The EU Framework Programme for Research and Innovation H2020
Research and Innovation Action

CENTAURO

Deliverable D3.4 Final Version of Control Station

Dissemination Level: Public

Project acronym:	CENTAURO
Project full title:	Robust Mobility and Dexterous Manipulation in Disaster Response by Fullbody Telepresence in a Centaur-like Robot
Grant agreement no.:	644839
Lead beneficiary:	SSSA – Scuola Superiore di Studi Universitari e di Perfezionamento Sant Anna
Authors:	Antonio Frisoli, Massimiliano Solazzi, Daniele Leonardis, Mine Sarac, Antonio Di Guardo, Domenico Chiaradia, Massimiliano Gabardi, Simone Marcheschi, Domenico Buongiorno, Torben Cichon, Tobias Klamt
Work package:	WP3 – Operator Interfaces
Date of preparation:	2018-07-25
Type:	Report
Version number:	1.0

Document History

Version	Date	Author	Description
0.1	2018-06-20	Antonio Frisoli, Massimiliano Solazzi, Daniele Leonardis, Mine Sarac, Antonio Di Guardo, Domenico Chiaradia, Massimiliano Gabardi, Simone Marcheschi, Domenico Buongiorno	First draft released
0.2	2018-07-18	Torben Cichon	General Setup of Control Station added
0.3	2018-07-19	Tobias Klamt	Material for additional operator interfaces added
0.4	2018-07-25	Tobias Klamt	Prepare for submission
0.5	2018-08-02	Daniele Leonardis	Conclusions and set to public version
1.0			Submitted version

Executive Summary

This deliverable presents the final version of the operator control station developed for teleoperating the Centauro robot. The bilateral arm exoskeleton for the first operator has been integrated with the wrist modules and with the hand exoskeleton, obtaining an innovative system to provide full control of the robot limbs directly through movements of the operator's limbs, and providing force feedback. The first operator can control robot navigation by two pedals with three degrees of freedom. Stereo visual feedback is provided by a Head Mounted Display.
In addition, we developed several visualization and control interfaces for the support operators. Those enable the user to keep an overview over the scene, including the robot state and the environment, and to trigger locomotion and manipulation actions on different levels of autonomy. With respect to the first version of the Control Station presented in Deliverable 3.3, the final version presented here has been validated and improved through test results obtained in the Centauro Evaluation Camp and other additional experimental activities. In particular, hardware and electronics of different modules (wrist and hand exoskeleton modules) have been improved towards higher robustness and integrated communication with the rest of the complex Control Station system, and with the teleoperated Centauro robot. Moreover, additional experimental activities have been carried out for improving control techniques used in the teleoperation of the arm and the hand exoskeletons. The bilateral teleoperation of the arm with force feedback has been improved in stability through further development of the Time Domain Passivity Approach based controller and of the position-drift compensation to deal with communication delays. Regarding teleoperation of the hand through the under-actuated hand exoskeleton, calibration of the device, estimation of the hand pose, matching of the kinematics with the different kinematics of the robotic hand, and force feedback have been further investigated for improving the performance of the teleoperation and the quality of the sensory feedback provided to the operator.

Contents

1 Introduction

This deliverable presents the development of the final version of the Centauro Control Station. It contains improvements on different modules of the system and their final integration in the second version of the Control Station. In its second version, the system is composed of bilateral robotic exoskeleton modules worn at the arms, wrists and hands respectively, an immersive visualization module, and pedals for controlling navigation of the Centauro robot.

Regarding the control station for the first operator, additional experimental activities and relevant improvements have been performed for the hand-exoskeleton module, in terms of kinematic mapping with the teleoperated robotic hand, and introduction of the force-feedback, useful for the operator for proper modulation of the grasping forces. Estimation of the pose of the user's hand in the under-actuated Handexos has been further investigated developing a novel, faster and more reliable calibration method. Mapping of the estimated pose with the different kinematics of the teleoperated Handexos has also been developed and evaluated experimentally. The introduction of force-feedback for finger grasping has been investigated with the aim of providing informative sensory feedback to the user regarding contact threshold with grasped objects and modulation of the exerted grasping forces. The experimental activity was conducted on the Handexos and the robotic right hand of the Centauro Robot, exploring the capabilities of the master-slave system which depend on the available sensors and actuators. A force feedback teleoperation loop was finally implemented and tested between the two devices (Section 3.2).

Regarding the wrist exoskeleton module, a second version of the WRES exoskeleton has been developed, based on the design of the first prototype with improved electronics, for implementation at both exoskeleton's arms (Section 2.2). It features custom-made electronics designed for being embedded in the wrist exoskeleton and with teleoperation capabilities.

The arm exoskeleton module of the Control Station has been further tested for robustness in continuous operation, considering the additional payload provided by the wrist and hand-exoskeleton modules in the final system setup. Moreover, stability of the arm exoskeleton in teleoperation was further tested with additional experimental sessions.

Besides the control station for the first operator which includes the exoskeleton and the immersive visualization through a head-mounted display, we developed several visualization and control interfaces for supporting operators. Processed and visualized sensory data enables those operators to keep an overview over the robot state and the environment. Several locomotion and manipulation controls on different levels of autonomy and respective user interfaces are presented. Those control functions are powerful in certain task classes and can be used to support and relieve cognitive load from the first operator.

2 Final Version of the Control Station

The Centauro Control Station is designed to provide immersive control of the Centauro robot by means of arm and hand movements tracked by an active exoskeleton, and immersive visual and audio feedback for the first person operator. Moreover, additional visualization modalities and control interfaces are provided to supporting third person operators in order to supervise robot actions performed on different levels of autonomy.

1. Control signals (from the Control Station to the Centauro Robot):
    - Control of the Centauro arms and hands directly via movements of the first person operator through the full upper-limb exoskeleton.
    - Control of the Centauro arms and hands from the third person operator using multi-DoF (degrees of freedom) input devices and/or GUIs in direct control mode, semi-autonomous or autonomous operation.
    - Navigation control of the Centauro robot through pedals from the first person operator.
    - Navigation control of the Centauro robot from the third person operator using multi-DoF input devices and/or GUIs in direct control mode, semi-autonomous or autonomous operation.
2. Feedback signals (from the Centauro Robot to the Control Station):
    - Visual feedback of the remote scene (raw and processed data) provided through an immersive Head Mounted Display (first person operator) and additional monitor interfaces (third person operators).
    - Audio Feedback of the remote scene.
    - Force Feedback of contact forces between Centauro arms and hands and the remote environment, applied to the exoskeleton.

In the following, the direct control of the Centauro robot with the full-body exoskeleton is described in detail including its force feedback capabilities. Support operator control functions and interface visualizations using classical input devices and ROS are described in detail in [12] and Section 4. As the visual feedback for the first-person operator and partly for the support operators is mainly done using the simulation in-the-loop (rendering raw and processed data onto stereoscopic and monoscopic output devices) this is described in deliverable D4.4 [5] in Sections 4 and 5. Audio feedback is based on the Microsoft Kinect's microphone and uses ROS to directly transmit the captured sound.

The control station for the first person operator follows the criteria of teleoperating the robot as if the operator is in place of the robot. Thus movements should be controlled and forces of arms and hands should be experienced as when moving his/her own limbs. Different exoskeleton modules are integrated to fully cover upper limb segments and a wide range of movements. Importantly, while on the one side movements of the user are tracked by the exoskeleton, force feedback of the robot interactions with the remote environment is provided to the user through the exoskeleton. Following the criteria of immersive teleoperation of the robot, the control station integrates an immersive visual and audio feedback module by means of an Head-Mounted-Display. Finally, pedals are used by the operator to control navigation of the Centauro Robot. Figure 2 shows the control station in development and application and the

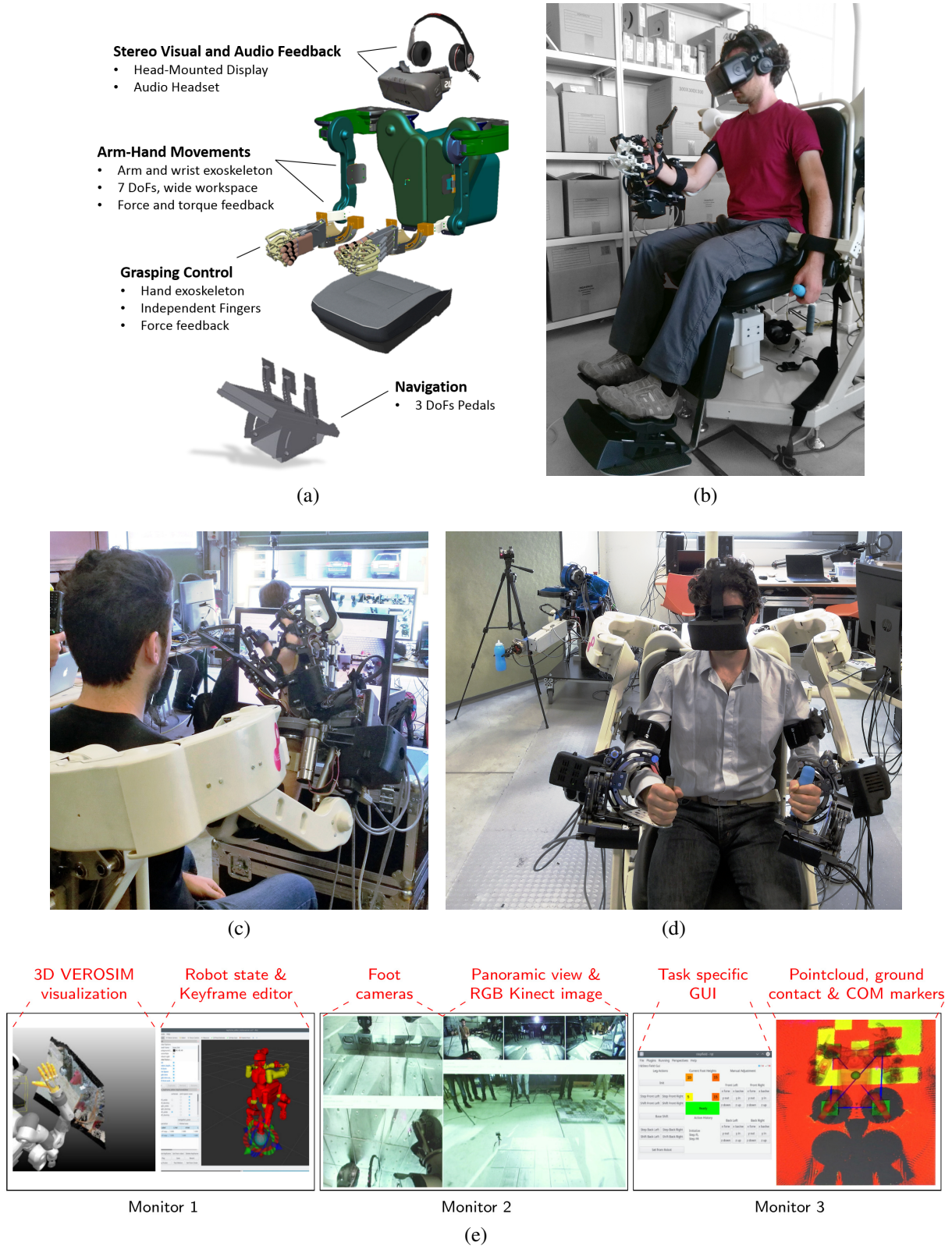


Figure 1: The Centauro Control Station: scheme (a), implementation (b), and application to the Centauro teleoperation scenarios (c),(d). Third person operator interface: environment and robot state visualization(e).

developed visualizations for the support operators.

The full upper limb exoskeleton is composed of:

- two upper limb active exoskeletons for the arms (4 DoFs each);
- two active wrist exoskeletons (3 DoFs each);
- an active hand orthosis (under-actuated, 1 DoF for each finger).

The upper limb arm-exoskeleton provides full support for shoulder and elbow rotation, implementing an innovative lightweight and backdrivable transmission for the complex 3 DoF shoulder joint. It is based on the ALEx (Active Lightweight arm Exoskeleton) bilateral robot [15],[16] described in Section 2.1. The whole upper limb exoskeleton can reach most of the natural workspace of the human arm without singularities, covering an extended range of motion for each DoF. The arm exoskeleton allows interaction at the level of the user's forearm, hence an active 3 DoFs wrist exoskeleton has been added (Section 2.2) to support and to provide torque feedback of wrist rotations [2]. Such additional DoFs are critical especially in bimanual tasks, where the same rigid body is grasped by both hands. Finally, a lightweight, under-actuated hand exoskeleton was implemented for teleoperation of grasping (Section 2.3), providing independent finger movements with force feedback.

2.1 Final Implementation of the Arm Exoskeleton

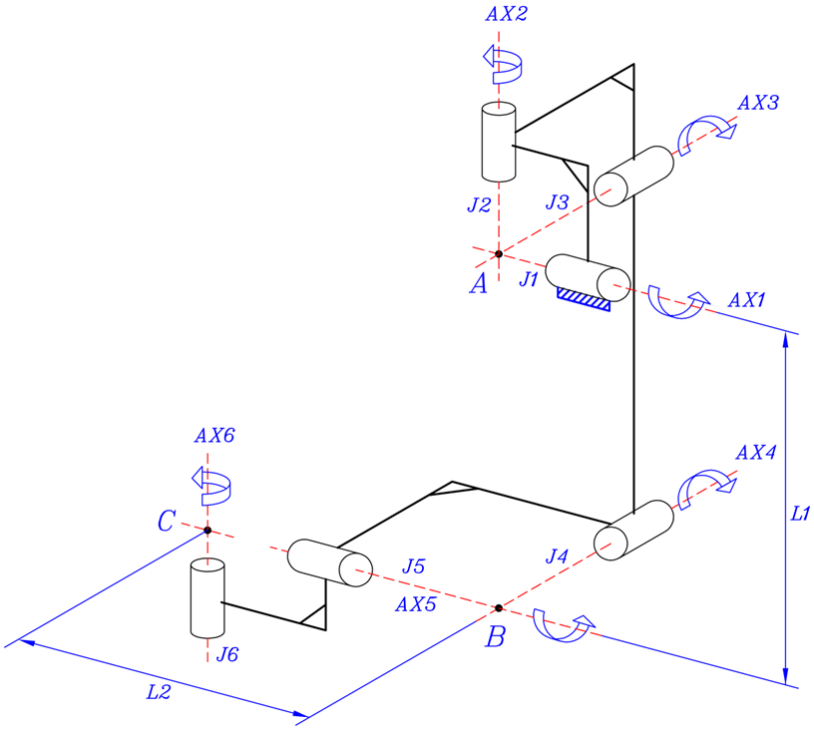
The dual arm active exoskeleton has been designed to feature

- a large range of motion to not restrict the movements of the operator,
- a high degree of compactness such that it does not interfere with the arms of the operator and avoids self-collisions between the two limbs during dexterous manipulation, and
- a low weight to reduce the inertia of the device and to improve dynamic response.

The Arm Lightweight Exoskeleton (ALEx) is a mechanically compliant robotic manipulator, actuated by electric actuators, remotely located with respect to the robotic joints. The remote location of the actuators allows to keep the masses and the encumbrances of the moving parts low, in order to achieve the highest possible dynamic performances and the widest admissible joint ranges of motion, without interference with the human body. For ALEx, the total weight of the moving parts is only 3 Kg, of which about 2 Kg are due to the first two proximal links.

The torque transmission from the actuators to the joints is achieved through idle pulleys and in-tension metallic tendons that are the main responsible for the intrinsic mechanical compliance of the device. This solution allows to implement high performance force control without the use of force and torque sensors located at the end-effector of the manipulator or at its joints. Conversely, two position sensors are used: one at the motor shaft, the other at the joint shaft, after the tendon transmission. The difference of the relative position measured by the two encoders is an indirect measurement of the tension of the tendon. Such method allows to achieve a higher degree of robustness in terms of the stability of the force control to the variability of the human limb mechanical impedance. To assure the best degree of reliability of the device, the sensory equipment of ALEx consists of only position sensors of two different technologies of

(a) (b)



(c)

Figure 2: The arm exoskeleton, featuring four actuated DoFs (shoulder and elbow) in a wide operative workspace

proven high robustness: optical incremental encoders, embedded in the electric actuators, and Hall-effect based absolute angular sensors, directly integrated in the robotic joints.

The arm exoskeleton kinematics is isomorphic to that of the human arm (i.e. the axes of rotation of the DoFs of the device are substantially aligned with those of the user physiological articulations) and has the important property of presenting no singularities in the natural workspace of the upper limb. Indeed, excluding some extreme postures that the human arm can reach but that are not of interest for the applications, in the natural workspace of the arm the rotational axis of the elbow articulation is sufficiently far from being oriented orthogonally to the frontal plane of the body. In order to leave enough space to fit for the user's shoulder encumbrance, without having any interference with the mechanical structure of the device, the

implementation of the 2nd DoF has been achieved using a remote center of rotation mechanism (RCRM) based on a four-link pantograph. The wide ranges of motion at the shoulder are $120°$, $95°$, $165°$, for the first, second and third DoF, respectively.

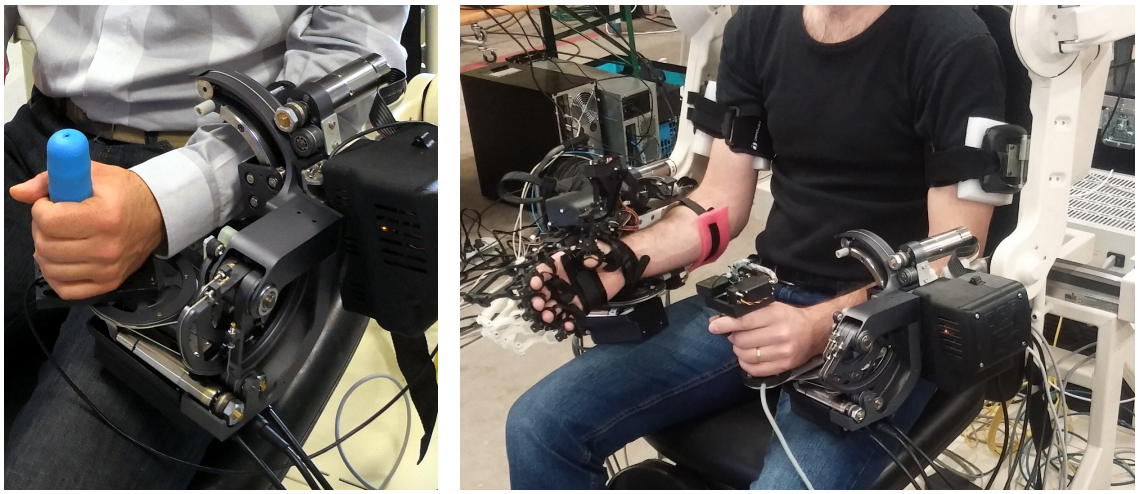
The most part of the control electronics of ALEx has been purposely developed to be embedded into the mechanics of the device, in order to minimize the electric wiring to the motors and to the sensors. The non-moving base link of the device (named the backpack) hosts the electric actuators (brushless DC motors) and the embedded control electronics and is connected to a rigid supporting structure that holds the power supply unit and the main control computing unit. A chair, having a manual hydraulic height regulation of the seat, allows the user to operate the device from a comfortable seated position.

2.2 Final Implementation of the Wrist Exoskeleton

The WRist ExoSkeleton (WRES), designed and implemented for the Centauro Control Station, is a purely rotational 3 DoF forearm-wrist exoskeleton based on serial kinematics (shown in Fig. 3 and in Fig. 4), and actuated by a tendon-based transmission. The device is able to elicit torques on the user's articulations: forearm pronation/supination (PS), wrist flexion/extension (FE) and radial/ulnar deviation (RU). Main requirements concerned the lightness of the device, its easiness to be worn, and the need to have an open structure in order to allow the user to manipulate real objects or avoid collision between the upper bilateral arms exoskeleton ALEx during bimanual operational tasks.

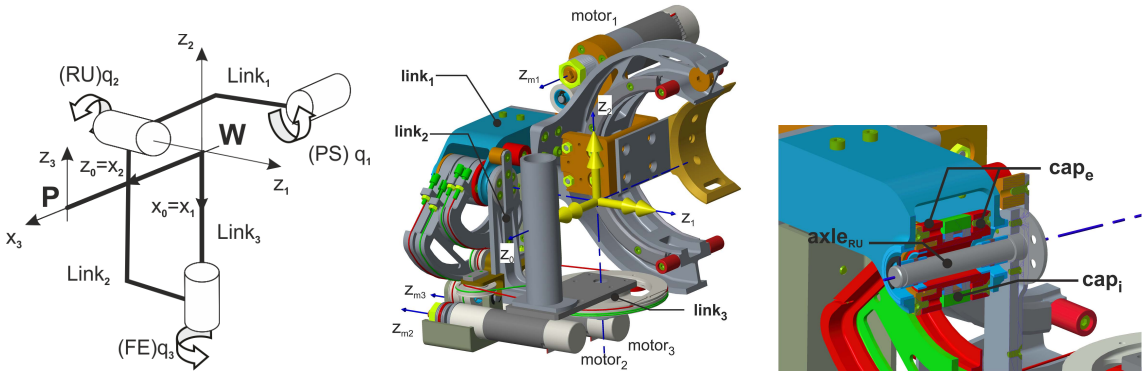
Kinematics & Mechanical design The basic kinematics structure of WRES, depicted in Fig. 4a, is characterized by a spherical 3 DoF kinematics based on a serial configuration that allows quasi-full RoM (range of motion) for the human wrist. A passive regulation of the handle position along the PS axis allows to adapt the last link length to the users hand size.

The PS joint has been designed to improve the wearability of the wrist device, by using an open curvilinear rail and rolling slider solution. Two $180\,\mathrm{deg}$ rail circular segments are mounted on both side of the moving cylindrical open hub. Two slider mechanisms, each mounted to a



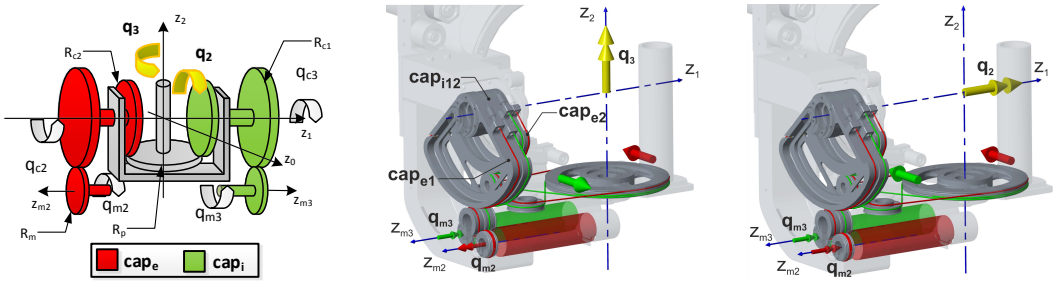
(a) (b)

Figure 3: Detail of the WRES device (a) and the two WRES devices in final configuration integrated in the Centauro Control Station and worn by the operator (b).



(a) (b) (c)

Figure 4: The schematic representation of the wrist device kinematics is depicted in (a); the correspondent CAD model is shown in (b).
The section from the CAD model in (c) shows the mechanical solution adopted to allow that the four capstans were mounted on the same side with respect to the $z_0 - z_2$ plane. The hollow shaft, which insists on the main RU axle through a couple of steel ball bearings, yokes both the external capstans (in red). At the same time, the hollow shaft supports the internal capstans pair (in green) through a couple of thin steel ball bearings.



(a) (b) (c)

Figure 5: The schematic representation of the differential transmission kinematics and the main geometric parameters are reported in (a), where: $R_{c1} = 69.5mm$ the driving capstan radius, $R_{c2} = 46mm$ the driven capstan radius, $R_p = 46mm$ the differential pulley radius and $R_m = 10mm$ the motor drum radius. The two CAD model representations show the cable routings for the motors 2 and 3. In (b) the configuration for $q_{m2} = |q_m|$ and $q_{m3} = -|q_m|$ is shown which results in $\boldsymbol{q_{FE}} = [0 \quad q_3]$, whereas in (c) for $q_{m2} = -|q_m|$ and $q_{m3} = -|q_m|$, we obtain $\boldsymbol{q_{RU}} = [q_2 \quad 0]$.

fixed frame, are used to support the required moment loads. The actuators 2 and 3 are remotely located with respect to the end-effector, both mounted on the link 1, in order to obtain the highest dynamic performances together with a simple configuration for the cable routing of the differential transmission. The kinematics adopted in the last two DoF (RU and FE) of WRES combine the output motion of two parallel actuators by using a differential transmission, where the output, the joint variables q_2 and q_3, are given by the linear combination of the two motor variables, q_{m2} and q_{m3}.

Fig. 5a shows a schematic representation of the transmission kinematics and the geometric notation adopted for its description. Both motors act in parallel on the central main pulley through a double-stage pulley transmission each. In order to make the structure both easy to be

worn and not interfering with workspace between the hand palms of the user, two double-stage transmissions (named in Fig. 5a as cap_i and cap_e for the internal and external one, respectively) have been arranged on the same side, with respect to the $z_0 - z_2$ plane. The first-stage pulley transmission for each motor is given through the capstans cap_{e1} and cap_{i1} (namely, driving capstans). The external capstan pair (red in Fig. 4c) is coupled through a hollow shaft, whereas the internal ones (green in Fig. 4c) are coupled rigidly. Both capstans, $2e$ and $2i$, directly drive the pulley of the link 3, and the link 2.

Each DoF is actuated by a BLDC gear motor controlled with a digital EtherCAT drive. In order to achieve an good transparency, the low-level control of the wrist exoskeleton takes into account the gear head efficiency and two main feed-forward compensation terms: gravity compensation and the gear motors' viscous friction compensation.

Final Implementation of the WRES in the Control Station Two WRES exoskeleton modules, for the right and left wrist, have been developed and integrated in the final version of the Centauro Control Station. In the final version, the two WRES modules feature a renewed control electronics fully interfaced with the rest of the Control Station modules and with the main control unit through the reliable and real-time EtherCAT communication protocol. The renewed electronics implement custom-made drivers, analogous to the units used for the actuators of the arm exoskeleton. This improves modularity and robustness of the whole Centauro Control Station system by using similar low level communication and electronics for the different actuated modules.

2.3 Final Implementation of the Hand Exoskeleton

The presented 5-fingered hand exoskeleton has been designed adopting the under-actuation concept. Each under-actuated finger component is attached to the corresponding finger of the user at two points to achieve 2-DoF mobility, and is controlled through a single actuator. The actuator applies forces to the finger phalanges to flex/extend the finger joints either to help users perform grasping various objects or to provide users the perception of interaction during teleoperation tasks. The parallel under-actuated mechanism adjusts the force transmission and the motion of the user based on the interaction forces acting on the finger phalanges or the guidance of users. In particular, covering the full mobility of the user's finger allows them to move their fingers freely with no physical limitation regarding the relation between the two assisted finger joints.

The hand exoskeleton is composed of 5 finger components and is equipped with 5 actuators in total. Thanks to the under-actuation concept, the exoskeleton is equipped with a lower number of actuators with respect to the total DoFs of the entire hand, without sacrificing the independent mobility of the assisted joints. Minimizing the number of actuators lets the hand exoskeleton to a lighter and a smaller structure. Doing so, the compact hand exoskeleton with 5 parallel finger chains and 5 actuators mounted on a ground link is rigidly connected to the back of user's hand as depicted in Figure 6. The ground link has been designed specifically considering the shape of the human hand. A foam layer has been inserted between the ground link and user's hand to improve the adaptability of the system for different hands and to favor user's comfort during interaction.

Kinematic design principles and structural design The finger components for the index, middle, ring and little fingers are connected to the proximal and the intermediate phalanges of the corresponding finger to assist the MCP and PIP joints [21]. In particular, the finger

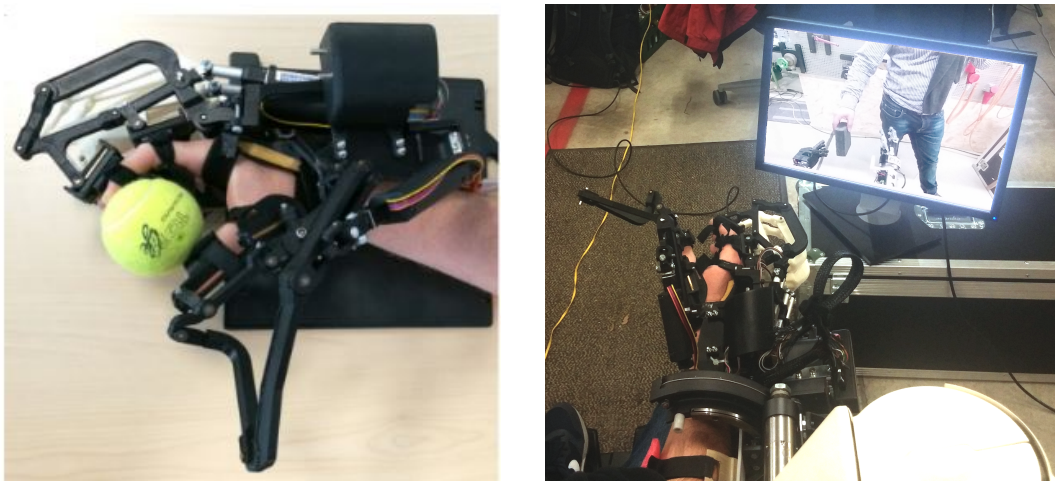


Figure 6: Close-up of the hand exoskeleton worn by the user performing the grasping of a standard tennis ball (left); hand exoskeleton attached to the telepresence station while teleoperating the Centauro robot (right).

components for the ring, middle and little fingers are limited to the planar axis, since the mechanism only allows flexion/extension for the MCP and the PIP joints. On the other hand, the index component is added an additional passive DoF for abduction/adduction of the MCP joint, which helps the finger to spread. Similarly, the finger component for the thumb is connected to the metacarpal and the distal phalanges to assist 5 DoF [7].

The design requirements of the presented hand exoskeleton has been defined as the capability of grasping objects with generic shape, effective transmission of forces to the phalanges and self-adjustability to different sizes of the fingers. Each requirement can be achieved by adopting certain design guidelines. Adopting a high level of under-actuation by the kinematics of the finger components allows the user to grasp objects with various forms and sizes with no prior adjustments. Designing the interaction points between the exoskeleton and the finger phalanges allows the tangential forces to be diminished, so that applying only perpendicular forces to the finger phalanges would increase the efficacy and naturalism of the force transmission. Furthermore, including the finger model to the kinematic loops of the finger components would handle the self-adaptability of the device's operation for different hand sizes.

The link lengths of the finger components have been optimized specifically to achieve the best performance in terms of force transmission and allowed range of motion for the finger joints. The device has been realized using 3-D printed parts, minimizing the overall weight and cost. For the actuation of each finger component, Firgelli $L16$ linear motors have been chosen with 50 mm stroke limit. The lack of magnetic hardware and the miniaturized hardware of the actuators allow them to be placed close to each other on top of the hand. The mechanical specifications of the device and the actuators have been summarized in Table 1.

Sensorization and backdriveability Despite of its practical advantages, adopting the under-actuation concept prevents the finger pose to be known during operation without using additional strategies. The index, middle, ring and little finger components are equipped with one additional potentiometer (BOURNS 3382G-1-103G) each. The flexion/extension of MCP and PIP joints are calculated using the direct kinematics of the overall system using the data coming from the additional potentiometer and the actuator encoder. The index component is equipped with one more sensor, to measure the abduction/adduction of the MCP joint. The thumb com-

Property	Value [mm]
Device Mass	≅ 350 g.
Motor gear ratio	35:1
Stroke of the motor	50 mm
Max. cont force of the motor	40 N
Backdrive force	31 N
Max. velocity	32 mm/s

Table 1: Specifications of the proposed exoskeleton and its actuators.

ponent is equipped with one potentiometer for the abduction/adduction of the CMC joint, while the flexion/extension of the thumb is extracted from the actuator encoder in general.

Using Firgelli linear actuators provide great advantages to the overall system, such as simplicity, portability and affordability. However, its internal mechanical gear box prevents the backdriveability of the motors. The backdriveability can be gained by inserting a force sensor to the mechanical loop and by implementing force control algorithms. In particular, a 1-DoF force sensor has been placed between the ground link and the actuator to measure the intentional forces applied by the user on each finger component. The control algorithm uses the measured forces to set the actuation direction and the speed of motion, which would open/close the finger as intended. The achieved highly transparent device with active backdriveability can be used as a master device for teleoperation applications.

2.4 Final Implementation of the Navigation Pedals

Navigation of the Centauro robot in remote environments is teleoperated from the Centauro Control Station through pedals integrated in the first person operator control platform. The module has been integrated in the final version of the Centauro Control Station. It is depicted in Figure 7.

The Centauro robot can move omnidirectionally on the ground plane which results in 3 DoF:

- I) moving longitudinally (forward and backward),
- II) rotating, and
- III) moving laterally (left and right)

and any combination of those. These DoF are controlled by the operator through an integrated pedal module making use of a couple of pedals with independent rotation on the horizontal axis, plus an additional rotation mechanism coupling the two pedals on the vertical axis. Thus, in the control scenario we envisaged, the right and left unidirectional pedals are used for moving the robot longitudinally and the rudder bar is used for turning to the right and left (e.g., moving the right foot forward and the left foot backward will rotate the robot to the left). Furthermore, the user can move the robot laterally by pressing both pedals simultaneously and then rotating the rudder left or right dependently on the desired direction. The hardware of the navigation module is composed by commercial pedals interfaced by custom made electronics with the control system of the Centauro Control Station. In particular, the position of each DoF of the

Table 2: Control strategy to address all three DoF of the omnidirectional Centauro robot driving capabilities with the 3 DoF pedal controller (rudder, left pedal and right pedal). The double "'+"' sign indicates that both pedals have to be pressed more than half of their maximum range

TASK	**RUDDER**	**LEFT**	**RIGHT**
No move	0	0	0
Move Forward	0	0	+
Move Backward	0	+	0
Turn Right	+	0	0
Turn Left	-	0	0
Move Right	+	++	++
Move Left	-	++	++

pedals is acquired by a microcontroller and forwarded to the main control system through fast UDP communication.

Figure 7: Pedals for controlling the navigation of the Centauro robot integrated in the Control Station

3 Teleoperation Experiments

3.1 Teleoperation of the Arm Exoskeleton

Teleoperation of the arm and wrist segments allows the operator to remotely control the position and the orientation of each of the two Centauro robot's limbs. The general teleoperation approach is based on the position-measured force ($P - F_m$) paradigm. For the sake of clarity, a simplified version of the teleoperation concept is shown in Fig. 8. It visualizes the all the exchanged information between the operator station (or master) and the Centauro robot (or slave). As in Fig. 8, the used paradigm considers the master velocity as the variable sent by the master to the slave, and the interaction force between the slave and the environment as the information sent in the opposite direction. Hence, one of the main characteristic of this approach is the need of force/torque sensing at the slave side.

Since the remote manipulation tasks are defined in the Cartesian space, the scheme $P - F_m$ (Fig. 8) has been applied independently at each of the six DoFs of each hand (three translations and three orientations), i.e., at the end-effector level. Although master and slave are based on the same kinematic (serial kinematic with 7 revolution joints), the exoskeleton and the Centauro robot have different link lengths and thus different workspaces. To deal with this asymmetry, a positional scaling has been introduced to map the slave's workspace into the master's one. It is worth noting that the master imposes only six of the seven slave arm DoFs. Thus, the slave arm final pose was set by using an inverse kinematic algorithm that exploits the additional DoFs for improving dexterous manipulation.

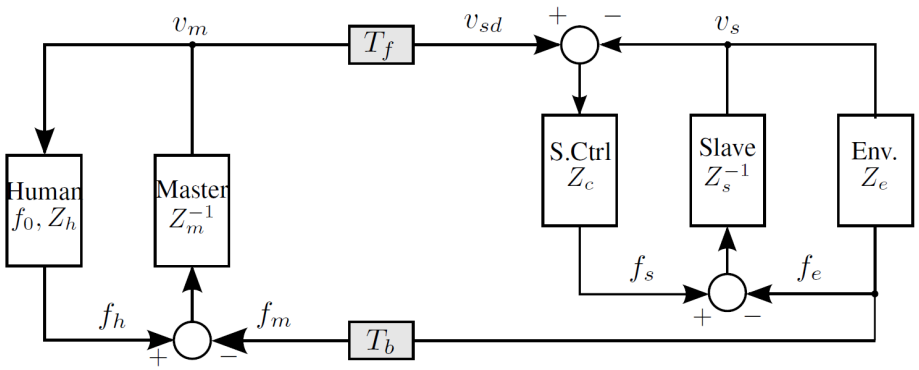


Figure 8: The block diagram of the P-F(measured) teleoperation architecture (contribution from [1]). v_m, v_{sd} and v_s are the master velocity, slave desired velocity and slave velocity; f_h, f_m, f_s and f_e are the human force, the desired force rendered by the master, the actuated slave force and the interaction force between the slave and the environment; finally, T_f and T_b are the forward and backward time delays. It is worth noting that, in the P-F(measured) paradigm, the desired slave velocity is set equal to the master velocity and the desired rendered force by the master is set equal to the slave/environment interaction force.

It is well known that communication delays in bilateral teleoperation can lead to an unstable behavior of the teleoperator system [1]. For this reason, the time domain passivity theory has been implemented independently for each of the six DoFs. The Time-Domain Passivity Approach (TDPA) consists of monitoring the energy of the system in real-time (by using Passivity-Observers – POs) and dissipating it only when the system presents an active behavior (by using Passivity-Controllers – PCs) [8, 19]. In this scenario, we have been able to exploit all the TDPA advantages. The main advantage of this approach is its model-free property, thus no slave robot dynamics have been included in the master exoskeleton control scheme and vice

(a) (b)

Figure 9: The operator opens a door through the Telepresence Station teleoperating the Centauro robot

versa. Then, the implemented control strategy stabilizes the master/slave interaction in presence of delay and in continuous contact with surfaces while forces/torques are exerted along several axes.

The whole teleoperator system has been successfully tested in a set of remote tasks such as: door opening Fig. 9, valve turning, fire hose connection, power plug connection, sample taking, snapping a hook, using an electrical screw driver, and cutting a cable.

The performance of the position-force (measured) TDPA scheme was evaluated numerically in two tasks involving a realistic asymmetrical multi-DoF setup (asymmetrical means the two robots are mechanically different). The experiments in this phase involved the master exoskeleton of the Centauro project (i.e. ALEx) and another exoskeleton used as slave that exhibits characteristics similar to the Centauro slave robot. The setup of these experiments is depicted in Fig. 10. The user that drives the master needs a device as light and transparent as possible. The slave robot has to exhibit a solid structure, the possibility to carry a relatively high weight and high dynamic bandwidth.

Regarding the master device, the local control is an open-loop force control. In detail, the master controller applies to the master device:

1. the torques/forces to be rendered (obtained from the slave measured forces),
2. the gravity compensation torques, and
3. both the static and viscous friction compensation terms to ensure the transparency.

The slave device implements a closed-loop position control, i.e., a PD control with a gravity and dynamic compensation. The dynamic compensation term is computed by using an acceleration observer that uses both joint torque sensors and encoder position readings. This setup was previously introduced in Deliverable 3.3 [6] at Section 7.

Both the master and slave controller ran with a time step of 200 μs, while the round-trip communication delay was less than 1 ms when no time delay was introduced. All the conducted experiments involved the master robot's left arm because the slave exoskeleton was configured as a left arm.

The new proposed benchmark is composed by two sets of experiments:

- (Experiment 1) Continuous contact with the edge of a stiff non-planar surface and
- (Experiment 2) Following a trajectory while carrying a load.

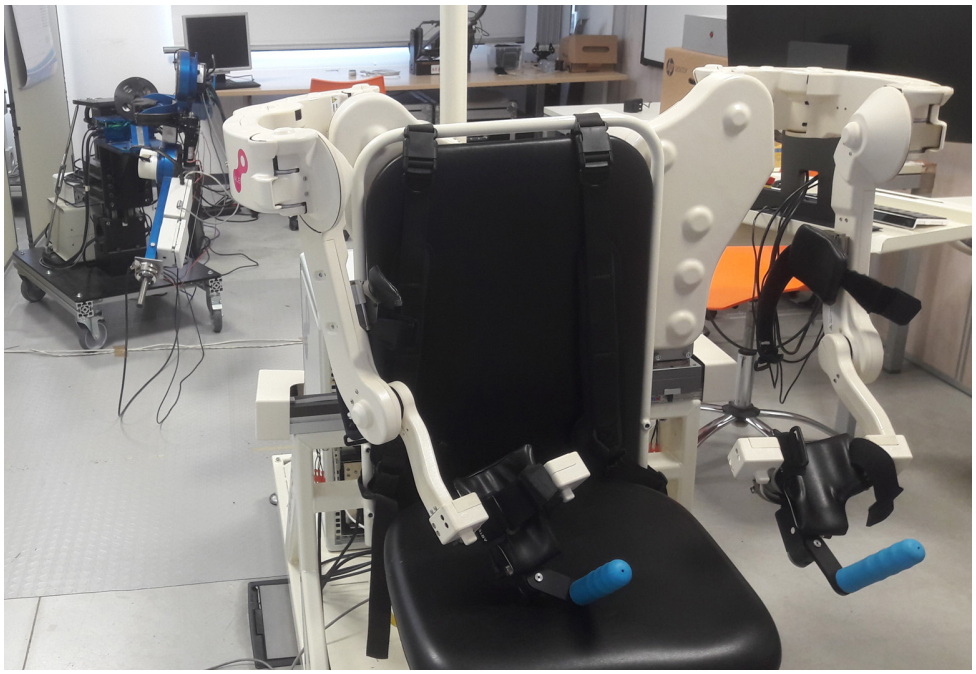
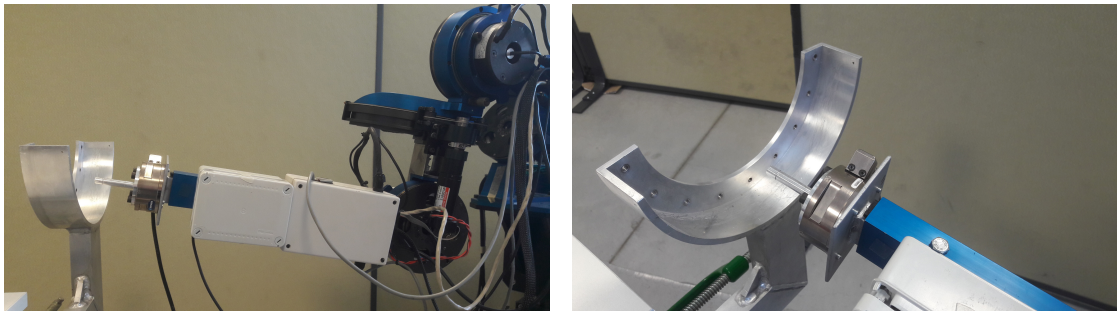


Figure 10: The setup of the experiments. In the back left: the slave device and on the right: the master device without the human operator.



(a) (b)

Figure 11: The setup of Experiment 1. In (a), the exoskeleton arm while approaching the round stiff surface. In (b), a particular of the aluminum round-shaped stiff surface.

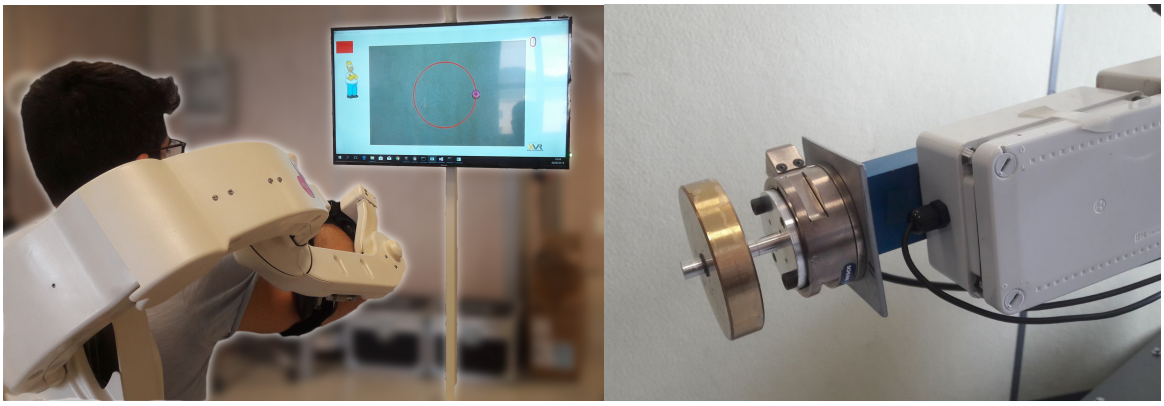
3.1.1 Experiments

Experiment 1 In the Experiment 1 the master exoskeleton operator was asked to teleoperate the slave robot while keeping continuous contact to the edge of a stiff non-planar surface (Fig. 11). The selected edge was a semi-circumference of an aluminum piece with a radius of $90\ [mm]$. This experiment has the objective to evaluate the control performance in the scenario of a continuous force exchange between the slave and the environment as for example in a door opening task or in a valve turning task. The Experiment 1 was articulated in two different tests:

- Continuous contact in case of enabled delay and passivity controller and
- continuous contact in case of disabled delay and passivity controller.

Experiment 2 Experiment 2 features load handling in a workspace. The master exoskeleton operator was asked to follow a visual circular reference trajectory at a constant angular speed

of 0.7 $[rad/s]$. At the end-effector handle of the slave robot a 1 $[Kg]$ cylindrical weight was attached, whilst the circumference diameter was 260 $[mm]$. The setup of Experiment 2 is depicted in Fig. 12. Again, Experiment 2 was articulated with enabled and disabled delay and passivity controller.



(a) (b)

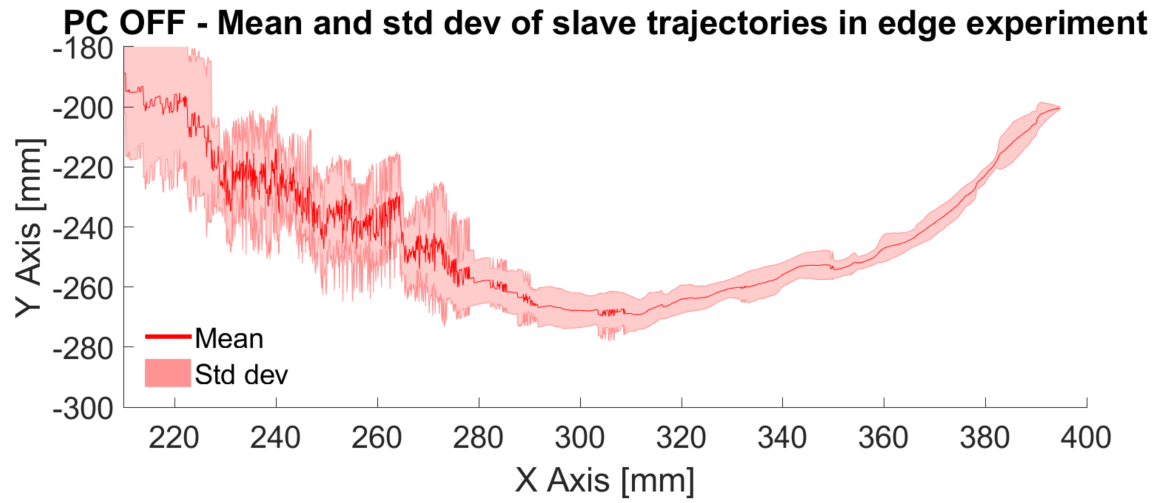
Figure 12: The setup of the Experiment 2. In (a), the master exoskeleton worn by the operator while following given trajectories at constant speed. The position references are given through a visual target displayed on a screen. In (b), a detailed capture of the slave end-effector on which the external mass was mounted.

3.1.2 Results

Continuous contact with a stiff edge (Experiment 1) This subsection presents the results of Experiment 1 that highlight the stabilizer effect of the time domain passivity controller also with an asymmetrical teleoperation setup in a challenging task, i.e., continuous contact where forces are exchanged over the three Cartesian axes. Figures 13a and 13b show the slave robot mean trajectories with a solid red line with the disabled/enabled PC respectively. The plot shows the aggregated position data of all trials. The pink area around the mean is the trajectories' standard deviation for a given x coordinate. A more large and spiky pink area is related with a more variable and oscillating signal. The experiment runs with disabled PC showed a noisier behavior and a lager standard deviation of the x coordinate between 200 and 280 $[mm]$ in comparison to those runs with enabled PC.

In the mentioned range, the whole teleoperation system with disabled passivity control started to oscillate at a relatively low frequency of 1.5 $[Hz]$. This behavior can be seen in Figures 14a and 14b for the x axis and the y axis respectively. Fig. 14 shows the power spectral density of the slave measured forces in more detail. In a continuous contact task, the frequency spectrum of the measured forces at the slave end-effector provides useful information on the stability of the contact. The plots highlight the presence of a resonance frequency at 1.5 $[Hz]$ when the delay is introduced in the communication loop and no passivity control is provided. The passivity control acts as a dissipative term: in this particular case it is in charge to dissipate the power generated at the resonant frequency.

The position and force data of a single trajectory of the edge test are provided in Fig. 15. In the range between 200 and 280 $[mm]$ of the x axis, the system exhibit an unstable behavior. The operator cannot stabilize the system which is underlined by the force plots in Figures 15c, 15d and 15e in the time range from 8 to 15 $[s]$. In the case with disabled PC, the slave contact



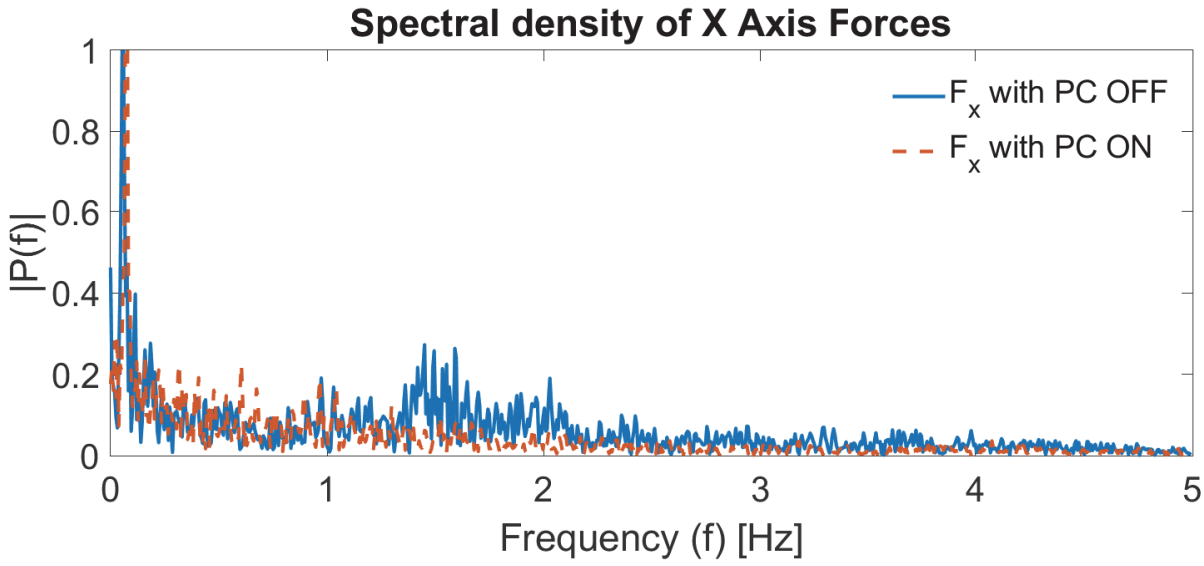
(a)

(b)

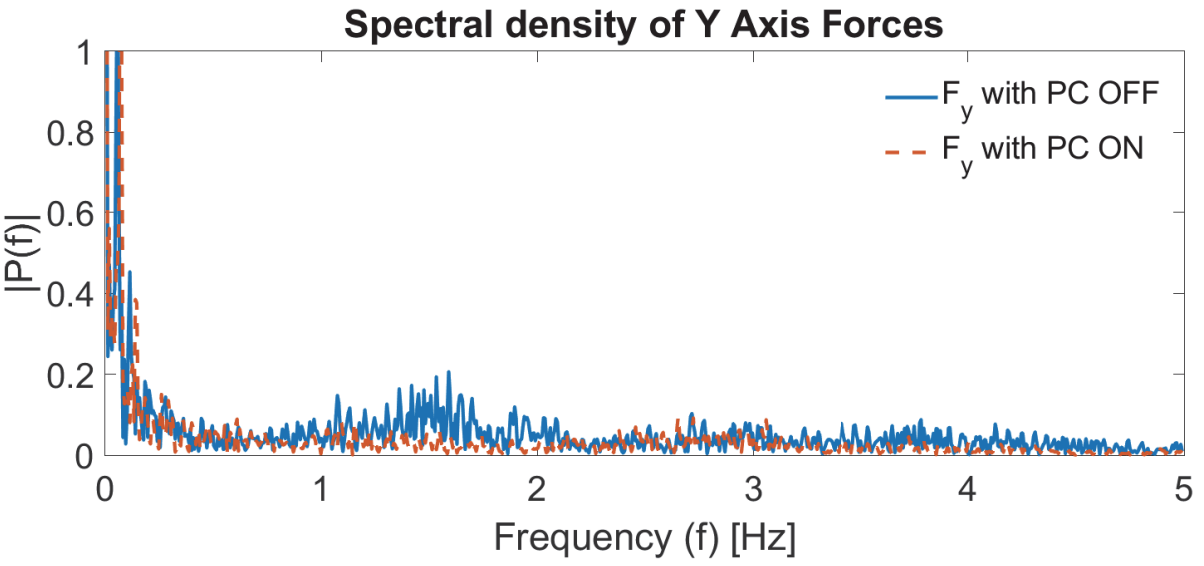
Figure 13: This figure aggregates the slave robot end-effector positions performing the Experiment 1. In (a), mean trajectory with disabled passivity controller, whereas in (b), PC is enabled. The pink area around the mean trajectory represents the standard deviation of the y coordinate for the given x axis coordinate. The larger and more irregular the standard deviation, the more oscillatory is the slave robot.

forces reached values that are two times larger than the ones in the case with enabled PC. Figures 15f, 15g and 15h show how the passivity controller dissipate the energy that is generated by the network delay through a cut of the command forces to render from the master robot. It is worth to notice that each axis contributes independently to instability and each passivity controller acts when the related axis become active from an energetic point of view; for example in the time range from 0 to1 $[s]$ only the x axis was passivated, while in the range from 1 to 6 $[s]$ only the y axis was passivated and finally, in the range from 6 to 9 $[s]$ the three passivity controllers acted simultaneously.

Load carrying over a trajectory (Experiment 2) Experiment 2 had the goal to demonstrate that the use of the passivity controller is also crucial in the case of load carrying. The obtained results are depicted in Fig. 16 which compares the results of disabled/enabled PC in the pro-



(a)



(b)

Figure 14: Frequency spectral density of slave end-effector forces. The blue solid line plots the case with disabled PC, while the dashed red line plots the case with enabled PC. (a) and (b) refer to the x axis and y axis forces respectively. The PC off case spectrum exhibits a resonant frequency at $1.5\ [Hz]$ the master driver was not able to compensate for.

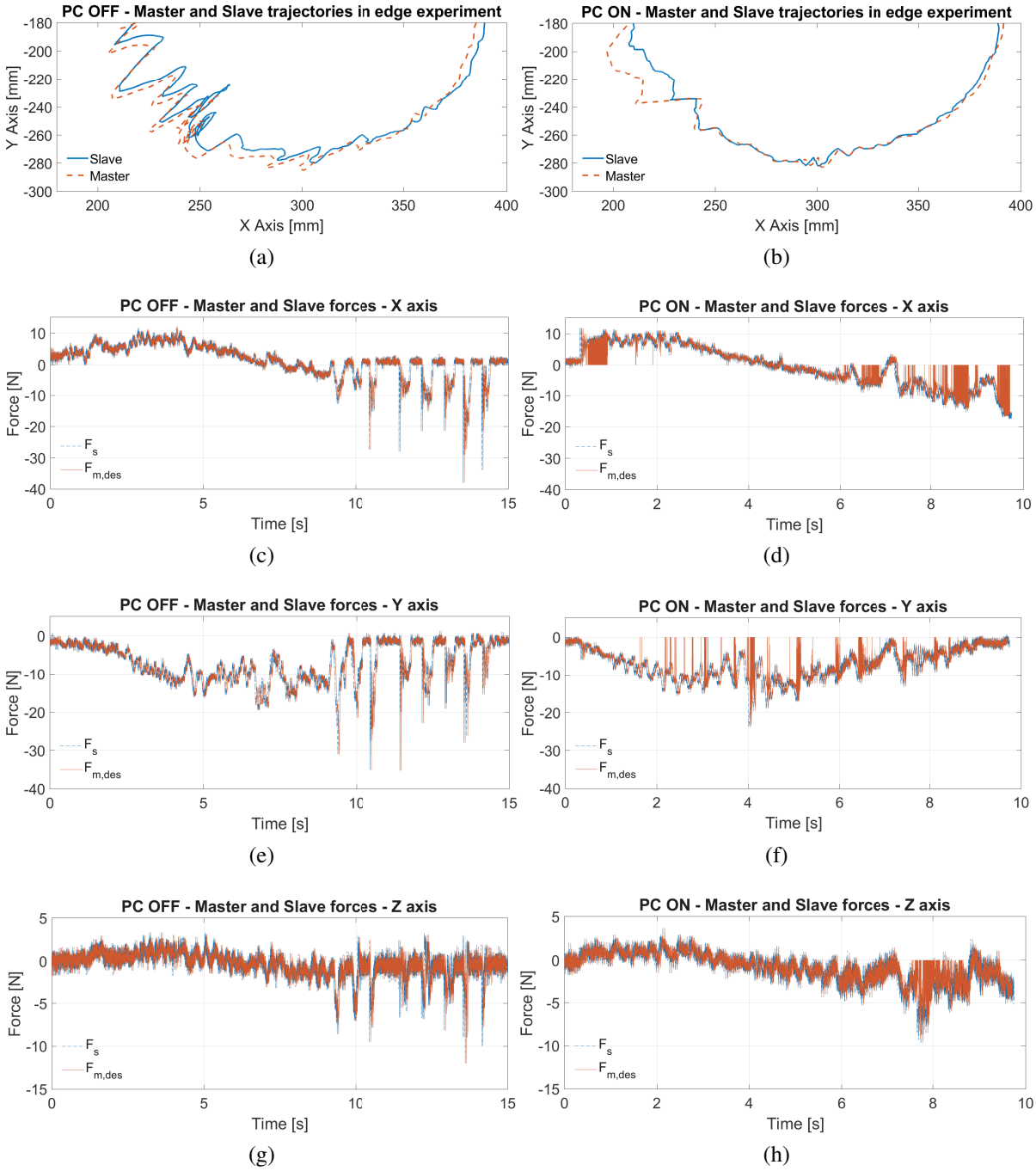


Figure 15: The results of a single trajectory during a continuous contact with a round-shaped stiff surface. The left side is devoted to the case with disabled PC, while for the results on the right side the PC was enabled. Figures (a) and (b) show the slave end-effector (blue solid) and master end-effector (red dashed) position. Figures (c)-(h) show the measured slave end effector forces (blue dashed) and the desired master forces (red solid) for the three Cartesian axes. The passivity controller increases the performances in terms of smoothness of both master and slave positions. It allows to keep a stable and continuous contact by decreasing the desired master forces when the system becomes active and dissipates the energy of the unwanted oscillations.

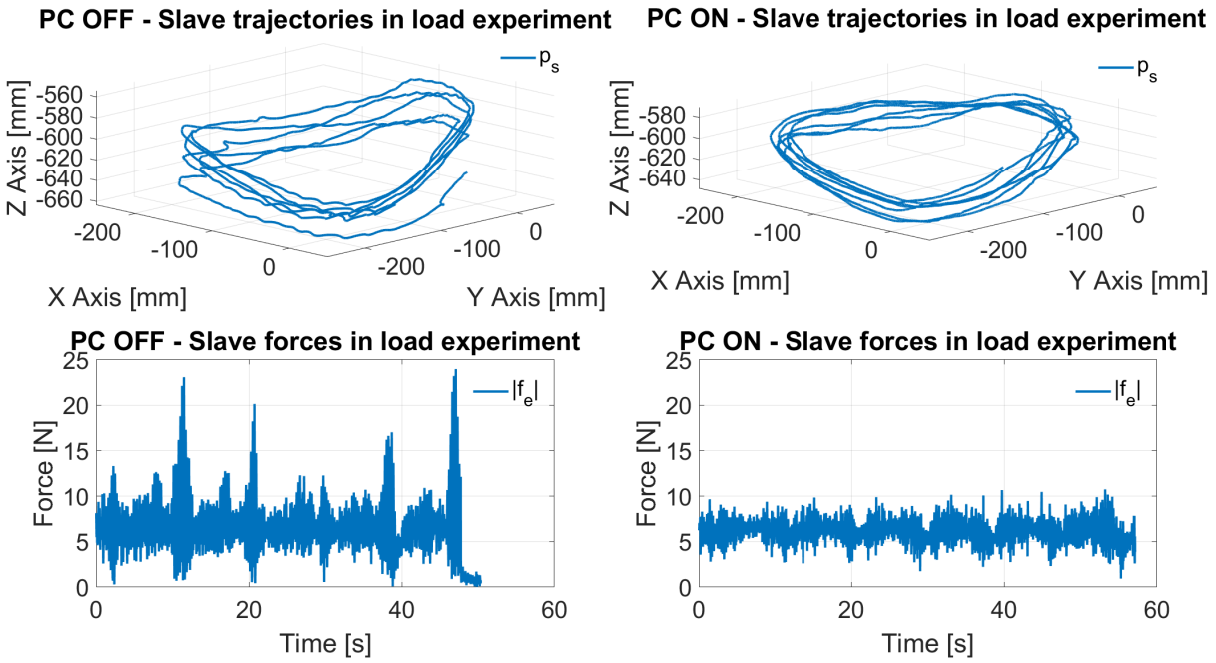


Figure 16: The results of Experiment 2 with disabled (left) and enabled (right) PC. Top plots show the measured slave end-effector position, whereas bottom plots show the forces due to the load handling. The passivity controller is crucial also in this task; in fact it eliminates the position oscillations and the force peaks that can be easily identified in the left side of the figure.

posed task. The force modulus were filter with a second order Butterworth low-pass filter, with a cut-off frequency of $50\ [Hz]$. The apparent noisy behavior is due to the large time scale.

The effect of the active behavior can be seen in the force signal, particularly at time $t = 10$, 20, 40 and $50\ [s]$. Moreover, the modulus of the force in the case of disabled PC is greater than the forces in the case of enabled PC for the entire trajectory. The oscillations originated from the active teleoperation system generate additive inertial forces of the attached load at the end-effector. The passivity controller applied on the slave end-effector positions results in smoother and more accurate positions compared to disabled PC. The master exoskeleton operator experienced a larger difficulty performing Experiment 2 in the case of disabled PC as highlighted in the position plot in the range from $120\ \text{deg}$ to $240\ \text{deg}$.

3.2 Teleoperation and Calibration of the Hand Exoskeleton

Teleoperation at the level of hand grasping and finger movements was developed in order to control the Schunk anthropomorphic robotic hand, mounted at the right arm of the Centauro robot. The aim of the teleoperation was to control finger movements and force modulation of the operator's hand directly on the remote robotic hand, and to provide force feedback of the grasping interaction with physical objects. One of the challenges in such teleoperation scenarios is that the master and slave systems (the hand exoskeleton and the robotic hand) share similar but not identical kinematics, and different actuation and control methods. In particular, the very high number of DoFs of the human hand introduces limits to the number of actuators both for the robotic hand and for the exoskeleton. Moreover, due to variability of human hand sizes, under-actuation and adaptive kinematics are design solutions implemented in the exoskeleton, further reducing the number of implemented actuators together with other wearability constraints. On the basis of the target task (in this study, fine grasping), different relations between DoFs of the

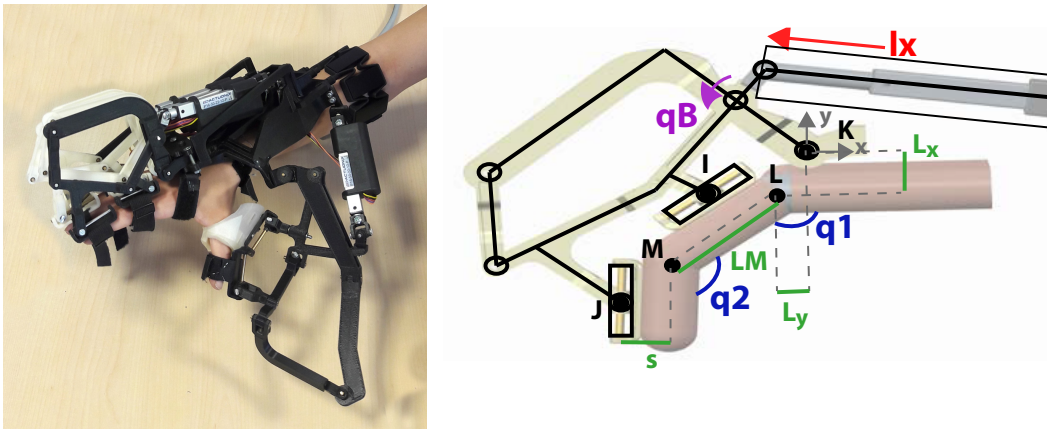


Figure 17: (a) Overview of the 5-fingered exoskeleton. (b) Kinematics model of the under-actuated hand exoskeleton with useful definitions depicted on it.

two devices have been implemented for teleoperation of finger movements, and for providing force feedback to the operator.

In this section we summarize the control approach implemented for teleoperating grasping of the Centauro robotic hands, further experimental investigations and improvements regarding the calibration and control methods are presented in Section 3.2.1 and Section 3.2 with additional details.

3.2.1 Hand Pose Estimation and Calibration

The mechanical modeling of the structure and the deterministic model of the user's finger are detailed in Figure 17b.

As mentioned before, the instant finger pose, which can be defined as q_{o1}, q_{o2}, can be calculated through direct kinematics [21, 20] using sensory measurements coming from the actuator encoder (l_x) and the additional potentiometer (q_B). These calculations also require the finger articulation parameters and the position of the MCP joint with respect to the exoskeleton to be known as constants. As presented in Figure 17b, $\overline{LM}$ is the length of the proximal phalange, s is the distance between the passive slider axes and the phalange axes along the proximal and middle phalanges, Lx and Ly are horizontal and vertical distances between the points K and L. The parameter s is supposed to be constant and equal for both phalanges. Lx and Ly drastically depend on the way the user is wearing the device and cannot be directly measured.

Even though these parameters need to be known for the kinematics analysis of the exoskeleton, it not clear at this point how much they affect the output of these analyses. In order to understand how accurate these parameters should be measured when the user wears the device, a sensitivity analysis has been performed.

The one-at-a-time (OAT) sensitivity analysis can be used to evaluate the effect of each parameter by means of a cost function in which other parameters are kept fixed. The basic idea behind this sensitivity analysis is to change each parameter individually by a predetermined amount and investigate the impact of this change as an output. For this study, the amount of change applied to each parameter has been selected as $1\ \mu m$. The change is selected minimum intentionally to highlight how much impact this little variation causes on the analysis outcome through different hand poses. The main interest of this sensitivity analysis is to understand the

impact of the aforementioned parameters on the pose analysis or Jacobian analysis.

Given a parameter E to be analyzed, the parameter values to be used for the sensitivity analysis have been set as $E_1 = E_{av} - 0.001$ and $E_2 = E_{av} + 0.001$, where E_{av} is the measurement of the base model. The output function to be investigated for the pose analysis have been computed as P_1 and P_2 using E_1 and E_2 respectively, based on $P_i = sqrt((q^i_{o1})^2 + (q^i_{o2})^2)$. Once we compute the output function for both parameter values, the impact of the parameter can be calculated simply by calculating the error rate err as depicted in Eqn. 1.

$$err = (P_2 - P_1)/(E_2 - E_1). \tag{1}$$

A similar analysis has been calculated as S_1, S_2 and S_av using the same parameter values E_1, E_2 and E_av by computing the determinant of the Jacobian matrix $S_i = det(J_i)$. Once we compute the output function for both parameter values, the impact of the parameter can be calculated simply by calculating the sensitivity index SI as depicted in Eqn. 2. The SI value greater 1 would signify that the parameter change causes a greater impact on the output function.

$$SI = \frac{(S_2 - S_1)/S_{av}}{(E_2 - E_1)/E_{av}} \tag{2}$$

At this point, it is important to emphasize that different analysis indices have been adopted for the pose analysis and for the Jacobian. Since the Jacobian values are not expected to be dramatically different from each other for different finger poses, the sensitivity index in Eqn. 2 has been formed following the theory of the sensitivity analysis. However, the output function for the pose estimation P_i changes dramatically for each finger pose, and sometimes might reach values of zero, adopting the conventional sensitivity index would be not fair for a comparison among different poses. That is why the analysis for the pose estimation has been adopted as a pose error rather than the rate of pose errors.

The pose estimation error has been computed for almost 100 possible pose combinations using Eqn. 1 for the the finger length ($\overline{LM}$), finger width (s), horizontal offset(Lx) and vertical offset (Ly) parameters. Figure 18 shows that the finger width and the vertical offset causes a higher error in deg/mm compared to others (two central plots).

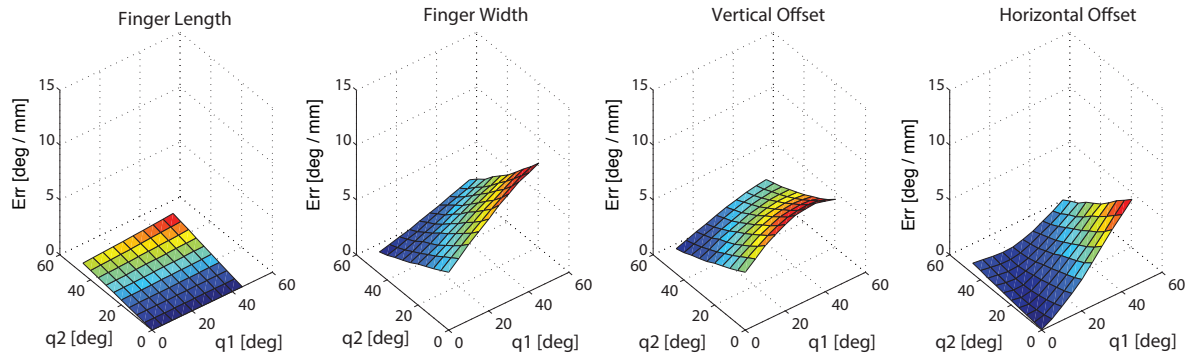


Figure 18: Parameter error estimation according to pose

Similarly, the Jacobian error has been computed for almost 100 possible pose combinations using Eqn. 2 for the the finger length ($\overline{LM}$), finger width (s), horizontal offset(Lx) and vertical offset (Ly) parameters. Please note that the sensitivity index in Eqn. 2 shows a dramatic impact of a parameter if the index is higher than 1. Figure 19 shows the output of SI for Jacobian matrix

for each finger pose, and the most crucial parameters can be found as finger width and vertical offset.

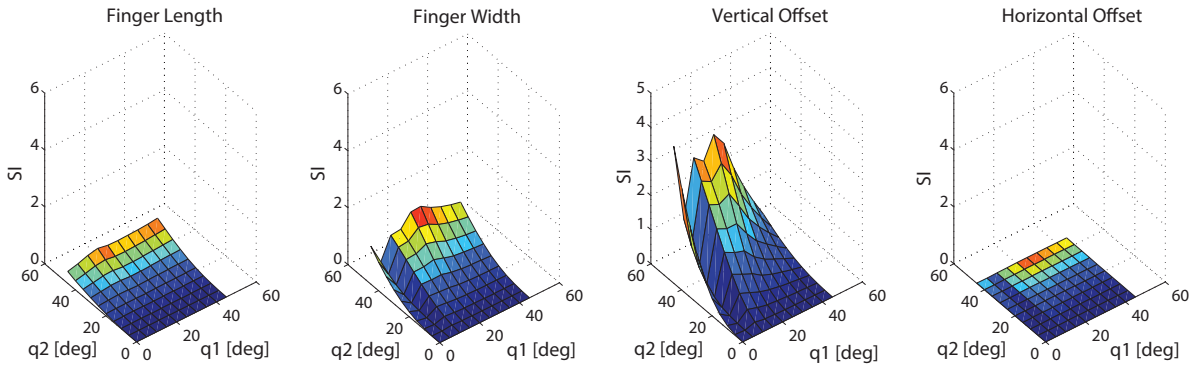


Figure 19: Sensitivity analysis output for single pose

The sensitivity analysis highlighted that all parameters, but specifically the vertical offset (Lx), become critical to compute accurate finger poses and Jacobian matrix. Since these parameters are related to the way of wearing the exoskeleton, they are difficult to be estimated before wearing the exoskeleton and definitely challenging to be measured once the exoskeleton is worn. That is why a calibration algorithm, to be run once the exoskeleton is already worn at the user's hand, has been developed.

Calibration of the Hand Exoskeleton The two-step algorithm made use of data acquired during a constrained calibration motion to geometrically compute the unknown parameters. In the first step s, L_x, L_y are assumed constant. Knowing the value of s and being $q_{o2} = 0$ close to zero, $\overline{IJ}$ will always be parallel to $\overline{ML}$ for every value of q_{o1} (Figure 20a).

For every system pose, the $x - y$ coordinates of I and J can be computed by only knowing the sensory readings (l_x, q_B) and the lengths of both finger and exoskeleton links. Therefore, $\forall(q^i_{o1}, q^i_{o2})$ identifies the straight line parallel to $\overline{IJ}$ and passing through the finger joint L as function of x, y and the offset s in the $x - y$ reference frame centered in K with the x-axis parallel to the metacarpal direction.

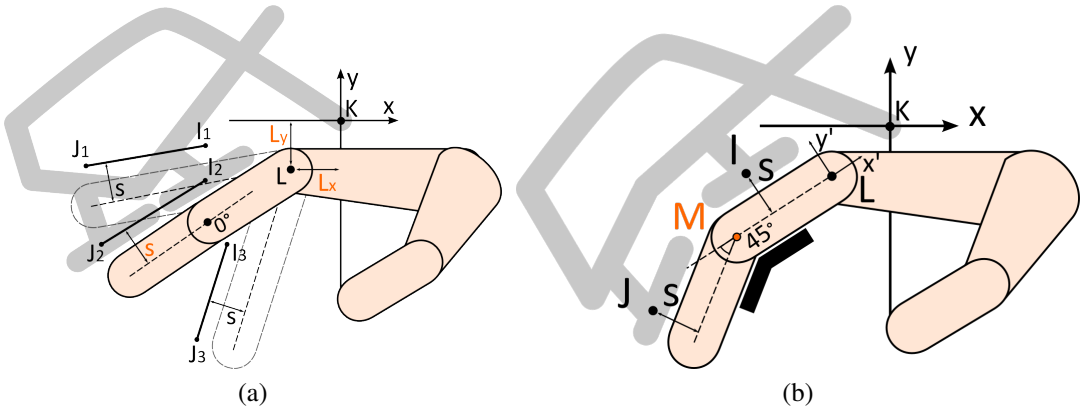


Figure 20: The calibration notations during (a) the first and (b) the second phase.

$$\begin{bmatrix} 1 & -m_i & \frac{-1}{\cos(\tan^{-1}(m_i))} \end{bmatrix} \begin{bmatrix} L_y \\ L_x \\ s \end{bmatrix} - \frac{y_{I_i}(x_{I_i} - x_{J_i}) - x_{J_i}(y_{I_i} - y_{J_i})}{x_{I_i} - x_{J_i}} = 0 \tag{3}$$

where $m_i = \frac{y_{I_i} - y_{J_i}}{x_{I_i} - x_{J_i}}$. As shown in Figure 20a, three poses [(q^1_{o1},0 deg), (q^2_{o1},0 deg), (q^3_{o1},0 deg)] are sufficient to determine the coordinates of L in the $x - y$ plane and the offset s. It should be noted that it is not necessary to know the three q^i_{o1} joint coordinates to perform this identification step. Since the elasticity and some mechanical backlash in the attachments allow little relative movements between the exoskeleton and the finger during the flexion/extension motion, in order to find out the closest solution for L_x, L_y and s to all the possible solutions obtained along the entire opening and closure path, 30 different poses equally spaced on the entire stroke of joint M have been taken into account. The least squares method has been used to solve the linear optimization problem defined by 30 equations of the type of Eqn. 3, each one derived from a different pose. Please note that L_x, L_y and s are sufficient to univocally determine q_{o1} independently on q_{o2} value.

The second estimation procedure step allows defining the phalange length $\overline{LM}$. As depicted in Figure 20b, knowing q_{o1} and the coordinates of L, the straight line passing through $\overline{LM}$ can be defined. Since, as previously mentioned, J and s are also known, both M and $\overline{LM}$ can be computed based on q_{o2}. In the experimental procedure q_{o2} has been fixed to $45°$ by locking the user's finger joint with a 3D-printed piece properly shaped. Eqn. 4 shows the mathematical expression for $\overline{LM}$ in the reference frame $L_{x'y'}$ (figure 20b)

$$\overline{LM} = \left| J_{x'} - \frac{J_{y'}}{\tan q_{o2}} - \frac{s}{\cos q_{o2}} \right|, \tag{4}$$

where (J'_x, J'_y) are the coordinates of J expressed in $L_{x'y'}$. In order to increase the algorithm robustness in the finger workspace, the $\overline{LM}$ value has been evaluated as the mean among 30 $\overline{LM^i}$ values obtained for different q^i_{o1} values equally spaced in the stroke of joint L.

In order to experimentally evaluate the estimation of hand parameters through the presented algorithm, an optical tracking system (Optitrack V120 Trio) has been used to measure phalange position and orientation. The precision of the tracking system, which depends on the distance of markers from the infrared cameras, has been experimentally evaluated to be below $0.1mm$ (estimated as repeatability of distance between two markers fixed to a solid body, which is moved and rotated across the workspace). Two markers were fixed to the exoskeleton base (one corresponding to the K point in Figure 20, two to the link fixed to the first phalanx, and two to the link fixed with the second phalanx. Through positions of these markers, the joint angles q_{o1} and q_{o2} could be computed. Three healthy volunteers participated in the experimental session (male, aged 25-31 years old) by providing written informed consent to participate. Length of the middle finger proximal phalange of the three subjects measured respectively $53mm$, $54mm$ and $66mm$. Each experimental session consisted in wearing the exoskeleton, calibrating the algorithm presented above through two opening-closing movements, and finally in accomplishing a free hand closing and opening for validation.

The two opening-closing movements for system calibration were executed fixing q_{o2} (to $0°$ and $45°$ respectively) by means of plastic supports. During such movements, sensor measurements were recorded and provided to the estimation algorithm offline. 3 presents the results.

During the free hand validation phase, the finger trajectory was computed by means of both the optical tracking system and the direct kinematic (Figure 21). The absolute error of q_{o1} is

Table 3: Measured ($_m$) and estimated ($_e$) parameters of the subjects. All data is stated in *mm*.

Subject	$\overline{LM_m}$	$\overline{LM_e}$	s_m	s_e	L_x	L_y
1	56 ± 1	54.98	8 ± 1	10.81	1.96	22.50
2	53 ± 1	51.77	9 ± 1	9.22	9.52	24.61
3	54 ± 1	49.98	8 ± 1	8.50	0.19	23.76

comparable with the angular backlash between the phalange and the exoskeleton. The error on q_{o2} estimation results higher, because the second phases of the algorithm not only propagates the error of the first phase but suffers the difficulty of keeping the angle between the proximal and median phalanges fixed due to the human skin softness.

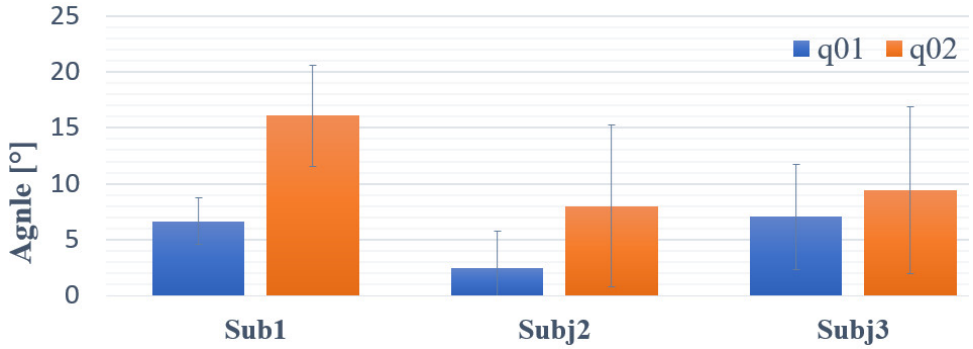


Figure 21: Bar plot representing errors along the test trajectory.

Results showed good estimation results, although the not perfectly rigid fixing of the phalanges to the exoskeleton introduces errors in the pose estimation (mean errors: $e_{q_{o1}} = 5.40°$ and $e_{q_{o2}} = 11.18°$). In addition, the softness of human tissues and complexity of articulations makes it very difficult to build an accurate model in which position and orientation of the axes of any human joint are assumed to be fixed [3]. On this regard, it should be noted that the adopted model considers a rigid link between the finger and the exoskeleton. Due to the fact that the exoskeleton applies only normal forces, the hand exoskeleton can be fixed on the user's hand efficiently using fastened soft belts and the displacement between the exoskeleton sliders and finger phalanges might be neglected.

3.2.2 Hand Teleoperation and Force Feedback

A teleoperation scenario has been developed using a local admittance control of the hand exoskeleton and the local impedance control of the Schunk hand as summarized in Figure 22.

During the experiment, the hand exoskeleton is used as a master device, while the estimated finger pose was sent to the Schunk hand as a position reference. A proportional-derivative position controller computes the required actuator torques based on the received reference. Then, the torques applied by the Schunk hand were sent back to the hand exoskeleton as a force reference in the local admittance control. Hence, the operator can wear the hand exoskeleton and move freely to control the position of fingers of the Schunk hand using the local transparency of the exoskeleton and the position control of the Schunk hand. Once the Schunk hand gets in contact with an object, the interaction forces between the object and the Schunk hand could be increased if the operator keeps closing his fingers further. This behavior can be explained with a virtual spring, due to the proportional component of the position control loop at the robotic

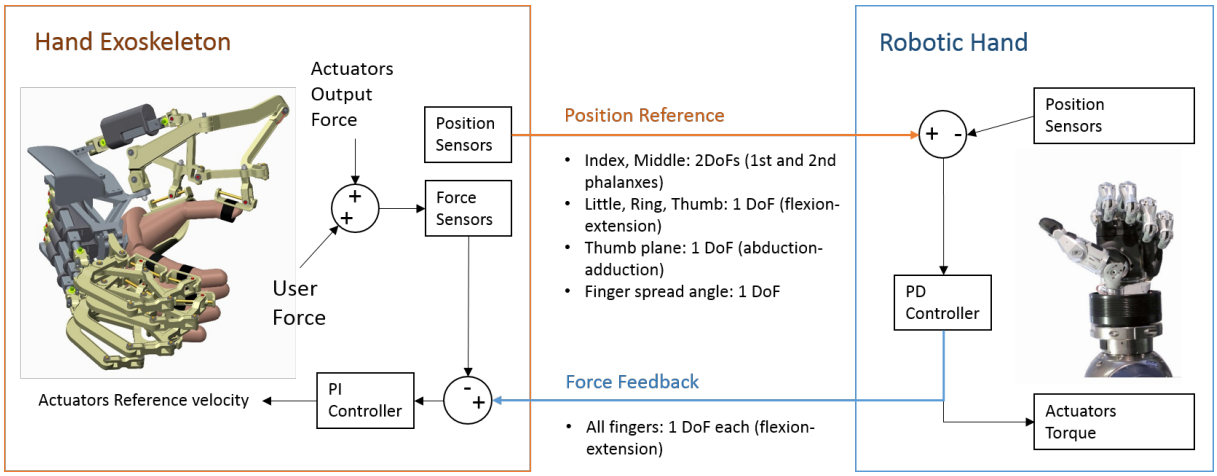
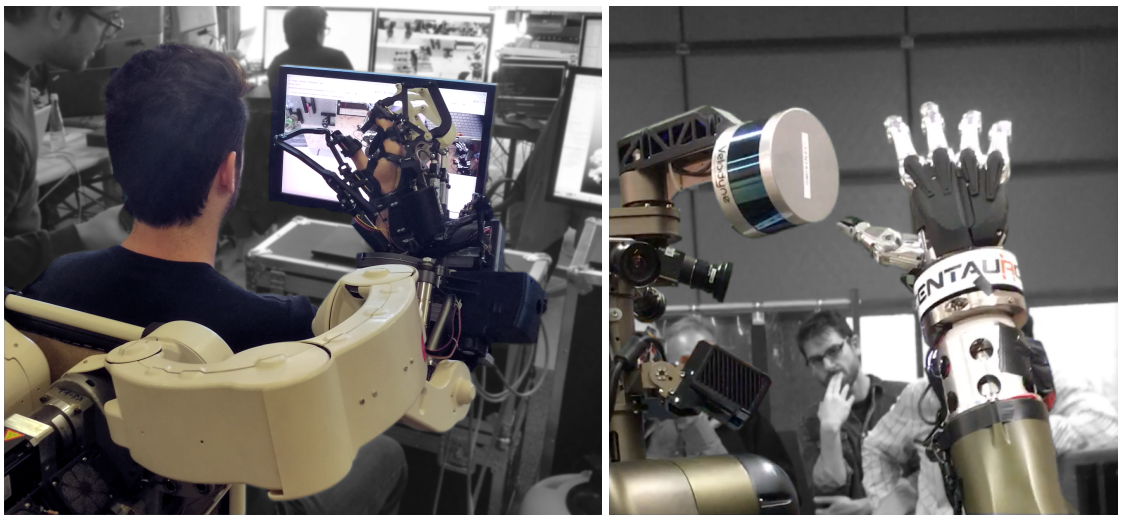


Figure 22: Teleoperation scheme between the Hand Exoskeleton and the Schunk robotic hand

hand. Meanwhile, these interaction forces are also perceived by the operator by adding them into the transparency control as a force reference. The operator controlling the Centauro robotic hand through the Telepresence Station is shown in Figure 23.



(a) (b)

Figure 23: Teleoperation between the hand exoskeleton of the Telepresence Station and the robotic hand of the Centauro Robot

The DoFs of the hand exoskeleton and the Schunk hand are slightly different: the thumb, little and ring fingers of the Schunk hand and the exoskeleton are operated as single DoF, considering only the finger opening/closing. The hand exoskeleton can measure the rotation of the MCP and PIP joints of the index and middle fingers for a better grasping, while the Schunk hand was position controlled with $2-DoFs$ correspondingly. However, the force feedback was given on $1-DoF$, since the under-actuated hand exoskeleton is controlled by a single actuator. The hand exoskeleton has an additional $2-DoFs$ measured mobility without force feedback to control spread between fingers and to rotate the plane for the thumb opening/closing.

The overall control structure for this teleoperation scenario, depicted in Figure 24, can be summarized as:

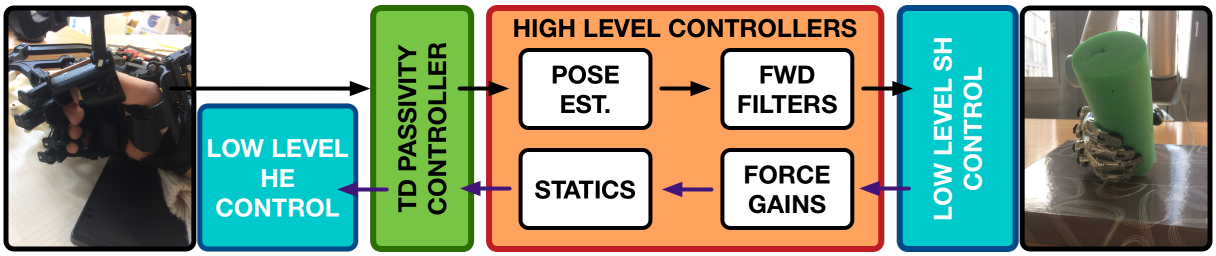


Figure 24: General view of the teleoperation control scheme.

1. **Low level exoskeleton control** provides active backdrivability for the hand exoskeleton controlling the force with high transparency.

2. **Low level Schunk Hand control** comprehends an inner PID current loop (at $100\ kHz$) and an external PD position loop (at $1\ kHz$) for the control of the Schunk hand. The PD gains are maximized for the capability of modulating interaction forces.

3. **High level teleoperation control** comprehends the pose estimation of the operator's hand, the resolution of the statics despite of the under-actuation of the exoskeleton, forward position filters and force reduction gains. An empiric approach is proposed to tune all force reduction gains.

The details of the master and slave structures will be detailed below. Then, the details of the overall teleoperation system will be discussed and the experimental results of the additional experimental activities regarding the teleoperation scenario and quality of force feedback given to the operator will be described.

Low Level Exoskeleton Control The low level exoskeleton control algorithm controls the grasping forces acting on the 5 fingers of the operator's hand. Since the hand exoskeleton can control each finger component independently, the control problem can be divided into 5 different problems to be considered similarly.

Figure 25 details the low level control structure of the master hand exoskeleton. F_{ref} is defined as the reference input signal for the linear actuator. F_{err} is the error between the reference and the actual force F_{sens}, which is measured by the force sensor. The error F_{err} is passed to the controller ($C(s)$) that generates the corresponding PWM input for the linear actuator. The physical loop focuses on the activity measured by the actuator encoder (*act*) and the revolute potentiometer (*b*). The mechanical structure has been analyzed through direct and inverse Jacobian matrices due to the human impedance $Z_h^* = J^T Z_h J$ depending on the exoskeleton pose.

The *physical loops* system shown in Figure 25 is simplified by considering it as a *black box*. For this reason, a calibration test was required such that:

- The user was asked to wear a single finger component of the exoskeleton while he was asked to keep his finger *completely extended* (Figure 26). A step function of $6\ N.$ was defined as a reference force input (F_{ref}) while the value of the proportional controller $C(s) = k_p$ was increased until the system reaches instability for the maximum control value (k_p^{max}), and

- The same experiment was repeated for different finger configurations (e.g., with the metacarpal and the interphalanx joints at about 45 degrees).

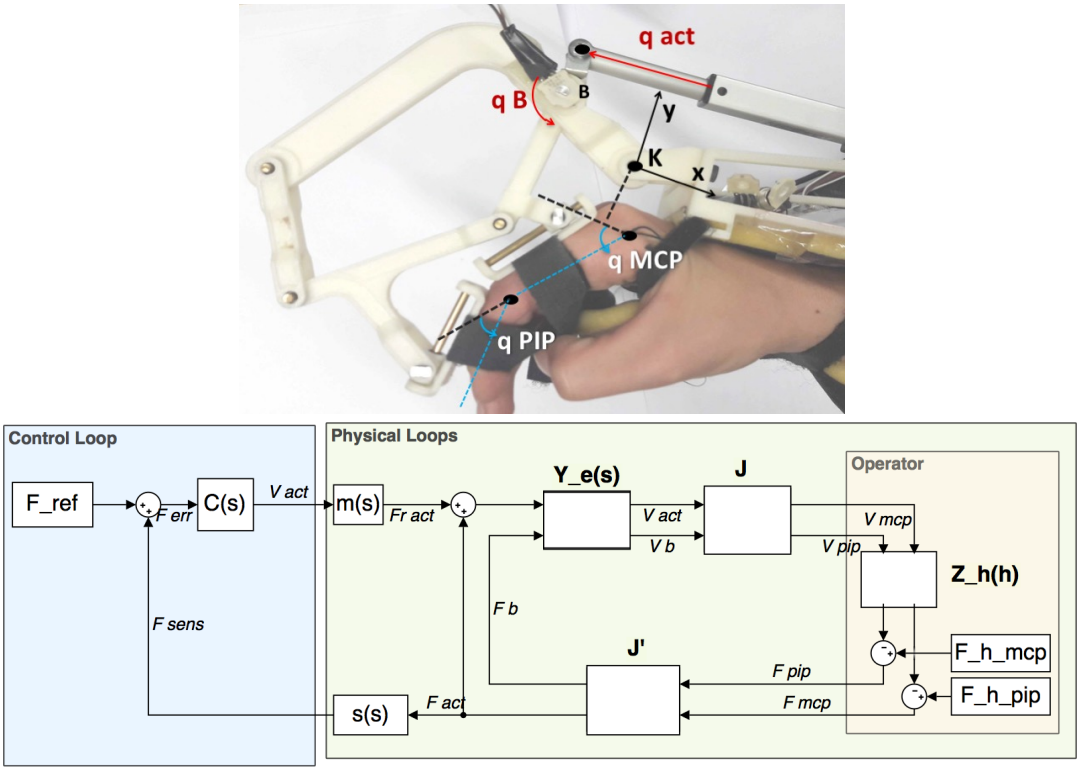


Figure 25: System composed of the human finger and the hand exoskeleton (top) and the relative system diagram (bottom).

Figure 26: Two different configurations used during ZieglerNichols PI calibration: the hand is *completely open* (left), the hand is *completely closed* (right).

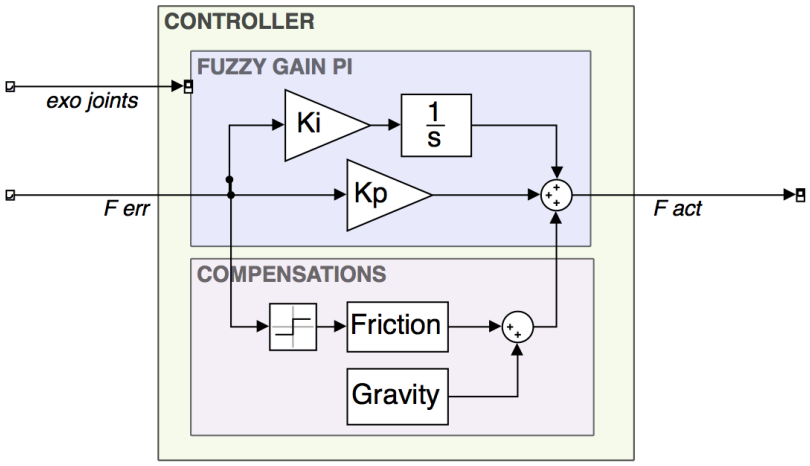


Figure 27: Finger exoskeleton explicit force control structure. The motor torque is the result of two contribution: the *friction compensation* that depends on the direction of the reference force; and the *PI* contribution. k_p and k_i values are fuzzy-tuned depending on actual *exoskeleton joints*.

The findings of the calibration test show that k_p^{max} has a trend to decrease as the joint angles increase. Such a trend can be explained in two ways:

- The transmitted impedance mathematically depends on the transmission Jacobian and
- the human finger might have different levels of *rigidity* in different configurations.

In this scenario the system has different gain margin values in different finger poses. That is why it is inadequate to develop a *static* controller that stabilizes the system in all possible configurations. Instead, a *fuzzy gain scheduling* PI controller was developed in which the control parameters depend on the finger configurations (see Figure 27). These dependencies might be explained further based on the following considerations:

- The instability arises from the finger's *level of closure* and the concept of *closure* strongly depends on motor displacements. The discrimination between *finger poses* is performed by comparing different actuator displacement.
- It is performed the ZieglerNichols method in two different *level of closure* (Figure 26 and Figure 28).
- The integral term is governed by an anti-windup mechanism that blocks the integration whenever the control variable saturates the PWM limits.
- Under the assumption of a black-box dynamic, non-linear terms are treated as follows:
  - The gravity is considered as constant and compensated with a fixed (pre-measured) term. This approach shows good results for all the fingers except the thumb, because the exoskeleton is very light-weight. Obviously, the fixed gravity compensation should vary at least according to the exoskeleton orientation. This improvement was not performed in the *first and second prototype* version because they required information coming from the upper limb exoskeleton. Experiments show that the lack of the complex gravity compensation does not jeopardize the system. Regarding

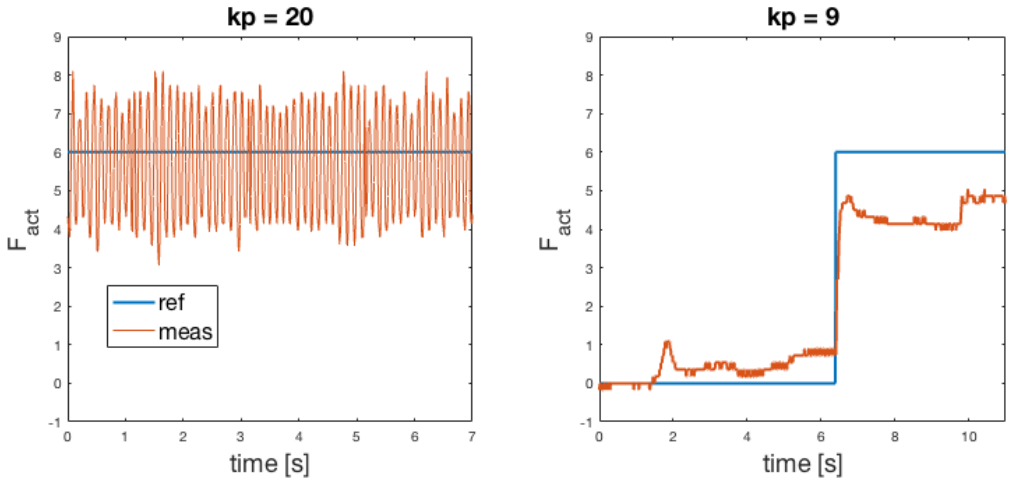


Figure 28: The index finger behavior in *completely open* configuration with a proportional controller with $k_p = 20$ (system marginally stable) and with $k_p = 9$ (value found using ZN method).

the thumb, a high level of under-actuation does not allow the gravity compensation to be sufficiently large, even when the hand exoskeleton is blocked in a stationary orientation. Therefore, no additional experiments were performed to improve the efficiency of gravitational effects.

- The inertial, centrifugal and Coriolis terms of the dynamic system are neglected because of the high motor reduction and the negligible exoskeleton mass.
- The actuator frictions are compensated by an additional fixed term that depends on the sign of the control variable. Since this approach causes chattering effects, a more efficient approach would be to module the friction compensation by means of an S-function in proximity of the discontinuity.

Low Level Robotic Hand Control Schunk Hand controllers are implemented directly on the device hardware driver. Control parameters are set during the calibration phase and cannot be changed online. Each motor is surrounded by a $100\ kHz$ current loop (PID controller) run by an external PID position loop at $1\ kHz$ (see Figure 29). Besides, a *thermal controller* ensures the heat dissipation limiting the motor currents. Looking at the SH step response using default parameters, it emerges that

- the device working point is in *currents saturation* (the position loop is very quick) and
- when the finger *touches* something, the motor provides the maximum available torque.

The last behavior is due to the presence of an integration component in the position control loop that cannot modulate forces. Force modulation is a necessary condition to perform safe grasping (see Figure 30). To achieve this goal, it is necessary to set the k_i value to zero. Doing so, the motor and its reference behave like a dumper-spring system, where the motor torque linearly depends on the error between the reference and the reached position. The K_d value focuses on the system bandwidth, filtering the control signal.

In this scenario, the following design choices are taken for each motor:

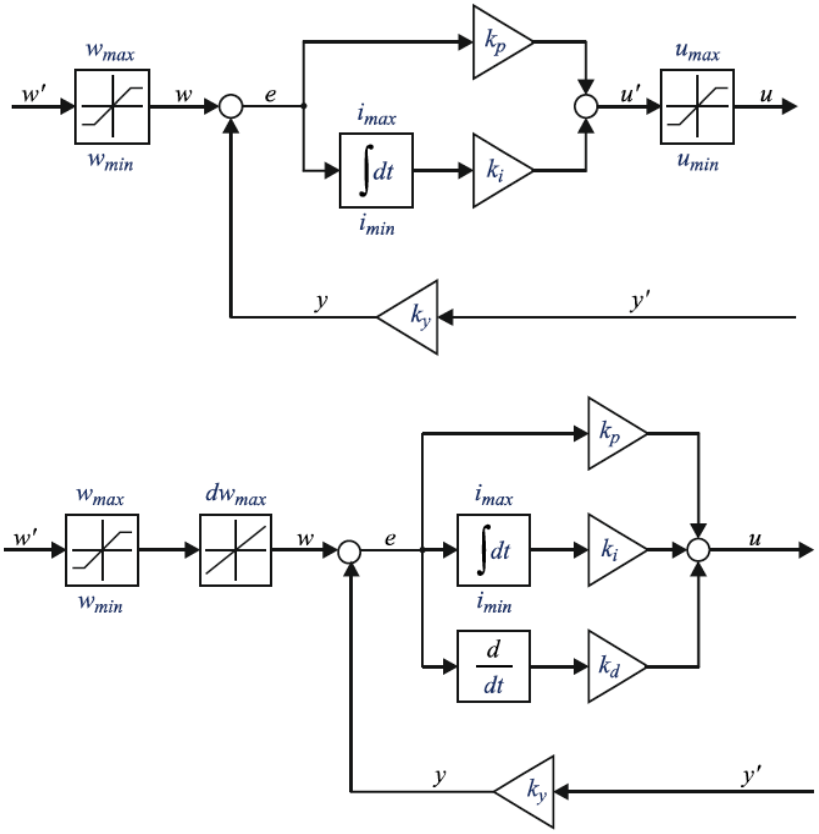


Figure 29: Schunk Hand inner current loop (top) and outer position loop (bottom), where the thermal model is ignored.

Figure 30: Schunk Hand is teleoperated to grasp very fragile object: without force modulation it would have been impossible to grasp biscuits without breaking them.

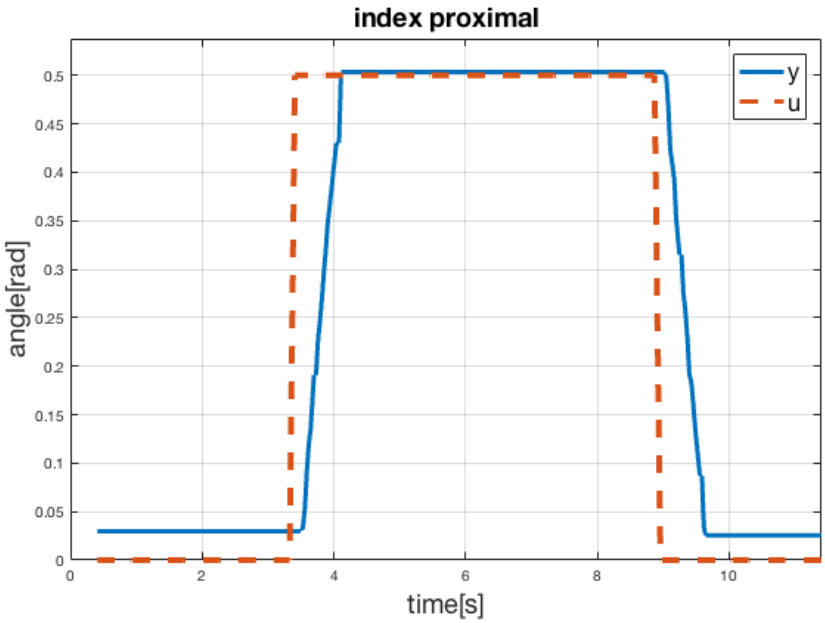


Figure 31: Response of the Schunk hand index proximal joint to a reference step of 0.5 radiants. It can be noticed that $1\ s$ is necessary to reach the final value. In addition, the absence of an integral component causes a nonzero steady state error.

- default *current loop* parameters are kept constant,
- the position loop k_i parameter is set to zero,
- the position loop k_p parameter is scaled such that a position error of about 20^o is necessary to saturate forces even though the Schunk Hand cannot provide its maximum force when the motor position is close to its limits,
- the position loop k_v parameter is scaled such that the proportion with the k_p remains the same of the default parameters and
- the spread position loop remains unaltered.

The final Schunk hand position control gains are presented in Table 4. The main drawback of the system bandwidth reduction is that eventually the SH hand moves *very slow* (see Figure 31), even though it is a necessary condition to allow force modularity.

How to measure Schunk Hand interaction forces A challenging problem is the estimation of Schunk hand interaction forces. Unfortunately, the Schunk hand is not equipped with force sensors: the only available data is the current coming from the DC motors. The estimation presents the following issues:

- Some of the Schunk Hand motors actuates more than one joint,
- Schunk Hand joints are not back-drivable, and
- the Schunk Hand dynamic model is not available, and the author has not found a procedure to identify it.

DoF name	$\mathbf{k_p}$	$\mathbf{k_v}$	$\mathbf{k_i}$
thumb flexion	0.08	160	0
thumb opposition	0.4	800	0
index finger distal	0.4	600	0
index finger proximal	0.05	200	0
middle finger distal	0.4	600	0
middle finger proximal	0.05	200	0
ring finger	0.05	200	0
pinky	0.05	300	0
finger spread	1	4000	0.36

Table 4: Position loop gains for each DoFs.

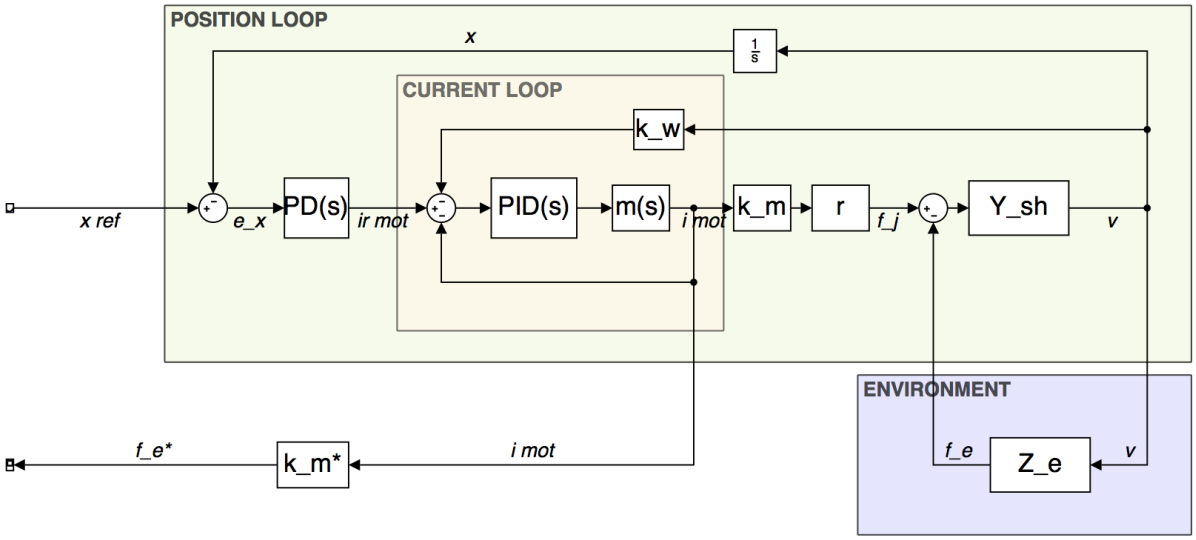


Figure 32: Control diagram of a Schunk Hand joint. m_s, k_m, r, Y_{sh}, Z_e and f_j are respectively the motor electric model, the motor constant torque, the transmission reduction factor, the Schunk Hand admittance the environment admittance and the joint torque. It can be noticed that f_e^* is an estimation of f_j but when $Z_e >> 1/Y_{sh}$ $f_e^* \rightarrow f_e$.

Non-backdrivability means that actuators are aware only of Schunk hand forces which come from Schunk hand active motions. Forces that come from the activity of the environment are not observable. The absence of a Schunk hand model requires a procedure to extract torques from currents. It should be underlined that the currents used to estimate Schunk hand joint torques does not only depend on the interaction with the environment, but they comprehends terms coming from the Schunk hand admittance (Figure 32). However, experimental results in the previous paragraph show that inertial terms can be neglected. In this scenario, if we suppose the current loops sufficiently quick, and the motor currents are considered dependent only on the environment impedance.

Finally, the motor constant k_m^* that comprehend the transition scale factor is estimated as follows: given, the maximum joint torque and the peak current for each motor (Schunk hand

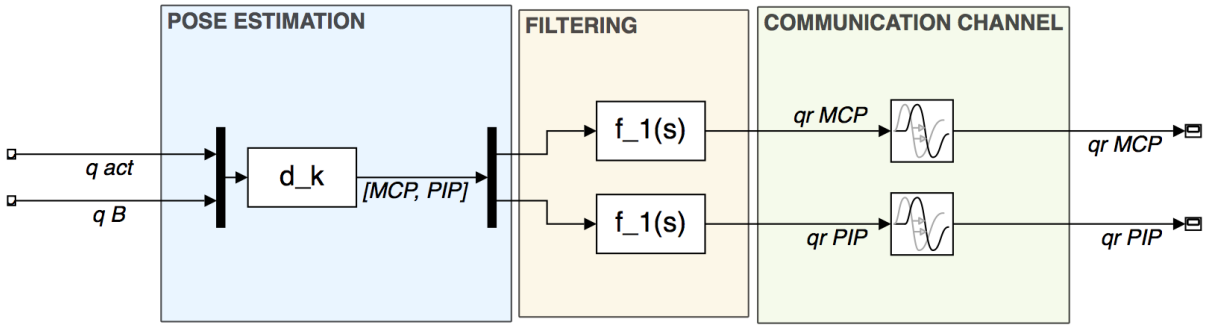


Figure 33: The forward controller of the *index* finger. While the *middle* finger has the same structure, in case of *ring* and *pinky* only the MCP command is sent. *Thumb* and *spread* forward controllers have analogous schemes.

documentation), k_m^* is approximated to

$$k_m^* = \frac{\tau_{max}}{i_{max}} \approx k_m r. \tag{5}$$

This consideration is precise only for motors that actuate a single joint (as in the case of index and middle proximal joints).

High-Level Teleoperation Controllers In this context, *High-Level Teleoperation Controllers* indicate the whole set of logic between the Handexos and the Schunk hand motors. The entire logic can be split into two parts:

- The **Forward Controller** computes the finger pose from Handexos sensors and generates Schunk Hand position commands.
- The **Backward Controller** estimates Schunk Hand torques from motor currents and, for each finger, solves the Handexos under-actuation problem.

The forward controller has a modularized structure. For each finger it estimates the finger pose computing the direct kinematics from the sensor measurements (see Figure 33). Regarding the thumb, all five pose angles are computed while for all other fingers both MCP and PIP angles are computed. Then, Schunk hand commands are generated according with the kinematic mapping, filtered and sent to the low-level Schunk hand control.

Regarding the backward controller, force feedback is rendered only on the operator index, middle, ring, and pinky degrees of freedom. Regarding the thumb, due to the empirical mapping and the very high grade of under-actuation, it is challenging to find a render procedure without introducing an unpredictable behavior. In this scenario, the backward controller is modularized in four different controllers, one for each finger. Once the Schunk hand interaction torques are estimated, the statics is solved through the null space optimization technique. Finally the motor force references are sent to the low-level exoskeleton controllers.

Transparency Validation The operator should feel forces which are as close as possible to the ones experienced by the Schunk hand during the interaction with the environment. The partial non-backdrivability of the manipulator does not allow to measure interaction forces accurately, thus the teleoperation fidelity relies on the Schunk hand currents that are affected by the inertia of the manipulator.

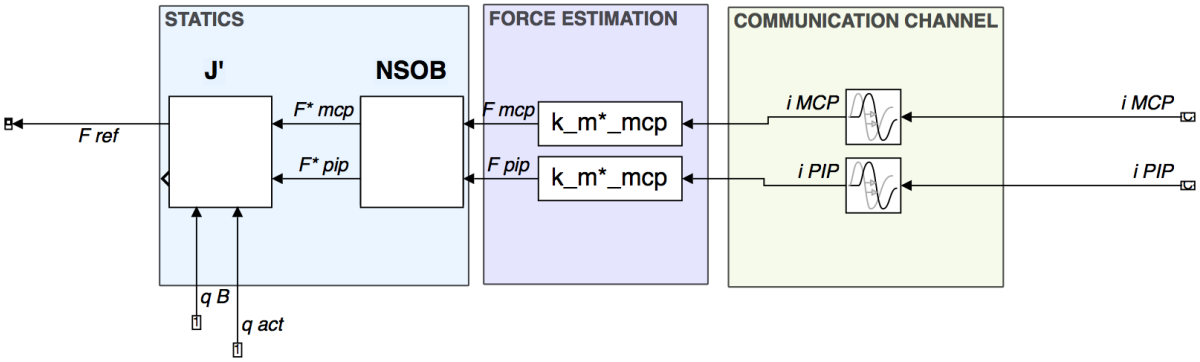


Figure 34: Index finger backward controller. The index and middle schemes are equal, the ring and pinky differs because only a current for both DoFs is available. In this case the torque is considered distributed in both joints according with the maximum available torque.

In addition, it is challenging to physically measure the torques that are rendered by the Handexos. However, their accuracy can be evaluated *a priori* assuming that:

- the exoskeleton differential kinematic is known within a neglectable level of uncertainty because of the links production method. Besides, the precision of the pose estimation is of about 5%.
- All sensors provide reliable measurements. All sensors are periodically calibrated and, in particular:
  - each motor potentiometer behaves linearly according to the product specifications (less than 2%),
  - each revolute potentiometer is periodically calibrated and it is linearity is verified before the installation,
  - each load cell is assumed to behave linearly and its characteristic curve is offline calibrated, and
  - the explicit force control does not guarantee zero steady-state error because the integrator was removed to preserve passivity. No improvements can be performed in this direction.

Another factor that can potentially jeopardize teleoperation transparency is the presence of the *null space optimization block* (NSOB). As previously explained, for each finger the block extracts the pair of Handexos feasible torques that are closer to the torques measured by the manipulator (and scaled by the *force reflection gain* k_f). This block distorts the operator forces, in contrast with the transparency requirement. However, the following paragraph shows the optimality of this solution. The *torque drift* is qualitatively evaluated through real data acquisitions.

Evaluation of NSOB effects For each finger, the *Null Space Optimization Block* is responsible to extract feasible torques from Schunk hand interaction torque. In fact, it is not possible to render torques on MCP and PIP joints independently, but the under-actuation provides an implicit relationship between them. Among the ∞^1 feasible pair of torques, NSOB selects the *closest* to the Schunk hand observed pair.

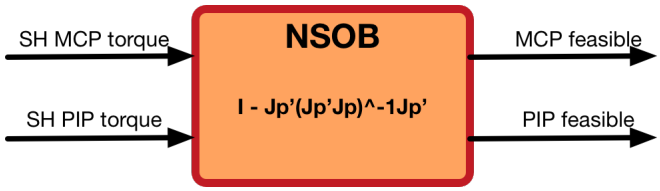


Figure 35: NSOB inputs (SH scaled torques) and outputs (feasible torques). Note that SH torques are scaled by the force reduction factor before being elaborated by NSOB.

Param name	**Big Hand [mm]**	**Small Hand [mm]**
phalanx lenght	54.0	44.0
phalanx width	10.5	8.5
offset x	0.0	0
offset y	20.0	20.0

Table 5: The table shows the operator-dependent parameters of the *Big Hand* and the *Small Hand* operators.

It is challenging to evaluate the effects of the NSOB analytically. The main reason is that the NSOB depend on the Jacobian. In addition, the Jacobian does not only depend on the finger pose, but it is strongly related to some operator parameters. For this reason, it was decided to evaluate the influence of NSOB qualitatively. The operator is asked to perform grasping with different levels of *grasping power* and then it is compared the drift between *feasible* and Schunk hand torque pair. For the sake of simplicity, only the index finger torques are compared. Two operators with very different finger sizes performed the test.

Looking at the result plots in Figure 36, the following considerations are obtained:

- It is not possible to modulate the force on the PIP joint: a difference between the operator and the Schunk hand angle fewer than four degrees is sufficient to saturate the Schunk hand motor current.
- There is a factor of 10 between maximum PIP and MCP Schunk Hand torques.
- The Schunk Hand PIP always works *in saturation*. In this scenario, the shape of *feasible* PIP plot depends only on the Schunk hand MCP. In particular, the amplitude of *feasible* PIP is always (at least two times) less than the Schunk hand MCP.

Summarizing, for what regards MCP torques, the *feasible* torque is always smaller than the Schunk hand (scaled) one. However, they have precisely the same shape. On the other hand, the *feasible* PIP torque plot trend does not depend on the Schunk hand (scaled) torque. Therefore, it is validated the design choice to neglect the possibility to modulate PIP forces.

Finally, the null space optimization block is compared with an easier block that solves the statics and neglects the torque component related to the passive revolute joint B. Figure 37 shows that the second approach is without any doubt worse (please note the presence of peaks). For this reason NSOB guarantees the *as good as possible* fidelity to Schunk hand torques.

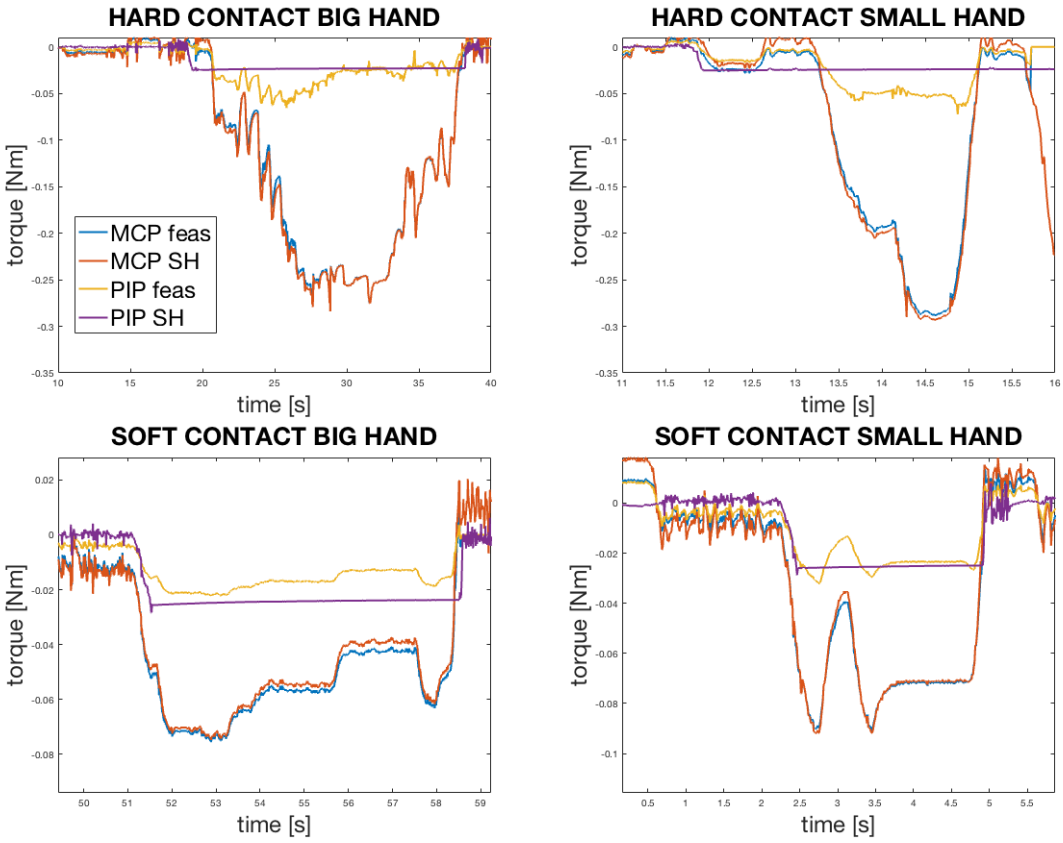


Figure 36: The plots compare the Schunk hand and the *feasible* torques pair in case of a *Big Hand* (left) and a *Small Hand* operator (right). On the top, it is shown the behavior in case of *hard* contact, while in the bottom the interaction forces are shallow.

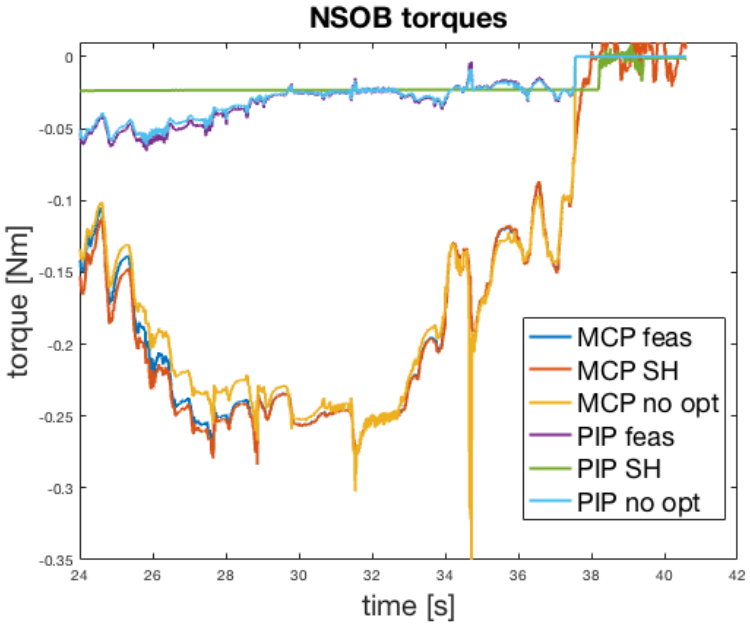


Figure 37: The graph shows the comparison between Schunk hand torques, *feasible* torques and the torques obtained without performing the optimization (*no opt*).

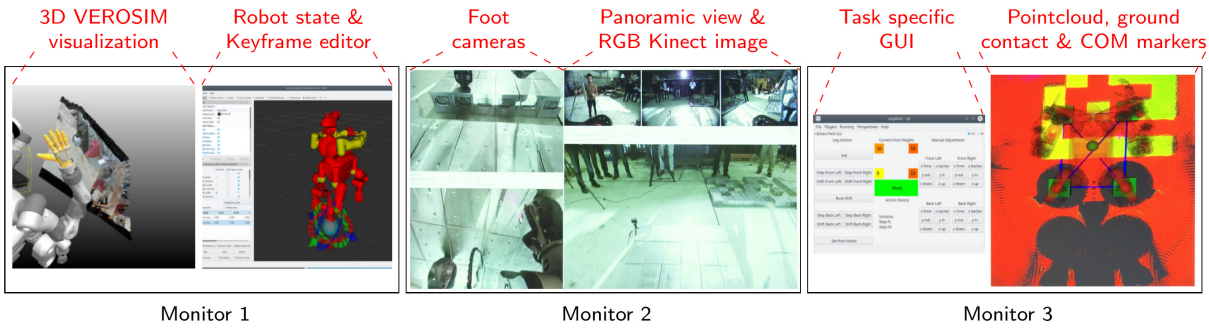


Figure 38: Environment and robot state visualization for the support operators.

4 Additional Operator Interfaces

Besides the main operator sitting in the telepresence suit, there are additional operators who

- can control the robot via control interfaces other than the telepresence suit,
- provide the required visualizations for the operator in the telepresence suit, and
- are responsible for keeping an overview over the whole situation while the operator in the telepresence suit might focus on task details.

Hence, those operators need a wide range of visualizations and control interfaces for both locomotion and manipulation control.

4.1 Visualizations

We provide visualizations by displaying processed and unprocessed data from several sensors on multiple monitors, as shown in Fig. 38. A panoramic view from the robot head perspective is helpful for general scene understanding. In addition, images from the two RGB cameras under the robot base are arranged to give a detailed assessment for the terrain under the robot base which was key to a save stepping locomotion operation (Fig. 38).

Moreover, ground contact for each foot is visualized by correspondent markers. Together with the visualized robot center of mass (CoM), this helped assessing robot stability. 3D *VEROSIM* visualizations further increased the scene understanding, especially in manipulation tasks where occluded areas are compensated by the simulation-based approach. Several control GUIs were developed for task-specific robot control.

4.2 Locomotion Control

The Centauro lower body provides a wide range of locomotion capabilities which require suitable interfaces to be controlled efficiently. Intuitive interfaces for omnidirectional driving control are a joystick and a pedal controller (Section 4.2.1). Leg movements can be controlled by a keyframe editor (Section 4.2.2). A higher degree of autonomy is reached by utilizing a semi-autonomous stepping controller (Section 4.2.3). Finally, we developed a hybrid driving-stepping locomotion planner which autonomously plans and executes locomotion to a goal pose specified by an operator using *VEROSIM*, as described in Deliverables D5.1 - D5.4 [9, 4, 11, 10].

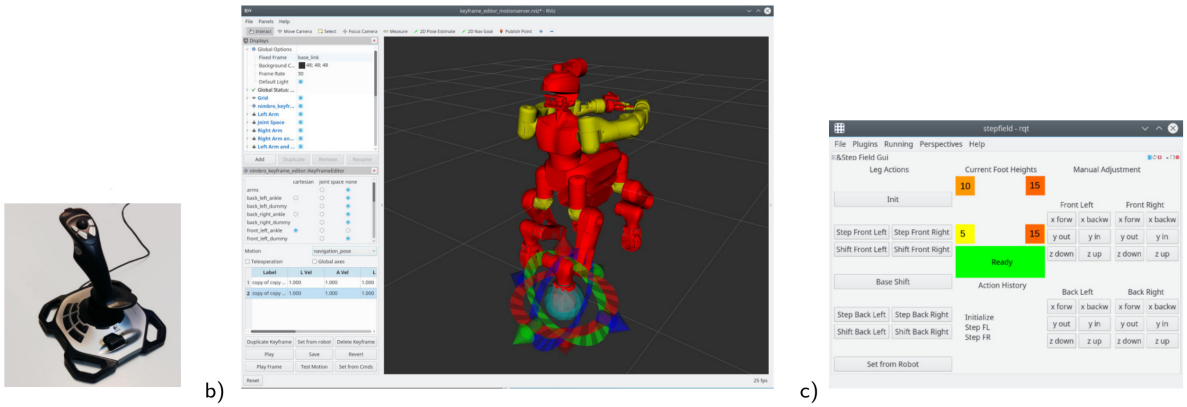


Figure 39: Locomotion control interfaces: a) 4D joystick, b) keyframe editor, c) semi-autonomous stepping controller GUI.

4.2.1 4D Joystick

Omnidirectional driving can be controlled by a joystick with four axis (see Fig. 39a). Robot base velocity components v_x, v_y and v_θ are mapped to the three corresponding joystick axis. Foot specific velocities and orientations are derived from this robot base velocity. The joystick throttle controller jointly scales all three velocity components.

4.2.2 Keyframe Editor

Leg motions can be controlled by a keyframe editor [22]. It allows for the control of joint groups (e.g. the front left leg) in joint space or Cartesian end-effector space live during the mission or through predefined keyframes. Keyframes can also be sequenced to motions. The GUI provides a graphical interface in which joints and end-effectors can be configured using interactive markers and the computer mouse (Fig. 39b). A numerical configuration is also possible. Finally, robot motions are generated by interpolating between given keyframes.

4.2.3 Semi-autonomous Stepping Controller

To efficiently control stepping locomotion in irregular terrain, we developed a semi-autonomous controller. It provides a set of stepping and stepping-related motions that can be triggered by the operator. The available motions are: step with a chosen foot, drive a chosen foot forward, and shift the robot base forward. If a stepping motion is triggered, the robot shifts its base longitudinally and laterally and rolls around its longitudinal axis to establish a stable stepping pose. The stepping foot is then lifted, extended by a given length and lowered. The latter stops as soon as ground contact is detected. Hence, the robot automatically adapts to the ground structure. Motions are represented as sequences of keyframes, as described for Section 4.2.2.

The controller is operated through an intuitive GUI which provides buttons to trigger the described motions for an individual foot (Fig. 39 c). Additional buttons allow the operator manual foot movement in Cartesian space, which is helpful to apply minor corrections. Furthermore, the GUI visualizes detected terrain heights under the individual feet and a history of the last triggered motions which helps to follow certain motion orders.

Figure 40: 6D input device (l.) and corresponding GUI (r.) for dexterous wrist control.

4.2.4 Motion Execution

Motions from any source need to be transformed to joint space trajectories to be executable by the robot. We use a keyframe interpolation method which was originally developed for Momaro [23]. The interpolation system generates smooth joint trajectories obeying velocity and acceleration constraints set per keyframe. Input are keyframes consisting of joint space or 6D Euclidean space poses for each of the robot limbs.

4.3 Manipulation Control

The Centauro system possesses several degrees of autonomy to control manipulation. The upper body can be controlled in joint or Cartesian space or by executing keyframe motions with the keyframe editor which is described in Section 4.2.2. Moreover, precise control of the wrist pose is provided by a 6D input device (Section 4.3.1). Finally, we propose an autonomous grasping functionality which is described in [18, 17, 13, 14].

4.3.1 6D Input Device for Wrist Control

A *3DConnexion SpacePilot Pro* 6D input device and a corresponding GUI provide a teleoperation interface for dexterous wrist control (see Fig. 40). The developed interface establishes the connection between the device and the motion player mentioned in Section 4.2.4. The *SpacePilot* movement is streamed as a desired 6D end-effector pose to the motion player which interpolates from the current to the desired pose and executes the motion. The GUI and buttons on the input device can be used to easily adjust the following control parameters:

- **End-effector:** A wrist for arm control, both wrists for bimanual arm control, or an ankle for leg control.
- **Reference frame:** End-effector frame, robot base frame, or a custom frame.
- **Enable axes:** Each translational and rotational axis can be enabled/disabled so the user input on this axis is considered/ignored.
- **End-effector speed:** Scalar speed factor.

In case of bimanual arm control the streamed mouse movement is applied on bot wrists relative to the right arm end-effector control pose. This control pose might be in a custom frame, for example the mean of both end-effector poses for symmetrical arm control. This interface

is well suited for manipulation tasks where very precise arm movement along certain axes is required (e.g. moving the arm along a plane surface, or turning an object around a specified axis).

5 Conclusions

In this Deliverable we presented the final version of the operator control station developed for teleoperating the Centauro robot. It consists of a telepresence suit for the first person operator and additional control and visualization interfaces for the supporting operators.

Regarding the control station for the first person operator, an advanced full exoskeleton for the upper limbs has been developed. It allows the operator to control robot limb movements directly through his/her own arm movements, and to perceive force feedback of the remote environment, while navigation of the Centauro robot is controlled by pedals. Stereo visual and audio feedback is provided by a Head Mounted Display.

The upper limb exoskeleton of the control station has been completed with modules for the arm (shoulder and elbow), wrist, and hand. Improved control electronics, teleoperation and calibration algorithms have been tested in particular for the hand exoskeleton module and for stable teleoperation of the whole upper limb exoskeleton.

Manipulation of the first person operator has been improved through additional experimental activities regarding control techniques used in the teleoperation of the arm and the hand exoskeletons. The bilateral teleoperation of the arm with force feedback has been improved in stability through further development of the Time Domain Passivity Approach based controller and of the position-drift compensation to deal with communication delays. The passivity controller applied on the slave end-effector positions resulted in smoother and more accurate movements compared to disabled passivity control. Also, the master exoskeleton operator experienced less effort in performing tasks in case of active passivity control.

Regarding teleoperation of the hand through the under-actuated hand exoskeleton, calibration of the device, estimation of the hand pose, matching of the kinematics with the different kinematics of the robotic hand, and force feedback have been further investigated for improving the performance of the teleoperation and the quality of the sensory feedback provided to the operator. An algorithm for online calibration of the hand pose has been developed for overcoming variability of different hand dimensions and unavoidable different offsets when fixing the hand to the exoskeleton links. Operator's hand pose estimation by means of the exoskeleton sensor has been then mapped to the different DoFs of the robotic hand of the Centauro robot.

Implementation of force feedback at the finger grasping has been further investigated developing specific control algorithms both at the master (hand exoskeleton) and at the slave (robotic hand) sides, given the available sensors and the different actuation methods of the two devices.

In addition, we developed several visualization and control interfaces in third person perspective for the supporting operators. Those enable the user to keep an overview over the scene, including the robot state and the environment, and to trigger locomotion and manipulation actions on different levels of autonomy. Regarding visualization of the remote robot and environment, the third person operator is provided visualizations of processed and unprocessed data acquired from several sensors and displayed on multiple monitors, helping the operator to better understand the remote scene.

Several control GUIs were developed for task-specific robot control from a third person perspective. Regarding locomotion, operator interfaces for efficient control of the wide range of locomotion capabilities provided by the Centauro robot have been developed. Besides joystick and pedal interfaces for intuitive omnidirectional driving control, interfaces for more advanced leg movements and stepping control have been implemented: a semi-autonomous stepping controller achieving a higher level of autonomy, and a hybrid driving-stepping locomotion planner which autonomously plans and executes locomotion to a goal pose specified by an operator using VEROSIM. Manipulation control for the third person operator has been developed by

means of keyframe motions in the Cartesian space and precise pose control through a 6D input device. Also, an autonomous grasping functionality has been proposed.

Overall, the final version of the control station shows improvements in control functionalities provided to operators in different modules, regarding locomotion, visualization, and manipulation features for both the first person operator and the third person support operators. Such improvements are motivated by results and experiences obtained in the past Centauro Evaluation Camp and in other additional experimental activities. The improvements have been partially tested and showed promising results. The final version of the control station will be validated in the next and last Centauro Evaluation Camp.

References

[1] Jordi Artigas, Jee-Hwan Ryu, Carsten Preusche, and Gerd Hirzinger. Network representation and passivity of delayed teleoperation systems. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 177–183. IEEE, 2011.

[2] Domenico Buongiorno, Edoardo Sotgiu, Daniele Leonardis, Simone Marcheschi, Massimiliano Solazzi, and Antonio Frisoli. WRES: a novel 3DoF WRist ExoSkeleton with tendon-driven differential transmission for neuro-rehabilitation and teleoperation. *IEEE Robotics and Automation Letters*, 2018.

[3] M. Cempini, S. M. M. De Rossi, T. Lenzi, M. Cortese, F. Giovacchini, N. Vitiello, and M.C. Carrozza. Kinematics and design of a portable and wearable exoskeleton for hand rehabilitation. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pages 1–6, 2013.

[4] X. Chen, F. Schilling, T. Klamt, and P. Jensfelt. Deliverable D5.2 CENTAURO Terrain Classification.

[5] Torben Cichon. Deliverable D4.4 switching between direct control and prediction mode.

[6] Antonio Frisoli, Massimiliano Solazzi, Domenico Buongiorno, Daniele Leonardis, Massimiliano Gabardi, Mine Sarac, Michael Felsberg, and Klas Nordberg. Deliverable D3.3 operator interface control, first version of control station.

[7] Massimiliano Gabardi, Massimiliano Solazzi, Daniele Leonardis, and Antonio Frisoli. Design and evaluation of a novel 5 DoF underactuated thumb-exoskeleton. *IEEE Robotics and Automation Letters*, 3(3):2322–2329, 2018.

[8] Blake Hannaford and Jee-Hwan Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, 2002.

[9] P. Jensfelt, J. Folkesson, G. Meneghetti, M. Felsberg, and S. Behnke. Deliverable D5.1 CENTAURO Navigation Concept.

[10] T. Klamt, X. Chen, H. Karaoguz, and S. Behnke. Deliverable D5.4 CENTAURO Walking Navigation.

[11] T. Klamt, P. Jensfelt, X. Chen, K. Nordberg, and D. Doreschel. Deliverable D5.3 CENTAURO Driving Navigation.

[12] Tobias Klamt, Max Schwarz, Christian Lenz, Lorenzo Baccelliere, Domenico Buongiorno, Torben Cichon, Xi Chen, Antonio Di Guardo, David Droeschel, Massimuliano Gabardi, Karl Holmquist, Felix Järemo-Lawin, Malgorzata Kamedula, Navvab Kashiri, Arturo Laurenzi, Daniele Leonardis, Luxa Muratore, Dmytro Pavlichenko, Arul Selvam Periyasamy, Andreas Robinso, Diego Rodriguez, Fabisn Schilling, Massimuliano Solazzi, Michael Felsberg, John Folkesson, Antonio Frisoli, Michael Gustmann, Patrix Jensfeld, Klas Nordberg, Jürgen Roßmann, Uwe Süss, Nikos G. Tsagarakis, and Sven Behnke. Solving disaster-response tasks through the centaur-like robot Centauro: Full-body telepresence and autonomous operator assistance. *Submitted to Journal of Field Robotics*, 2018.

[13] D. Pavlichenko, D. Rodriguez, and M. Schwarz. Deliverable D6.3 Autonomous Single-Arm Pick and Place Manipulation Skills.

[14] D. Pavlichenko, D. Rodriguez, M. Schwarz, C. Lenz, A. Periyasamy, and S. Behnke. Deliverable D6.4 Autonomous Dual-Arm Pick and Place Manipulation Skills.

[15] Elvira Pirondini, Martina Coscia, Simone Marcheschi, Gianluca Roas, Fabio Salsedo, Antonio Frisoli, Massimo Bergamasco, and Silvestro Micera. Evaluation of a new exoskeleton for upper limb post-stroke neuro-rehabilitation: Preliminary results. In *Replace, Repair, Restore, Relieve–Bridging Clinical and Engineering Solutions in Neurorehabilitation*, pages 637–645. Springer, 2014.

[16] Elvira Pirondini, Martina Coscia, Simone Marcheschi, Gianluca Roas, Fabio Salsedo, Antonio Frisoli, Massimo Bergamasco, and Silvestro Micera. Evaluation of the effects of the arm light exoskeleton on movement execution and muscle activities: a pilot study on healthy subjects. *Journal of neuroengineering and rehabilitation*, 13(1):9, 2016.

[17] A. Robinson, F. Jremo Lawin, A. Milan, K. Nordberg, and Felsberg M. Deliverable D6.2 Object and Workspace Perception.

[18] T. Rodehutskors, M. Schwarz, G. Meneghetti, and S. Behnke. Deliverable D6.1 CENTAURO Manipulation Concept.

[19] Jee-Hwan Ryu, Dong-Soo Kwon, and Blake Hannaford. Stable teleoperation with time-domain passivity control. *IEEE Transactions on robotics and automation*, 20(2):365–373, 2004.

[20] Mine Sarac, Massimiliano Solazzi, Daniele Leonardis, Edoardo Sotgiu, Massimo Bergamasco, and Antonio Frisoli. Design of an underactuated hand exoskeleton with joint estimation. In *Advances in Italian Mechanism Science*, pages 97–105. Springer, 2017.

[21] Mine Sarac, Massimiliano Solazzi, Edoardo Sotgiu, Massimo Bergamasco, and Antonio Frisoli. Design and kinematic optimization of a novel underactuated robotic hand exoskeleton. *Meccanica*, 52(3):749–761, 2017.

[22] Max Schwarz, Marius Beul, David Droeschel, Sebastian Schüller, Arul Selvam Periyasamy, Christian Lenz, Michael Schreiber, and Sven Behnke. Supervised autonomy for exploration and mobile manipulation in rough terrain with a centaur-like robot. *Frontiers in Robotics and AI*, 3:57, 2016.

[23] Max Schwarz, Tobias Rodehutskors, David Droeschel, Marius Beul, Michael Schreiber, Nikita Araslanov, Ivan Ivanov, Christian Lenz, Jan Razlaw, Sebastian Schüller, et al. NimbRo Rescue: Solving disaster-response tasks with the mobile manipulation robot Momaro. *Journal of Field Robotics*, 34(2):400–425, 2017.