A kinaesthetic hand exoskeleton system toward robot-augmented rehabilitation therapies in the Health 4.0 era

Lorenzo Bartalucci
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

Chiara Brogi
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

Alessandro Bucci
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

Alberto Topini
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

Andrea Della Valle
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

Nicola Secciani
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

Alessandro Ridolfi
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

Benedetto Allotta*
Department of Industrial Engineering, University of Florence, via di Santa Marta 3,50139, Florence, Italy

ABSTRACT

In the new era of Industry 4.0, digitization and automation are the tipping points for addressing rapidly evolving challenges. The momentum of this movement has underscored the critical importance of developing new technologies to collect, understand and use the enormous amount of data created. This revolutionary wave has also taken root in the healthcare sector, giving rise to the so-called Health 4.0, based on smart machines to provide patients with better, more value-added and more efficient healthcare services. This scenario has contributed to making innovative robot-augmented therapies a hot research topic with potentially significant societal impact. In this context, the authors present the implementation of a new prototype of a kinaesthetic exoskeleton to enhance and recover sensorimotor skills to perform grasping and manipulation via force stimuli from serious rehabilitation games. The discussion includes an initial overview of the system's hardware, both from a mechanical and electronic point of view, motivating the choices made in the design phase. It continues by moving on to a description of the system's low-level and high-level software architecture, along with the control strategy applied to implement progressive resistant force therapy. Finally, an example of serious game was developed as an application case for evaluating the system's usability and extrapolating empirical parameters indicative of responsiveness and accuracy. The results showed that the proposed system can reproduce constant forces with an average error of 0.75 N and standard deviation of 0.3 N with an average reaction time of 0.5 s and standard deviation of 0.4 s.

*Corresponding author's email: lorenzo.bartalucci@unifi.it

ICBRA 2023, September 22–24, 2023, Barcelona, Spain

ACM ISBN 979-8-4007-0815-2/23/09…$15.00
https://doi.org/10.1145/3632047.3632068

CCS CONCEPTS

• **Applied computing**; • **Physical sciences and engineering**; • **Computer-aided design**;

KEYWORDS

Hand exoskeleton, Rehabilitative robotics, Robot-augmented therapy, Health 4.0, Rehabilitative serious game

ACM Reference Format:
Lorenzo Bartalucci, Chiara Brogi, Alessandro Bucci, Alberto Topini, Andrea Della Valle, Nicola Secciani, Alessandro Ridolfi, and Benedetto Allotta. 2023. A kinaesthetic hand exoskeleton system toward robot-augmented rehabilitation therapies in the Health 4.0 era. In *2023 the 10th International Conference on Bioinformatics Research and Applications (ICBRA) (ICBRA 2023), September 22–24, 2023, Barcelona, Spain.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3632047.3632068

1 INTRODUCTION

The advent of the so-called *Industry 4.0* has radically reshaped the healthcare sector by prompting a technological revolution that has led to the coining of the term *Health 4.0* [1] [2] [3]. This concept refers to the use of *Industry 4.0* technologies to provide patients with better, more value-added, and more cost-effective services while improving the healthcare industry's efficacy and efficiency. *Health 4.0*, thus, outlines a new and innovative vision for the health sector with smart machines that get access to large amounts of data. The unstoppable surge of this change has paved the way for the advancement of Robotics for Medicine and Healthcare as a powerful tool for enhancing assistive technology. Examples of such systems are active prostheses [4] [5], controlled robotic tools for surgery [6], and wearable robots (e.g., exoskeletons) for assistive or rehabilitative tasks [7] [8] [9]. Incorporating automatic controls, sensors, computing power and storage capabilities, robotic devices show the potential to open innovative clinical paths, taking the name of robot-augmented therapies. Lately, research efforts in robot-augmented therapy have addressed the development of haptic devices capable of interfacing patients with serious games, a branch of Virtual Reality (VR) technology. Serious games are interactive and designed to train patients in active movements, refine motor control and coordination skills, and improve user engagement. Many studies have proven the huge benefits provided by this type of therapy [10] [11] [12] [13], identifying it as one of the most promising emerging practices and making it a hot research topic. However, developing devices for this application represents a tough challenge as they are requested a twofold behavior: they should act, on one side, as rehabilitative devices, providing and supervising therapy and, on the other side, as haptic devices, replicating the kinaesthetic interactions with the VR, i.e., forces and movement stimuli, coming from a serious game. Of all robotics devices for medicine and health care, exoskeletons have emerged as the best suitable to achieve this goal, being wearable for a better VR rendering capability and capable of applying the large amount of force typically required by rehabilitation tasks. However, the state of the art of these devices, both from a mechatronic and control software architecture point of view, is still in the midst of development and expansion despite being started in the 1980s. This interest is due to the vast intrinsic potential of these devices, but to date, they still need to be effectively and efficiently implemented due to the complexity of performing high-level kinaesthetic tasks while working closely with the human body and, consequently, dealing with the demands of wearability and portability. This demand becomes much more complicated when developing Hand Exoskeleton Systems (HESs) because of the complex anatomy, the vast set of movements required and the very restricted space available. Despite the multiple adversities, many examples of HESs meant to work with serious games for robot-augmented therapy are proposed in the literature. Commonly the implemented control strategies are based on position feedback, and the serious games are designed to lead the user to perform rehabilitative movements and particular hand gestures [14] [15] [16]. In contrast, very few HESs feature the capability of applying force feedback [17] [18] [19] for providing patients with a better VR experience and more value-added therapy, such as assistance as needed [20] or progressive resistance [21].

1.1 Contribution and structure of the paper

This manuscript presents a kinaesthetic HES designed to replicate on the user's fingers the force references elaborated within a serious game. This functionality is specifically designed to improve the user's sensorimotor and proprioceptive functions by applying a progressive force resistance to module the therapy according to the user's progresses.

This paper provides in Sections 2 and 3 an overview of the proposed device by covering the mechatronic aspects: Section 2 is concerned with mechanics, while Section 3 is with electronics. Section 4 introduces the software architecture and control strategy implemented to provide serious game-assisted progressive force therapy. Afterwards, Section 5 reports an evaluation conducted on a pilot study for the empirical extrapolation of valuable data to characterize the performance of the proposed system in terms of reactivity and accuracy in force traceback. Finally, Section 6, concludes the paper.

2 MECHANICS

The proposed HES is designed to move simultaneously and independently, in opening and closing, three fingers: index, thumb, and middle finger, as these three fingers are the minimum set that allows replication of almost all power and precision grips.

The system features a Remote Actuation System (RAS) containing the heaviest and bulkiest elements, such as actuators, power supplies, control boards and accessory electronic components. The motion transmission is based on Bowden cables; flexible, lightweight, and cost-effective elements that enable high torques to be transmitted to the lighter and less bulky wearable exoskeleton. The result is a device that is not overly burdensome on the user's limb and capable of delivering considerable force on the user's fingers, reaching up to 15 N per finger. The architecture of the exoskeleton finger mechanism, responsible for guiding the user's fingers, is rigid, single-DOF, with a single interaction point in the middle of the medial phalanx. This mechanism results from years of research and development at the University of Florence [24] [25], and its usability is guaranteed in close contact with humans.

As visible in Figure 2, the HES features a magnetic encoder and a load cell to recreate the hand pose and interaction force within the serious game. The pose of the mechanism end-effector (i.e., the interaction point) can be derived through kinematic synthesis starting from the angular measure provided by the encoder. The interaction force (i.e., the force applied to the finger by the finger mechanism) can be derived from the force measured by the load cell using a dynamic model; a detailed discussion is reported in [23].

The RAS used in this work results from the system's evolution presented in [23]. Such a RAS, as shown in Figure 2, features a closed case consisting of the assembly of stainless-steel sheets, two mm thick, appropriately bent. Sizing was conducted to accommodate the necessary components for three-finger implementation and functional electronics. In addition, provision was made for housing a screen at the top, two fans to ensure adequate cooling, and the housing of connectors for data exchange with an external computer. The whole system, both exoskeleton and RAS, is designed to be completely modular so that the HES can be used in seven different configurations derived from all the possible commutations of the three finger mechanisms (3 x single-finger configurations, 3 x double-finger configurations, and 1 x three-finger configuration). Switching from one configuration to another is made possible by magnetic quick couplings that make the procedure completely tool-free.

3 ELECTRONICS

Compared with the version presented in [23], the electronic hardware has been modified by implementing an onboard computer (Raspberry Pi 4) to perform low-level control for managing the three fingers and communicating with an external computer, which will provide high-level commands.

As shown in Figure 3, a Sparkfun development board (AST-CAN485 from SparkFun Electronics), a driver (the HTPTZ model from Haitai Mechanical and Electrical Equipment Co.) and a signal amplifier (the FSC 5-24 V model from Forsentek Co.) for the load cell are provided for the functioning of each finger.

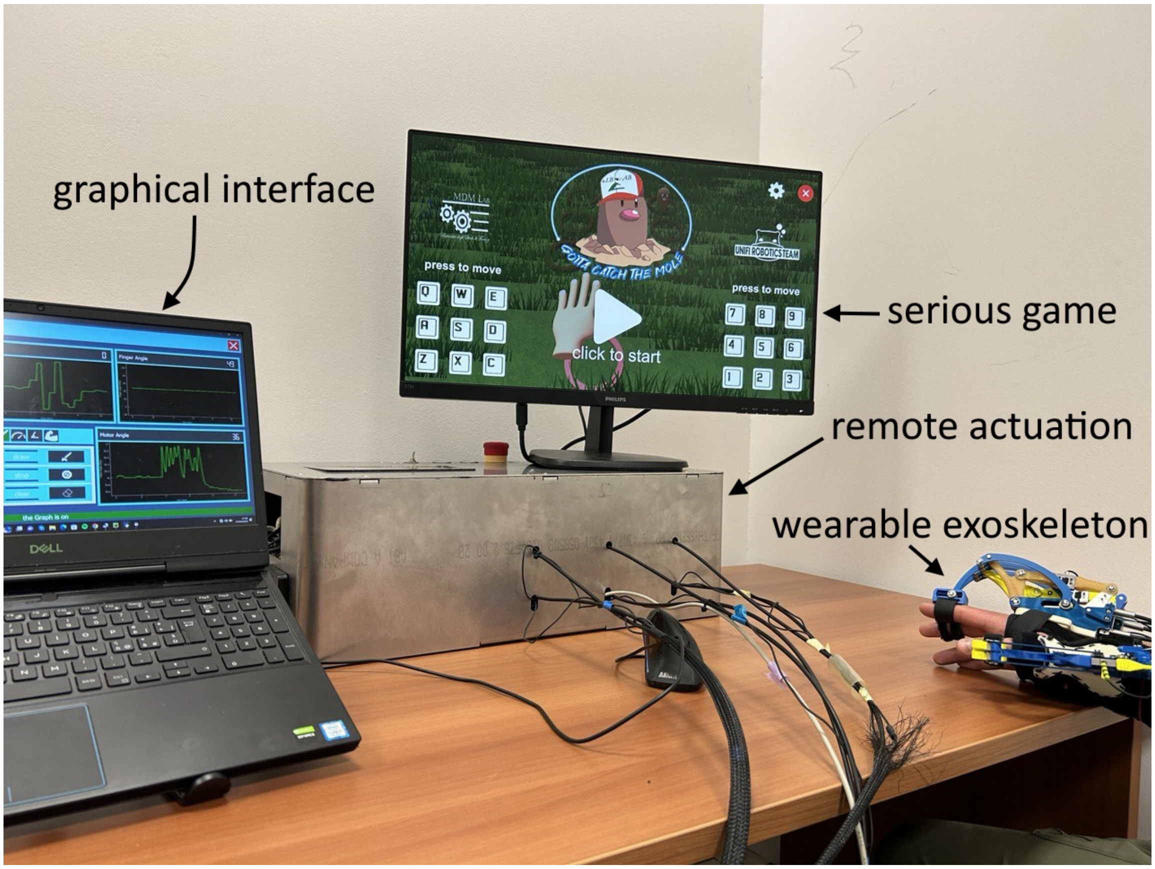

Figure 1: – The presented HES composed of the RAS and the wearable exoskeleton. The left monitor shows the real-time status of the device (i.e., finger speeds, transmitted torques, etc.), the right one instead shows the VR-based serious game.

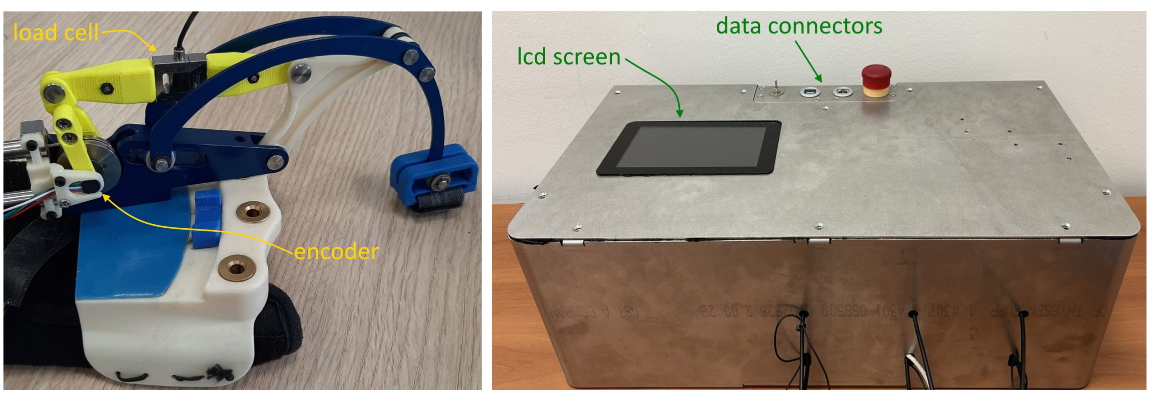

Figure 2: - wearable Part (on the left), and RAS (on the right).

The Sparkfun development board handles low-level control via the ATmega32U4 microcontroller with a 16 MHz processor. Precisely, it samples and processes analogue signals from the load cell, through the amplifier, and from the exoskeleton encoder.

The motor driver contains the firmware to drive the Haitai motor by means of a PID controller in speed. A proprietary communication protocol allows the monitoring of various parameters, including current consumption, supply voltage, Revolute Per Minute (RPM), and absolute position. The electrical/electronic scheme requires a 220V supply line transformed by an AC-DC transformer to a constant 24 V voltage. Then distributed to the Sparkfun drivers,

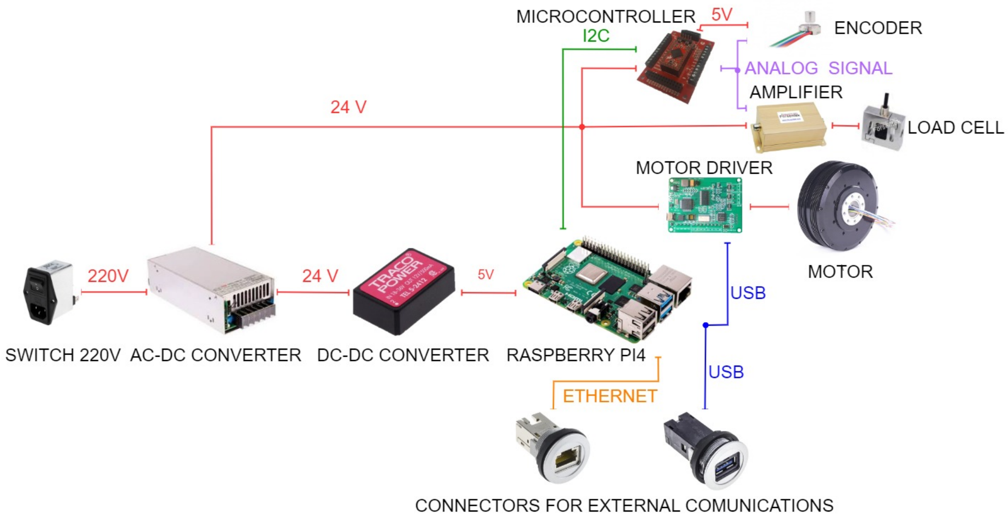

Figure 3: - HES electronic scheme.

amplifiers, and development boards, which power the motors, encoders, and load cells. To power the Raspberry Pi 4 a 5 V DC-DC was inserted on the 24V line.

As for the communication network, the onboard computer interacts with the Sparkfun boards in I2C and serial with the motor drivers and an external computer.

4 SOFTWARE ARCHITECTURE

The software architecture of the exoskeleton consists of a code structure divided in two parts designed to perform mid and low-level control, and high-level control independently but interactively. The onboard computer executes the mid and low-level control consisting of managing the actuation and data collection of the three finger modules in parallel. An external computer is instead in charge of the high-level control by means of a dedicated Graphic User Interface (GUI) and a virtual reality. In the application under consideration, the two computers communicate via a serial USB communication channel, and through this, the low level sends the status parameters of the device, while the high level sends the commands to be executed by the device. The strategy of keeping these two types of control separate brings an advantage both at the security level, in that any crashes or anomalies arising from the GUI or the serious game are less likely to trigger abnormal and dangerous behaviors of the device, and at the functional level, in that it is possible to manage the GUI and the exoskeleton at different frequencies. The middle level performs the control and processing methods of signals acquired at a high frequency (300 Hz), while the high level, with higher computational burdens, dealing with interaction and visualization, is updated at significantly lower frequencies (50 Hz).

4.1 Onboard firmware

The firmware running on the onboard computer is a Python module named ExoRunfile, which runs automatically when the Raspberry is turned on.

An object-oriented programming was pursued by defining a custom ExoFinger class. Thus, at startup, for each physical finger, an ExoFinger object containing the necessary instructions to control and supervise the management of a single physical finger is instantiated according to the system configuration exploited. This class has a mainloop() method, called cyclically, to sequentially execute the instructions necessary for: (i) the initialization process at startup; (ii) collecting and sending state parameters; (iii) receiving external commands; (iv) sending the speed command for the motor. At each iteration cycle, the status parameters are measured by the Sparkfun and driver and consequently collected by the onboard computer through the local communication network. The state variable of a finger exoskeleton can be represented by a vector of five elements: the motor torque, motor current, motor speed, onboard encoder angle and load cell measurement. The latter is processed through a moving-average filter over 30 samples and, at initialization or following a specific command, can undergo software calibration, i.e., the current average value is subtracted from all subsequent samplings. When required by the high-level control the status variables are packed into a serial buffer and sent to the external computer. A digital PID has been implemented in the ExoFinger class that, by exploiting the feedback loops on the position or force signals, allows the calculation of the reference speed to be passed to the motor driver. This method can be triggered following an appropriate GUI command to follow a reference in position or force. The equation describing the specific algorithm implemented is reported in Equation 1.

$$\begin{aligned} u_k = & \frac{T_N+1}{T_N} u_{k-1} - \frac{1}{T_N} u_{k-2} + K_p \frac{T_N+N}{T_N} e_k \\ & + \frac{K_p}{T_N} \left(T_N \frac{T_c - T_i}{T_i} - 2N + 1 \right) e_{k-1} \\ & + \frac{K_p}{T_N} \left(1 + N - \frac{T_c}{T_i} \right) e_{k-2} \end{aligned} \quad (1)$$

Where u_k is the velocity calculated at the k-th interaction, e_k is the error between the reference signal and the signal measured at the k-th iteration, K_p is the proportional gain, T_i the integral time

Figure 4: - An overview of the ExoGUI with colored frames highlighting the various panels. The setting panel, in green; the control panel, in red; the data logger panel, in orange; the graphics area, in blue; the graph control panel, in yellow; and the text box, in white.

constant, T_c time difference between the k-th sampling and the previous sampling, N is the value of the additional pole introduced to make the PID transfer function causal, and, finally, T_N is a constant dependent on the derivative constant T_d, as described by Equation 2.

$$T_N = \frac{N\,T_c}{T_d} + 1 \quad (2)$$

It is worth noting that K_p, T_i, T_d, and N are parameters of the digital PID and their respective values are derived following an appropriate calibration process.

4.2 exoGUI

The GUI under consideration, named exoGUI, was developed on Qt framework using PyQt5 libraries. The version presented is not optimized for the end user yet, instead is an intermediate version employed as a tool for the development and testing phase. The exoGUI is structured as a single-window application with a purpose-oriented layout, dividing the window area into panels with specific purposes. As shown in Figure 4, the following are distinguished: the setup panel, the control panel, the panel for saving data, and the panel for managing plotted graphs in the dedicated graph area. Finally, there is a dedicated text box in the lower area of the window for sending messages to the user following specific events, such as certifying the correct execution of the sent command or reporting errors. The Setting panel allows the user to enter, via text boxes, all the parameters necessary for communication with the onboard computer and the serious game application. The Start panel contains the Run Exo button that executes instructions to communicate startup to the onboard computer, while the Kill Exo button communicates termination of the session. The control panel has been implemented using a stacked widget that allows changing the contents of a certain portion in the panel area by clicking on the specific buttons, located on the left side of the panel. As their assigned names suggest, the speed, position, and force buttons bring up all the necessary widgets (buttons and text boxes) to activate the PID controller with speed, position, or force reference, respectively. The Catch the Mole button opens the widgets needed to initialize and start the application executable containing the serious game. The graphical control panel contains buttons to manage the display of some of the system state variables in the form of Cartesian graphs in the dedicated areas, with the sampling times on the x-axis and the values of the state variables on the y-axis. The state variables that can be displayed are exoskeleton angle, end-effector force, motor torque, and motor angle, which in the process of experimentation, evaluation, and tuning, are considered of greater interest and usefulness than the others. The data logger panel allows the state variables to be saved in a csv format file generated within the folder containing the application executable for possible post-processing analysis of the collected data.

4.3 Catch the Mole

A serious game was entirely developed in the Unity environment to stimulate and train the sensorimotor apparatus necessary to perform the kinaesthetic gesture of ball grip. The application was structured considering progressive resistant force therapy, whose effectiveness for initiating active hand movements was demonstrated in [22], recording a significant increase in grip strength following therapy (up to 31.8%). This treatment consists of repetitive flexion and extension of the fingers with various loads for periods of 15 minutes, which can be tedious and unstimulating for the patient without the support of a serious game. The proposed game consists of a digital reimagining of mole-catching with a virtual hand which is a digital-twin of the user's one. Trivially, the game aims to have the patient catch the most moles in a given time interval. The virtual hand pose is controlled by the signal from the exoskeleton's encoder: each digital finger reproduces the 1-dof flexion/extension motion of the real finger. At the application's refresh rate (60 Hz), the Catch the Mole application reports to the GUI, via an appropriate flag, whether the avatar is in catch mode, i.e., whether the avatar has caught a mole. When not in this mode, the exoskeleton is controlled with a zero-force reference, so it only compensates for its inertia, aiming to be as transparent as possible for the user. The capture mode is triggered when the three fingers: index, thumb, and middle finger, all come in contact with the mole, and when this happens, the reference force changes. In order to simulate the effort required to hold grip onto the mole, the finger mechanisms are requested to perform an opening motion (i.e., exerting an extension force on the fingers) with constant preset intensity ranging from 0.5 to 15 N so that it is possible to train with predetermined levels of force in line with the therapy outlined in [22]. The game scene is fixed and set in a portion of a meadow where 10 discs can be distinguished: 9 brown and one red. The brown ones represent the areas where the digital mole can appear and be reachable by the avatar by pressing specific buttons. The red disk awards a point when the user drops the captured mole into it. At the start of the game: the avatar is placed on top of the red disk, the countdown starts (highlighted by a colored bar on top left corner), and a digital mole is randomly generated in one of 9 predetermined locations. After the avatar has reached the position where the mole has appeared, the user can proceed to capture it by closing the hand. Once the capture mode is triggered, the avatar automatically travels from the capture zone to the red disk, dragging the digital mole with it, in a predetermined amount of time depending on the chosen training intensity. During this time window, the exoskeleton attempts to force the user's fingers open as described above. The moment one of the digital fingers, driven by the behavior of the real ones, exits the mole's collider (i) the capture mode ceases, (ii) the reference force returns to zero, (ii) the digital mole falls toward the ground simulating the action of gravity, disappears and randomly reappears in one of 9 possible areas. Thus, the user gets the point only if he or she can counter the action of the exoskeleton until the avatar stands over the red disk and then drops the mole inside it. User involvement is induced by the stimulus to score the most points, triggering the goal of improving with each run. It is important to note that during the gaming session, the state parameters, indicative of the user's efforts, are directly visible in real-time thanks to the exoGUI and can be exported to the CSV file allowing the clinical operator to quantitatively and objectively evaluate the rehabilitation session.

5 APPLICATION CASE

A pilot study was conducted on a healthy user to evaluate the usability and ability of the presented device in replicating the force tracking required during the performance of the serious game. Figure 6 shows a recording of the interaction force of a finger during the performance of two catches in the Catch the Mole application. The green line represents the interaction force measured by the exoskeleton, while the red line represents the reference force from 0 N, when no interaction with the mole is recorded in the application, to 8 N during the capture mode. The blue horizontal lines identify the average interaction force calculated during contact and the deviations from the reference, denoted by Δe, which indicate the accuracy with which the device replicates the desired force. Also reported are the device reaction time intervals Δt, calculated as the difference between the instant of contact initiation or cessation and when the device achieves the desired interaction force. With the intent of deriving values of Δe and Δt, with a minimum statistical significance, 30 captures were evaluated, obtaining normal distributions of these parameters with mean value and standard deviation reported in Table 1. The results obtained can be considered more than satisfactory for a rehabilitation application, being a kinaesthetic application with not excessively stringent demands for accuracy and responsiveness.

6 CONCLUSION

Although the many benefits of robot augmentation therapy have been demonstrated through multiple studies, there are still very few examples in the literature of devices that can offer such therapy specifically aimed at hand rehabilitation. The overview presented in this article is certainly interesting from a scientific-technical point of view as it deals in detail with implementing a new prototype of a kinaesthetic HES to rehabilitate grasping and manipulation using force stimuli from serious rehabilitation games. The discussion provided a general overview at both hardware and software levels, introducing the sensing and control strategies to deliver the specific progressive resistant force therapy. In addition, an application example of a serious game, Catch the Mole, was also developed and presented, allowing empirical evaluation of the device's usability while also deriving useful parameters to quantify the level of accuracy and responsiveness of the system. Although the present study is only preliminary, early tests have shown promising results, positively contributing to state of art in biomedical robotics. To date, the natural continuation of the research effort will be to evaluate the device's effectiveness in practice through a dedicated clinical study and to develop a control strategy that replicates more complicated force profiles.

ACKNOWLEDGMENTS

We acknowledge the support of the European Union by the Next Generation EU project ECS00000017 'Ecosistema dell'Innovazione' Tuscany Health Ecosystem (THE, PNRR, Spoke 9: Robotics and Automation for Health), the PNRR project Fit for Medical Robotics

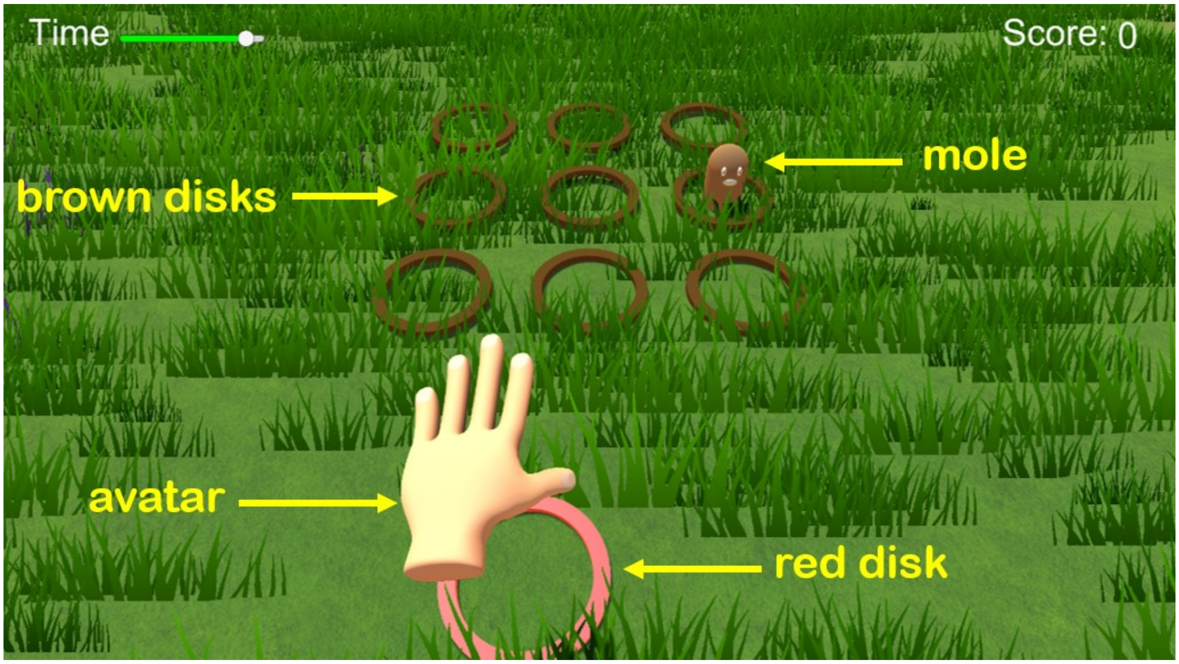

Figure 5: - Opening scene of the Catch the mole application.

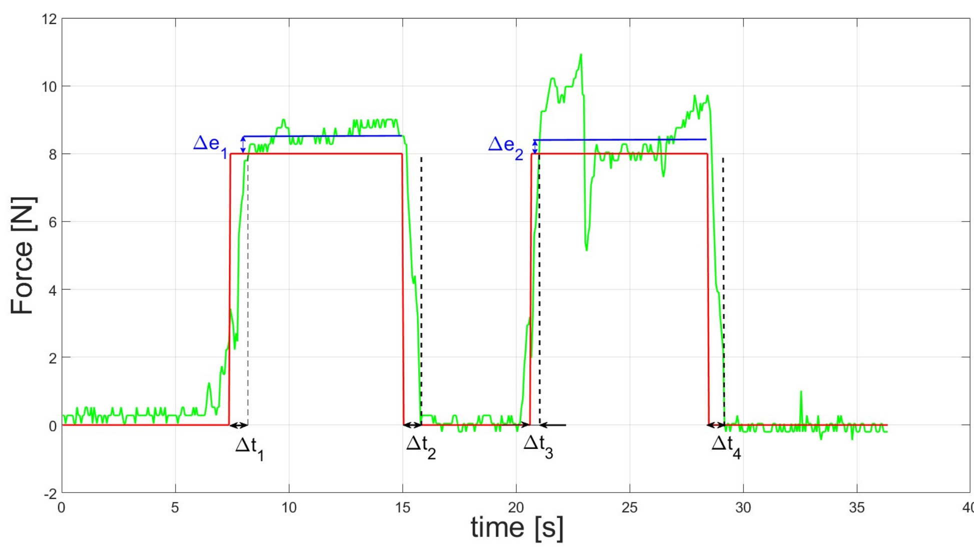

Figure 6: - example of a force profile.

Table 1: the average value and the standard deviation of Δe and Δt parameters

parameter	Definition	Average value	Standard deviation
Δe	Error from the reference value	0.75 N	0.3 N
Δt	Exoskeleton reaction time	0.5 s	0.4 s

(Fit4MedRob) and the BMIFOCUS project, funded by the Tuscany Region (POR FESR 2014–2020).

REFERENCES

[1] Tortorella, Guilherme Luz, *et al.* "Healthcare 4.0: trends, challenges and research directions." *Production Planning & Control* 31.15 (2020): 1245-1260.
[2] Estrela, Vania V., *et al.* "Health 4.0: applications, management, technologies and review: array." *Medical Technologies Journal* 2.4 (2018): 262-276.
[3] Yang, Geng, *et al.* "Homecare robotic systems for healthcare 4.0: visions and enabling technologies." *IEEE journal of biomedical and health informatics* 24.9 (2020): 2535-2549.
[4] Toledo, Cinthya, *et al.* "Upper limb prostheses for amputations above elbow: A review." *2009 pan American health care exchanges* (2009): 104-108.
[5] Weiland, James D., and Mark S. Humayun. "Retinal prosthesis." *IEEE Transactions on Biomedical Engineering* 61.5 (2014): 1412-1424.
[6] Mylonas, George P., *et al.* "CYCLOPS: A versatile robotic tool for bimanual single-access and natural-orifice endoscopic surgery." *2014 IEEE international conference on robotics and automation (ICRA).* IEEE, 2014.
[7] Gopura, R. A. R. C., Kazuo Kiguchi, and D. S. V. Bandara. "A brief review on upper extremity robotic exoskeleton systems." *2011 6th international Conference on Industrial and Information Systems.* IEEE, 2011.
[8] Lovrenovic, Zlatko, and Marc Doumit. "Review and analysis of recent development of lower extremity exoskeletons for walking assist." *2016 IEEE EMBS international student conference (ISC).* IEEE, 2016.
[9] Malvezzi, Monica, *et al.* "Design, development, and preliminary evaluation of a highly wearable exoskeleton." *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN).* IEEE, 2020.
[10] Hadjadj, Zineb, *et al.* "Leap motion controller for upper limbs physical rehabilitation in post-stroke patients: a usability evaluation." *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD).* IEEE, 2022.
[11] Jimeno, Miguel, *et al.* "Video games development as an opportunity for social impact." *2020 IEEE Games, Multimedia, Animation and Multiple Realities Conference (GMAX).* IEEE, 2020.
[12] Kim, Gerard Jounghyun. "A SWOT analysis of the field of virtual reality rehabilitation and therapy." *Presence* 14.2 (2005): 119-146.
[13] Sveistrup, Heidi. "Motor rehabilitation using virtual reality." *Journal of neuroengineering and rehabilitation* 1 (2004): 1-8.
[14] Aruanno, Beatrice, and Mario Covarrubias. "Handy novel hand exoskeleton for personalized rehabilitation." *Comput. Aided Des. Appl* 19.3 (2022): 405-425.
[15] Bouteraa, Yassine, Ismail Ben Abdallah, and Ahmed M. Elmogy. "Training of hand rehabilitation using low cost exoskeleton and vision-based game interface." *Journal of Intelligent & Robotic Systems* 96 (2019): 31-47.
[16] Borboni, Alberto, Maurizio Mor, and Rodolfo Faglia. "Gloreha—hand robotic rehabilitation: Design, mechanical model, and experiments." *Journal of Dynamic Systems, Measurement, and Control* 138.11 (2016).
[17] Sarakoglou, Ioannis, *et al.* "Hexotrac: A highly under-actuated hand exoskeleton for finger tracking and force feedback." *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2016.
[18] Jo, Inseong, and Joonbum Bae. "Design and control of a wearable hand exoskeleton with force-controllable and compact actuator modules." *2015 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2015.
[19] Yun, Youngmok, *et al.* "Accurate torque control of finger joints with UT hand exoskeleton through Bowden cable SEA." *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2016.
[20] Lum, Peter S., *et al.* "Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke." *Archives of physical medicine and rehabilitation* 83.7 (2002): 952-959.
[21] Taylor, Nicholas F., Karen J. Dodd, and Diane L. Damiano. "Progressive resistance exercise in physical therapy: a summary of systematic reviews." *Physical therapy* 85.11 (2005): 1208-1223.
[22] Bütefisch, Cathrin, *et al.* "Repetitive training of isolated movements improves the outcome of motor rehabilitation of the centrally paretic hand." *Journal of the neurological sciences* 130.1 (1995): 59-68.
[23] Bartalucci, Lorenzo, *et al.* "An original mechatronic design of a kinaesthetic hand exoskeleton for virtual reality-based applications." Mechatronics 90 (2023): 102947.
[24] Secciani, Nicola, *et al.* "Tailor-made hand exoskeletons at the university of florence: from kinematics to mechatronic design." Machines 7.2 (2019): 22.
[25] Bianchi, Matteo, *et al.* "Optimization-based scaling procedure for the design of fully portable hand exoskeletons." Meccanica 53 (2018): 3157-3175.