is shown row-wise. Reference measurements are marked with *Ref.*

Task ID	First	Middle	Last
T1	$M = 0.10$; $SD = 0.04$	$M = 0.07$; $SD = 0.03$	$M = 0.05$; $SD = 0.01$
	$T(20) = 3.39$; $p^* = .0029$	$T(21) = 2.37$; $p^* = .0302$	*Ref.*
T2-2X	$M = 0.08$; $SD = 0$		$M = 0.06$; $SD = 0$
	$T(4) = 4.75$; $p^* = .0090$		*Ref.*

differences between those two tasks. Considering the different repetitions (trials) for each task, there is a clear tendency to navigate the RoboTrainer more precisely towards the end of the task. These differences are statistically measurable for *first–middle* and *first–last* comparisons in task T1 and between *first–middle* comparison in task T2-2X (cf. table 7.17). There are no other significant differences with respect to average deviation.

The participants' time performance (see figure 7.11) also follows the trend mentioned earlier, i.e., they were consecutively faster over tasks T1, T2, and T2-2X. This confirms the observations about the average velocity. Table 7.18 shows the significant differences in trial duration for different repetitions. The average velocity measurements do not show these differences but average deviation values (table 7.17). The observations regarding RoboTrainer's average velocity and time performance should not be entitled to the influence of the *SCA-Force Area*. They are probably happening because of increasing users' experience and confidence in the interaction with the device. Therefore, those outcomes are probably a consequence of the task's order.

Figure 7.13 compares the average disturbance force between different repetitions. The disturbance force had a trapezoidal profile (cf. figure 6.5) and the resulting disturbance velocity was calculated using the steady-state relation of the admittance equation (equation (2.23)). Therefore, the influence of the RoboTrainer's velocity on disturbance was reduced, as shown by comparing the tasks' repetitions. The discrete nature of the controller causes the residual and unavoidable "noise" in disturbance forces when using *SCA-Force Area*. The average values for task T2-2X are approximately two times higher, confirming the proper functionality of this spatial control action.

The users' answers do not differ between tasks T2 and T2-2X because of the small sample in the latter case. Therefore, those data are classified together in tables 7.21 and 7.22. Looking at the overall results (table 7.22), there are no detectable differences

Table 7.18.: Significant differences ($p^* < .05$) between the repetitions for time measurements (duration). The measured statistics are *T-test* and *Mann-Whitney U-test* depending on the variables' normal distribution. Comparison is shown row-wise. Reference measurements are marked with *Ref.*

Task ID	First	Middle	Last
T1	$M = 10.57$ $SD = 2.63$ $U = 100$; $n_{1,2}$; $p^* = .0104$	$M = 9.89$ $SD = 3.25$ $U = 106$; $n_1 = 12$; $n_2 = 11$; $p^* = .0151$	$M = 7.73$ $SD = 1.58$ *Ref.*
T2	$M = 8.98$ $SD = 2.15$ *Ref.*		$M = 6.76$ $SD = 1.23$ $U = 64$; $n_{1,2}$; $p^* = .0423$
T2-2X	$M = 8.20$ $SD = 0.28$ *Ref.*	$M = 6.61$ $SD = 0.50$ $T(4) = 3.88$; $p^* = .0178$	$M = 6.24$ $SD = 0.43$ $T(4) = 5.35$; $p^* = .0059$

between tasks `T1` and `T2`. In a detailed view on the number of answers in each category (table 7.21), there is a tendency of users to experience the task with the *SCA-Force Area* as (1) more complex, (2) less interesting, (3) more demanding, i.e., needing more effort. In general, the users' did not experience major changes between tasks `T1` and `T2`. To the question, how many forces they felt, 11 participants answered "one" and 2 answered "two". Regarding the direction, 10 participants were correct by detecting the disturbance toward the left. The other 3 participants answered both directions, left and right. The reason for this can probably be that, when a disturbance vanishes, it can feel like there is one to the opposite direction.

GCA-Inverted Controls

The *GCA-Inverted Controls* is evaluated in tasks `T3-C` and `T3-C2`. This evaluation is done on the "curve"-path, and, since in tasks `T4-DA`, `T4-NL`, and `T5` variables were manipulated, there is no ground-truth task. Comparing the tasks with the *GCA-Inverted Controls* and their repetitions, no significant differences can be found. This indicates that the reference path was not suitable to evaluate this control action. The average deviation (figure 7.10a) shows a large upper limit for the *first* repetition of task `T3-C2`. Such results are expected when using *GCA-Inverted Controls*. The last repetition values are comparable with the other tasks because users adapted themselves to the control action. Still, since task `T3-C2` had the sample of only three participants, measurements may be distorted.

The users' answers were recorded together for tasks `T3-C` and `T3-C2`. The users experienced the task as the most complex and demanding than all other tasks (table 7.21). The intuitiveness of the RoboTrainer's control was also reduced, confirming the desired impact on the user. The speed of the device was perfectly adjusted for all participants. Nevertheless, the users did not find the tasks with *GCA-Inverted Controls* more interesting compared to the others, but they were less assured in their performance. This could also be a motivating factor for training with RoboTrainer v2.

Comparison of the Adaptive Control Strategies

The two evaluated adaptive control strategies show a significant reduction in users' average input forces against all other tasks (cf. figure 7.7). The concrete values are given in table 7.19. The velocity-based non-linear adaption (`T4-NL`) from this thesis also significantly reduces the interaction force against the state-of-the-art approach with linear damping adaption (`T4-DA`) from Yu, Spenko, and Dubowsky [185]. This is the intended purpose of these strategies. The difference between tasks `T4-DA` and `T4-NL` is influenced by two factors. First, the parameters of each adaption which are tuned to provide agile RoboTrainer's movement without oscillation between the user-walker system. This tuning is done experimentally in preparations ahead of the study. The second reason is that the non-linear approach also reduces the mass factor in the admittance equation (equation (2.12)) and enables a direction change with less force since RoboTrainer has lower inertia on higher velocities.

The average velocity is not significantly affected by the adaptive controllers[3]. There is a tendency to lower velocity than in the first two tasks, probably influenced by the different reference paths (cf. table 7.12). On the other hand, compared to other tasks done along the "curve", there is a tendency to higher velocities. Those results practically confirm the adaption concepts, where the goal is to reduce users' effort, i.e., input force, when strolling with a higher velocity.

The average deviation values (figure 7.10) are comparable to the other tasks, showing that the adaption does not significantly influence the users' control precision. The same can be concluded for the duration (figure 7.11), comparable to the other tasks along the "curve". The average users' distance to RoboTrainer (figure 7.12) also does not differ from other tasks.

Comparing users' answers in tasks `T4-DA` and `T4-NL` (table 7.21), the following is disclosed: in the `T4-NL` RoboTrainer's behavior, compared to `T4-DA`, it tends to be (1) less safe, (2) less intuitive, (3) with more appropriate speed and weight. Otherwise, the users' subjective estimation of these tasks is comparable to the other tasks. The only

[3] There is a significant difference toward task `T5` evaluated in the next section.

Table 7.19.: Significant differences ($p^* < .05$) for average interaction force between the tasks with adaptive (`T4-DA` and `T4-NL`) and conventional control. The measured statistics are *T-test* and *Mann-Whitney U-test* depending on the variables' normal distribution. Comparison is shown column-wise.

Task ID	`T4-DA`	`T4-NL`
`T4-DA`	$M = 45.96; SD = 4.40$	
`T4-NL`	$T(42) = 4.15; p^* = .0002$	$M = 37.52; SD = 8.47$
`T1`	$M = 52.64; SD = 12.51$	
	$T(52) = 2.78; p^* = .0079$	$T(56) = 5.39; p^* < .0001$
`T2`	$M = 61.11; SD = 14.06$	
	$T(45) = 5.16; p^* < .0001$	$T(49) = 7.21; p^* < .0001$
`T2-2X`	$M = 70.10; SD = 7.21$	
	$T(27) = 10.69; p^* < .0001$	$T(31) = 9.92; p^* < .0001$
`T3-C`	$M = 61.04; SD = 6.99$	
	$U = 24; n_{1,2}; p^* < .0001$	$U = 12; n_1 = 24; n_2 = 20; p^* < .0001$
`T3-C2`	$M = 55.59; SD = 9.36$	
	$U = 20; n_1 = 20; n_2 = 6; p^* = .0162$	$U = 15; n_1 = 24; n_2 = 6; p^* = .0034$
`T5`	$M = 56.69; SD = 7.83$	
	$T(39) = -5.30; p^* < .0001$	$T(43) = -7.67; p^* < .0001$

exception is reported by the two participants who had to watch out for collision with their legs against the RoboTrainer. This is probably caused by different dynamics than in previous tasks and more agile movement of the device.

When asked directly to compare the adaption algorithms, 6 participants preferred non-linear and linear-damping algorithms. One participant said that the RoboTrainer's control felt the same.

Influence of the RoboTrainer v2's Wheel-Setup

The RoboTrainer's footprint was reconfigured from the position "open-short" (figure 4.4c) to the position "closed-long" (figure 4.4b) in task `T5`. The goal was to show how users interact with the RoboTrainer when having less space for legs.

The average interaction force (figure 7.7) has similar values as for task `T1`. There is a tendency to use less force in this configuration than in other tasks along the "curve" with conventional controller (`T3-C` and `T3-C2`). There are no significant differences to any other task, except those using an adaptive controller (cf. table 7.19). Table 7.20 shows all other significant differences between the tasks.

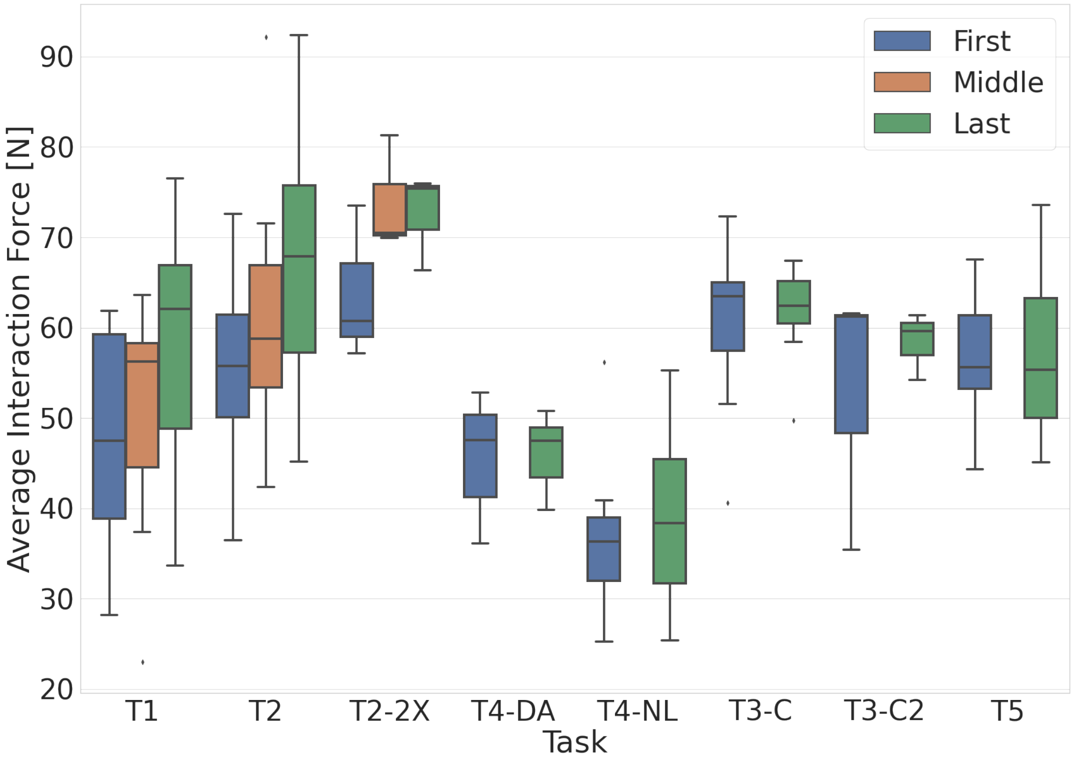
Although it was the last task, in `T5`, the participants were significantly slower than in all other tasks. The exception is the first task `T1` and `T3-C` with *GCA–Inverted Controls*. Nevertheless, further investigations have to be done to understand why a smaller area for users' feet reduces the walking speed. Maybe the users will be faster again after some practice, as happened for tasks `T1` and `T2`.

During `T5`, the average deviation was larger compared to all tasks except `T3-C` and `T3-C2`, where only a tendency is present. The average duration of a trail was significantly longer compared to tasks `T4-DA` and `T4-NL`. This result is probably caused by the higher average velocities when using adaptive-controlled RoboTrainer. One interesting outcome is the smaller distance between the user and RoboTrainer v2 when using "closed-long" configuration. A significant difference is shown for all tasks except `T3-C2`. This exception is probably caused by the small sample of three users in `T3-C2`. A reason for this effect is not known and should be investigated in the future.

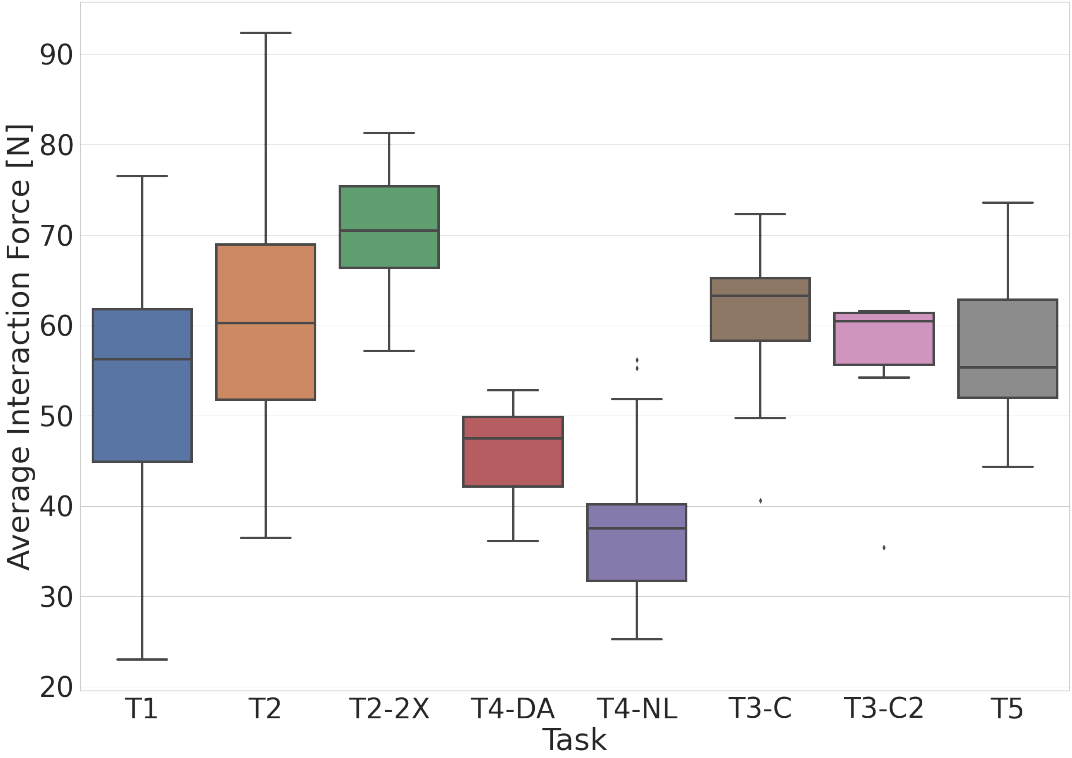
In the questionnaire (tables 7.21 and 7.22), users reported that they had to watch out for collisions between their feet and the RoboTrainer. This is an expected outcome since the change was considerable after interaction using the "open-short" footprint setup.

Table 7.20.: Significant differences ($p^* < .05$) showing the influence of RoboTrainer's footprint setup (task T5) on different variables. The measured statistics are *T-test* and *Mann-Whitney U-test* depending on the variables' normal distribution. Comparison is shown column-wise.

Task ID	Velocity	Deviation	Duration	Distance
T5	$M = 0.92$ $SD = 0.12$	$M = 0.10$ $SD = 0.04$	$M = 12.32$ $SD = 1.48$	$M = 0.53$ $SD = 0.04$
T1		$M = 0.07$ $SD = 0.04$ $U = 230$ $n_1 = 34$ $n_2 = 21$ $p^* = .0284$		$M = 0.58$ $SD = 0.06$ $U = 561$ $n_1 = 34$ $n_2 = 21$ $p^* = .0004$
T2	$M = 1.07$ $SD = 0.18$ $T(46) = 3.36$ $p^* = .0016$	$M = 0.06$ $SD = 0.03$ $U = 119$ $n_1 = 27$ $n_2 = 21$ $p^* = .0007$		$M = 0.60$ $SD = 0.07$ $U = 435$ $n_1 = 27$ $n_2 = 21$ $p^* = .0017$
T2-2X	$M = 1.18$ $SD = 0.09$ $T(28) = 5.54$ $p^* < .0001$	$M = 0.07$ $SD = 0.01$ $T(28) = -2.68$ $p^* = .0125$		$M = 0.60$ $SD = 0.07$ $U = 157$ $n_1 = 9$ $n_2 = 21$ $p^* = .0050$
T4-DA	$M = 1.03$ $SD = 0.14$ $T(39) = 2.70$ $p^* = .0103$	$M = 0.07$ $SD = 0.05$ $U = 122$ $n_1 = 20$ $n_2 = 21$ $p^* = .0225$	$M = 10.97$ $SD = 1.88$ $U = 100$ $n_1 = 20$ $n_2 = 21$ $p^* = .0043$	$M = 0.59$ $SD = 0.06$ $U = 334$ $n_1 = 20$ $n_2 = 21$ $p^* = .0013$
T4-NL	$M = 1.02$ $SD = 0.15$ $T(43) = 2.41$ $p^* = .0204$	$M = 0.07$ $SD = 0.04$ $U = 149$ $n_1 = 24$ $n_2 = 21$ $p^* = .0197$	$M = 11.18$ $SD = 1.75$ $T(43) = -2.29$ $p^* = .0269$	$M = 0.58$ $SD = 0.06$ $U = 384$ $n_1 = 24$ $n_2 = 21$ $p^* = .0028$
T3-C				$M = 0.59$ $SD = 0.07$ $U = 312$ $n_1 = 20$ $n_2 = 21$ $p^* = .0081$

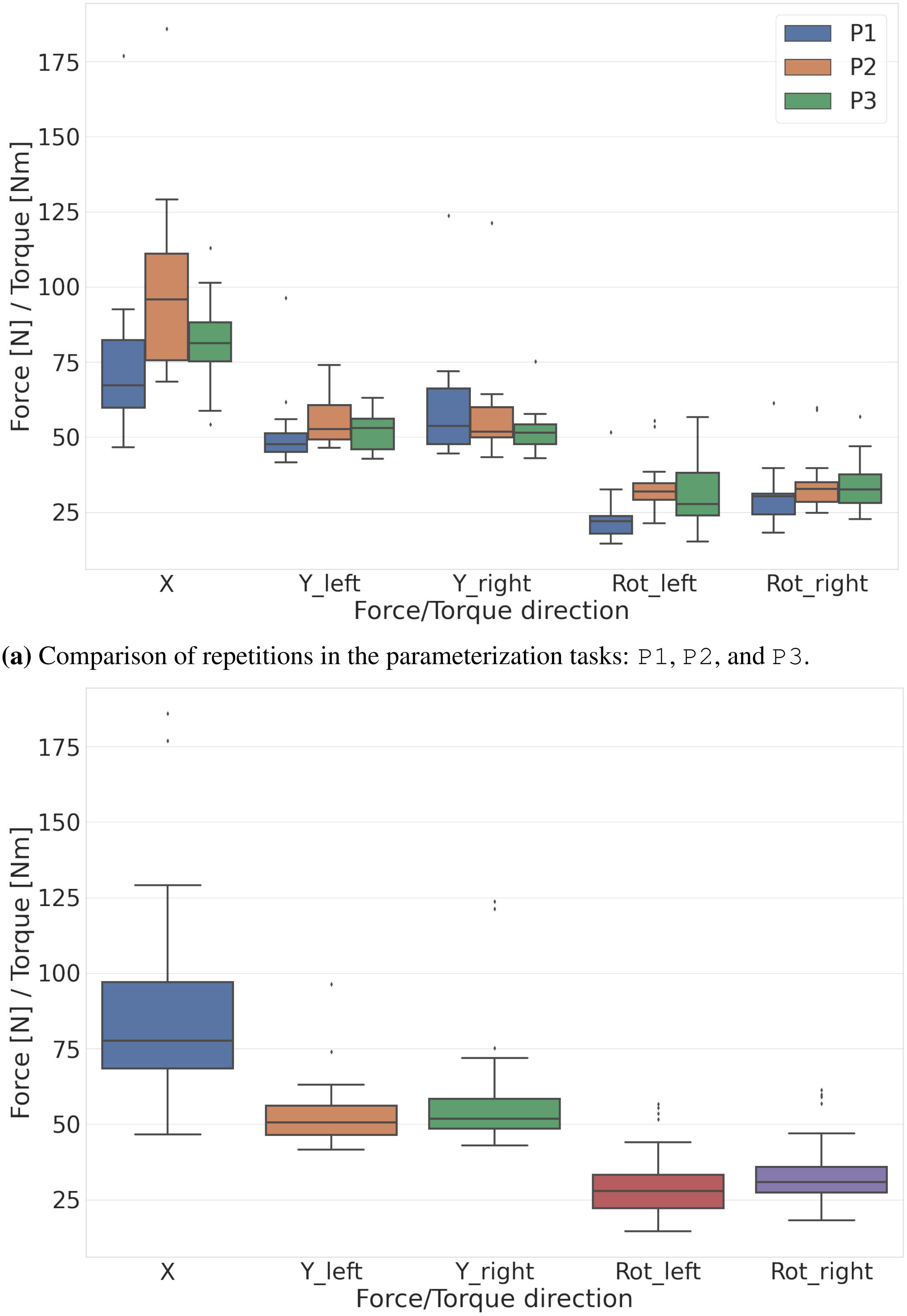


(a) Average input forces for different repetitions. The colors indicate different repetitions.



(b) Average input force for the tasks.

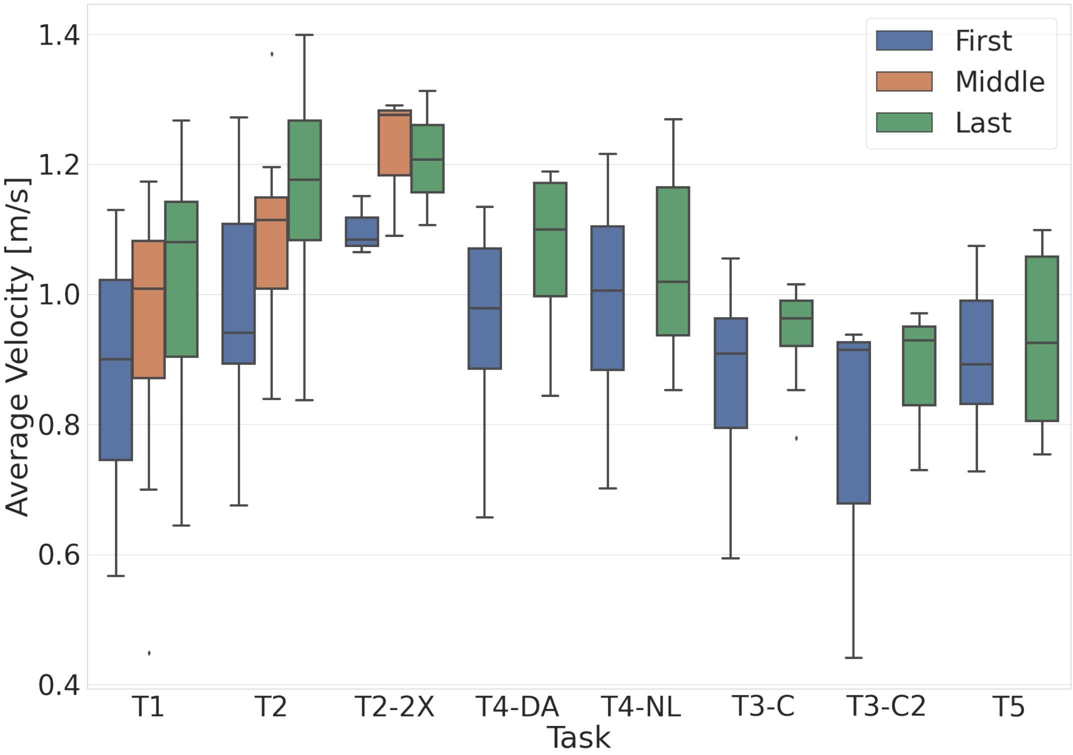
Figure 7.7.: Average interaction force of all users showed for the different repetitions of a task and as the average value of a task.



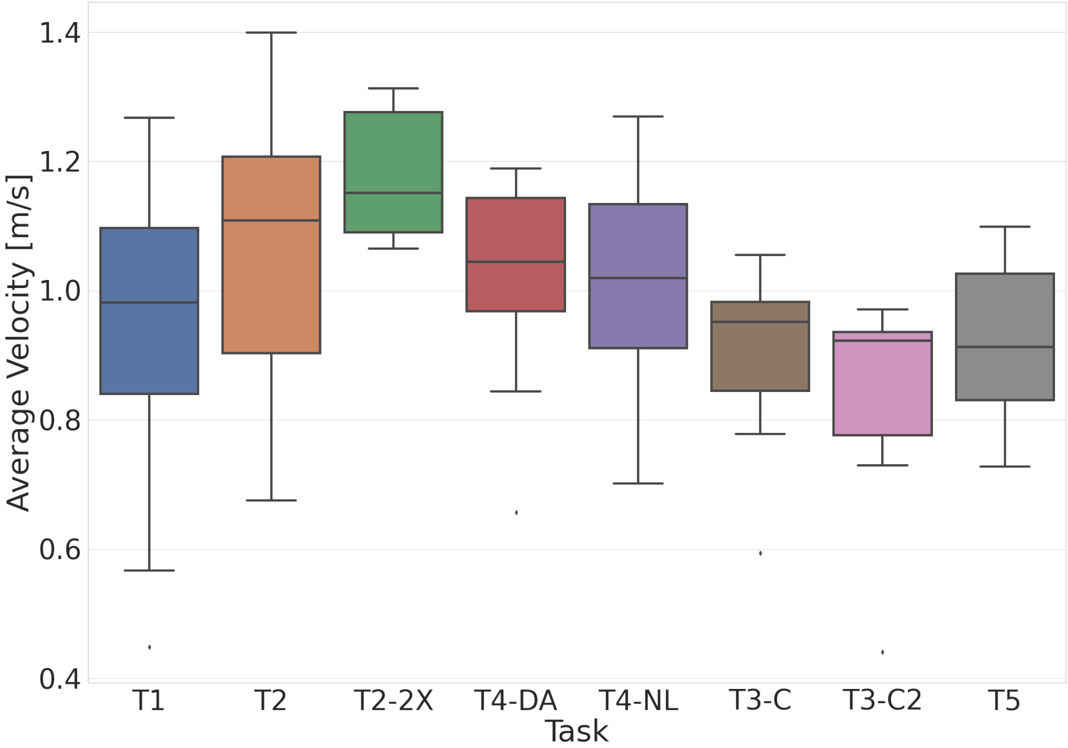
(a) Comparison of repetitions in the parameterization tasks: `P1`, `P2`, and `P3`.

(b) Average forces for all parameterizations for different degrees of freedom.

Figure 7.8.: Average input force of all users showed for the different repetitions of a task and as the average value of a task.

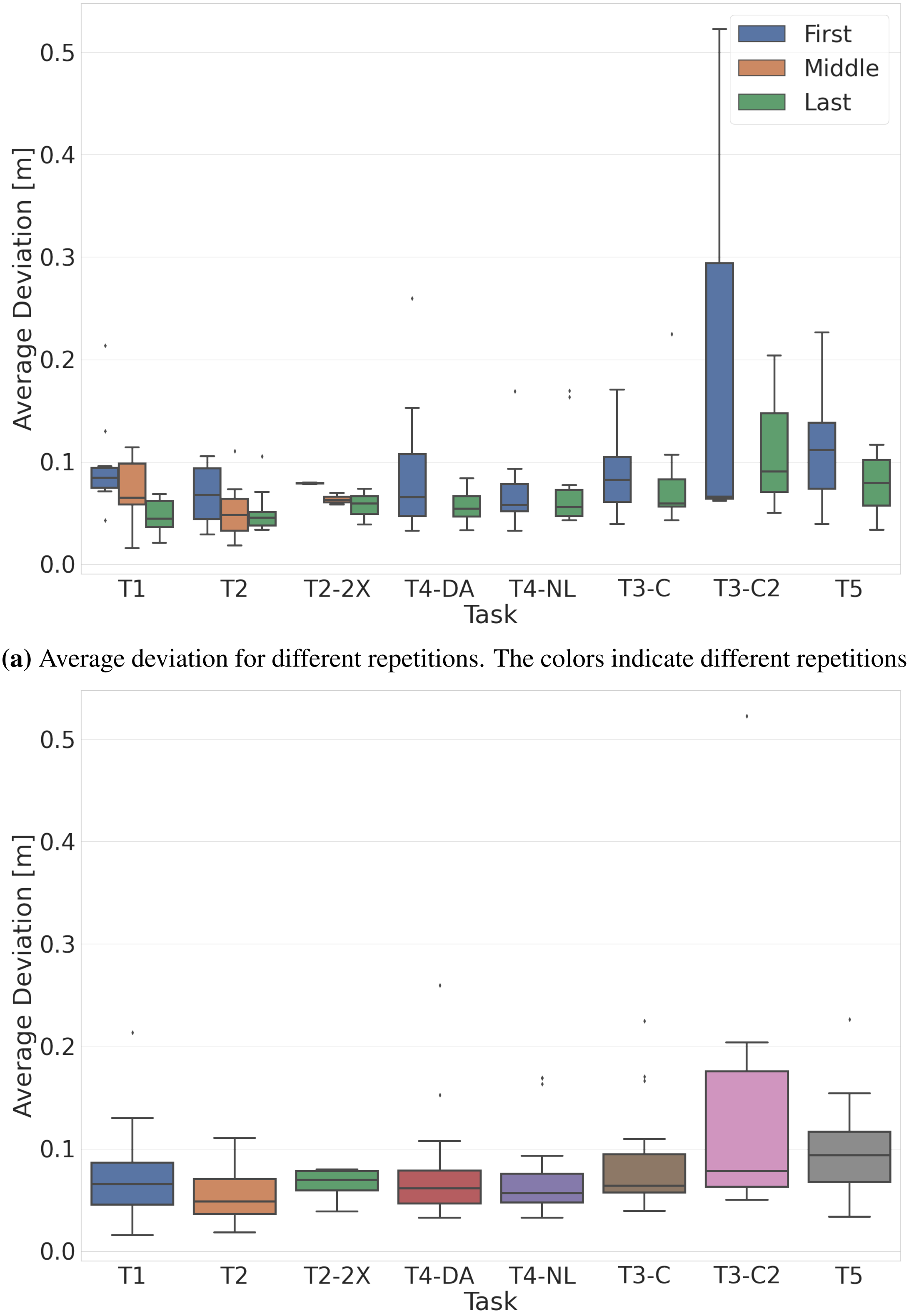


(a) Average velocity for different repetitions. The colors indicate different repetitions.



(b) Average velocity for the tasks.

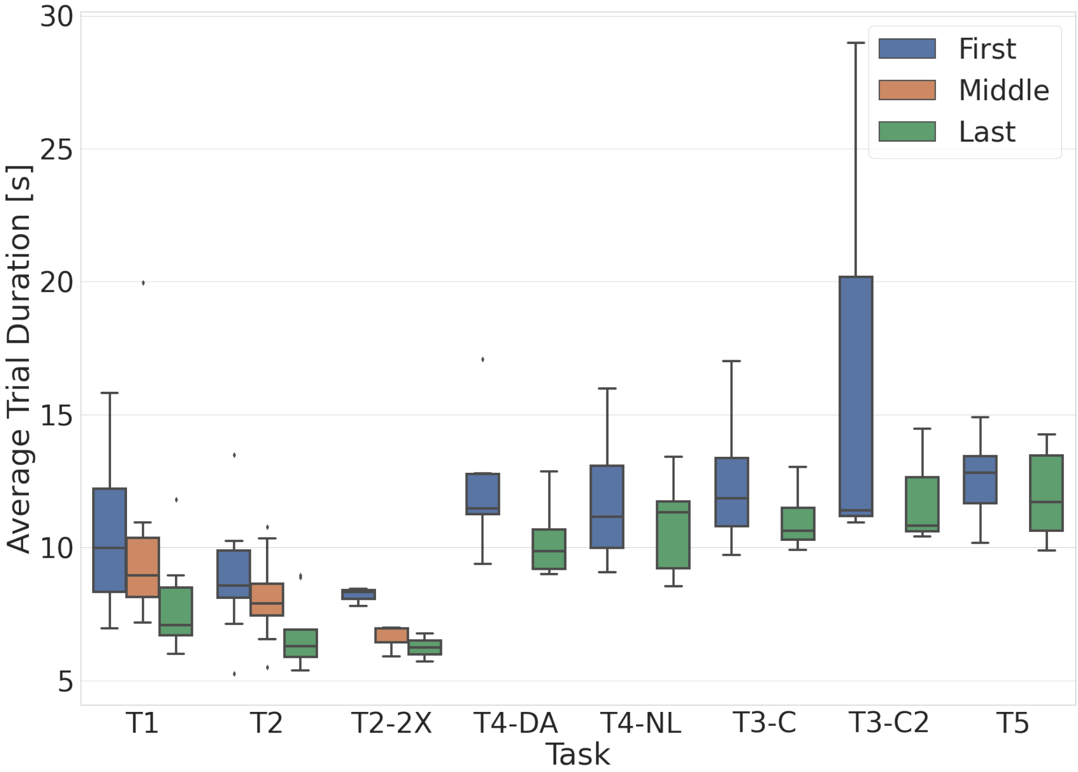
Figure 7.9.: Average velocity of all users showed for the different repetitions of a task and as the average value of a task.



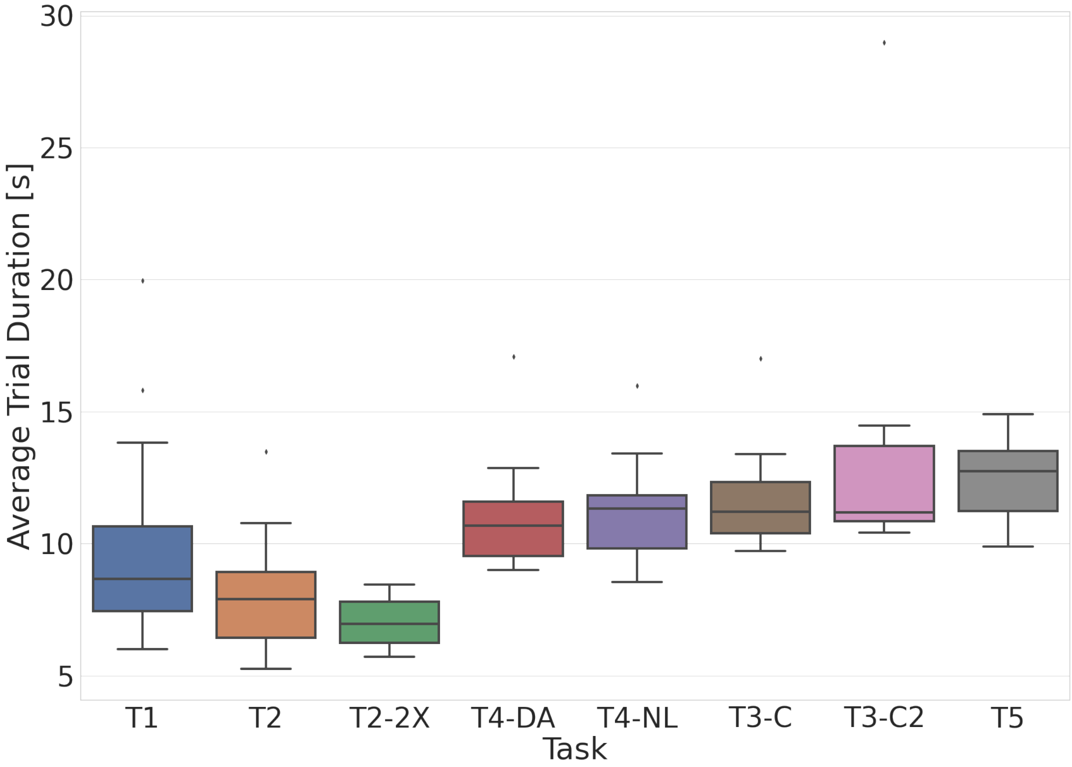
(a) Average deviation for different repetitions. The colors indicate different repetitions.

(b) Average deviation for the tasks.

Figure 7.10.: Average deviation of all users showed for the different repetitions of a task and as the average value of a task.

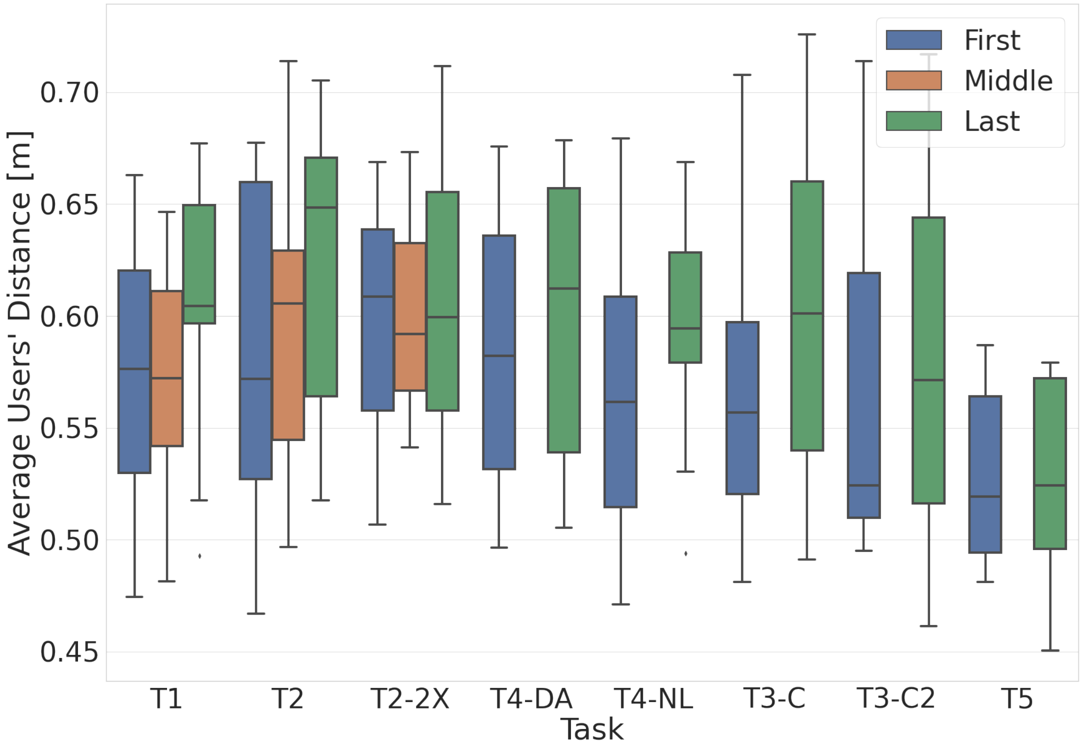


(a) Duration for different repetitions. The colors indicate different repetitions.

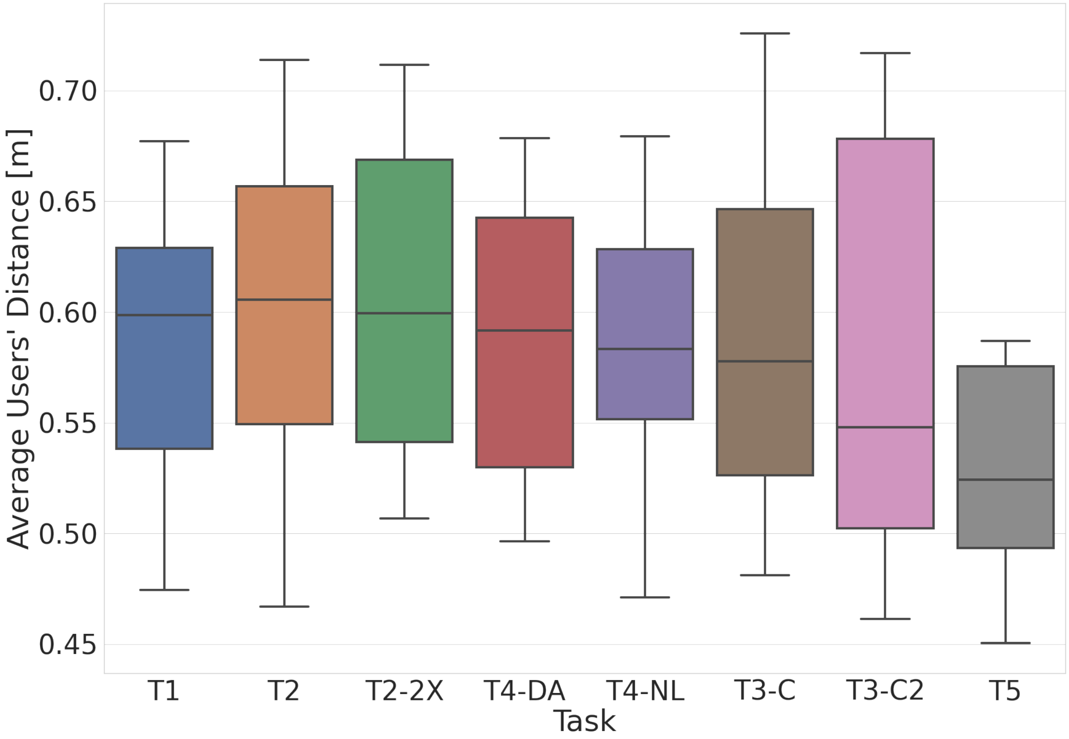


(b) Duration for the tasks.

Figure 7.11.: Duration of all users showed for the different repetitions of a task and as the average value of a task.



(a) Average users' distance for different repetitions. The colors indicate different repetitions.



(b) Average users' distance for the tasks.

Figure 7.12.: Average users' distance of all users showed for the different repetitions of a task and as the average value of a task.

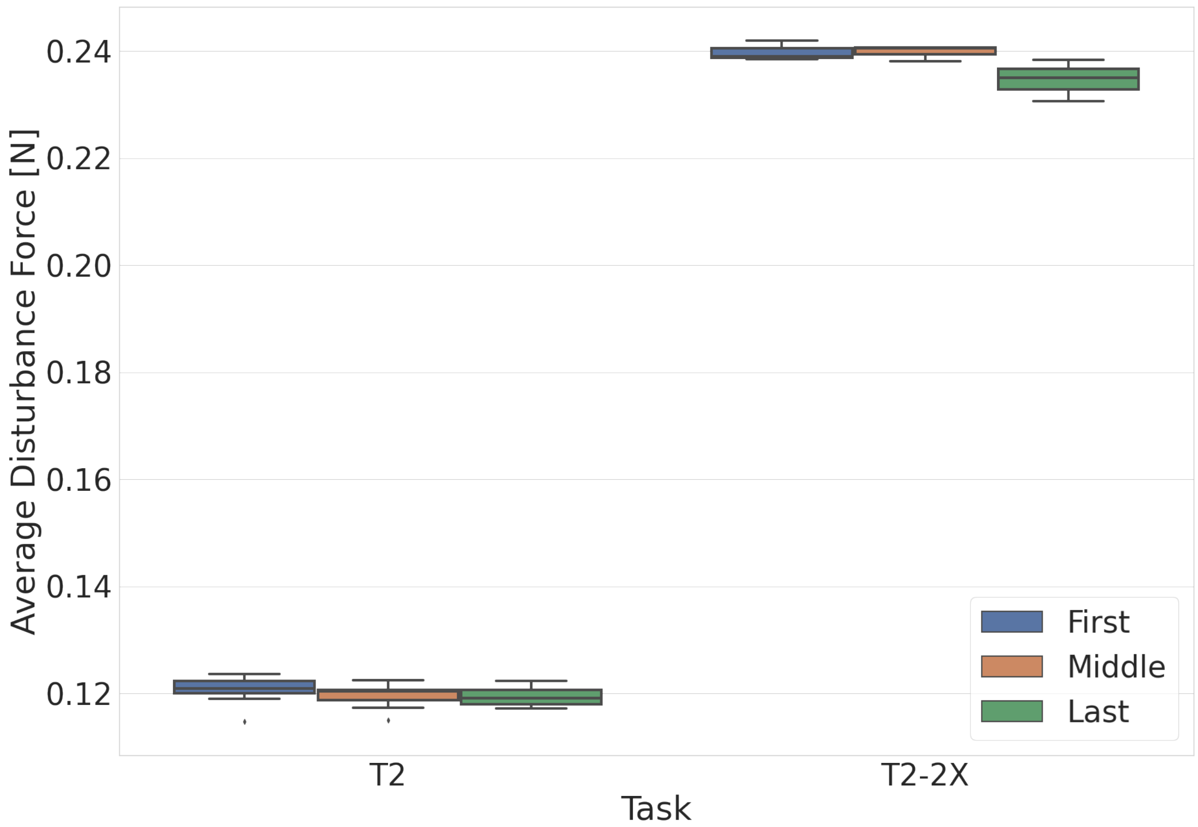


Figure 7.13.: Average disturbance force for different repetitions. The colors indicate different repetitions.

Table 7.21.: User's answers during the RoboTrainer v2's evaluation. The numbers represent the absolute number of answers per category on a Likert-type scale with five categories (cf. table 7.13). The sum of answers is not always equal to $n = 13$ since not all participants answered every question.

Task ID	**Complexity**					**Self-Assessment**					**Interest**					**Effort**					**Safety**					**Intuitiveness**					**Speed**					**Weight**					**Collision**				
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
P1	0	1	3	6	3																0	0	2	1	10	1	1	5	3	3															
T1	0	0	0	5	8	0	0	2	8	3	0	1	1	3	7	3	7	2	1	0	0	0	2	1	10	0	0	4	5	3	0	8	3	2	0	1	7	4	1	0	9	2	0	2	0
T2	0	0	2	5	6	0	0	2	7	4	0	2	2	2	7	4	5	1	3	0	0	0	1	2	10	0	0	0	8	5															
P2	0	0	1	6	6																0	0	0	2	11	0	0	1	8	4															
T4-DA																					0	0	0	3	10	0	0	1	9	3	0	3	8	2	0	0	5	8	0	0					
T4-NL																					0	0	1	2	10	0	0	4	4	5	0	1	10	2	0	0	2	10	1	0					
T4	0	0	3	9	1	0	0	3	6	4	0	0	2	2	9	2	4	4	3	0																					6	5	0	1	1
T3-C	1	3	5	4	0	0	0	6	4	2	0	0	2	2	9	0	3	2	4	3	0	0	2	2	9	0	1	9	2	1	0	0	13	0	0	0	4	9	0	0	8	4	1	0	0
P3	0	0	0	3	9																0	0	0	1	11	0	0	0	4	8															
T5	0	0	1	9	2	0	0	0	10	2	0	1	2	1	8	2	6	2	2	0	0	0	1	3	8	0	0	2	4	6	0	3	8	0	0	0	5	7	0	0	1	0	1	4	6

Table 7.22.: Median and range of users' answers for each evaluated variable. The responses are recorded on a Likert-type scale with five response categories (cf. table 7.13).

Task ID	**Complexity**		**Self-Assessment**		**Interest**		**Effort**		**Safety**		**Intuitiveness**		**Speed**		**Weight**		**Collision**	
	Median	Range	Median	Range	Median	Range	Median	Range	Median	Range	Median	Range	Median	Range	Median	Range	Median	Range
P1	4	2 − 5							5	3 − 5	3	1 − 5						
T1	5	4 − 5	4	3 − 5	5	2 − 5	2	1 − 4	5	3 − 5	4	3 − 5	2	2 − 4	2	1 − 4	1	1 − 4
T2	4	3 − 5	4	3 − 5	5	2 − 5	2	1 − 4	5	3 − 5	4	4 − 5						
P2	4	3 − 5							5	4 − 5	4	3 − 5						
T4-DA									5	4 − 5	4	3 − 5	3	2 − 4	3	2 − 3		
T4-NL									5	3 − 5	4	3 − 5	3	2 − 4	3	2 − 4		
T4	4	3 − 5	4	3 − 5	5	3 − 5	3	1 − 4									2	1 − 5
T3-C	3	1 − 4	3.5	3 − 5	5	3 − 5	4	2 − 5	5	3 − 5	3	2 − 5	3	3 − 3	3	2 − 3	1	1 − 3
P3	5	4 − 5							5	4 − 5	5	4 − 5						
T5	4	3 − 5	4	4 − 5	5	2 − 5	2	1 − 4	5	3 − 5	4.5	3 − 5	3	2 − 3	3	2 − 3	4.5	1 − 5

7.2.3. Findings from the Evaluation of RoboTrainer v2

Overall, all participants were able to finalize the evaluation of RoboTrainer v2 without any major issues. They felt safe during the trials and managed to control the device in all configurations. The users found the interaction reasonably intuitive. After the trials, a participant reported that they had to adapt themselves to the RoboTrainer, rather than the device to them. This means that the evaluation fulfills its goal, i.e., to provide data about changes in users' interactions over a period of time.

From the software perspective, the RoboTrainer behaved as intended during the whole time. Except for explained issues with the safety-fields configuration, there was only one safety-relevant situation where the wireless emergency stop was utilized. This situation was that the device started to move forward slowly without a user's intention. The reason was an offset drift of the force-torque sensor measurements over time. This is a known drawback of force-torque sensors that happens with changing sensor's temperature when in operation. After repeated calibration, the evaluation was continued without any further issues. In this concrete case, the user was in danger, but the user was wired to EEG- and EMG-hardware placed on RoboTrainer. To avoid such situations, an automatic sensor offset calculation should be done. The main challenge to realize this is recognizing that there is no external force influence when calculating sensors' offsets.

Before the evaluation, the hypothesis was set that interaction forces would reduce and converge toward the end of the evaluation session. Nevertheless, this is not unambiguously confirmed with the results. On the other side, the analysis of interaction forces shows that the controller's adaption significantly influences them. So, this knowledge can be used in the future to investigate how the user's effort can be influenced.

The control actions were working as expected and lead to measurable changes in observed variables. The *SCA–Force Area* compared in tasks `T1` and `T2` does not show significant changes regarding RoboTrainer's deviation from the reference paths. Probably, the disturbance forces were too cautious, providing only a small impulse for the participants. This follows the study design to only slightly disturb the participant's balance and detect their reaction in cognitive and muscular activity. Using the steady-state equation to calculate disturbance velocity from disturbance force, showed coherent RoboTrainer's behavior. Although using the trapezoidal profile, jerk at the entrance and exit of the SCA's influence area was not detected. This was probably because of the small maximum value of disturbance. Nevertheless, for using this approach in the future, any twice differentiable function, e.g., Gaussian or exponential, could be used as disturbance's profile. Use of those would assure continuous jerk changes.

In the study, the *GCA–Inverted Controls* showed its confusing effect on most participants at the end of the reference path, when RoboTrainer had to turn for 180°. After using higher velocity limits in `T3-C2`, the effects were more obvious. Still, in the future, tasks should be designed differently to unfold the full potential of this SCA.

The adaptive control strategies strongly impacted the user's interaction, especially considering the user's input force. The acquired measurements confirm the relevance of using an adaptive concept. The results are ambiguous regarding the use of concrete adaption type, e.g., non-linear or linear-damping.

The impact of RoboTrainer's rear wheels' configuration on participants' performance is insignificant for almost all variables. This unexpected result demands a further investigation in multiple tasks to check how user interaction develops over time. Nevertheless, the preference by the users for the open-wheel setup was apparent.

Overall, the study shows that the intended training is feasible using RoboTrainer v2. The device showed very high mechanical robustness and the functionality of all components was flawless.

8 Conclusions

This thesis focuses on the design and control of a robotic device for targeted strength and coordination training, called *RoboTrainer*. The research on neurocognitive aging models shows that physical activity has a significant positive impact on older adults. This opens the questions “if” and “how” robotic technology could support such activity, aiming at the motor and cognitive engagement of elderly people in everyday life. Robotized walking assistance devices, called Smart Walkers (SWs), focus on the physical and sensory support of older adults or persons with disabilities. Classically, research on Smart Walkers investigates human-robot interaction interfaces and shared-control approaches. Only in rare cases, those devices address specialized gait-rehabilitation scenarios. Therefore, it is still unclear if a SW-like device can be used for motor activation of its user and what kind of functionalities are needed for challenging and safe training.

Addressing this central question, this thesis

1. proposes training with a robotic device and investigates the feasibility and acceptance of elderly adults in chapter 3;
2. designs a novel device providing mechanical adaption, increased safety, and various degree of physical support for its users in chapter 4;
3. develops a parameterization approach for an individual and adaptive per-user control, and extends state-of-the-art passivity concepts for active Smart Walkers in chapter 5; and
4. offers concepts for the variability of robot training supporting various complexity levels and interaction schemes in chapter 6.

Chapter 3: Proof of Concept of a Smart Walker for Training

This chapter introduces neuromuscular training with a robotic device and investigates this training and the device itself in a pilot study with ten older adults with mild cognitive impairment. The training is derived from research on motor activation of persons through physical activity. Additionally, the training concept considers human-robot interaction possibilities and technical constraints of a mobile robot. The robotic device from this chapter, the RoboTrainer Prototype, is based on a research mobile platform extended with a force-torque sensor and a handlebar to enable physical interaction with users. The chapter's main contributions are high-level control concepts and a pilot study to evaluate the overall approach. The results show that the target group accepts the proposed training and that the training's complexity can be adjusted using control concepts. At the end of the training week, the participants were significantly faster and more precise at guiding the RoboTrainer along training paths.

Chapter 4: Design of a Device for Active Training

During the pilot study, *RoboTrainer Prototype* showed several drawbacks regarding its footprint, mechanical construction, and safety. Therefore, a novel device was developed to address these issues. *RoboTrainer v2* places the user closer to its center of mass in order to reduce the risk of tipping over. Furthermore, it uses a newly designed fixture for the user interface to increase its stiffness and to enable the handle's height adjustment. The main distinction from the state-of-the-art devices is the possibility to modify the device's footprint in two degrees of freedom. This means that the positions of its rear wheels can be changed, thus enabling the adjustment of users' support area during training. The chapter provides a detailed discussion of safety measures for RoboTrainer v2 and Smart Walkers in general. So far, such a discussion does not yet exist in literature.

Chapter 5: Control of Devices for Active Training

This chapter presents concepts for individual and adaptive control of Smart Walkers based on admittance dynamics. First, the novel device's dynamics and concrete physical human-robot interaction are analyzed. This analysis results in an extended user-walker interaction model considering the influence of users' sensorimotor performance on the model's stability. The analysis increases the understanding of occasional oscillations between user and walker and enables the extension of state-of-the-art to avoid these oscillations by taking into account users' movement intentions. A further contribution is a unique controller's parameterization strategy to determine the per-user dynamics of the device. The parameterization also adjusts the limits of non-linear and velocity-dependent adaption of walkers parameters. This adaption modifies the user's input force to implicitly adjust admittance dynamics when a walker moves and, at the same time, influences the control actions in the same manner. The control approach is implemented using the Robot Operating System (ROS) framework enabling its reuse across the RoboTrainer versions.

Chapter 6: Control Actions as Modifiers of Smart Walkers' Behavior

The *control actions* are fundamental building blocks of training with a Smart Walker. They modify the RoboTrainer's behavior either by generating a virtual force field with specific properties or by influencing the internal controller's states using a custom function. The CAs influence RoboTrainer in two ways: in the whole training environment by using *global* CAs, and in specific areas using *spatial* CAs. The main conceptual difference from state-of-the-art is the alteration of the device's velocity and not the users' input forces. The advantage of this approach is the separation of users' conscious movement and control actions' influence, which enables detection and avoidance of dangerous situations. Detection and avoidance are achieved by using the *passivity* approach for control actions and by reducing *safety* risks with methods developed in this thesis.

Chapter 7: User Evaluations

This chapter presents two user studies that evaluate the concepts and developments presented in chapters 4 to 6. The first study with 22 participants assesses the parameterization and adaption approaches for the admittance controller. The results show that participants preferred parameters that were calculated automatically by RoboTrainer. Furthermore, they preferred the non-linear adaptive controller over the controller with fixed parameters. After initial interaction, the participants favored the tasks with control actions as they provided a greater challenge. The second study with 13 participants focuses on the overall evaluation of the RoboTrainer v2 and compares non-linear and state-of-the-art adaptive controllers. The novel device was well accepted, and it is shown that its footprint configuration significantly influences the user's experience during training. An interesting result is that the RoboTrainer's wheel setup "closed-long" leads to reduced distance between user and RoboTrainer. The tested control actions worked as expected, but their moderate parameterization provided ambiguous results. The adaptive controller considerably reduces users' interaction forces while keeping the other variables comparable to the conventional controller. The participants' preference regarding adaptive-controller type was even. Overall, the RoboTrainer v2 operated reliably at all times during the evaluation.

Outlook

The thesis's main contributions culminate in a novel device for research of physical human-robot interaction applied in training with adults. Now, after the achievement of this integration, the interplay of individual functionalities can be further optimized. Furthermore, several research directions could be based on the key results of this thesis.

Some ideas for those directions are given at the end of each chapter, and some more general research topics are given in the remainder.

Safety From the safety perspective, there are still a few situations where users protection can be optimized to realize high-performance training. Current safety standards limit challenging training for younger people and athletes. Those standards have to be upgraded with novel concepts allowing physical human-robot interaction with agile robotic devices.

User Perception The current approach does not exhaust the full potential of RoboTrainer's sensors. For example, the force-torque sensor could be used for gait tracking and measurement of users' walking stability or cameras for body tracking could be integrated into the control loop. Some of these possibilities are investigated during the theses of Weber [176], Zimmermann [188], and Pelcz [129] but not yet mature for online use during training.

Control The current approach proposes to model users' sensorimotor performance as a first-order dynamic, but it is still unclear how to determine the model's parameters for each user. This requires an extension of the parameterization process, probably including other sensors into RoboTrainer. Once those values are estimated, a further question is how to use them for the device's safer operation. Also, further investigation on adaption strategies is needed, as, from the user studies, it is not clear which one is preferred.

Evaluations The total value of the presented concepts has to be further investigated regarding influence on persons' physical and cognitive state. This should be done from the perspective of an individual functionality, as done in the user-study at German Sport University Cologne for the *SCA–Force Area*. On the other hand, the RoboTrainer concepts should be evaluated as a whole in multidisciplinary studies, where possible treatments and diagnostic procedures are investigated. The device could be especially interesting for the fields of biomechanics, changes is motor capabilities during aging, and treatments of mild cognitive impairment.

These further advances would open a path for robotic devices, specifically Smart Walkers, offering support to aging societies in order to bear upcoming challenges.

Bibliography

[1] Hiroshi Akima et al. “Muscle function in 164 men and women aged 20-84 yr”. In: *Medicine & Science in Sports & Exercise* 33.2 (2001). ISSN: 0195-9131. DOI: 10.1097/00005768-200102000-00008.

[2] Alin Albu-Schäffer and Antonio Bicchi. “Actuators for Soft Robotics”. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 499–530. ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_21.

[3] J. Alves et al. “Considerations and Mechanical Modifications on a Smart Walker”. In: *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. May 2016, pp. 247–252. DOI: 10.1109/ICARSC.2016.30.

[4] J. Alves et al. “Overview of the ASBGo++ Smart Walker”. In: *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*. Feb. 2017, pp. 1–4. DOI: 10.1109/ENBENG.2017.7889420.

[5] M. Alwan et al. “Passive derivation of basic walker-assisted gait characteristics from measured forces and moments”. In: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 1. Sept. 2004, pp. 2691–2694. DOI: 10.1109/IEMBS.2004.1403772.

[6] Majd Alwan et al. “Basic walker-assisted gait characteristics derived from forces and moments exerted on the walker’s handles: results on normal subjects”. In: *Medical engineering & physics* 29.3 (May 2007), pp. 380–389. DOI: 10.1016/j.medengphy.2006.06.001.

[7] Majd Alwan et al. “Stability Margin Monitoring in Steering-Controlled Intelligent Walkers for the Elderly”. In: *Caring Machines: AI in Eldercare, Papers from the 2005 AAAI Fall Symposium, Arlington, Virginia, USA, November 4-6, 2005*. Ed. by Timothy W. Bickmore. Vol. FS-05-02. AAAI Technical Report. AAAI Press, 2005, pp. 1–8.

[8] American Psychological Association. *Publication manual of the American Psychological Association*. 6th ed. Washington, DC, 2010. ISBN: 978-1-4338-0561-5.

[9] Samira Anderson et al. "Reversal of age-related neural timing delays with training". In: *Proceedings of the National Academy of Sciences of the United States of America* 110.11 (Feb. 2013), pp. 4357–4362. DOI: `10.1073/pnas.1213555110.`.

[10] Oliver Armbruster. "Robot-based therapeutic options for persons with early stage dementia". Master's Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Oct. 2015.

[11] Daniel Azanov. "Dynamic Following of Human Legs Using a 2D Laser-Scanner". Bachelor's Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Apr. 2018, p. 41.

[12] Hamid Bateni and Brian E. Maki. "Assistive devices for balance and mobility: Benefits, demands, and adverse consequences". In: *Archives of Physical Medicine and Rehabilitation* 86.1 (2005), pp. 134–145. ISSN: 0003-9993. DOI: `10.1016/j.apmr.2004.04.023`.

[13] Anne S. Berry et al. "The influence of perceptual training on working memory in older adults". In: *PloS one* 5.7 (July 2010), e11537. DOI: `10.1371/journal.pone.0011537`.

[14] Richard W. Bohannon. "Comfortable and maximum walking speed of adults aged 20—79 years: reference values and determinants". In: *Age and Ageing* 26.1 (Jan. 1997), pp. 15–19. ISSN: 0002-0729. DOI: `10.1093/ageing/26.1.15`. eprint: `https://academic.oup.com/ageing/article-pdf/26/1/15/78378/26-1-15.pdf`.

[15] J. Borenstein and Y. Koren. "Histogramic in-motion mapping for mobile robot obstacle avoidance". In: *IEEE Transactions on Robotics and Automation* 7.4 (Aug. 1991), pp. 535–539. ISSN: 2374-958X. DOI: `10.1109/70.86083`.

[16] J. Borenstein and I. Ulrich. "The GuideCane-a computerized travel aid for the active guidance of blind pedestrians". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 2. Apr. 1997, 1283–1288 vol.2. DOI: `10.1109/ROBOT.1997.614314`.

[17] Alison Bowes et al. "Physical activity for people with dementia: a scoping study". In: *BMC Geriatrics* 13.1 (2013), pp. 129–138. ISSN: 1471-2318. DOI: `10.1186/1471-2318-13-129`.

[18] William L. Brogan. *Modern Control Theory (3rd Ed.)* USA: Prentice-Hall, Inc., 1991. ISBN: 0135897637.

[19] I. Caetano et al. "Development of a Biofeedback Approach Using Body Tracking with Active Depth Sensor in ASBGo Smart Walker". In: *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. May 2016, pp. 241–246. DOI: 10.1109/ICARSC.2016.34.

[20] I. Caetano et al. "Development of a data base to gait analysis and data organization in ASBGo Smart Walker". In: *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*. Feb. 2017, pp. 1–4. DOI: 10.1109/ENBENG.2017.7889485.

[21] Richard Camicioli et al. "Talking while walking". In: *Neurology* 48.4 (1997), pp. 955–958. ISSN: 0028-3878. DOI: 10.1212/WNL.48.4.955.

[22] Georgia Chalvatzaki et al. "Comparing the Impact of Robotic Rollator Control Schemes on Elderly Gait using on-line LRF-based Gait Analysis". In: *A workshop at ICRA 2019 - MoRobAE - Mobile Robot Assistants for the Elderly*. May 2019, p. 6.

[23] Patrick M. Charron, R. Lee Kirby, and Donald A. MacLeod. "Epidemiology of walker-related injuries and deaths in the United States: A Commentary". In: *American Journal of Physical Medicine & Rehabilitation* 74.3 (1995), pp. 237–239. ISSN: 0894-9115. DOI: 10.1097/00002060-199505000-00011.

[24] Sachin Chitta et al. "ros_control: A generic and simple control framework for ROS". In: *Journal of Open Source Software* 2.20 (2017), p. 456. DOI: 10.21105/joss.00456.

[25] Wan Kyun Chung, Li-Chen Fu, and Torsten Kröger. "Motion Control". In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 163–194. ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_8.

[26] O. Chuy, Y. Hirata, and K. Kosuge. "A New Control Approach for a Robotic Walking Support System in Adapting User Characteristics". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 36.6 (Nov. 2006), pp. 725–733. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2006.879396.

[27] O. Chuy, Y. Hirata, and K. Kosuge. "Active type robotic mobility aid control based on passive behavior". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2007, pp. 165–170. DOI: 10.1109/IROS.2007.4399408.

[28] O. Chuy, Y. Hirata, and K. Kosuge. "Control of walking support system based on variable center of rotation". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. Sept. 2004, 2289–2294 vol.3. DOI: 10.1109/IROS.2004.1389750.

[29] O. Chuy, Y. Hirata, and K. Kosuge. “Environment Feedback for Robotic Walking Support System Control”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Apr. 2007, pp. 3633–3638. DOI: 10.1109/ROBOT.2007.364035.

[30] O. Chuy, Y. Hirata, and K. Kosuge. “Online approach in adapting user characteristic for robotic walker control”. In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* June 2005, pp. 139–142. DOI: 10.1109/ICORR.2005.1501070.

[31] O. Y. Chuy et al. “A Control Approach Based on Passive Behavior to Enhance User Interaction”. In: *IEEE Transactions on Robotics* 23.5 (Oct. 2007), pp. 899–908. ISSN: 1552-3098. DOI: 10.1109/TRO.2007.907920.

[32] C. A. Cifuentes et al. “Multimodal Human–Robot Interaction for Walker-Assisted Gait”. In: *IEEE Systems Journal* 10.3 (Sept. 2016), pp. 933–943. ISSN: 1932-8184. DOI: 10.1109/JSYST.2014.2318698.

[33] Carlos A. Cifuentes and Anselmo Frizera, eds. *Human-Robot Interaction Strategies for Walker-Assisted Locomotion*. Vol. 115. Springer Tracts in Advanced Robotics. Springer International Publishing, 2016. ISBN: 978-3-319-34063-0.

[34] Gery Colombo et al. “Treadmill training of paraplegic patients using a robotic orthosis”. In: *Journal of rehabilitation research and development* 37.6 (Nov. 2000), pp. 693–700.

[35] Radu Constantinescu et al. “Assistive devices for gait in Parkinson’s disease”. In: *Parkinsonism & Related Disorders* 13.3 (2007), pp. 133–138. ISSN: 1353-8020. DOI: 10.1016/j.parkreldis.2006.05.034.

[36] DIN 18040-1. *Construction of accessible buildings - Design principles - Part 1: Publicly accessible buildings*. de. Standard DIN 18040-1. Belin, Germany: Deutsche Institut für Normung e.V. (DIN), Oct. 2010.

[37] DIN 18040-2. *Construction of accessible buildings - Design principles - Part 2: Dwellings*. de. Standard DIN 18040-2. Berlin, Germany: Deutsche Institut für Normung e.V. (DIN), Sept. 2011.

[38] Directive 2006/42/EC. *Machinery Directive 2006/42/EC*. en. Tech. rep. Directive 2006/42/EC. European Parliament and Council, June 2006.

[39] H.H. Dodge et al. “In-home walking speeds and variability trajectories associated with mild cognitive impairment”. In: *Neurology* 78.24 (2012). For HEiKA Part, pp. 1946–1952. ISSN: 0028-3878. DOI: 10.1212/WNL.0b013e318259e1de.

[40] S. Dubowsky et al. “PAMM - a robotic aid to the elderly for mobility assistance and monitoring: a "helping-hand" for the elderly”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. Vol. 1. Apr. 2000, 570–576 vol.1. DOI: 10.1109/ROBOT.2000.844114.

[41] Claire Dune, P. Gorce, and Jean-Pierre Merlet. “Can smart rollators be used for gait monitoring and fall prevention?” In: *EEE/RSJ International Conference on Intelligent Robots and Systems –Workshop Assistance and Service Robotics in a Human Environment*. Oct. 2012. DOI: 10.13140/2.1.2267.5843.

[42] Andrew S. Duxbury. “Gait disorders and fall risk: Detection and prevention”. In: *Comprehensive Therapy* 26.4 (2000), pp. 238–245. ISSN: 1559-1190. DOI: 10.1007/s12019-000-0024-7.

[43] *European Union: Horizon 2020 - Work Programme 2016-2017: General Annexes: G.Technology readiness levels (TRL)*. Web. Oct. 2016-2017.

[44] V. Faria et al. “Dynamical system approach for obstacle avoidance in a Smart Walker device”. In: *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. May 2014, pp. 261–266. DOI: 10.1109/ICARSC.2014.6849796.

[45] Tim Fleiner et al. “A Structured Physical Exercise Program Reduces Professional Caregiver’s Burden Caused by Neuropsychiatric Symptoms in Acute Dementia Care: Randomized Controlled Trial Results”. In: 74 (2020). 2, pp. 429–433. ISSN: 1875-8908. DOI: 10.3233/JAD-191102.

[46] Fraunhofer-Institut für Produktionstechnik und Automatisierung. *Care-O-bot 4*. Online; accessed 3rd Dec 2020. 2019.

[47] Fraunhofer-Institut für Produktionstechnik und Automatisierung. *rob@work*. Online; accessed 3rd Dec 2020. 2016.

[48] J. Frémy, F. Michaud, and M. Lauria. “Pushing a robot along - A natural interface for human-robot interaction”. In: *2010 IEEE International Conference on Robotics and Automation*. May 2010, pp. 3440–3445. DOI: 10.1109/ROBOT.2010.5509584.

[49] J. Frémy et al. “Force-controlled motion of a mobile platform”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2010, pp. 2517–2518. DOI: 10.1109/IROS.2010.5653861.

[50] Anselmo Frizera et al. “Smart Walkers: Advanced Robotic Human Walking-Aid Systems”. In: vol. 106. Jan. 2015, pp. 103–131. DOI: 10.1007/978-3-319-12922-8_4.

[51] Anselmo Frizera et al. “The Smart Walkers as Geriatric Assistive Device. The SIMBIOSIS Purpose”. In: *Gerontechnology* 7.2 (2008), pp. 108–108. ISSN: 1569-1101.

[52] Anselmo Frizera Neto et al. “Extraction of user’s navigation commands from upper body force interaction in walker assisted gait”. In: *BioMedical Engineering OnLine* 9.1 (Aug. 2010), p. 37. ISSN: 1475-925X. DOI: `10.1186/1475-925X-9-37`.

[53] Anselmo Frizera-Neto et al. “Empowering and Assisting Natural Human Mobility: The Simbiosis Walker”. In: *International Journal of Advanced Robotic Systems* 8.3 (2011), p. 29. eprint: `https://doi.org/10.5772/10666`.

[54] W. R. Frontera et al. “A cross-sectional study of muscle strength and mass in 45- to 78-yr-old men and women”. In: *Journal of Applied Physiology* 71.2 (1991). PMID: 1938738, pp. 644–650. DOI: `10.1152/jappl.1991.71.2.644`.

[55] BEMOTEC GmbH. *beactive+e - E-Rollator*. https://www.my-beactive.de/. Online; accessed 13th Feb. 2019. BEMOTEC GmbH, 2019.

[56] eMovements GmbH. *ello – Der elektrische Rollator*. https://ello.wmt.team/. Online; accessed Oct. 30, 2020. WMT GmbH, 2020.

[57] B. Graf. “Reactive navigation of an intelligent robotic walking aid”. In: *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591)*. Sept. 2001, pp. 353–358. DOI: `10.1109/ROMAN.2001.981929`.

[58] Birgit Graf and Martin Hägele. “Dependable interaction with an intelligent home care robot”. In: *In Proceedings of ICRA Workshop on Technical Challenge for Dependable Robots in Human Environments*. 2001, p. 2.

[59] Gilbert Groten. “Development of a graphical tool for the definition of robot-based training scenarios”. Bachelor’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Nov. 2017, p. 44.

[60] R. Guadagno. *Writing up your results – Guidelines based on APA style*. Web. 2010.

[61] Sami Haddadin and Elizabeth Croft. “Physical Human–Robot Interaction”. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 1835–1874. ISBN: 978-3-319-32552-1. DOI: `10.1007/978-3-319-32552-1_69`.

[62] Patricia Heyn, Beatriz C. Abreu, and Kenneth J. Ottenbacher. “The effects of exercise training on elderly persons with cognitive impairment and dementia: A meta-analysis”. In: *Archives of Physical Medicine and Rehabilitation* 85.10 (2004), pp. 1694–1704. ISSN: 0003-9993. DOI: `10.1016/j.apmr.2004.03.019`.

[63] Y. Hirata, A. Hara, and K. Kosuge. "Motion Control of Passive Intelligent Walker Using Servo Brakes". In: *IEEE Transactions on Robotics* 23.5 (Oct. 2007), pp. 981–990. ISSN: 1552-3098. DOI: 10.1109/TRO.2007.906252.

[64] Y. Hirata, A. Hara, and K. Kosuge. "Motion Control of Passive-type Walking Support System based on Environment Information". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Apr. 2005, pp. 2921–2926. DOI: 10.1109/ROBOT.2005.1570557.

[65] Y. Hirata, A. Hara, and K. Kosuge. "Passive-type intelligent walking support system "RT Walker"". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 4. Sept. 2004, 3871–3876 vol.4. DOI: 10.1109/IROS.2004.1390018.

[66] Y. Hirata, S. Komatsuda, and K. Kosuge. "Fall prevention control of passive intelligent walker based on human model". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sept. 2008, pp. 1222–1228. DOI: 10.1109/IROS.2008.4651173.

[67] Yasuhisa Hirata, Asami Muraki, and Kazuhiro Kosuge. "Passive-type Intelligent Walker Controlled Based on Caster-like Dynamics". In: *Rehabilitation Robotics*. Ed. by Sashi S Kommu. Vienna, Austria: Itech Education and Publishing, Aug. 2007. ISBN: 978-3-902613-04-2. DOI: 10.5772/5161.

[68] Yasuhisa Hirata et al. "Human-adaptive motion control of active and passive type walking support system". In: *IEEE Workshop on Advanced Robotics and its Social Impacts, 2005.* 2005, pp. 139–144. DOI: 10.1109/ARSO.2005.1511640.

[69] K. H. Hoang et al. "Gait performance indicators for elderly humans interacting with robotic mobility assistance devices". In: *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*. Aug. 2015, pp. 758–763. DOI: 10.1109/ICORR.2015.7281293.

[70] Marc Hofmann et al. "Interactive computer-training as a therapeutic tool in Alzheimer's disease". In: *Comprehensive Psychiatry* 44.3 (2003), pp. 213–219. ISSN: 0010-440X. DOI: 10.1016/S0010-440X(03)00006-3.

[71] Cunjun Huang et al. "Shared Navigational Control and User Intent Detection in an Intelligent Walker". In: *Caring Machines: AI in Eldercare, Papers from the 2005 AAAI Fall Symposium, Arlington, Virginia, USA, November 4-6, 2005*. Ed. by Timothy W. Bickmore. Vol. FS-05-02. AAAI Technical Report. AAAI Press, 2005, pp. 59–67.

[72] D. Hyndman and A. Ashburn. ""Stops walking when talking" as a predictor of falls in people with stroke living in the community". In: *Journal of Neurology, Neurosurgery & Psychiatry* 75.7 (2004), pp. 994–997. ISSN: 0022-3050. DOI: 10.1136/jnnp.2003.016014.

[73] IEC 60204-1:2016. *Safety of machinery - Electrical equipment of machines - Part 1: General requirements*. en. International Standard IEC 60204-1:2016. Geneva, Switzerland: International Electrotechnical Commission (IEC), Oct. 2016, p. 277.

[74] IEC 61496-3:2008. *Safety of machinery - Electro-sensitive protective equipment - Part 3: Particular requirements for active opto-electronic protective devices responsive to diffuse Reflection (AOPDDR)*. en. International Standard IEC 61496-3:2008. Geneva, Switzerland: International Electrotechnical Commission (IEC), Feb. 2008, p. 139.

[75] IEC 61496-3:2018. *Safety of machinery - Electro-sensitive protective equipment - Part 3: Particular requirements for active opto-electronic protective devices responsive to diffuse Reflection (AOPDDR)*. en. International Standard IEC 61496-3:2018. Geneva, Switzerland: International Electrotechnical Commission (IEC), Dec. 2018, p. 183.

[76] International Federation of Robotics. *Service Robots Statistics*. http://www.ifr.org/service-robots/statistics/. Online; accessed: 19th November 2015. 2015.

[77] S. Irgenfried and H. Wörn. "Motion Control and Fall Prevention for an Active Walker Mobility Aid". In: *New Advances in Mechanisms, Transmissions and Applications*. Ed. by Victor Petuya, Charles Pinto, and Erwin-Christian Lovasz. Vol. 17. Mechanisms and Machine Science. Dordrecht: Springer Netherlands, 2014, pp. 157–164. ISBN: 978-94-007-7484-1, 978-94-007-7485-8. DOI: 10.1007/978-94-007-7485-8_20.

[78] ISO 10218-1:2011. *Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots*. en. Standard ISO 10218-1:2011. Geneva, Switzerland: International Organization for Standardization, July 2011.

[79] ISO 11199-2:2005. *Walking aids manipulated by both arms — Requirements and test methods — Part 2: Rollators*. en. Standard ISO 11199-2:2005. Geneva, Switzerland: International Organization for Standardization, Apr. 2005.

[80] ISO 12100:2010. *Safety of machinery — General principles for design — Risk assessment and risk reduction*. en. Standard ISO 12100:2010. Geneva, Switzerland: International Organization for Standardization, Nov. 2010, p. 77.

[81] ISO 13482:2014. *Robots and robotic devices — Safety requirements for personal care robots*. en. Standard ISO 13482:2014. Geneva, Switzerland: International Organization for Standardization, Feb. 2014.

[82] ISO 13849-1:2015. *Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design*. en. Standard ISO 13849-1:2015. Geneva, Switzerland: International Organization for Standardization, Dec. 2015.

[83] ISO/TR 14121-2:2012. *Safety of machinery — Risk assessment — Part 2: Practical guidance and examples of methods*. en. Standard ISO/TR 14121-2:2012. Geneva, Switzerland: International Organization for Standardization, June 2012, p. 38.

[84] ISO/TS 15066:2016. *Robots and robotic devices — Collaborative robots*. en. Standard ISO/TS 15066:2016. Geneva, Switzerland: International Organization for Standardization, Feb. 2016, p. 33.

[85] Y. Jiang and S. Wang. "Adapting directional intention identification in running control of a walker to individual difference with fuzzy learning". In: *2010 IEEE International Conference on Mechatronics and Automation*. Aug. 2010, pp. 693–698. DOI: 10.1109/ICMA.2010.5589091.

[86] Jan-Christoph Kattenstroth et al. "Superior sensory, motor, and cognitive performance in elderly individuals with multi-year dancing activities". In: *Frontiers in Aging Neuroscience* 2 (2010), p. 31. ISSN: 1663-4365. DOI: 10.3389/fnagi.2010.00031.

[87] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2. Mar. 1985, pp. 500–505. DOI: 10.1109/ROBOT.1985.1087247.

[88] Andrew J. Ko, Thomas D. LaToza, and Margaret M. Burnett. "A practical guide to controlled experiments of software engineering tools with human participants". In: *Empirical Software Engineering* 20.1 (Feb. 2015), pp. 110–141. ISSN: 1573-7616. DOI: 10.1007/s10664-013-9279-3.

[89] Ljubomir Kuljača. *Automatsko upravljanje - analiza linearnih sustava /*. 1. izd. Udžbenici Sveučilišta u Zagrebu = Manualia Universitatis studiorum Zagrabiensis / Sveučilišna Naklada Liber. Kigen d.o.o. Zagreb, 2005.

[90] G. J. Lacey and D. Rodriguez-Losada. "The Evolution of Guido". In: *IEEE Robotics Automation Magazine* 15.4 (Dec. 2008), pp. 75–83. ISSN: 1070-9932. DOI: 10.1109/MRA.2008.929924.

[91] Gerard Lacey and Kenneth M. Dawson-Howe. "Evaluation of Robot Mobility Aid for the Elderly Blind". In: *Proceedings of the Fifth International Symposium on Intelligent Robotic Systems*. 1997.

[92] Gerard Lacey and Kenneth M. Dawson-Howe. "The application of robotics to a mobility aid for the elderly blind". In: *Robotics and Autonomous Systems* 23.4 (July 1998). Intelligent Robotics Systems - SIRS'97, pp. 245–252. ISSN: 0921-8890. DOI: 10.1016/S0921-8890(98)00011-6.

[93] Gerard Lacey, Shane Mac Namara, and Kenneth M. Dawson-Howe. “Personal adaptive mobility aid for the infirm and elderly blind”. In: *Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing*. Ed. by Vibhu O. Mittal et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 211–220. ISBN: 978-3-540-68678-1. DOI: 10.1007/BFb0055980.

[94] A. G. Leal-Junior et al. “Polymer Optical Fiber-Based Sensor System for Smart Walker Instrumentation and Health Assessment”. In: *IEEE Sensors Journal* 19.2 (Jan. 2019), pp. 567–574. ISSN: 1530-437X. DOI: 10.1109/JSEN.2018.2878735.

[95] G. Lee et al. “JAIST Robotic Walker control based on a two-layered Kalman filter”. In: *2011 IEEE International Conference on Robotics and Automation*. May 2011, pp. 3682–3687. DOI: 10.1109/ICRA.2011.5979784.

[96] G. Lee et al. “Walking Intent-Based Movement Control for JAIST Active Robotic Walker”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.5 (May 2014), pp. 665–672. ISSN: 2168-2232. DOI: 10.1109/TSMC.2013.2270225.

[97] Geunho Lee, Takanori Ohnuma, and Nak Young Chong. “Design and control of JAIST active robotic walker”. In: *Intelligent Service Robotics* 3.3 (2010), pp. 125–135. ISSN: 1861-2784. DOI: 10.1007/s11370-010-0064-5.

[98] Martin Lehmann et al. “Vitamin B12–B6–Folate Treatment Improves Blood-Brain Barrier Function in Patients with Hyperhomocysteinaemia and Mild Cognitive Impairment”. In: *Dementia and Geriatric Cognitive Disorders* 16.3 (2003), pp. 145–150. DOI: 10.1159/000071002.

[99] Stefan Leutenegger et al. “Flying Robots”. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 623–670. ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_26.

[100] Jan Lexell, Charles C. Taylor, and Michael Sjöström. “What is the cause of the ageing atrophy?: Total number, size and proportion of different fiber types studied in whole vastus lateralis muscle from 15- to 83-year-old men”. In: *Journal of the Neurological Sciences* 84.2 (1988), pp. 275–294. ISSN: 0022-510X. DOI: 10.1016/0022-510X(88)90132-3.

[101] Ulrich Lindemann et al. “Problems of older persons using a wheeled walker”. In: *Aging Clinical and Experimental Research* 28.2 (2016), pp. 215–220. ISSN: 1720-8319. DOI: 10.1007/s40520-015-0410-8.

[102] Lillemor Lundin-Olsson, Lars Nyberg, and Yngve Gustafson. ““Stops walking when talking” as a predictor of falls in elderly people”. In: *The Lancet* 349.9052 (Mar. 1997), p. 617. DOI: 10.1016/s0140-6736(97)24009-2.

[103] Lars Lünenburger, Gery Colombo, and Robert Riener. “Biofeedback for robotic gait rehabilitation”. In: *Journal of neuroengineering and rehabilitation* 4 (Jan. 2007), pp. 1–1. ISSN: 1743-0003. DOI: `10.1186/1743-0003-4-1`.

[104] Sara M. Bradley and Cameron R. Hernandez. “Geriatric Assistive Devices”. In: *American family physician* 84 (Aug. 2011), pp. 405–11.

[105] S. MacNamara and G. Lacey. “A smart walker for the frail visually impaired”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065).* Vol. 2. Apr. 2000, 1354–1359 vol.2. DOI: `10.1109/ROBOT.2000.844786`.

[106] Henry W. Mahncke et al. “Memory enhancement in healthy older adults using a brain plasticity-based training program: a randomized, controlled study”. In: *Proceedings of the National Academy of Sciences of the United States of America* 103.33 (Aug. 2006). 16888038[pmid], pp. 12523–12528. ISSN: 0027-8424. DOI: `10.1073/pnas.0605194103`.

[107] William C. Mann et al. “An Analysis of Problems with Walkers Encountered by Elderly Persons”. In: *Physical & Occupational Therapy In Geriatrics* 13.1-2 (1995), pp. 1–23. DOI: `10.1080/J148v13n01_01`.

[108] M. Martins, C. Santos, and A. Frizera. “Online control of a mobility assistance Smart Walker”. In: *2012 IEEE 2nd Portuguese Meeting in Bioengineering (ENBENG).* Feb. 2012, pp. 1–6. DOI: `10.1109/ENBENG.2012.6331388`.

[109] M. Martins et al. “A new integrated device to read user intentions when walking with a Smart Walker”. In: *2013 11th IEEE International Conference on Industrial Informatics (INDIN).* July 2013, pp. 299–304. DOI: `10.1109/INDIN.2013.6622899`.

[110] M. Martins et al. “Design, implementation and testing of a new user interface for a smart walker”. In: *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC).* May 2014, pp. 217–222. DOI: `10.1109/ICARSC.2014.6849789`.

[111] M. Martins et al. “Legs tracking for walker-rehabilitation purposes”. In: *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics.* Aug. 2014, pp. 387–392. DOI: `10.1109/BIOROB.2014.6913807`.

[112] M. Martins et al. “Smart walker use for ataxia’s rehabilitation: Case study”. In: *2015 IEEE International Conference on Rehabilitation Robotics (ICORR).* Aug. 2015, pp. 852–857. DOI: `10.1109/ICORR.2015.7281309`.

[113] Maria Martins et al. “A review of the functionalities of smart walkers”. In: *Medical Engineering & Physics* 37.10 (2015), pp. 917–928. ISSN: 1350-4533. DOI: `10.1016/j.medengphy.2015.07.006`.

[114] Maria M. Martins et al. “Assistive mobility devices focusing on Smart Walkers: Classification and review”. In: *Robotics and Autonomous Systems* 60.4 (2012), pp. 548–562. ISSN: 0921-8890. DOI: 10.1016/j.robot.2011.11.015.

[115] *MATLAB*. The Mathworks, Inc. Natick, Massachusetts, 1994-2020.

[116] Alexander Mayer. “Design of a Reconfigurable Mobile Training System”. Bachelor’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Mar. 2018, p. 68.

[117] S. McLachlan et al. “A multi-stage shared control method for an intelligent mobility assistant”. In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* June 2005, pp. 426–429. DOI: 10.1109/ICORR.2005.1501134.

[118] D. W. Meer and S. M. Rock. “Coupled-system stability of flexible-object impedance control”. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. Vol. 2. May 1995, 1839–1845 vol.2. DOI: 10.1109/ROBOT.1995.525536.

[119] Anat Mirelman et al. “Virtual Reality for Gait Training: Can It Induce Motor Learning to Enhance Complex Walking and Reduce Fall Risk in Patients With Parkinson’s Disease?” In: *The Journals of Gerontology: Series A* 66A.2 (Nov. 2010), pp. 234–240. ISSN: 1079-5006. DOI: 10.1093/gerona/glq201. eprint: https://academic.oup.com/biomedgerontology/article-pdf/66A/2/234/1611741/glq201.pdf.

[120] A. Morris et al. “A robotic walker that provides guidance”. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. Vol. 1. Sept. 2003, 25–30 vol.1. DOI: 10.1109/ROBOT.2003.1241568.

[121] J. C. Morris, A. Heyman, R. C. Mohs, et al. “Clinical and neuropsychological assessment of Alzheimer’s disease. Neurology”. In: *The Consortium to Establish a Registry for Alzheimer’s Disease (CERAD) Part I.* Vol. I. 1989. Chap. 39, pp. 1159–1165.

[122] K. Mun et al. “Design of a novel robotic over-ground walking device for gait rehabilitation”. In: *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*. Mar. 2014, pp. 458–463. DOI: 10.1109/AMC.2014.6823325.

[123] Dominik Muth. “Adaptive Force-Torque Control for a Mobile Training Device”. Bachelor’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Oct. 2017, p. 65.

[124] Manuel Muth. “Compliant control for a Robot-Based Training System”. Bachelor’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, June 2019, p. 75.

[125] Y. Nemoto et al. "Power-assisted walking support system for elderly". In: *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286)*. Vol. 5. Nov. 1998, 2693–2695 vol.5. DOI: `10.1109/IEMBS.1998.745229`.

[126] Tiia Ngandu et al. "A 2 year multidomain intervention of diet, exercise, cognitive training, and vascular risk monitoring versus control to prevent cognitive decline in at-risk elderly people (FINGER): a randomised controlled trial". In: *The Lancet* 385.9984 (June 2015), pp. 2255–2263. ISSN: 0140-6736. DOI: `10.1016/S0140-6736(15)60461-5`.

[127] Bondhan Novandy, Jungwon Yoon, and Auralius Manurung. "Interaction control of a programmable footpad-type gait rehabilitation robot for active walking on various terrains". In: July 2009, pp. 372–377. DOI: `10.1109/ICORR.2009.5209542`.

[128] Denise C. Park and Patricia Reuter-Lorenz. "The adaptive brain: aging and neurocognitive scaffolding". In: *Annual review of psychology* 60 (2009), p. 173. DOI: `10.1146/annurev.psych.59.103006.093656`.

[129] Philipp Pelcz. "Bestimmung der 6D Position von Füßen und Unterbeinen eines Menschen bei der Interaktion mit einem mobilen Trainingssystem". Bachelor's Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Oct. 2019, p. 48.

[130] A. Pereira, N. F. Ribeiro, and C. P. Santos. "A Preliminary Strategy for Fall Prevention in the ASBGo Smart Walker *". In: *2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG)*. Feb. 2019, pp. 1–4. DOI: `10.1109/ENBENG.2019.8692533`.

[131] Frederico Pieruccini-Faria, Yanina Sarquis-Adamson, and Manuel Montero-Odasso. "Mild Cognitive Impairment Affects Obstacle Negotiation in Older Adults: Results from "Gait and Brain Study"". In: *Gerontology* 65 (Oct. 2018), pp. 1–10. DOI: `10.1159/000492931`.

[132] Lynn Poynter et al. "Does Cognitive Impairment Affect Rehabilitation Outcome?" In: *Journal of the American Geriatrics Society* 59.11 (2011), pp. 2108–2111. DOI: `10.1111/j.1532-5415.2011.03658.x`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1532-5415.2011.03658.x`.

[133] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. Jan. 2009, pp. 1–6.

[134] Andrew Rentschler et al. "Clinical evaluation of Guido robotic walker". In: 45 (Feb. 2008), pp. 1281–93.

[135] Andrew Rentschler et al. "Intelligent walkers for the elderly: Performance and safety testing of VA-PAMAID robotic walker". In: *Journal of rehabilitation research and development* 40 (Sept. 2003), pp. 423–31. DOI: 10.1682/JRRD.2003.09.0423.

[136] Robot Care Systems. Online; accessed Oct. 31, 2020.

[137] M. J. Scherer. "The Impact of Assistive Technology on the Lives of people with Disabilities". In: *Designing and Using Assistive Technology-The Human Perspective-* (1998), pp. 99.–116. DOI: 10.3109/17483107.2015.1027295.

[138] Michaela Schimpl et al. "Association between Walking Speed and Age in Healthy, Free-Living Individuals Using Mobile Accelerometry—A Cross-Sectional Study". In: *PLOS ONE* 6.8 (Aug. 2011), pp. 1–7. DOI: 10.1371/journal.pone.0023299.

[139] Michael Schwenk et al. "An Intensive Exercise Program Improves Motor Performances in Patients with Dementia: Translational Model of Geriatric Rehabilitation". In: 39 (2014). 3, pp. 487–498. ISSN: 1875-8908. DOI: 10.3233/JAD-130470.

[140] Michael Schwenk et al. "Dual-task performances can be improved in patients with dementia". In: *Neurology* 74.24 (2010). Therapy by dual-Tasking, pp. 1961–1968. ISSN: 0028-3878. DOI: 10.1212/WNL.0b013e3181e39696. eprint: https://n.neurology.org/content/74/24/1961.full.pdf.

[141] Michael Schwenk et al. "Sensor-based balance training with motion feedback in people with mild cognitive impairment". English (US). In: *Bulletin of Prosthetics Research* 53.6 (2016), pp. 945–958. ISSN: 0007-506X. DOI: 10.1682/JRRD.2015.05.0089.

[142] Scott W. Shaffer and Anne L. Harrison. "Aging of the Somatosensory System: A Translational Perspective". In: *Physical Therapy* 87.2 (Feb. 2007), pp. 193–207. ISSN: 0031-9023. DOI: 10.2522/ptj.20060083. eprint: http://oup.prod.sis.lan/ptj/article-pdf/87/2/193/9984278/ptj0193.pdf.

[143] D. C. Shah, M. Evans, and D. King. "Prevalence of mental illness in a rehabilitation unit for older adults". In: *Postgraduate Medical Journal* 76.893 (2000), pp. 153–156. ISSN: 0032-5473. DOI: 10.1136/pmj.76.893.153. eprint: https://pmj.bmj.com/content/76/893/153.full.pdf.

[144] Fei Shi et al. "Based on force sensing-controlled human-machine interaction system for walking assistant robot". In: *2010 8th World Congress on Intelligent Control and Automation*. July 2010, pp. 6528–6533. DOI: 10.1109/WCICA.2010.5554167.

[145] J. Shin et al. “SmartWalker: Towards an Intelligent Robotic Walker for the Elderly”. In: *2015 International Conference on Intelligent Environments*. July 2015, pp. 9–16. DOI: 10.1109/IE.2015.10.

[146] Jiwon Shin, Andrey Rusakov, and Bertrand Meyer. “SmartWalker: An intelligent robotic walker”. In: *Journal of Ambient Intelligence and Smart Environments* 8 (July 2016), pp. 383–398.

[147] Anke H. Snijders et al. “Neurological gait disorders in elderly people: clinical approach and classification”. In: *The Lancet Neurology* 6.1 (2007), pp. 63–74. ISSN: 1474-4422. DOI: 10.1016/S1474-4422(06)70678-0.

[148] Page Solenne et al. “Smart walkers: an application-oriented review”. In: *Robotica* -1 (Feb. 2016), pp. 1–20. DOI: 10.1017/S0263574716000023.

[149] Kim Sooyeon. “Exploring the field application of combined cognitive-motor program with mild cognitive impairment elderly patients”. In: *J Exerc Rehabil* 14.5 (2018), pp. 817–820. DOI: 10.12965/jer.1836418.209. eprint: http://www.e-jer.org/journal/view.php?number=2013600583.

[150] M. Spenko, Haoyong Yu, and S. Dubowsky. “Analysis and design of an omnidirectional platform for operation on non-ideal floors”. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*. Vol. 1. May 2002, 726–731 vol.1. DOI: 10.1109/ROBOT.2002.1013444.

[151] M. Spenko, H. Yu, and S. Dubowsky. “Robotic Personal Aids for Mobility and Monitoring for the Elderly”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14.3 (Sept. 2006), pp. 344–351. ISSN: 1534-4320. DOI: 10.1109/TNSRE.2006.881534.

[152] Matthew J. Spenko. “Design and analysis of the SmartWalker : a mobility aid for the elderly”. MA thesis. 2001.

[153] D. Stogl, B. Hein, and M. Mende. “Overview of a Robot for a Neuromuscular Training – RoboTrainer”. In: *2019 European Conference on Mobile Robots (ECMR)*. Sept. 2019, pp. 1–6. DOI: 10.1109/ECMR.2019.8870955.

[154] D. Stogl et al. “Robot-Based Training for People With Mild Cognitive Impairment”. In: *IEEE Robotics and Automation Letters* 4.2 (Apr. 2019), pp. 1916–1923. ISSN: 2377-3766. DOI: 10.1109/LRA.2019.2898470.

[155] D. Stogl et al. “Spatial Control Actions for a Physical Training with a Smart Walker”. In: *2019 International Symposium ELMAR*. Sept. 2019, pp. 131–134. DOI: 10.1109/ELMAR.2019.8918658.

[156] Denis Stogl et al. “A Technical system for physical activation of persons with mild cognitive impairment”. In: *Technische Unterstützung für Menschen mit Demenz : Symposium 30.09. - 01.10.2013. Hrsg.: T. Schultz.* Ed. by Tanja Schultz, Felix Putze, and Andreas Kruse. Karlsruhe: KIT Scientific Publishing, Karlsruhe, 2014, pp. 123–143. ISBN: 978-3-7315-0258-6. DOI: `10.5445/KSP/1000042907`.

[157] Denis Stogl et al. “Control of a Smart Walker for Training Using Interaction-Energy and Personalized Parameters”. In: *21st IFAC World Congress (Proceedings Preprint).* 21th IFAC World Congress. Berlin, Germany, July 2020.

[158] R. Tan et al. “Adaptive control of an omni-directional walker considering the forces caused by user”. In: *2013 IEEE International Conference on Mechatronics and Automation.* Aug. 2013, pp. 761–766. DOI: `10.1109/ICMA.2013.6618012`.

[159] R. Tan et al. “Adaptive controller for motion control of an omni-directional walker”. In: *2010 IEEE International Conference on Mechatronics and Automation.* Aug. 2010, pp. 156–161. DOI: `10.1109/ICMA.2010.5589099`.

[160] R. Tan et al. “Path tracking control of an omni-directional walker considering pressures from a user”. In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* July 2013, pp. 910–913. DOI: `10.1109/EMBC.2013.6609649`.

[161] Renpeng Tan et al. “Adaptive control method for path-tracking control of an omni-directional walker compensating for center-of-gravity shifts and load changes”. In: *International journal of innovative computing, information & control: IJICIC* 7 (July 2011), pp. 4423–4434. eprint: `http://www.ijicic.org/10-06053-1.pdf`.

[162] Renpeng Tan et al. “Path tracking control considering center of gravity shift and load change for an omni-directional walker”. In: July 2010, pp. 672–675. DOI: `10.1109/ICINFA.2010.5512419`.

[163] M. Tani et al. “Internal model control for assisting unit of wheeled walking frames”. In: *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.* Vol. 2. Sept. 2004, 928–933 Vol.2. DOI: `10.1109/CCA.2004.1387488`.

[164] David Tomsu. “Integration der Messung von Bio-Signalen in ein roboterbasiertes Trainings- system”. Bachelor’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, 1. Juni 2019.

[165] United Nations, Department of Economic and Social Affairs Population Division. *World Population Ageing 2015.* Electronic. New York: United Nations, 2015.

[166] United Nations, Department of Economic and Social Affairs, Population Division. *World Population Ageing 2019 (ST/ESA/SER.A/444)*. Electronic. New York: United Nations, 2020.

[167] Carlos Valadão et al. "A New Controller for a Smart Walker Based on Human-Robot Formation". eng. In: *Sensors (Basel, Switzerland)* 16.7 (July 2016). 27447634[pmid], p. 1116. ISSN: 1424-8220. DOI: 10.3390/s16071116.

[168] Michael Vassallo et al. "A prospective observational study of outcomes from rehabilitation of elderly patients with moderate to severe cognitive impairment". In: *Clinical Rehabilitation* 30.9 (2016). PMID: 27496699, pp. 901–908. DOI: 10.1177/0269215515611466.

[169] Luigi Villani and Joris De Schutter. "Force Control". In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 195–220. ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_9.

[170] M. Wada and H. H. Asada. "Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs". In: *IEEE Transactions on Robotics and Automation* 15.6 (Dec. 1999), pp. 978–989. ISSN: 1042-296X. DOI: 10.1109/70.817663.

[171] J. M. H. Wandosell and B. Graf. "Non-holonomic navigation system of a walking-aid robot". In: *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*. 2002, pp. 518–523. DOI: 10.1109/ROMAN.2002.1045674.

[172] Xingbo Wang. "Online approach for assessing the interaction of a user with a robot-based training system". Master's Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Mar. 2019, p. 65.

[173] G. Wasson et al. "A physics-based model for predicting user intent in shared-control pedestrian mobility aids". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 2. Sept. 2004, 1914–1919 vol.2. DOI: 10.1109/IROS.2004.1389677.

[174] G. Wasson et al. "Effective Shared Control in Cooperative Mobility Aids". In: *In Proceedings of the Fourteenth international Florida Artificial intelligence Research Society Conference (May 21 - 23*. AAAI Press, 2001, pp. 509–513.

[175] G. Wasson et al. "User intent in a shared control framework for pedestrian mobility aids". In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. Vol. 3. Oct. 2003, 2962–2967 vol.3. DOI: 10.1109/IROS.2003.1249321.

[176] Alex Weber. “Bestimmung der Gangparameter eines Menschen aus Kraftdaten bei der Interaktion mit einem Smarten Rollator”. Karlsruhe, Germany: Karlsruhe Institute of Technology, p. 82.

[177] David Wechsler et al. *Wechsler Memory Scale-Revised Edition. Manual New York*. The Psychological Corporation, 1987.

[178] Peter René Wern. “High-Level Control Concepts for a Robot-Based Training System”. Bachelor’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Mar. 2018, p. 66.

[179] B. Winblad et al. “Mild cognitive impairment – beyond controversies, towards a consensus: report of the International Working Group on Mild Cognitive Impairment”. In: *Journal of Internal Medicine* 256.3 (Aug. 2004), pp. 240–246. DOI: 10.1111/j.1365-2796.2004.01380.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2796.2004.01380.x.

[180] David A. Winter. *The biomechanics and motor control of human gait : normal, elderly and pathological*. English. Waterloo (Ontario): Waterloo Biomechanics, 1991. ISBN: 0-88898-105-8 978-0-88898-105-9.

[181] World Health Organization. *10 facts on dementia*. Online; accessed 19th November 2015. World Health Organization, 2015.

[182] Yunyou Yang et al. “Design of mechanical structure and tracking control system for lower limbs rehabilitative training robot”. In: *2009 International Conference on Mechatronics and Automation*. Aug. 2009, pp. 1824–1829. DOI: 10.1109/ICMA.2009.5246401.

[183] J. Ye et al. “Development of a width-changeable intelligent walking-aid robot”. In: *2012 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*. Nov. 2012, pp. 358–363. DOI: 10.1109/MHS.2012.6492438.

[184] Haoyong Yu, Steven Dubowsky, and Adam Skwersky. “Omni-Directional Mobility Using Active Split Offset Castors”. In: *In Proceedings of 2000 ASME IDETC/-CIE 26th Biennial Mechanics and Robotics Conference*. Baltimore, Maryland, Sept. 2000.

[185] Haoyong Yu, Matthew Spenko, and Steven Dubowsky. “An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly”. In: *Autonomous Robots* 15.1 (July 2003), pp. 53–66. ISSN: 1573-7527. DOI: 10.1023/A:1024488717009.

[186] Haoyong Yu, Matthew Spenko, and Steven Dubowsky. “Omni-Directional Mobility Using Active Split Offset Castors”. In: *Journal of Mechanical Design - J MECH DESIGN* 126 (Sept. 2004). DOI: 10.1115/1.1767181. eprint: http://www.ijicic.org/10-06053-1.pdf.

[187] Wuxiao Zhou, Li Xu, and Jie Yang. “An intent-based control approach for an intelligent mobility aid”. In: *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*. Vol. 2. Mar. 2010, pp. 54–57. DOI: `10.1109/CAR.2010.5456618`.

[188] Robert Zimmermann. “Position Estimation of the Upper Body during Human-Robot-Interaction”. Bachelor’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Feb. 2018, p. 63.

[189] Daniel Zumkeller. “Adaptive Control of a Robotic System for Training and Walking Support”. Master’s Thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology, Oct. 2019, p. 80.

Appendix

A. Overview of Smart Walker

This section provides a condensed overview in table form of the SWs relevant to this thesis. Most of those SWs are mentioned in the overview in section 2.3.1. The following tables present general data and functionalities (table A.1), design characteristics (table A.2), and control properties (table A.3). Some of the criteria used in the tables are inspired by SWs overviews presented by Martins et al. [113], Solenne et al. [148], and Alves et al. [4]. Though all these overviews mention general functionalities, this overview associates those with the corresponding publication and provides detailed insights into SWs' mechatronic design and control details. The investigated criteria with their description for each table are listed hereunder.

General data and functionalities (Table A.1)

Walker SW's name, citation to the relevant publication introducing it, and reference to the SW's figure in this thesis.

Institution / Year – Publications Name of the institution(s) at which the SW was developed. Years of active development and a list of publications related to SW.

Main Purpose The main purpose for which SW is designed.

Target Population The population for which the SW is designed.

Functionalities List of functionalities the SW realizes with a list of publications where those functionalities are presented or mentioned. If there is no publication listed, then functionality is presented in the default publication.

User-Studies List of user studies done with the SW and the corresponding publication.

Design characteristics (Table A.2)

Base Type / User Position Type of the SW's base, e.g., mobile robot base, WW-shaped.

Kinematics / Actuation The kinematics, actuation, and list of passive/active wheels of SW.

Handles-Type Type of handles used for SW, e.g., forearm support, bicycle handles, WW handles.

Mechanical adaption Possibilities for the SW's mechanical adaption.

Mechatronic Properties Some specific design and mechatronic properties if provided in publications.

Control properties (Table A.3)

Sensors and Data for Control Sensor and data most relevant for SW's control.

Control Algorithm List of control algorithms used for the SW.

Control Modes List of control modes with the corresponding publication.

Sensors / Functionalities Other sensors and through them supported functionalities of the Sw.

Safety Measures List of implemented safety measures in the SW.

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	Institution / Year – Publications	Main Purpose	Target Population	Functionalities	User-Studies
PAM-AID concept prototype[91]	Trinity College, Dublin, Ireland 1995-1997 [91, 93] figure 2.2a	navigation assistance for visualy impaired	visually impaired elderly persons	○ physical support ○ wall following ○ slow-down and stop before collision ○ audio/speech feedback	○ evaluation by able bodied non-technical persons
PAM-AID second prototype [92]	Trinity College, Dublin, Ireland 1997-1999 [92, 93] figure 2.2b	navigation assistance for visually impaired	visually impaired elderly persons	○ physical support ○ stairs detection ○ obstacle avoidance	○ eight subjects from residential homes for visually impaired
PAM-AID [105]	Trinity College, Dublin, Ireland 2000 [105]	navigation assistance for visualy impaired	frail visually impaired persons	○ adaptive braking ○ assistive mode ○ audio/speech feedback ○ corridor recognition ○ collision avoidance	○ twelve subjects from a residential home for visually impaired
PAMM SmartWalker [184] figure 2.3b	MIT, USA 2000-2006 [40, 152, 150, 185, 186, 151]	mobility assistance and monitoring	Elderly with mobility difficulty due to physical frailty and/or disorientation due to aging and sickness.	○ physical stability ○ localization system using markers on the celing ○ adaptive shared control (free to autonomous) ○ health monitoring (pulse) ○ gait monitor (velocity power spectrum)	○ six subjects from elderly care center (84 to 95 years of age)

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	Institution / Year – Publications	Main Purpose	Target Population	Functionalities	User-Studies
MARC Smart Walker[174] / COOL Aide [71] figure 2.4	Medical Automation Research Center 2001-2007 [175, 173, 5, 7, 71, 6]	navigation assistance for elderly and visually imparied	elderly or visualy imparied	○ obstacle and cliff detection ○ path following ○ shared control - assistance and collision avoidance ○ user's intent [173] ○ 2D grid stype mapping – HIMM [71] ○ gait characteristics from forces[6]	○ lab experiments with 8 healty subjects (3 above 65) (Vicon Tracking)[173] ○ lab experiemnts with 22 healty subjects (15 above 65) [7]
RT Walker [65] figure 2.5a	Tohoku University, Japan 2004-2007 [64, 68, 29, 63, 67, 66]	passive, controllable walker	elderly, handicapped and blind persons	○ rehabilitation - increase the load to the user ○ collision avoidance ○ cliff detection [64] ○ slope detection [63] ○ fall prevention [66]	○ lab experiments with 5 healthy blindfolded persons[65]
Walking Helper [28] figure 2.5b	Tohoku University, Japan 2004-2007 [30, 68, 26, 27, 31, 29]	active walker to compensate its weight	elderly	○ collision avoidance ○ individual CoR adaption ○ automatic CoR adaption [26] ○ passive control [31, 29]	○ lab experiments [28] ○ lab experiments with elderly simulator (TMI2000-Japan) [30] ○ lab experiments [31]

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	**Institution /** **Year – Publications**	**Main Purpose**	**Target Population**	**Functionalities**	**User-Studies**
GUIDO [134] Figure 2.2d	Human Engineering Research Laboratories, Department of Veterans Affairs (VA); Trinity College, Dublin Ireland; Haptica Ltd. 2001-2008 [134, 90]	navigation assistance for visually impaired	frail visually impaired persons	○ physical support ○ CleanSweep collision avoidance - shared control ○ audio/speech feedback ○ SLAM	○ test with subjects from a residential home for visually impaired; device is certified product
SIMBIOSIS [51] figure 2.6a	Instituto de Automática Industrial — CSIC, Spain [52, 53]	walking support, gait monitoring	persons with mobility issues	○ physical support ○ gait monitoring	○ clinical validation with eight patients [53]
AZIMUT-3 [48] figure 2.7	Université de Sherbrooke, Canada 2010 [49]	development of a natural interface to interact with a robot	elderly or disabled persons		○ lab evaluation by the authors
JARoW [97] figure 2.8	Japan Advanced Institute of Science and Technology (JAIST), Japan 2010-2014[95, 96]	active walker with a natural user interface	elderly	○ physical support ○ collision avoidance (velocity reduction)	○ five elderly subjects in everyday situations – controlled environment [96]
Omni-Directional Walker (ODW) [85] figure 2.9	Kochi University of Technology, Japan [182, 159, 162, 161, 158, 160]	rehabilitation of lower limbs	disaled or injured persons	○ path tracking ○ gait training programs	○ laboratory experiments only

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	Institution / Year – Publications	Main Purpose	Target Population	Functionalities	User-Studies
Adaptive System Behaviour Group [108] figure 2.10a figure 2.10b [4]	Minho University Guimarães, Portugal [109, 44, 110, 111, 112, 19, 20, 130]	rehabilitation of patients with Ataxia	elderly and disabled persons	○ user-walker distance (IR sensors) and detect falls (FRS) ○ body-motion feedback [19] ○ database of gait analysis [20]	○ lab experiments with 11 healthy volunteers [108] ○ lab experiments with 10 healthy subjects [111] ○ one patient with ataxia, 3 weeks use [112]
UFES Walker [50] figure 2.6b	Federal University of Espirito Santo, Brazil [32, 167, 33, 94]	gait analysis	elderly and persons with gait issues	○ physical stability and motion support ○ biomechanical monitoring ○ health assessment	○ lab evaluations
SmartWalker [145] figure 2.11	ETH Zürich, Creative Computer Software AG, Switzerland; Innopolis University, Kazan, Russia [146]	assistance to elderly in care facilities	elderly	○ gesture control ○ autonomous drive to park position ○ assistance on the slopes ○ gait tracking	○ evaluation with 23 residents from 5 elderly care facilities [146]
ASBGo++ [4] figure 2.10c figure 2.10d	Minho University Guimaraes, Portugal; Orthos XXI 2016- [3, 130]	rehabilitation of patients with Ataxia	elderly and disabled persons	○ physical support ○ navigation and localization ○ bio-mechanical monitoring ○ remote control ○ fall prevention [130]	almost a product
			Commercial Smart Walkers		

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	Institution / Year – Publications	Main Purpose	Target Population	Functionalities	User-Studies
ello – Der elektrische Rollator [56] figure 2.12a	WMT GmbH, Stuttgart, Germany 2017-2019	gait assistance	elderly	○ gait assistance ○ assistance on slopes ○ emergency call ○ folding possibility ○ gps tracker ○ use with empty battery	product
beactive+e E-Rollator [55] figure 2.12c	BEMOTEC GmbH, Reutlingen, Germany 2013-	geriatric assistance and rehabilitation	elderly and persons after a stroke	○ gait assistance ○ assistance on slopes ○ assistance on curbs ○ folding possibility ○ rehabilitation programs	product

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	Institution / Year – Publications	Main Purpose	Target Population	Functionalities	User-Studies
LEA [136] figure 2.12b	Robot Care Systems, Delft, Netherlands (bankruptcy in 2019) 2014-2019	geriatric assistance and rehabilitation	elderly and persons with dementia, traumatic brain injury, Parkinson disease	○ gait assistance ○ single arm use ○ assistance on slopes ○ dancing with the robot ○ rehabilitation programs ○ heath status monitoring ○ sit-to-stand assistance ○ autonomous navigation and localization ○ emergency and personal assistance services (integrated tablet)	product
		Smart Walker from this thesis			

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	Institution / Year – Publications	Main Purpose	Target Population	Functionalities	User-Studies
RoboTrainer Prototype [156]	Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics 2014-2019 [156, 154, 155, 157]	training for persons with MCI	elderly with MCI	○ gait assistance ○ mapping, localization and navigation ○ training programs – virtual forces [10] ○ GUI for editing training scenarios [153, 59] ○ training programs – control actions [155, 178] ○ posture monitoring [188, 176] ○ gait monitoring [11] ○ health monitoring [164] ○ passive and per-user control [124, 189] ○ user-feedback using RGBD-LED stripe	○ evaluation with 10 subjects [154]

Table A.1.: Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature.

Walker	Institution / Year – Publications	Main Purpose	Target Population	Functionalities	User-Studies
RoboTrainer v2 [153]	Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics 2018- [153, 116, 157]	force and motor training	healthy adults, elderly and persons with MCI	○ gait assistance ○ mapping, localization and navigation ○ adaptable footprint [155, 116] ○ GUI for editing training scenarios [153, 59] ○ training programs – control actions [155, 178] ○ posture monitoring [188, 176] ○ gait monitoring [11, 129] ○ health monitoring [164] ○ passive, adaptive and per-user control [157, 124, 189] ○ user-feedback using RGBD-LED stripe and display	○ evaluation with 13 subjects (this thesis)

Table A.2.: Overview of the design properties of the Smart Walkers from the literature.

Walker	Base Type / User Position	Kinematics / Actuation	Handles-Type	Mechanical adaption	Mechatronic Properties
PAM-AID concept prototype[91]	○ mobile robot base ○ rectangular (Labmate robot base) ○ behind rear axis	○ active	handrail		
PAM-AID second prototype [92]	○ motorized off-the-shelf rollator ○ rear axis	○ differential ○ motorized rear fixed wheels; front castor wheels ○ active	handles	height adjustable handles	
PAM-AID [105]	○ WW-shaped - as conventional walker ○ rear axis	○ differential ○ passive fixed rear wheels, active steering with front castor wheels ○ passive	bicycle-like handles		
PAMM SmartWalker [184]	○ WW-like - custom frame ○ rear axis	○ omni-directional ○ passive castor rear-wheels; ASOC front wheels ○ active	bicycle-like handles	handle height	only for SmartCane: ○ $M = 15\,\text{kg}$ ○ $v_{max} = 0.5\,\text{m/s}$ ○ $M_{ctrl} = 10\,\text{kg}$ ○ $D_{ctrl} = 30\,\text{N s/m}$
MARC [174]	○ three wheeled, v-shaped off-the-shelf walker (Invacare) ○ behind rear axis	○ differential ○ passive	walker handles	handles height	
RT Walker [65]	WW-shaped, custom frame behind rear axis	passive	○ differential ○ fixed wheels behind; two castor wheels in front	walker-like handles	

Table A.2.: Overview of the design properties of the Smart Walkers from the literature.

Walker	Base Type / User Position	Kinematics / Actuation	Handles-Type	Mechanical adaption	Mechatronic Properties
Walking Helper [28]	mobile robot based - custom hand rests behind rear axis	omni-directional - swedish wheels active	forearm support		○ $M = 80\,\mathrm{kg}$ ○ $V_{x_{max}} = 0.75\,\mathrm{m/s}$ ○ $V_{y_{max}} = 1.3\,\mathrm{m/s}$ ○ $\omega_{max} = 3\,\mathrm{rad/s}$ ○ Stable: 60 kg; 30 N s/m ○ Unstable: 20 kg; 30 N s/m
GUIDO [134]	○ U-shaped ○ rear axis	○ differential ○ active steering with front wheels ○ passive	bicycle-like handles		LRF - 38 cm, 30° to the floor - dropdown/stairs detection
SIMBIOSIS [51]	WW-like – custom frame rear axis	○ differential ○ passive castor front wheels; fixed motorized back wheels ○ active	forearm support	handle height	
AZIMUT-3 [48]	mobile robot base – custom frame gehind rear axis	○ omni-directional (quasi-holonomic) ○ four steerable ”Azimut“ wheels – castor wheel: DC brushless drive motor and differential elastic actuator as steer actuator ○ active	improvised handles		○ $M = 34\,\mathrm{kg}$ payload ○ NiMH batteries ○ ROS

Table A.2.: Overview of the design properties of the Smart Walkers from the literature.

Walker	Base Type / User Position	Kinematics / Actuation	Handles-Type	Mechanical adaption	Mechatronic Properties
JARoW [97]	circular shape - custom frame inside the shape	○ omni-directional (3 omni-wheels) ○ active	forearm support	forearm support plate height 825 mm to 1000 mm	○ $V_{max} = 6.58\,\mathrm{km\,h^{-1}}$ ○ WxLxH - 880 mm x 770 mm x 1000 mm ○ $M = 20\,\mathrm{kg}$
Omni-Directional Walker (ODW) [85]	WW-like – custom frame rear axis	○ omni-directional (4 omni-wheels) ○ active	forearm support	height forearm support	○ touch panel ○ physical parameters in [159]
Adaptive System Behaviour Group [108]	WW-shaped - modified frame rear axis	○ differential ○ passive castors front; motorized fixed rear wheels ○ active	forearm support with vertical handgrips and horizontal handgrips	height of the forearm support	○ low-Cost development: Arduino controller ○ motors: $V_n =$ 40 revolution/min, $T_n = 5\,\mathrm{N\,m}$ ○ $V_{max} = 0.3$ m/s ○ ROS
UFES Walker [50]	WW-shaped - custom frame; three wheels rear axis	○ differential ○ passive castor front; motorized fixed rear wheels ○ active	forearm support with vertical handles		○ Some technical detils in [50]
SmartWalker [145]	WW-shaped - modified frame	○ differential ○ passive castor front; motorized fixed rear wheels ○ active	handles		○ touch screen for interaction with a user ○ Roboscoop and ROS

Table A.2.: Overview of the design properties of the Smart Walkers from the literature.

Walker	Base Type / User Position	Kinematics / Actuation	Handles-Type	Mechanical adaption	Mechatronic Properties
ASBGo++ [4]	WW-like – custom frame rear axis	○ differential ○ passive castors front; motorized fixed rear wheels ○ active	forearm support with vertical handgrips and horizontal handgrips	○ electrical height adjustment ○ back posture ○ handlebare width adjustment	○ gait's area: 58 cm (W) x 69 cm (L)
		Commercial Smart Walkers			
ello – Der elektrische Rollator [56]	WW-like	○ differential ○ passive castors front; motorized fixed rear wheels ○ active	handles	handle height	○ $M = 14\,\mathrm{kg}$ ○ $V_{max} = 5.5\,\mathrm{km/s}$ ○ $T_{battery} = 3\,\mathrm{h}$ ○ basket and seat
beactive+e E-Rollator [55]	WW-like	○ differential ○ passive castors front; motorized fixed rear wheels ○ active	handles	handle height	○ $T_{battery} = 3$ h to 9 h ○ basket, seat, different battery sizes
LEA [136]	WW-like	○ differential ○ passive castors front; motorized fixed rear wheels ○ active	○ handles ○ sit-to-stand handles	handle height	○ display, seat, lights different battery sizes
		Smart Walker from this thesis			
RoboTrainer Prototype [156]	rectangular – mobile robot base behind rear axis	omni-directional – 4 steer-drive modules active	bike handlebar, ergonomic grips		○ LED Stripe for user's feedback

Table A.2.: Overview of the design properties of the Smart Walkers from the literature.

Walker	Base Type / User Position	Kinematics / Actuation	Handles-Type	Mechanical adaption	Mechatronic Properties
RoboTrainer v2 [153]	triangle (behind) rear axis – depends on rear-wheels configuration	omni-directional – 3 steer-drive modules active	bike handlebar, ergonomic grips	○ handle height ○ footprint – reconfiguration of the rear wheels	○ LED Stripe for user's feedback ○ display for a user ○ laser-pointer markers on the floor

Table A.3.: Overview of the control features of the Smart Walkers from the literature.

Walker	Sensors and Data for Control	Control Algorithm	Control Modes	Sensors / Functionalities	Safety Measures
PAM-AID concept prototype[91]	joystick button for mode-switching		direct joystick control wall following	○ infrared proximity switches – wall following ○ bumpers – collision ○ sonar – navigation	
PAM-AID second prototype [92]	○ Custom – micro switches detecting handle's movements ○ buttons for choice of direction ○ button for mode switching		○ manual ○ assistive	○ sonar – distance, collision ○ bumpers – collision	○ safety confirmation switch
PAM-AID [105]	○ spring loaded handle rotation ±15° – hall sensor ○ switch for "rotation-on-the-spot" function	transfer function	○ manual ○ assistive – obstacle avoidance	○ LRF – SICK LMS200 ○ sonar ring ○ encoders on rear wheels	○ passive system
PAMM SmartWalker [184]	6D – FTS	adaptive admittance control [185]	○ free driving, but safety monitoring ○ adaptive shared control ○ full computer control	○ camera – localization ○ sonar array – obstacle detection ○ ECG – pulse monitor	○ collison and stability implementd in software

Table A.3.: Overview of the control features of the Smart Walkers from the literature.

Walker	Sensors and Data for Control	Control Algorithm	Control Modes	Sensors / Functionalities	Safety Measures
MARC [174]	○ – [174] ○ 6D FTSs in each handle [173]	○ rule-based algorithms [174] ○ stability optimization [173]	○ safety control ○ navigational aid ○ goal achievement assistance	○ sonar ○ infrared sensors ○ LRF 180° (PBS-03JN Hokuyo) [71] ○ heading encoder on the front wheel ○ wheel encoders on the rear wheels	○ passive system ○ break levers
RT Walker [65]	2004-2007 [64, 29, 63, 67]	○ control based on brakes – admittance model ○ artificial potential field for collision avoidance	○ human-adaptive motion – variable dynamics ○ environment-adaptive motion ○ gravity compensation algorithm [64, 63] ○ caster-like dynamics controller [67]	○ encoders on the rear wheels ○ tilt-angle sensor – slope detection[64] ○ LRF tilted towards floor – stairs detection [64] ○ LRF toward user – fall prevention [66]	○ passive system
Walking Helper [28]	FTS under support frame – “body FTS”	○ admittance controller ○ variable CoR fuzzy controller ○ variable apparent dynamics based on environment	○ CoR training ○ admittance control ○ passivity monitor [31] ○ passive environment influence [29]	○ laser scanner – CoR training, collision avoidance	○ collision detection using “body FTS”

Table A.3.: Overview of the control features of the Smart Walkers from the literature.

Walker	Sensors and Data for Control	Control Algorithm	Control Modes	Sensors / Functionalities	Safety Measures
GUIDO [134]	○ FTS for steering ○ turn buttons on eah handlebar ○ switches for drive modes	○ proportional controller for steering ○ Bayesian network to interpret user goals	○ manual ○ assistive – shared control ○ park mode – front wheels locking	○ LRF (SICK LMS200) – mapping, stairs ○ 16 sonar sensors (Polaroid 7000) – short range obstacle ○ optical encoders on the fixed wheels ○ potentiometers on steering wheel	○ brake levers
SIMBIOSIS [51]	FTSs in handles	fuzzy controller after gait filtering from FTS data	manual	○ ultrasonic sensors – user-walker distance, gait evolution ○ intertial kinematics sensors – placed on the user ○ FTS under each forearm support – user's load, cadence estimation	
AZIMUT-3 [48]	Moment-Sensors in the wheels – force-torque measurements	admittance control scheme with measuring forces in the wheels	○ direct interaction	○ 6D FTS – ground truth	

Table A.3.: Overview of the control features of the Smart Walkers from the literature.

Walker	Sensors and Data for Control	Control Algorithm	Control Modes	Sensors / Functionalities	Safety Measures
JARoW [97]	infrared and LRF sensors – leg-positions	○ KF-based motion generation from user's legs positions ○ PID-based orientation from leg orientation	○ cooperative movement	○ rotating IR sensors for lower limbs (12-180 cm) ○ two LRF Hokuyo URG-04LX – user's legs, obstacle detection	
Omni-Directional Walker (ODW) [85]	four force sensors sensors under the forearm support (7 Hz sampling)	○ fuzzy control[85] ○ PID adaptive control (tracking minimization) [159] ○ model-based adaptive control (tracking minimization) [158]	○ compliance control ○ adaptive control	○ external cameras – path tracking	○ limited velocity to 0.25 m/s
Adaptive System Behaviour Group [108]	○ joystick [108] ○ potentiometers in custom made handlebars[109];	fuzzy logic	○ software calibration of the interface [109]	○ infrared (IR) sensor-user-walker distance ○ Force sensors resistive in handles and forearm support ○ LRF – leg tracking, gait monitoring [111] ○ camera – user's posture [19]	○ only forward movement ○ IR and force sensors detect distance and contact with a user
UFES Walker [50]	3D FTS under each forearm support and leg position from LRF	adaptive force filtering anf fuzzy logic	user manual mode	○ LRF – gait monitoring ○ IMU– slope detection	○ emergency braking base on software evaluations

Table A.3.: Overview of the control features of the Smart Walkers from the literature.

Walker	Sensors and Data for Control	Control Algorithm	Control Modes	Sensors / Functionalities	Safety Measures
SmartWalker [145]	○ camera – autonomous mode ○ LRF – assistive mode	k-NN classifier with dynamic time wraping	○ assistive mode ○ autonomus mode	LRF – gait tracking	○ manual break levers
ASBGo++ [4]	custom handlebars with potentiometers	fuzzy logic	○ user control mode	○ infrared (IR) sensor-user-walker distance ○ Force sensors resistive in handles and forearm support ○ LRF – leg tracking, gait monitoring ○ camera – user's posture monitoring	○ emergency-Stop Button ○ obstacle avoidance ○ fall detection
Commercial Smart Walkers					
ello – Der elektrische Rollator [56]	buttons	velocity control	○ buttons – lights, horn, sos call		○ manual breaks
beactive+e E-Rollator [55]	buttons	velocity control	○ user controlled ○ rehabiliation programs	○ IMU– slope detection	○ manual breaks ○ enable buttons
LEA [136]	FTS	admittance	○ user controlled ○ rehabiliation programs	○ IMU– slope detection (probably) ○ sonar – obstacle detection ○ 3D environment camera	○ manual breaks ○ emergency buttons on the sides ○ safety stop base on sonar sensors
Smart Walker from this thesis					

Table A.3.: Overview of the control features of the Smart Walkers from the literature.

Walker	Sensors and Data for Control	Control Algorithm	Control Modes	Sensors / Functionalities	Safety Measures
RoboTrainer Prototype [156]	6D FTS- ATI Mini 45	admittance control	○ adaptive force control [123, 189] ○ per-user individual control [157, 189] ○ passive control [124]	○ laser-scanner – mapping, localization and navigation ○ LRF – gait monitoring [11] ○ upper body 3D camera – posture estimation [188, 176] ○ feet 3D camera – gait monitoring [129] ○ O2/heart rate – health monitoring [164]	○ laser-scanner fields (not certified) ○ emergency stop buttons (not accessible easily)
RoboTrainer v2 [153]	6D FTS- ATI Mini 45	admittance control	○ adaptive force control [189] ○ per-user individual control [157, 189] ○ passive control [124]	○ laser-scanner – mapping, localization and navigation ○ LRF – gait monitoring [11] ○ upper body 3D camera – posture estimation [188, 176] ○ feet 3D camera – gait monitoring [129] ○ O2/heart rate – health monitoring [164]	○ laser-scanner fields ○ emergency stop buttons ○ wreless emergency stop button

B. RoboTrainer's Hardware Technical Specification

B.1. Safety Hardware

RoboTrainer v2 uses the following safety hardware:

Safety System SICK Flexi Soft Safe EFI-pro System

- **CPU** FX3-CPU000000
- **Gateway** FX3-GEPR00000
- **GPIO** FX3-XTIO84002 (2x)

Safety Laserscanner microScan3 Pro – EFI-pro: MICS3-CBAZ55ZA1P01 (3x)

Motor Controllers ELMO Gold Whistle - CAN (concrete model depends on motor type used in Care-o-Bot drive-steer modules)

Wireless Emergency Stop System

- **Transmitter** Tyro Remotes Indus 1S switch
- **Receiver** Tyro Remotes Gemini 1S

Emergency Stop Switches DECA A20L (4x)

B.2. Drive Steer Modules

The modules weight 8.0 kg, have a payload capacity of 75 kg (i.e., 50 kg in continuous operation), can achieve a maximal linear velocity of 1.595 m/s, maximal torque of 16.8 N m, and have a wheel diameter of 160 mm (RW6).

B.3. Force Torque Sensor

In both systems, RoboTrainer Prototype and RoboTrainer v2, Mini58 from ATI Industrial Automation force-torque sensor (FTS) is used. Technical details can be found on the manufacturer's website: `https://www.ati-ia.com/products/ft/ft_models.aspx?id=Mini58`.

The most important specifications for this thesis are listed in table B.4.

Table B.4.: Technical Specification of the ATI Mini58 force-torque sensor (FTS)

	Parameter	Value
Physical Specifications	Weight	0.345 kg
	Diameter	58 mm
	Height	30 mm
Single-Axis Overload	F_{xy}	±21 000 N
	F_z	±4800 N
	T_{xy}	±590 N m
	T_z	±800 N m
Measurement Range (SI-700-30 calibration)	F_{xy}	700 N
	F_z	1700 N
	T_{xy}	30 N m
	T_z	30 N m
Resolution (SI-700-30 calibration)	F_{xy}	1/6 N
	F_z	7/24 N
	T_{xy}	9/1600 N m
	T_z	1/320 N m

C. Data representation

C.1. Boxplots

The samples in the boxplots are represented as follows: inside the box are 50 % of the data, whiskers are 1.5 times the inner-quartile range or indicate minimum/maximum values, and outliers are represented with a red cross or a dot. The middle, somewhere red, line, and the area inside the rectangle represent the mean and standard deviation of measurements.

C.2. Data significance

If not stated explicitly, a two-tailed hypothesis is used, and a difference between two data sets is considered significant for $p < 0.05$.

The tests calculated a two-tailed hypothesis with a significance level of 0.05.

D. Documents for Evaluation of Individual Parameters and Adaptive Control

This section gives concrete details on evaluation with users regarding individual parameters and non-linear adaptive control. In the first subsection (section D.1), original scripts (in german) provided to users before the evaluation are provided.

D.1. Task description and clarification of use of user's data (in german)

Die Aufgabe

Bei dem Training soll ein Benutzer den RoboTrainer möglichst genau über einen vordefinierten Parcours entlang navigieren. Der gewünschte Pfad ist auf dem Boden durch eine Linie angezeigt. Zur vereinfachten Positionierung des Roboters auf die Bodenmarkierungen sind drei Referenzmarker auf dem RoboTrainer angebracht.

Gesammelte Daten

Sind sie mit dem Sammlung folgenden Daten einverstanden?

Während des Versuchs werden folgende von Ihnen stammende Daten gesammelt:

- Eingabekräfte in den RoboTrainer
- Position Ihrer Unterschenkel relativ zum RoboTrainer (Laserscanner-Daten)
- Position und Geschwindigkeit des RoboTrainers

Weiterhin werden wir auf Basis Ihrer Daten die folgenden Werte berechnen:

- Präzision der Navigation des RoboTrainers
- Virtuelle Kraftanpassung relativ zur Geschwindigkeit
- Benötigte Zeit für die Aufgabe
- Ihre Leistung während des Trainings (abstrakte Zahl aus mehreren Variablen)

Hinweise zur Bedienung des RoboTrainer

1. Art der Interaktion mit dem RoboTrainer (mit Beschreibung relevanter Komponenten):
   - Der RoboTrainer wird über den Lenker mit Anwendung von Kraft bewegt.
   - Der RoboTrainer berechnet ausgehend von der Eingabekraft eine Ausgabegeschwindigkei nach welcher er sich fortbewegt. Höhere Eingabekräfte resultieren hierbei in höheren Geschwindigkeiten. Die maximale Geschwindigkeit ist durch eine maximal zulässige Kraft beschränkt.
   - Bei Loslassen des Lenkers stoppt der RoboTrainer. Er bewegt sich nur unter Kraftaufwendung auf den Lenker fort.
2. Der RoboTrainer besitzt um sich herum ein Schutzfeld, der ihm von der Kollision mit statischen und dynamischen Hindernissen schützt (das Feld mit der Hand auslösen).
   - Das Feld kann während des Versuchs unabsichtlich verletzt werden, was zum einem sofortigen Stopp des RoboTrainers führt. In diesem Fall lassen sie den Robotrianer los und lassen den Versuchsleiter den RoboTrainer wieder in Start-Position zu fahren, von welcher aus die Aufgabe wiederholt wird.
3. Falls Sie sich in irgendeiner Situation gefährdet oder überfordert fühlen, lassen Sie einfach den Lenker des RoboTrainers los und bleiben ruhig stehen. Der RoboTrainer wird von selbst zum Halten kommen.

Hinweise zum Versuchsablauf

1. Der Versuch besteht aus drei Abschnitten, in welchen der Parcours insgesamt 5 mal durchlaufen werden soll. Zu Beginn der ersten und zweiten Phase wird außerdem noch eine kurze Parametrierung des Roboters an den Benutzer durchgeführt.
   - Vor dem Anfang der Parametrierungen wird der Versuchsleiter zunächst den Ablauf erklären und den Roboter zum Startpunkt fahren.
   - Vor dem Anfang der Parcoursdurchläufe wird der Versuchsleiter ebenfalls zunächst den Roboter zum Startpunkt bewegen.
2. Vor dem Start des praktischen Versuchsablaufes sollen Sie zunächst ein paar Basisfragen beantworten.
3. Nach Ablauf jeder Phase sollen Sie ein paar Aufgabenbezogene Fragen beantworten.

4. Auf die Aufforderung des Versuchsleiters sollen Sie mit der Parametrierungsaufgabe beziehungsweise dem Parcoursdurchlauf starten. Nachdem Sie den Zielpunkt des Parcours erreichen, bringen Sie den RoboTrainer zum Stehen und lassen den Lenker des RoboTrainers los, um die Aufgabe abzuschließen.
5. Bitte bewegen Sie den RoboTrainer erst nach Anweisung durch den Versuchsleiter fort.

Glossary

Ataxia Medical term for loss of voluntary coordination of muscle movements. Ataxia is frequently caused by cerebellar injuries.. 34, 49, 250

control action Smart Walker's behavior modifies used as building blocks for training. They modify users' input of SW's velocity to make training more challenging or help users. viii, 7, 8, 23, 42, 50, 58, 85, 114, 115, 125, 138, 147, 149–151, 162–164, 169, 173–175, 196, 204, 217, 220, 221, 273, 279

global control action Global behavior modifiers that change SW's behavior independently from its position in the training environment. 7, 115, 149–151, 169, 173, 176, 193, 274, 281

physical human-robot interaction An art of Human-Robot Interaction where a robot and a human have physical contact.. 5, 24, 126, 275

RGBD camera Type of a camera that provides depth data, i.e., distances to objects, togeather with RGB image. Such cameras are often used in robotics.. 97

RoboTrainer Prototype The first of the two devices used in this thesis. The device's base is the rob@work mobile platform which was extended with handles and a force-torque sensor for feasibility evaluation. iv, vii, viii, xv, 3–5, 7, 8, 23, 31, 34, 35, 41, 43, 45, 49, 50, 54–59, 61, 66, 70, 83–85, 87, 88, 94–97, 99–101, 113–116, 119, 126, 127, 131, 140, 141, 145, 150–152, 156–158, 176, 178, 180, 183, 189, 220, 253, 258, 265, 266, 275, 277, 281

RoboTrainer v2 The second of the two devices used in this thesis. This device is designed and developed to research neuromuscular training with robotic devices presented in this thesis. iv, viii, xvi, xvii, 5–8, 18, 23, 31, 35, 41–43, 45, 50, 87–111, 113–116, 118–124, 131, 140, 145–148, 152, 160, 175, 189, 193–201, 203, 205–207, 209, 211, 213, 215–218, 220, 221, 254, 259, 265, 266, 278, 279, 281, 282

Safe Torque Off Safety function in motor controllers to stop the motor. It kills the torque-generating energy of a motor and prevents unintentional starting. If a motor has an electromagnetic brake, it should be activated by this function. 105, 110, 275

Smart Walker A mobile robotic device aimed at the physical support of its user for walking. vii, viii, xv–xvii, 2–5, 7, 8, 11, 12, 14, 17, 18, 20, 23, 24, 27, 30–33, 35, 42–45, 49, 50, 52, 53, 62, 83, 85, 90, 91, 102, 113, 114, 117–120, 125–127, 129, 132, 136, 140, 143, 144, 148–150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 200, 219–222, 245, 247–265, 275, 277, 282, 283

spatial control action Spatially-limited behavior modifiers placed in the training environment. xvi, 7, 88, 115, 145, 149–151, 153, 154, 163, 164, 167, 169–173, 175, 176, 182, 189, 193, 194, 203, 275, 281, 285

Technology Readiness Level "Technology Readiness Levels (TRL) are a type of measurement system used to assess the maturity level of a particular technology. Each technology project is evaluated against the parameters for each technology level and is then assigned a TRL rating based on the projects progress. There are nine technology readiness levels. TRL 1 is the lowest and TRL 9 is the highest." Source: NASA[1]. In this document definition from European Union (EU) is used [43]. 275

[1] `https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html`

Acronyms

AD Alzheimer's disease. 13

ASBGo Adaptive System Behaviour Group. 2, 26–29, 45, 49, 250, 257, 258, 263, 264, 277

ASOC Active Split Offset Castor. 19, 33, 34, 255

CA control action. viii, 7, 8, 23, 42, 50, 58, 85, 114, 115, 120, 138, 147, 149–152, 162–164, 169, 171, 173–175, 196, 204, 217, 220, 221, 273, 279, *Glossary:* control action

CANTAB Cambridge Neuropsychological Test Automated Battery. 13, 63

CERAD-Plus Consortium to Establish a Registry for Alzheimer's Disease. 13, 63

CG control group. 13

CIMH Central Institute of Mental Health. iv, 3, 49, 62–64, 277

CoR center of rotation. 22, 33, 41, 42, 85, 141, 150, 152, 177, 180, 183, 184, 189, 248, 261

CPU Central Processing Unit. 105, 106, 108

DK direct kinematics. 114

DOF degree of freedom. 8, 9, 24, 37, 46, 57, 117, 119, 144, 152, 157, 164, 167

DSHS German Sport University Cologne. iv, 160, 175, 193–195, 197, 222, 279

ETH Zürich Eidgenössische Technische Hochschule Zürich. 28

EU European Union. 91, 100, 108

FOW field of view. 101

Fraunhofer IPA Fraunhofer-Institut für Produktionstechnik und Automatisierung. 54, 95

FTS force-torque sensor. vii, 3, 19, 22, 27, 34–38, 52–55, 57, 83, 93, 96, 97, 100–102, 109, 113, 118, 119, 129, 130, 144, 146, 147, 152, 217, 220, 222, 260–267, 271, 283

GCA global control action. 7, 115, 149–153, 169, 173, 176, 177, 193, 196, 200, 202, 204, 205, 207, 217, 274, 281, *Glossary:* global control action

GPIO general-purpose input/output. 105, 106, 266

GUI graphical user interface. 149, 170, 174

HMI human-machine interface. 97, 101

HRI human-robot interaction. 5, 6, 17, 50, 52, 53, 111, 219–222

IAR-IPR Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics. iv, 3, 58–60, 89, 103, 108, 175, 177, 253, 254, 277, 278

IG intervention group. 13

IK inverse kinematics. 114

IMU Inertial Measurement Unit. 24, 31, 36, 43, 263, 264

IR infrared. 21, 43

JAIST Japan Advanced Institute of Science and Technology. 249

JARoW JAIST active robotic walker. 25, 26, 33, 249, 277

KF Kalman filter. 25, 43, 263

LiPo lithium-ion polymer. 97, 99

LRF laser range finder. 21, 22, 25, 36, 41–43, 55, 58, 59, 93, 94, 256, 260–265

MARC Medical Automation Research Center. 21, 33, 248, 255, 261

MCI mild cognitive impairment. vii, viii, 1, 3, 7, 12, 13, 49, 50, 61–63, 83, 90, 149, 182, 219, 222, 253, 254

MIT Massachusetts Institute of Technology. 19, 247

MMSE-K Mini-Mental States Examination-Korea. 13

MOCA Montreal Cognitive Assessment. 13

ODW Omni-Directional Walker. 26

PAM-AID Personal Adaptive Mobility Aid. 17–20, 45, 247, 255, 260, 277

PAMM *Personal Aids for Mobility and Monitoring*. 19, 21, 40, 45, 247, 255, 260, 277

PD Parkinson's disease. 13

pHRI physical human-robot interaction. 5, 24, 126, 275, *Glossary:* physical human-robot interaction

PID Proportional-plus-Integral-plus-Derivative. 26, 263

PL Performance Level. 92, 93, 104, 105, 107–109

PLC Programmable logic controller. 85

ROS Robot Operating System. 55, 57, 58, 64, 69, 145–148, 169, 170, 220, 279

ROS-C ros_control. xvi, 114, 145–147

RoSylerNT Learning Robotic-Assistance Systems for Neuro-Muscular Training. 193, 195, 197, 279

SCA spatial control action. xvi, 7, 88, 115, 145, 149–151, 153–173, 175–177, 182, 189, 193–198, 202–204, 217, 222, 275, 279, 281, 282, 285, *Glossary:* spatial control action

SLAM Simultaneous Localization and Mapping. 4, 249

STO Safe Torque Off. 105, 107, 110, 275, *Glossary:* Safe Torque Off

SW Smart Walker. vii, viii, xv–xvii, 2–5, 7, 8, 11, 12, 14, 17–24, 26, 27, 30–46, 49–53, 62, 83, 85, 90, 91, 102, 113, 114, 117–120, 125–132, 136, 138–144, 148–150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 200, 219–222, 245–265, 275, 277, 278, 281–283, *Glossary:* Smart Walker

TF transfer function. 38, 39, 120, 122–124, 260

TRL Technology Readiness Level. 275, *Glossary:* Technology Readiness Level

TUG Time Up & Go. 13

v1 RoboTrainer Prototype. iv, vii, viii, xv, 3–5, 7, 8, 23, 31, 34, 35, 41, 43, 45, 49, 50, 54–59, 61, 66, 70, 83–85, 87, 88, 94–97, 99–101, 113–116, 119, 126, 127, 131, 140, 141, 145, 150–152, 156–158, 176, 178, 180, 183, 189, 220, 253, 258, 265, 266, 275, 277, 281, *Glossary:* RoboTrainer Prototype

VA-PAMAID Veterans Affairs Personal Adaptive Mobility Aid. 19, 20

VFF virtual force field. viii, 4, 7, 21, 22, 41, 42, 52, 59, 149, 153, 157, 160, 162, 173, 221

WMS-R Wechsler Memory Scale – Revised. 13, 63

WW wheeled walker. 2, 3, 11, 28, 33, 36, 85, 149, 245, 255–258

ZOH zero-order hold. 47

List of Figures

1.1. Structure of the thesis with the main contributions. 6

2.1. Examples of conventional assistive and training devices for mobility. . . . 16
2.2. Research and commercial versions of PAM-AID Smart Walker. 20
2.3. Versions of *Personal Aids for Mobility and Monitoring* (PAMM). 21
2.4. MARC Smart Walker. Source: [174] 22
2.5. Smart Walkers developed at Tohoku University, Japan. 23
2.6. Smart Walkers developed by Anselmo Frizera [51, 50]. 24
2.7. AZIMUT-3 Smart Walker. Source: [49] 25
2.8. JAIST active robotic walker (JARoW) [95] 26
2.9. The Omni-directional walker from Kochi University of Technology. Source: [159] 27
2.10. Versions of the Adaptive System Behaviour Group (ASBGo) Smart Walker from Minho University, Portugal. 29
2.11. SMARTWALKER. Source: [146] 30
2.12. Commercial Smart Walkers 32
2.13. The SW's representation in block diagram form. 39
2.14. Effect of the mass and damping parameters on SW's behavior. 40

3.1. *rob@work 3* mobile platform – the base for the RoboTrainer Prototype. . 55
3.2. The RoboTrainer Prototype as used in the HEiKA pilot study. 56
3.3. The RoboTrainer Prototype's control architecture. 58
3.4. The RoboTrainer Prototype's test environment at Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics. 59
3.5. The virtual representation of the test environment at Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics 60
3.6. The concept of *artificial forces* developed for the RoboTrainer Prototype. 61
3.7. Individual steps of the *HEiKA experiment*. 63
3.8. The premises at CIMH where the training with the RoboTrainer is done. . 64
3.9. Virtual representation of the paths used in the pilot study in a 2D floor plan. 65

3.10. Path configurations in the pilot study (1/2). 67
3.11. Path configurations in the pilot study (2/2). 68
3.12. HEiKA Experiment: deviation for all participants and exercises. 71
3.13. HEiKA Experiment: deviation for all participants and exercises – boxplot. 72
3.14. HEiKA Experiment: deviation for all participants. 74
3.15. HEiKA Experiment: time performance for all participants and exercises. 75
3.16. HEiKA Experiment: time performance for all participants and exercises – boxplot. 76
3.17. HEiKA Experiment: time performance for all participants. 79
3.18. User-Experience Assessment – all tasks. 80
3.19. Users' Experience – assessment of the training day. 81
3.20. Deviation comparison for the path with and without *artificial forces* concept. 82

4.1. RoboTrainer v2 and its user on the test parkour at the IAR-IPR. 89
4.2. RoboTrainer v2 CAD model 95
4.3. RoboTrainer v2 toolless handle adaption real-world realization. 96
4.4. Extreme positions of the RoboTrainer v2's rear wheels configurations. . . 98
4.5. RoboTrainer v2's toolless concept for reconfiguring of the rear wheels. . 99
4.6. RoboTrainer v2's custom parts. 100
4.7. RoboTrainer v2's non-safety components 102
4.8. RoboTrainer v2's safety components. 106
4.9. Laser scanner's safety fields of RoboTrainer v2 107

5.1. Overview of the RoboTrainer v2 control architecture. 115
5.2. RoboTrainer v2 - controller coordinate system 118
5.3. Control loop of the interaction of a user and a SW. 119
5.4. RoboTrainer v2 – dynamics of input force filters, admittance controller, and the hardware's velocity. 121
5.5. RoboTrainer v2's – hardware dynamics. 122
5.6. RoboTrainer v2's – test of the identified dynamics. 123
5.7. RoboTrainer v2's – identified hardware dynamics using interaction data. . 124
5.8. Minimal values for the controller's desired mass M_d depending on the coupling factor h for the RoboTrainer (M_a = 84.5 kg). 127
5.9. The extended model of the user-walker system used for the stability analysis. 128
5.10. A user tries to control RoboTrainer v2 with stiff arms. 131
5.11. Comparison of the state of the art interaction energy calculation and the approach from this work. 133
5.12. Influence of the sliding integral on a sudden change of RoboTrainer's direction. 135

5.13. The adapted method for scaling factor with the detection of the user's intention to keep the RoboTrainer's dynamics when the user suddenly changes direction. 137
5.14. Non-linear scaling of user's input force in respect to velocity (5.21), with $r \in [0.1, 1.0]$ (step 0.1), $s = 9.0$. 139
5.15. RoboTrainer's velocity response using the controller with static parameters. 140
5.16. RoboTrainer's velocity response using the controller with damping adaption. 141
5.17. RoboTrainer's velocity response using the controller with non-linear adaption. 142
5.18. RoboTrainer v2 control architecture – internal classes. 146
5.19. RoboTrainer v2 control architecture – ROS packages. 147

6.1. Overview of the control actions in the RoboTrainer's controller. 151
6.2. Artistic representation of a scenario with the RoboTrainer using SCAs. . 154
6.3. Concept of the *SCA–Area* with different resulting behaviors. 155
6.4. Idea of the *SCA–Force Area* 156
6.5. *SCA–Force Area* – Force profile inside the area. 157
6.6. *SCA–Force Area* velocity compensation. 159
6.7. Effect of the *SCA–Force Area* on users' velocity. 159
6.8. Idea of the *SCA–Path Following* 161
6.9. Real-world example of the SCA–Path Tracking. 161
6.10. Idea of the *SCA–Wall* 162
6.11. A dangerous situation without the passivity concept for SCAs. 165
6.12. A dangerous situation with the passivity concept for SCAs. 166
6.13. A dangerous situation without the safety concept for SCAs. 167
6.14. A dangerous situation with the safety concept for SCAs. 168
6.15. *RoboTrainerEditor* – configuration GUI for SCAs. 170

7.1. Parkours used in the evaluation of adaptive and individualized control. . . 179
7.2. Boxplot of user's force and torque measurements during parameterization of "maximal user's force" (P1). 190
7.3. Measurements of user's force and torque during parameterization of "maximal user's force" (P1). 191
7.4. Parameterization of velocity adoption: Distance between the users' legs and RoboTrainer. 192
7.5. RoSylerNT evaluation environment at DSHS. 195
7.6. Virtual representation of the RoSylerNT evaluation environment at DSHS. 197
7.7. Average interaction force of all users showed for the different repetitions of a task and as the average value of a task. 209
7.8. Average input force of all users showed for the different repetitions of a task and as the average value of a task. 210

7.9. Average velocity of all users showed for the different repetitions of a task and as the average value of a task. 211
7.10. Average deviation of all users showed for the different repetitions of a task and as the average value of a task. 212
7.11. Duration of all users showed for the different repetitions of a task and as the average value of a task. 213
7.12. Average users' distance of all users showed for the different repetitions of a task and as the average value of a task. 214
7.13. Average disturbance force for different repetitions. 215

List of Tables

2.1. Overview of SotA-SWs design features and functionalities. 45

3.1. HEiKA Experiment: Overview of the training with RoboTrainer Prototype. 66
3.2. HEiKA Experiment: statistical differences between the first attempt on the first day and the last attempt on the last day for error score and time measurements. 74
3.3. User-Experience Assessment – all tasks. 78
3.4. Users' Experience – assessment of the training day. 80

5.1. Physical properties and default controller parameters of the RoboTrainer v2. 116

6.1. Parameters definition for global control actions. 169
6.2. Parameters definition for spatial control actions. 171

7.1. Demographics of participants in the evaluation of individual control parameters and non-linear adaptive control. 178
7.2. Experience of participants in use of a conventional walker and RoboTrainer.178
7.3. List of tasks in the evaluation with 22 participants. 184
7.4. User's answers regarding the parameterization processes for each evaluation category. 184
7.5. Median and range of user's answers regarding the parameterization processes for each evaluation category. 185
7.6. User's answers regarding the tasks for each evaluation category. The numbers represent the absolute number of answers. 185
7.7. Median and range of user's answers regarding the tasks for each evaluation category. 186
7.8. Significant differences in the participant's answers between baseline task T2 and tasks T6 and T7. 187
7.9. User's answers to general questions asked at the end of the evaluation session. 188

7.10. Demographics of participants in the evaluation of RoboTrainer v2. 194
7.11. Experience of participants in the evaluation of RoboTrainer v2. 194
7.12. List of tasks in the RoboTrainer v2's evaluation with 13 participants. . . . 196
7.13. List of questions and statements used in the questionnaires for the RoboTrainer v2 evaluation. 199
7.14. Significant differences for interaction force during parameterization. . . . 201
7.15. Significant differences for the average input force. 201
7.16. Significant differences in interaction force for the tasks with *SCA-Force Area*. 202
7.17. Significant differences for average deviation between repetitions. 203
7.18. Significant differences for duration between repetitions. 204
7.19. Significant differences for average interaction force of the tasks with adaptive and conventional control. 206
7.20. Significant differences for different variables regarding RoboTrainer's footprint setup. 208
7.21. RoboTrainer v2's evaluation – number of answers per category. 216
7.22. RoboTrainer v2's evaluation – median and range of users' answers. . . . 216

A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 247
A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 248
A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 249
A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 250
A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 251
A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 252
A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 253
A.1. Overview of the institutions, target population and high-level functionalities of the Smart Walkers from the literature. 254
A.2. Overview of the design properties of the Smart Walkers from the literature.255
A.2. Overview of the design properties of the Smart Walkers from the literature.256
A.2. Overview of the design properties of the Smart Walkers from the literature.257
A.2. Overview of the design properties of the Smart Walkers from the literature.258
A.2. Overview of the design properties of the Smart Walkers from the literature.259
A.3. Overview of the control features of the Smart Walkers from the literature. 260
A.3. Overview of the control features of the Smart Walkers from the literature. 261

A.3. Overview of the control features of the Smart Walkers from the literature. 262
A.3. Overview of the control features of the Smart Walkers from the literature. 263
A.3. Overview of the control features of the Smart Walkers from the literature. 264
A.3. Overview of the control features of the Smart Walkers from the literature. 265
B.4. Technical Specification of the ATI Mini58 force-torque sensor (FTS) . . 267

Listings

6.1. Example Scenario Description with spatial control actions (SCAs) 172