Remote Mobile Manipulation with the Centauro Robot: Full-body Telepresence and Autonomous Operator Assistance

Tobias Klamt[1]   Max Schwarz[1]   Christian Lenz[1]   Lorenzo Baccelliere[4]
klamt@ais.uni-bonn.de

Domenico Buongiorno[2]   Torben Cichon[3]   Antonio Di Guardo[2]   David Droeschel[1]

Massimiliano Gabardi[2]   Malgorzata Kamedula[4]   Navvab Kashiri[4]   Arturo Laurenzi[4]

Daniele Leonardis[2]   Luca Muratore[4,6]   Dmytro Pavlichenko[1]   Arul Selvam Periyasamy[1]

Diego Rodriguez[1]   Massimiliano Solazzi[2]   Antonio Frisoli[2]   Michael Gustmann[5]

Jürgen Roßmann[3]   Uwe Süss[5]   Nikos G. Tsagarakis[4]   Sven Behnke[1]

Abstract

Solving mobile manipulation tasks in inaccessible and dangerous environments is an important application of robots to support humans. Example domains are construction and maintenance of manned and unmanned stations on the moon and other planets. Suitable platforms require flexible and robust hardware, a locomotion approach that allows for navigating a wide variety of terrains, dexterous manipulation capabilities, and respective user interfaces. We present the CENTAURO system which has been designed for these requirements and consists of the Centauro robot and a set of advanced operator interfaces with complementary strength enabling the system to solve a wide range of realistic mobile manipulation tasks. The robot possesses a centaur-like body plan and is driven by torque-controlled compliant actuators. Four articulated legs ending in steerable wheels allow for omnidirectional driving as well as for making steps. An anthropomorphic upper body with two arms ending in five-finger hands enables human-like manipulation. The robot perceives its environment through a suite of multimodal sensors. The resulting platform complexity goes beyond the complexity of most known systems which puts the focus on a suitable operator interface. An operator can control the robot through a telepresence suit, which allows for flexibly solving a large variety of mobile manipulation tasks. Locomotion and manipulation functionalities on different levels of autonomy support the operation. The proposed user interfaces enable solving a wide variety of tasks without previous task-specific training. The integrated system is evaluated in numerous teleoperated experiments that are described along with lessons learned.

[1]Autonomous Intelligent Systems, University of Bonn, Bonn, Germany
[2]PERCRO Laboratory, TeCIP Institute, Scuola Superiore Sant'Anna, Pisa, Italy
[3]Man-Machine Interaction, RWTH Aachen University, Aachen, Germany
[4]Department of Advanced Robotics, Italian Institute of Technology, Genoa, Italy
[5]Kerntechnische Hilfsdienst GmbH, Karlsruhe, Germany
[6]School of Electrical and Electronic Engineering, The University of Manchester, Great Britain

3D laser scanner

Cameras

RGB-D sensor

Base with CPUs,
router and battery

7 DoF arm

9 DoF dexterous hand
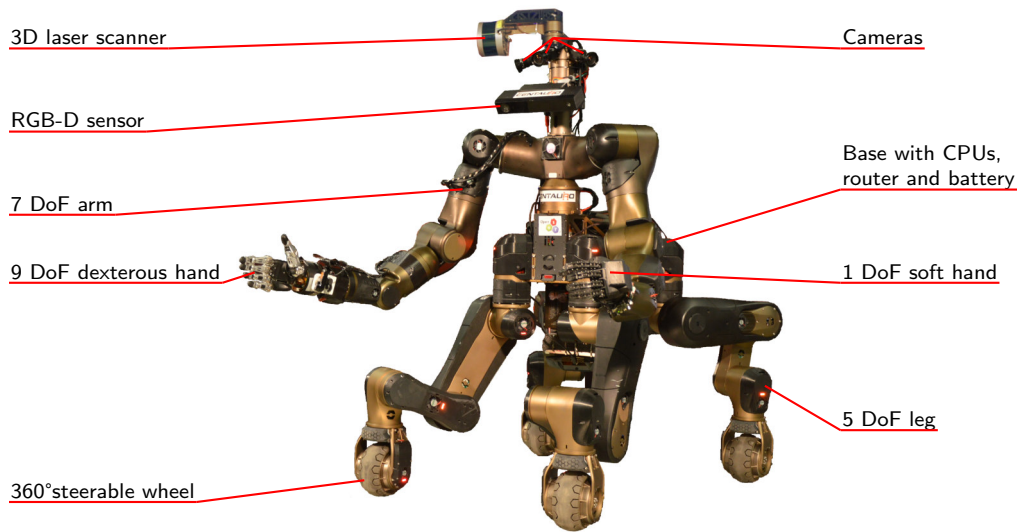
1 DoF soft hand

5 DoF leg

360°steerable wheel

Figure 1: The Centauro robot.

1 Introduction

Capable mobile manipulation robots are desperately needed in environments which are inaccessible or dangerous for humans. Missions include construction and maintenance of manned and unmanned stations, as well as exploration of unknown environments on the moon and other planets. Furthermore, such systems can be employed in search and rescue missions on earth. It applies to all these missions that human deployment is impossible or dangerous, and depends on extensive logistical and financial effort.

To address the wide range of possible tasks, a suitable platform needs to provide a wide range of capabilities. Regarding locomotion, exemplary tasks are to overcome a variety of obstacles which can occur on planetary surfaces and in man-made environments, e.g., in space stations. Regarding manipulation, tasks may be to use power tools, to physically connect and disconnect objects such as electrical plugs, or to scan surfaces, e.g., for radiation. Since maintenance is not possible during missions, a high hardware and software reliability is necessary. Furthermore, suitable operator interfaces are key to enable the control of a system that must solve such a large variety of tasks. In the European H2020 project CENTAURO[1], we develop a system according to the above requirements. It consists of the Centauro robot, operator interfaces including a full-body telepresence suit, autonomous locomotion and manipulation functions, and modules for communication and simulation.

Centauro, shown in Fig. 1, has a centaur-like body plan with four articulated legs and an anthropomorphic upper body. It is driven by torque-controlled series-elastic actuators. Each leg has five degrees of freedom (DoF) and ends in a directly driven, 360° steerable wheel. This leg design allows for both omnidirectional driving and stepping locomotion and combines their advantages. Sufficiently flat terrain can be traversed easily by driving, which is fast, energy efficient, and provides high stability. Omnidirectional motion enables the robot to navigate precisely in narrow spaces and to position itself accurately for manipulation tasks. More challenging terrain and height differences can be overcome by stepping, which only requires isolated footholds. The design further enables motions that are neither possible for pure driving nor for pure walking robots, such as changing the robot footprint under load.

Centauro's anthropomorphic upper body consists of a yaw joint in the spine and two 7 DoF arms. In combination with the adjustable base height and orientation, this results in a workspace equal to an adult person. The arms end in two different anthropomorphic five-finger hands to provide a wide range of manipulation

---

[1] https://www.centauro-project.eu

capabilities. A SoftHand, driven by a single actuator, is mounted on the left arm. Through its compliant design, it allows for robust manipulation. The right arm ends in a 9 DoF Schunk hand with 20 joints, capable of dexterous, human-like manipulation.

Environment perception is realized through a variety of sensors. A continuously 3D rotating scanner with a spherical field-of-view provides range measurements. Three wide-angle RGB cameras capture images from the robot head perspective. An additional RGB-D sensor is mounted on a pan-tilt unit below these cameras. For some tasks, the robot can be equipped with additional RGB cameras at suitable positions such as under the robot base.

A system with this amount of kinematic capabilities and sensory input is novel and directs the focus to an effective operator interface. This needs to be intuitive to enable fast and safe operation and flexible to address unknown tasks. Situation awareness for the Centauro operators is realized through a simulation-based environment, generated from sensor data and enriched with semantic information. A digital twin of the robot is placed in this representation. This allows for flexible views on the current scene. A full-body telepresence suit allows an operator to intuitively control the whole robot. Key component is an upper-body exoskeleton, which transfers the operator's arm, wrist and hand movements to the robot. It provides force feedback and, thus, allows the operator to solve a large variety of manipulation tasks.

For operator support, we developed autonomous locomotion and manipulation functions. This is especially helpful when operating the robot with a high latency and low bandwidth connection. A locomotion planner provides hybrid driving-stepping paths and a respective controller executes them. The only required operator input is a desired robot goal pose. An autonomous manipulation interface detects and categorizes objects, plans and executes optimized arm trajectories towards these objects and finally provides grasping movements. The latter even applies to unknown objects, since suitable grasping poses are derived from known instances of the same object class. Support operator can further control the robot through a set of locomotion and manipulation control interfaces which range from joint space control to a semi-autonomous stepping controller and possess complementary strengths. Operators can choose from the provided interfaces depending on the individual task and resulting in an overall large flexibility and wide range of controllable capabilities. This is especially helpful when tasks are previously unknown. While many of these components have been individually presented in other works, we believe that the integration into a holistic remote mobile manipulation system which is evaluated in a wide range of realistic tasks is novel and goes beyond the state of the art.

The system was evaluated during an intensive testing period at the facilities of Kerntechnische Hilfsdienst GmbH (KHG), Karlsruhe, Germany, which is part of the German nuclear disaster response organization. KHG operates a variety of remote-controlled manipulator vehicles and has deep knowledge of the application domain. Locomotion capabilities were evaluated in tasks like driving up a ramp, overcoming a gap, moving through an irregular step field, and autonomously climbing a flight of stairs. Semi-autonomous and autonomous manipulation was evaluated in tasks like drilling a hole with a power drill, screwing a screw with an electrical screw driver, cutting a cable with an electrical cutting tool, mounting a snap-hook, opening and closing lever-type and gate-type valves, connecting and disconnecting a 230V power plug and a fire hose, scanning a surface with a radiation measurement device, and grasping and operating an electrical screw driver. A combination of locomotion and manipulation capabilities was required for solving the task of opening and passing a regular door. Visual contact between the operators and the robot was not provided for any of the tasks, so the operators had to rely on the environment representation provided by the robot sensors. Most of the tasks were performed successfully and without previous training during this evaluation period which demonstrates the wide range of capabilities the system provides and the flexibility to adapt to unknown tasks. The project has led to many important lessons learned regarding the design of the hardware, the operator interfaces, and the autonomous functionalities, which will be discussed after reporting the evaluation results.

2   Related Work

Mobile manipulation robots have been developed for a variety of fields such as search and rescue, planetary exploration, or personal assistance. Those robots vary in their locomotion strategy which can be realized by driving with wheels or tracks, by walking with multiple legs, or by a combination of both. Further differences between mobile manipulation robots can be found in their manipulation setup regarding the number of arms, their DoFs, and their effectors. Key to the applicability in a wide range of scenarios are teleoperation interfaces which enable teleoperators to utilize large parts of the robot capabilities while keeping the cognitive load low and the interface flexibility high.

Driving robots have the capability to overcome long distances quickly with low energy consumption and high robot stability. Examples for mobile manipulation robots with a wheeled base are the NASA Centaur (Mehling et al., 2007) and the NimbRo Explorer (Stückler et al., 2016). Tracked mobile manipulation robots, as the iRobot Packbot (Yamauchi, 2004), are capable of overcoming more challenging terrain at the cost of less energy efficiency. However, all driving platforms are limited to suitable terrain.

Legged robots have the ability to overcome more challenging terrain since they only require isolated footholds. Examples for biped manipulation robots are the Boston Dynamics Atlas provided to some teams for the DARPA Robotics Challenge (DRC), e.g. IHMC (Johnson et al., 2015), or the Walk-Man robot (Tsagarakis et al., 2017) which features a powerful actuation concept with actively controllable compliance. Recently, Toyota unveiled their humanoid T-HR3 robot[2]. It is controlled by an exoskeleton and mimics the operator's behavior. Examples for quadruped platforms are the Titan-IX (Hirose et al., 2005), the Boston Dynamics BigDog (Raibert et al., 2008) with an attached powerful arm (Abe et al., 2013), the HyQ (Semini et al., 2011) with an attached HyArm (Ur Rehman et al., 2015), and the Boston Dynamics SpotMini with an attached arm[3]. A platform with six legs, of which some can also be used as arms, is the Lauron-V (Roennau et al., 2014). However, legged robots are often slow, compared to driving robots, less energy efficient, and require a more complex control strategy since the robots are less stable while walking.

Hybrid driving-stepping platforms combine the advantages of both locomotion strategies. Long distances of sufficiently even terrain can be traversed by driving, while stepping is performed on rough terrain where driving is infeasible. This combination allows for quick and energy-efficient locomotion in a wide variety of terrain types. Hybrid locomotion designs result in platforms with many DoFs, though, requiring the development of suitable teleoperation approaches to utilize all of their capabilities. An early example for a hybrid locomotion manipulation platform is WorkPartner (Halme et al., 2003). A recent bipedal platform is the Boston Dynamics Handle[4]. In 2015, the DRC pushed research teams to develop robots that are capable of performing several manipulation tasks and overcoming terrain with variable complexity. Four of the five best teams chose a hybrid driving-stepping platform which might be an indication for the advantages of such a locomotion strategy.

RoboSimian (Hebert et al., 2015) is a statically stable quadrupedal robot with four generalized limbs, developed for the DRC. Each limb consists of seven joints with identical actuators and ends in an under-actuated hand. Both stepping locomotion and manipulation tasks can be performed with each limb. Furthermore, RoboSimian has two active wheels at its trunk and two caster wheels at two of its limbs which allow for driving on flat terrain. To switch between driving and stepping locomotion, the robot has to lower itself to a predefined sitting pose. The environment is perceived with multiple stereo cameras and the operator interface is a standard laptop from which the operator can design, parametrize and sequence predefined behaviors. Recently, Motiv Robotics unveiled its successor RoboMantis[5]. At first sight, its design shows many similarities to RoboSimian. However, RoboMantis' kinematic concept moved away from the idea of generalized limbs. Its four legs end in wheels allowing for hybrid locomotion while additional arms can be mounted on the platform for manipulation. In addition, a laser scanner is utilized for environment perception.

---

[2]http://corporatenews.pressroom.toyota.com/releases/toyota+unveils+third+generation+humanoid+robot+thr3.htm
[3]https://www.youtube.com/watch?v=tf7IEVTDjng
[4]https://www.youtube.com/watch?v=-7xvqQeoA8c
[5]https://www.youtube.com/watch?v=x2s_ufwxRJE

The DRC winner robot DRC-HUBO (Zucker et al., 2015) is a humanoid robot with 32 DoF. In addition to bipedal walking, it is able to lower itself to a kneeing pose and perform driving locomotion over flat terrain due to four small wheels at its knees and ankles. The robot uses two 7-DoF arms which end in hands with three fingers. The right hand is extended by an additional trigger finger. The environment is perceived by a laser scanner, three RGB cameras and a RGB-D camera. Operation is apportioned among three operators with different tasks which control the robot by selecting and adapting predefined poses.

Another hybrid locomotion platform from the DRC is CHIMP (Stentz et al., 2015). It has a roughly anthropomorphic body with two arms and two legs providing 39 DoF. All four limbs are equipped with powered tracks which enable driving with two or four ground contact points. Each arm has seven DoF and ends in a three-finger gripper. Environment perception is done with two laser scanners, a panomorphic camera and four stereo cameras. The CHIMP operator interface is a combination of manual and autonomous control. Task-specific motions are configured through wizards by the operator before being executed.

Finally, the centaur-shaped mobile manipulation robot Momaro (Schwarz et al., 2017) of team NimbRo was one of the top participants at the DRC. With its 4-DoF passively compliant legs ending in 360° steerable pairs of wheels, it provides hybrid driving-stepping locomotion as well. In contrast to the previously presented systems, Momaro is capable of omnidirectional driving which is helpful for precise navigation. Since no posture change is necessary to switch between driving and walking mode, it is possible to perform unique motions such as changing its configuration of ground contact points under load. Momaro's upper body consists of two arms with seven DoF each and two grippers with eight DoF and exchangeable fingers. Environment perception is realized through a continuously rotating laser scanner with spherical field-of-view, seven RGB cameras and an infrared distance sensor mounted at one hand. The operator interface offers a variety of options to control the 56 DoF robot. Omnidirectional driving is controlled by a joystick. Legs and arms can be moved with a keyframe editor in Cartesian and joint space. In addition, the operator can access predefined motions. For stepping locomotion, a semi-autonomous stepping strategy was presented for stair climbing which relies on perceived terrain heights (Schwarz et al., 2016b), and manipulation tasks can be performed via bimanual telemanipulation (Rodehutskors et al., 2015). A teleoperator wears a head-mounted display which shows an egocentric view of the scene from the robot perspective and includes head movements in the displayed scene. Two hand-held controllers with magnetic trackers project the operators hand movements to the robot arms. Momaro was also used at the DLR SpaceBot Cup 2015 (Schwarz et al., 2016a). In a planetary exploration scenario, it demonstrated navigation and omnidirectional driving in rough terrain, the identification and transportation of objects, taking a soil sample, and performing assembly tasks. All tasks were performed with high degree of autonomy and only few operator interventions.

Centauro combines Momaro's kinematic design with the compliant actuation concept of Walk-Man. While the centaur-like hybrid wheeled-legged body plan is still used, several aspects were improved. Momaro's off-the-shelf actuators tended to overheat in some situations and the passively compliant leg links made precise end-effector control difficult. For the Centauro design, we use custom built series-elastic actuators which provide the option of compliant behavior that can be actively adapted to the situation. In contrast to Momaro, which only provided leg movement in the sagittal plane, Centauro's legs contain an additional hip yaw joint which increases movement capabilities, especially while stepping. Since teleoperating such hybrid-driving stepping locomotion is very complex and puts a high cognitive load on the teleoperator, we introduce a respective planning approach which generates and executes hybrid-driving stepping locomotion paths autonomously. Regarding manipulation, the bimanual teleoperation approach showed the capabilities to transmit the operator's behavior to the robot but due to imprecision and the lack of force feedback, the operation was still challenging and limited in its capabilities. Centauro's manipulation can be teleoperated by a full-body telepresence suit which eliminates those weak points. Autonomous manipulation functionalities for arm movements and grasping further support the operator. To increase manipulation capabilities, we utilize an anthropomorphic Schunk hand (Ruehl et al., 2014) and a flexible SoftHand (Catalano et al., 2014) with complementary characteristics. Finally, a simulation-based visualization system provides an operator interface with high flexibility.
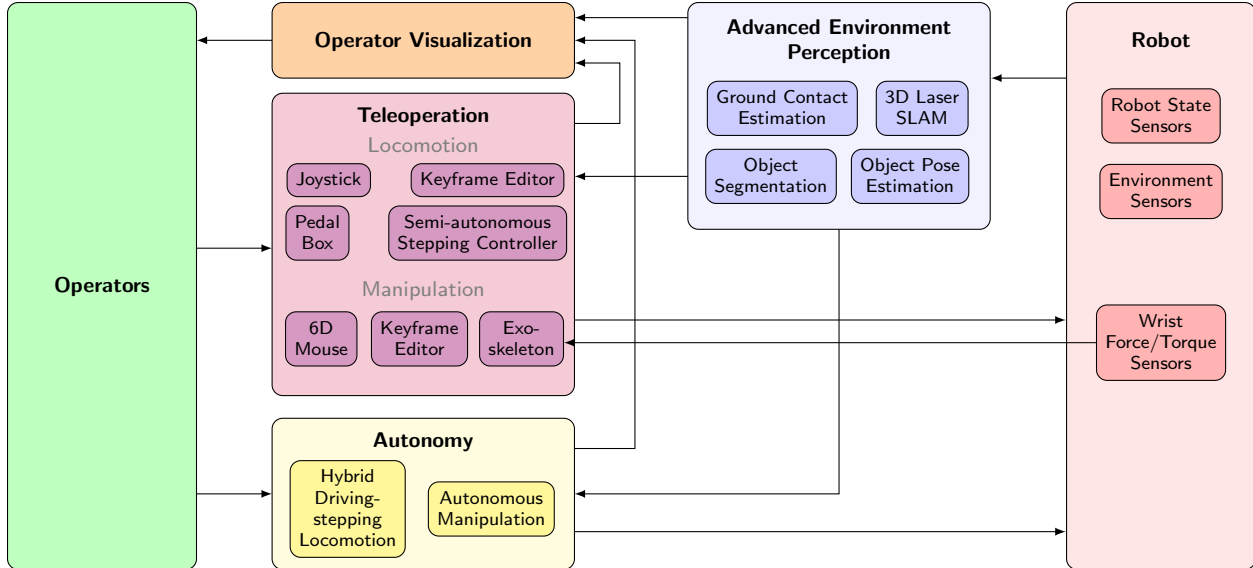
Figure 2: System architecture overview.

3   System Overview

The proposed system architecture possesses multiple hardware and software components. An overview over these components and the communication architecture between them is given in this section.

Starting point is the Centauro robot. It perceives information about its internal state and the environment through a 3D rotating laser scanner, a RGB-D sensor, multiple RGB cameras, and internal sensors. The hardware and sensor setup is described in Section 4.1.

Sensor measurements are processed by the advanced environment perception components to obtain ground contact estimation for each foot, registered point clouds for 3D mapping and localization, as well as object segmentation and respective pose estimation. Those components are described in Section 5. In addition, raw RGB images are used for operator visualization and measurements from the force/torque sensors at the robot wrists are used for force feedback in the telepresence suit.

To provide situation awareness, the operator visualization displays RGB images, task specific control interfaces as well as a simulation-based digital twin of the robot. The latter uses data from RGB(-D) cameras and registered point clouds to generate an environment and robot model. Details can be found in Section 6.2.

Regarding the robot control, we propose a set of teleoperation interfaces with different levels of autonomy to provide locomotion and manipulation control for a wide range of tasks. Driving locomotion can be controlled by a joystick while stepping locomotion is addressed by a keyframe editor and a semi-autonomous controller for stepping locomotion in rough terrain. The latter uses the ground contact estimation to autonomously adapt to the ground structure. Details are given in Section 6.3. Manipulation can be controlled by the same keyframe editor as well as a 6D input device. Major effort has been spent in the development of a full-body telepresence station to provide advanced teleoperation. It consists of a pedal box for driving control and an upper-body exoskeleton for intuitive bimanual teleoperation of the robot arms, wrists, and end-effectors while providing force-feedback. The exoskeleton hardware is described in Section 4.2 while Section 6.4 describes the manipulation control interfaces. While, besides the telepresence suit, most operator interfaces are well known, we see a novelty of our work in the combination and the integration of these interfaces to enable the operator-robot team to solve a considerably wide range or realistic tasks.

Autonomous control functions require less input from the operator which decreases the dependency on the communication link while relying on additional data from the robot sensors. A hybrid driving-stepping locomotion planner which uses registered point clouds for the environment representation and plans and executes paths to goal poses that are defined by the operator. It is described in Section 7. For autonomous manipulation, objects from the object segmentation component and their respective poses from the pose estimation component are used to plan grasps. Arm trajectories towards the grasp poses are optimized using registered point clouds for environment representation. Details are given in Section 8. All control interfaces require knowledge about the current robot state and output desired poses to the robot as well as respective information for the operator visualization.

4 Mechatronic Design of Centauro Robot and Exoskeleton

The CENTAURO system is designed for mobile manipulation missions in challenging environments. Since the task details in such missions are in general unknown in advance, the system has to cover a wide variety of navigation and manipulation capabilities. Regarding navigation capabilities, addressed environments are either planetary surfaces or man-made and thus, contain, e.g., flat and rough surfaces, ramps, and stairs. Those environments may, however, be affected, e.g., by cluttered debris. Regarding manipulation, the usage of tools and objects is essential for many tasks such as drilling holes or cutting objects. Moreover, it is desirable to use tools which are designed for human usage since those are easily available, and sufficiently tested to provide a high reliability. The robot hardware design is described in Section 4.1. The design of a full-body telepresence suit for the operator is described in Section 4.2.

4.1 The Centaur-like Robot Centauro

According to the requirements of considered scenarios, the Centauro robot (see Fig. 1) is designed as a centaur-like platform embodying an anthropomorphic upper-body and a quadrupedal lower-body—bringing together Momaro's kinematics and Walk-Man's compliant actuation. The legs are equipped with active wheels at their lower end and the two arms possess two hands with complementary capabilities as end-effectors. Section 4.1.1 describes the kinematics in detail, Section 4.1.2 explains the developed actuators and Section 4.1.3 focuses on the software architecture. Moreover, the robot platform includes a head module comprising a set of cameras and sensors. It encompasses a *Microsoft Kinect V2* RGB-D sensor (Fankhauser et al., 2015) whose tilt and pen angle can be actively controlled. In addition, an array of three *PointGrey BlackFly BFLY-U3-23S6C* wide angle color cameras and a rotating *Velodyne Puck* 3D laser scanner with a spherical field-of-view provide further environment perception. A *VectorNav VN-100* inertial measurement unit (IMU) is mounted in the torso. Two additional RGB cameras are mounted under the robot base to get a view on the feet.

Furthermore, the robot base incorporates three computing units. One unit is responsible for the real-time control of the robot. It possesses an *Intel Core i7-6820EQ*@2.8 Ghz, 16 GB RAM and a 120 GB SSD and runs with the XENOMAI RT development kit[6]. The other two computing units are used for perception and high-level robot control. They both possess an *Intel Core i5-7500T*@2.7 GHz, 32 GB RAM, a *GeForce GTX 1070*, and a 500 GB SSD. In addition, to allow for untethered operations, the robot pelvis accommodates a *Netgear Nighthawk X10 R900* wireless communication router and a 7.5 kg Lithium-Ion polymer battery of 34.6 Ah capacity supplying 48 V with 80 A max current discharge (limited by PCM). This provides power for about two hours of operation when performing standard manipulation and locomotion tasks.
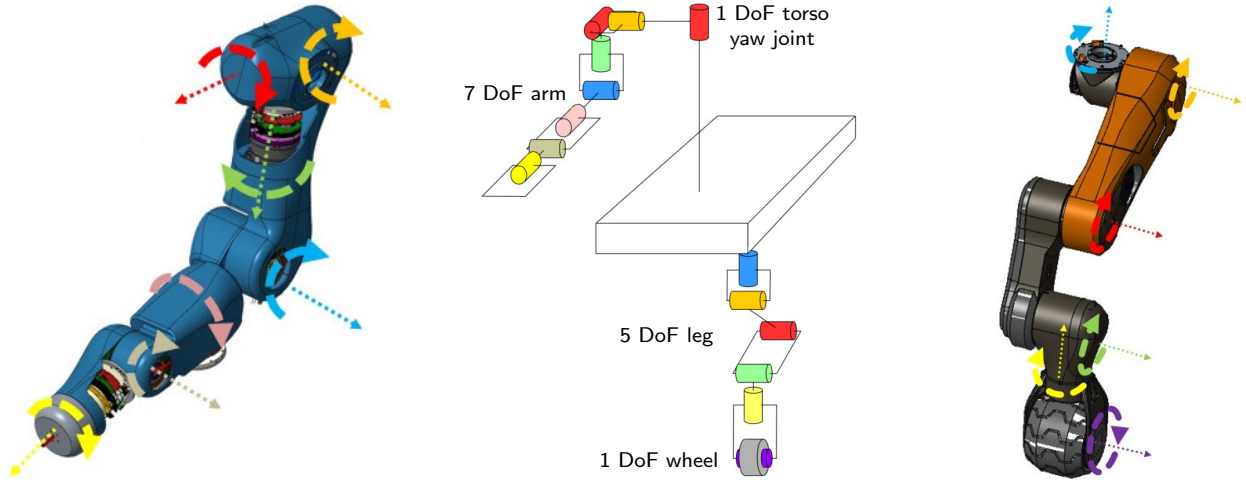
---

[6]https://xenomai.org/

Figure 3: Kinematic layout. Left: The right arm. Joint axes are marked with colored lines. Center: Kinematic tree. For clarity, only one arm and one leg are pictured. The hands are neglected. Proportions are not to scale. Right: The front left leg.

4.1.1 Kinematics

The lower-body comprises four wheeled legs of 81 cm length, whose first joints are accommodated in the four corners of the base structure. Fig. 3 illustrates the kinematic scheme and joint positioning. To allow for versatile locomotion, each leg consists of five DoF in a spider-like configuration, which can be more beneficial in terms of stability required for the manipulation of powerful tools, as shown in (Kashiri et al., 2016). Furthermore, in this configuration, the first leg joint has to deliver substantially lower effort and power compared to a mammal-like configuration. According to the chosen spider-like configuration, each hip module consists of a yaw and a pitch joint, followed by another pitch joint in the knee. Each ankle consists of a pitch and a yaw joint which allow for steering the wheel and adjusting its steering axis to the ground. Finally, each leg ends in an actively drivable wheel with 78 mm radius. The described configuration allows for omnidirectional driving as well as for articulated stepping locomotion. Since no posture change is needed to switch between the two, it is even possible to perform motions which are unique for that design such as moving a foot relative to the base while under load. Thus, a wide range of locomotion capabilities is provided. To permit versatile leg articulation in difficult terrains, the ranges of the leg joints were maximized while taking into account the mechanical and electrical interfacing constraints.

The robot torso incorporates two arms with seven DoF each and an additional rotational joint in the waist, to endow the upper-body with yaw rotation. The kinematics of the two arms closely resembles an anthropomorphic arrangement to provide a large workspace, to enable dexterous manipulation, and to simplify teleoperation. Each arm is 73 cm long and comprises of three DoFs at the shoulder, one DoF at the elbow and another three DoFs at the wrist. The degree of redundancy helps to overcome possible constraints that may be introduced in the task space by the surrounding environment. Even though this is a traditional design that aims at replicating the anthropomorphic structure of the human arm with seven DoF, it is only approximately equivalent for the human arm kinematic structure. To extend the range of motion of the elbow joint, an off-center elbow configuration was chosen. Similarly, for the wrist, a non-anthropomorphic configuration with non-intersecting axes was considered to maximize the range of motion of the wrist flexion and abduction motions. Finally, humans have the ability to elevate (upward/downward) and to incline (forward/backward) the shoulder joint, utilizing supplementary kinematic redundancy of the arm to achieve certain goals in task coordinates. This, however, would require the addition of two more DoF to each arm, increasing the complexity/weight and dimensions of the overall platform. To avoid this, while at the same time obtain, to some extent, the benefits provided by the elevation (upward/downward) and inclination (forward/backward) of the shoulder, a fixed elevated and inclined shoulder arrangement was selected based
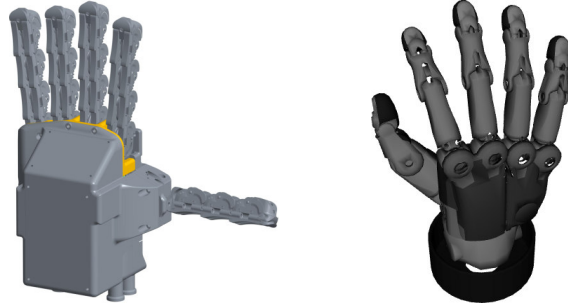
Figure 4: Centauro manipulation end-effectors: 1 DoF SoftHand (l.) and anthropomorphic 9 DoF Schunk hand (r.).

on the optimization study in which important manipulation indices were considered and evaluated in a prioritized order (Baccelliere et al., 2017).

The two arms end in different end-effectors with complementary properties to provide an overall wide range of manipulation capabilities. Both hands are shown in Fig. 4. On the left side, a SoftHand with one DoF provides compliant and robust manipulation. The right arm utilizes an anthropomorphic Schunk hand with nine DoF for dexterous manipulation tasks. A customized force-torque sensor between the right arm wrist and the Schunk hand measures 6D forces/torques which are applied to the end-effector and can be used for force feedback by the exoskeleton.

Overall, the robot possesses 52 DoFs and weighs 92 kg. The base width and length are both 61 cm. The whole robot is 171 cm tall when legs are fully extended, while the robot height can be reduced to 112 cm when folding the legs.

4.1.2 Compliant Actuation

Having defined the kinematics of the limbs, the required actuation performance for each joint could be derived. To get an estimation of the actuation needs, a series of simulation studies was executed using an initial model of the robot based on estimated rigid body dynamic properties (Kamedula et al., 2016). A set of trajectories with different frequencies which explored the overall robot workspace was executed, while carrying a 10 kg payload per arm. Respective joint torques were examined, resulting in the identification of a set of actuator classes with different sizes and torque levels ranging from about 30 to 270 Nm. Fig. 5 shows the manufactured actuation units of the identified classes. Table 1 shows the respective specifications. For the medium and small sized actuators there are two versions each with either a higher maximum velocity or higher continuous torque.

The series-elastic actuation (SEA) technology is utilized to protect the reduction gear against impacts—improving the system sturdiness, while at the same time being used for measuring the actuator torque

Figure 5: Manufactured actuation units of the identified classes (f.l.t.r.): Large, Medium, Small.

Table 1: Actuator specifications.

Type	Gear ratio	Joints	Maximum velocity [rad/s]	Torque [Nm] peak - continuous	Power [W] peak - continuous	Torque sensing resolution [Nm]	Mass [kg]
Large	120	Hip, Knee	8.8	268 - 92	2096 - 778	0.2	1.73
Medium A	160	Ankle Pitch, Elbow, Shoulder yaw	6.1	147 - 46	820 - 259	0.07	1.28
Medium B	160	Torso, Shoulder pitch, Shoulder roll	3.9	147 - 81	1014 - 295	0.07	1.45
Small A	100	Forearm yaw, Forearm pitch	11.7	55 - 17	556 - 179	0.07	1.0
Small B	100	Wheel, Ankle yaw, Wrist yaw	20.4	28 - 9	518 - 167	0.07	0.87

through the monitoring of the elastic element deflection. Considering the influence of different joints' stiffness levels on the natural dynamics and control of the robot, as discussed by (Kashiri et al., 2013) and (Roozing et al., 2017), and taking into account the available space for the different actuators, two technologies were utilized for joint torque measurement based on strain-gauge and deflection-encoder principles (Kashiri et al., 2017). The stiffness of the SEA deflection-encoder-based sensor is defined with respect to the required torque measurement resolution across the different joints. It was set ten times lower than the cogging torque of the motor drive when deflection of the sensor flexure is measured using a high resolution 19-bit absolute encoder.

4.1.3 Low-Level Software Architecture

For the low-level control of the Centauro platform, we developed *XBotCore* (*Cross-Bot-Core*), a light-weight, real-time (RT) software platform for robotics (Muratore et al., 2017). *XBotCore* is open-source[7] and is designed to be both an RT robot control framework and a software middleware. It satisfies hard RT requirements, while ensuring a 1 kHz control loop even in complex multi-DoF systems. The *XBotCore* Application Programming Interface (API) enables an easy transfer of developed software components to multiple robot platforms (cross-robot feature), inside any robotic framework or with any kinematics/dynamics library as a back-end. Out-of-the-box implementations are available for the *YARP* and *ROS* software frameworks and for the *RBDL* and *iDynTree* dynamics libraries.

A Robot Hardware Abstraction Layer (R-HAL), introduced in (Rigano et al., 2018) that permits to seamlessly program and control any robotic platform powered by *XBotCore* is also provided by the framework. Moreover, a simple and easy-to-use middleware API, for both RT and non-RT control frameworks is available. The *XBotCore* API is completely flexible with respect to the external control framework the user wants to utilize. Inside the Centauro robot, an RT Cartesian control plugin based on the OpenSoT (Hoffman et al., 2017), a hierarchical whole-body control library and the built-in *ROS* support are used. As shown in Fig. 6, *XBotCore* spawns three threads in the Linux Xenomai[8] RTOS:

- The R-HAL RT thread is running at 1 kHz and is responsible to manage and synchronize the Ether-CAT slaves in the robot, i.e., the electronic boards responsible for motor control and sensor data acquisition.
- The Plugin Handler RT thread is running at 1 kHz and is responsible to start all the loaded plugins, execute them sequentially and close them before unload. It is possible to dynamically load and unload

---

[7]https://github.com/ADVRHumanoids/XBotCore
[8]https://xenomai.org

one or more plugins in the Plugin Handler. As an example, the above-mentioned RT Cartesian control plugin is running inside the Plugin Handler. A shared memory communication mechanism is used to share data between this component and the R-HAL at 1 kHz.

- The Communication Handler non-RT thread is running at 200 Hz and is responsible for the communication with external frameworks. This component provides the option to send the desired robot state from the non-RT API to the chosen communication framework and to receive the reference, respectively. The Communication Handler uses XDDP (Cross Domain Datagram Protocol) for the asynchronous communication between RT and non-RT threads, guaranteeing a lock-free IPC (Inter-Process Communication). The run loop of this component is quite simple: it updates the internal robot state using the XDDP pipe with the non-RT robot API, sends the robot state to all the communication frameworks, receives the new reference from the requested "master" (we avoid to have multiple external frameworks commanding the robot) and finally, sends the received reference to the robot using the XDDP non-RT robot API.

4.2 The Full-body Telepresence Suit

An intuitive teleoperation interface is key to control a robot, as complex as Centauro, with all its capabilities. A full-body telepresence suit (see Fig. 7) allows for immersive control of the whole robot, especially for manipulation tasks. A high degree of intuition is reached by giving the operator the feeling of being present in the robot by perceiving the scene from the robot head perspective and moving the robot arms and hands as own limbs. This is realized through a head-mounted display for visual and acoustic situation awareness, pedals for basic locomotion control, and—most important—an upper-body exoskeleton. The latter enables the operator to transfer his upper body movements to the robot. Hence, a wide range of movement that covers most of the human workspace is required. The presented telepresence suit can reach about 90% of the human's arm workspace without singularities. Moreover, the upper-body exoskeleton provides force feedback by displaying the interaction forces between the robot and its environment. Thus, it allows the operator to transfer his or her experience and capability of situation assessment into the scene. The upper-body exoskeleton consists of

- two upper limb active exoskeletons for the arms with four DoF each (see Section 4.2.1),
- two active wrist exoskeletons with three DoF each (see Section 4.2.2),
- and an underactuated active hand orthosis with one DoF for each finger (see Section 4.2.3).

4.2.1 Arm Exoskeleton

The dual arm exoskeleton is a mechanically compliant robotic manipulator that provides full support for shoulder and elbow rotation (Pirondini et al., 2014). It features an innovative and lightweight backdrivable transmission with electric actuators that are located behind the operator's back. The torque transmission from the actuators to the joints is achieved through idle pulleys and in-tension metallic tendons. This design allows for light moving parts which achieve a high dynamic performance due to low inertia. Furthermore,
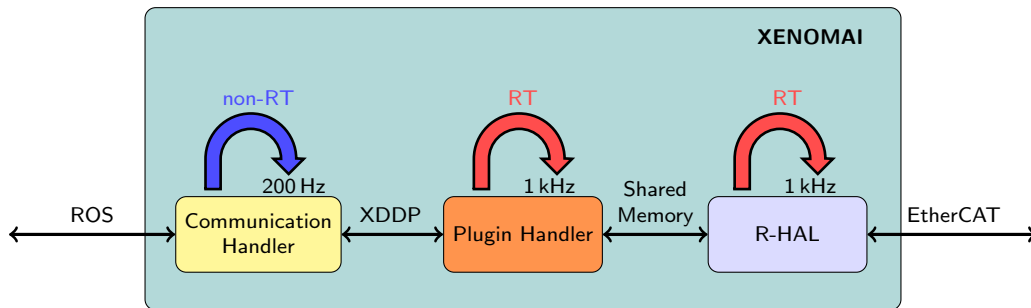

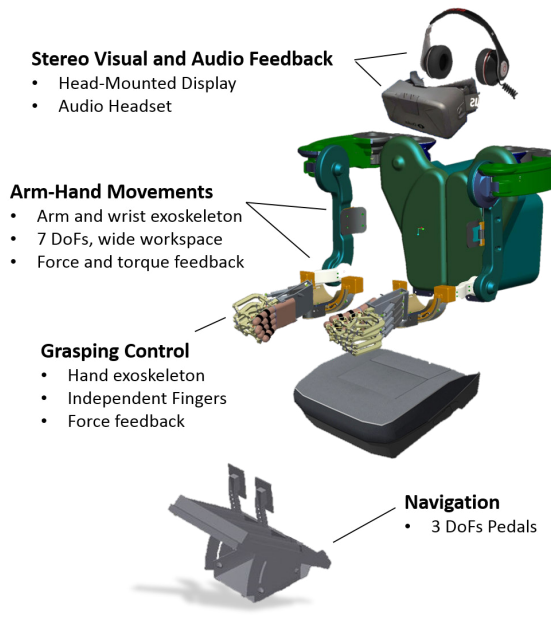Figure 6: *XBotCore* threads and communication architecture.

Figure 7: The full-body telepresence suit, components and implementation.

since neither actuators nor force-torque sensors are positioned on the joints or end-effectors, a slim design is possible that results in a large motion range and a low risk of either self-collision or collision with the human body. The total weight of the moving parts is only $3\,\text{kg}$, of which about $2\,\text{kg}$ belong to the first two proximal links.

However, the utilized metallic tendons show some compliant behavior which would result in inaccurate motion tracking or force transmission. Thus, two position sensors are used: one at the motor shaft, the other at the joint shaft, after the tendon transmission. The difference of the measured relative position is an indirect measurement of the tension of the tendon. Such method allows a higher degree of robustness regarding the force control stability to the variability of the human limb mechanical impedance. To assure a high degree of reliability, the sensory equipment of the arm exoskeleton consists of only two kinds of position sensors with a proven high robustness: optical incremental encoders are embedded in the electric actuators, and Hall effect-based absolute angular sensors are directly integrated in the robotic joints. Fig. 8 shows the manufactured arm exoskeleton and its kinematic design.

The arm exoskeleton kinematic is isomorphic to that of the human arm, i.e., the device's axes of rotation are aligned with those of the operator's physiological articulations. It has the important property of presenting no singularities in the natural workspace of the upper limb. In order to leave enough space for the user's shoulder without having any interference with the mechanical structure of the device, the implementation of the 2nd DoF utilizes a remote center of rotation mechanism (RCRM) based on a four-link pantograph. The ranges of motion for the first, second and third DoF are 120°, 95° and 165°, respectively.

The major part of the control electronics has been custom built to be embedded into the mechanics of the device, in order to minimize the electrical wiring effort at the motors and 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 support 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 users of different size to operate the device from a comfortable seating position.
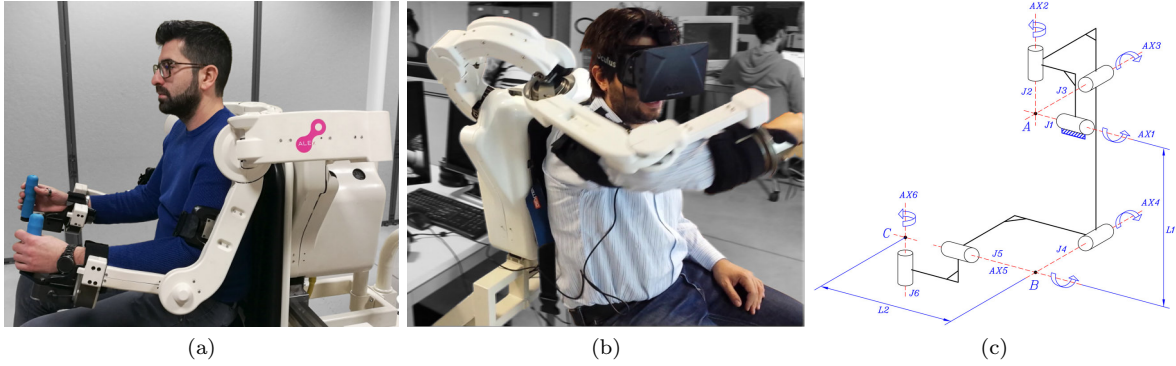
Figure 8: (a),(b) The arm exoskeleton features four actuated DoFs (shoulder and elbow) in a wide operative workspace, (c) kinematic scheme.
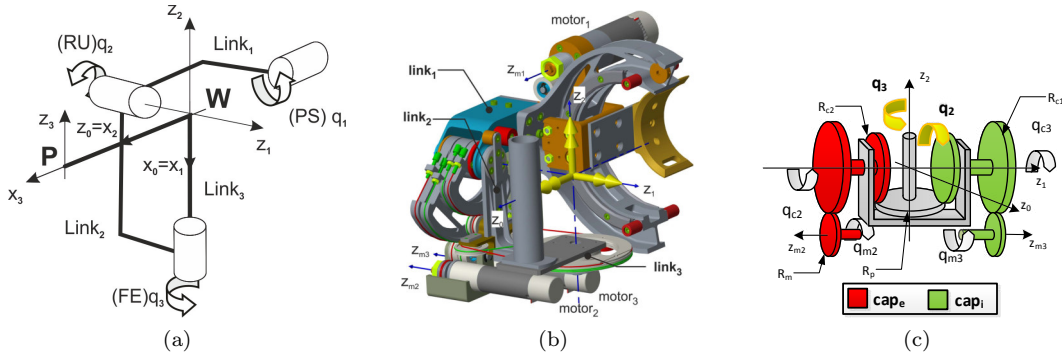

Figure 9: Wrist exoskeleton design: (a) kinematic scheme, (b) CAD model, (c) schematic representation of the FE/RU differential transmission.

4.2.2 Wrist Exoskeleton

For successful manipulation, the wrist movement is important, as it determines the orientation of the hand. Consequently, wrist motions are realized by an active wrist exoskeleton (Buongiorno et al., 2018). It covers the three rotational DoF of the wrist and provides respective torque feedback. The covered articulations are: forearm pronation/supination (PS), wrist flexion/extension (FE), and radial/ulnar deviation (RU).

The design is based on serial kinematics as shown in Fig. 9. Again, actuators are located remotely at the non-moving parts and forces are transmitted via metallic tendons to obtain good dynamic behaviors. Main design requirements concern the lightness of the device, its easiness to be worn, and the need to have an inwards open structure to avoid collision with other parts of the telepresence suit during bimanual teleoperation tasks. The PS joint has been designed to improve the wearability of the wrist device by using an open curvilinear rail and rolling slider solution. FE and RU motions are jointly transmitted by two parallel actuators using a differential transmission as shown in Section 4.2.2. A passive regulation of the handle position along the PS axis allows to adapt the last link length to the user's hand size. The total weight of the wrist exoskeleton is 2.9 kg. Fig. 10 shows the wrist exoskeleton worn by an operator.

4.2.3 Hand Exoskeleton

Finally, a lightweight, underactuated hand exoskeleton was designed for the teleoperation of grasping motions, providing independent finger motion tracking with force feedback. Each finger is connected to an

Figure 10: The wrist exoskeleton worn by an operator. Left: Isolated test. Right: Mounted on the arm exoskeleton.
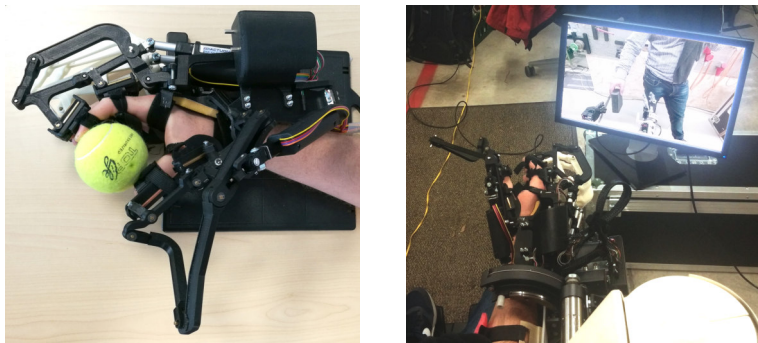

Figure 11: The hand exoskeleton worn by an operator.

individual underactuated parallel kinematic in two points which can adapt to the individual finger length as shown in Fig. 11. The hand exoskeleton uses two parameters to describe the pose of each of the four long fingers—index, middle finger, ring finger, and pinkie (Sarac et al., 2017). Those parameters do not directly map the fingers' joint positions. In addition, one parameter describes the spread between those four fingers. The thumb configuration is described with five parameters (Gabardi et al., 2018).

Force feedback is applied to each of the five fingers by an individual actuator. A total of five actuators is low in comparison to the DoF a human hand provides, but since a low device weight is an important design goal, a trade-off had to be found. Grasping teleoperation with force feedback from one actuator per finger showed promising results in previous experiments.

The five parallel kinematic chains are mounted on a ground link which is rigidly connected with the back of the user's hand, as shown in Fig. 11. The ground link shape is modeled in order to follow the shape of the back of a human hand while a layer of foam between the hand surface and the ground link improves the adaptability of the system to different hands and increases comfort. The hand exoskeleton has been realized with 3D printed parts, resulting in a low-cost and lightweight device which only weighs about 350 g. For the actuation, *Firgelli L16* linear motors with 50 mm stroke have been used. Moreover, the lack of magnetic hardware, the small case and the low weight of the actuators allow the motors to be placed close to each other to fit all the actuators on top of the hand. For sensing, potentiometers of the type *Bourns 3382G-1-103G* are utilized in addition to the potentiometers that are integrated in the actuators.

The utilized linear motors include a mechanical gearbox which prohibits backdriveability. However, backdriveability is fundamental to allow the operator to move its fingers. Consequently, force sensors are positioned between the ground link and the actuators. Those sensors measure the intentional forces applied by the user's fingers and a control algorithm is utilized to let the actuators follow the intended finger motion.
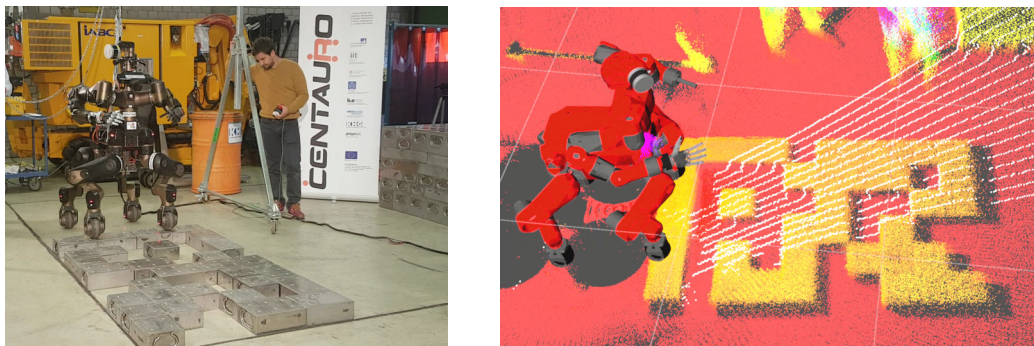
Figure 12: Centauro overcoming a step field: Scenario (left), localized robot and registered point cloud color coded by height (right).

5 Advanced Environment Perception

The chosen sensor setup generates data of several types which differs in the direct usability of the data. While some sensor data requires processing, other percepts, such as foot camera images, can be directly shown to the operators. The resulting representations are input to autonomous control functions or are visualized to the operators.

5.1 Ground Contact Estimation

When navigating in challenging terrain, it is helpful to detect if a foot has ground contact. This helps the operator to assess the robot stability since established ground contact might be hard to recognize from camera images in some situations. In addition, such ground contact estimation enables reliable semi-autonomous stepping with few additional required environment knowledge.

Inputs are joint torques of the legs which are provided by the actuators. Via forward dynamics, a 6D force vector is computed for each foot. If, after gravity compensation, the foot exerts a force onto the ground, contact is detected. This feature is e.g., used within the semi-autonomous stepping controller. Ground contact estimates are also visualized to the operators to facilitate robot stability assessment.

5.2 Laser-based 3D Mapping and Localization

The 3D rotating *Velodyne PUCK* laser scanner with spherical field-of-view provides about 300,000 range measurements per second with a maximum range of 100 m. It is rotated at 0.1 rotations per second, resulting in a dense omnidirectional 3D scan per halve rotation every 5 seconds. Slower rotation is possible if a higher angular resolution is desired. Laser range measurements are aggregated to a dense 3D map of the environment. We use our local multiresolution surfel grid approach for this (Droeschel et al., 2017). We compensate for sensor motion during acquisition by incorporating measurements of the IMU.

3D scans are aggregated in a robot-centric local multiresolution map by registering consecutive scans to a denser egocentric map. To allow for allocentric mapping of the environment, the resulting egocentric maps from different view poses form nodes in a pose graph. Graph edges connect these view poses and represent spatial constraints, which result from aligning these maps with each other. The global registration error is minimized using graph optimization. The resulting 3D map allows for localizing the robot in an allocentric frame which is used autonomous locomotion and manipulation panning but also to generate informative visualizations for the operators. Fig. 12 shows an example of a generated point cloud and a localized robot.

Figure 13: Turntable capture and scene synthesis. Left: Different drills on the turntable as captured by a DSLR camera. Right: Synthetic training scene generated by inserting new objects. The top image shows the resulting color image, the bottom shows generated ground truth for training the segmentation.

5.3 Object Segmentation

To enable autonomous or semi-autonomous manipulation of the workspace, the Centauro robot has to detect and determine the pose of useful objects in its environment, for example tools. For the purpose of object segmentation, we train a CNN-based model on a dataset of tools. The main sensor for perceiving the workspace is the chest-mounted *Kinect V2* sensor. This sensor provides registered RGB color and depth estimates, which is especially useful for pose estimation. A multi-frequency phase unwrapping algorithm improves the depth estimates (Lawin et al., 2016).

Our semantic segmentation pipeline directly produces pixel- (or point-)wise class labels. Utilizing the recent RefineNet (Lin et al., 2017) architecture, we create a representation of the input image with both highly semantic information and high spatial resolution. Based on this representation, a final convolutional layer with softmax transfer function outputs class probabilities.

Deep learning methods require large amounts of training data. We address this problem by generating new training scenes using data captured from a turntable setup. Automatically extracted object segments are inserted into pre-captured scenes, as shown in Fig. 13. For details on the capturing and scene synthesis pipeline, we refer to (Schwarz et al., 2018).

5.4 Pose Estimation

To facilitate robust autonomous grasping of objects, we need their 6D pose. To this end, we developed a 5D pose predictor network for efficient pose estimation on a rough scale. Again, RGB-D data from the *Kinect V2* serves as input.

For efficient pose prediction, we augmented the semantic segmentation pipeline (Section 5.3) with an additional CNN to estimate the 5D object pose (rotational, and x and y of translation) from the RGB crops from the scene. Based on the bounding boxes of detected object contours, those crops are extracted. Pixels classified as non-object are pushed towards red to encode the segmentation results (see Fig. 14). By providing the network this representation, it is able to focus on the specific object of which the pose should be determined. The pretrained RefineNet network from Section 5.3 is used to extract features. We extend the
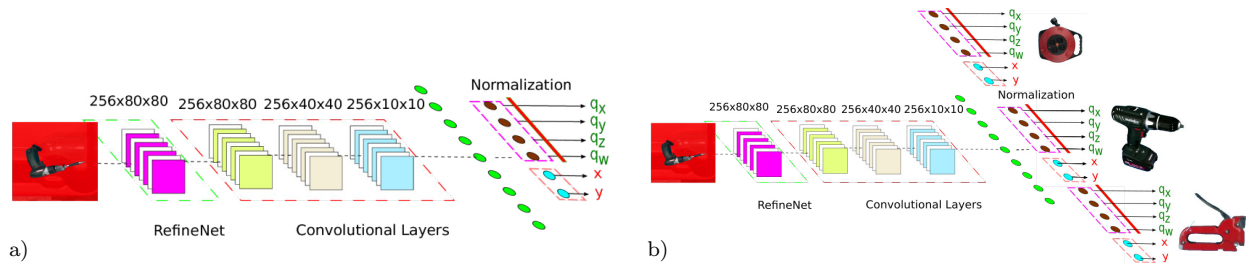
Figure 14: Pose estimation network architecture. a) Single-block output variant; b) Multi-block output.

data acquisition pipeline described in (Schwarz et al., 2018) to record turntable poses automatically and fuse captures with different object poses or different objects (see Fig. 13). This is used to generate the ground truth poses for network training.

We developed two different types of CNN architectures shown in Fig. 14. The single-block output variant predicts six values (rotation represented as a unit quaternion and x and y of translation), whereas the multi-block output variant predicts these for each object category. Our evaluation showed that the single-block variant performed slightly better in the presence of occlusion, and was thus chosen for use in the Centauro system. The predicted 5D pose can be projected into 6D using the mean depth measurement d in a window around the object center. The 6D object pose is e.g., input to the autonomous manipulation pipeline.

6 Operator Interfaces

Although the considered environments are too dangerous for humans to work in, the human capabilities of situation assessment, mission planning, and task experience are key to a successful mission, e.g., when maintaining a moon or planet base. The operator interfaces enable the operators to transfer these desired capabilities into the scene by giving them an awareness of the situation and enabling them to control the robot. Both require a communication infrastructure, since visual contact is not available.

Regarding locomotion and manipulation control, we aim at enabling the operators to solve as many previously known and unknown typical construction, maintenance, and exploration tasks as possible. Hence, we propose a set of control functions with advantages in different task classes to perform successful teleoperation. A key requirement is to utilize the whole range of the robot's kinematic capabilities while keeping the control itself intuitive and flexible to adapt to unforeseen situations. Different degrees of autonomy are used to fulfill this requirement. Parts of the operator interfaces have already been described in (Klamt et al., 2018).

6.1 Communication

We use a wired Ethernet connection or a standard IEEE 802.11ac 5 GHz WiFi link for data transmission between the operator station and the robot, Depending on the type of data, different protocols are used. The low-latency control and feedback messages for the exoskeleton use raw UDP packets. All other communication is realized via *ROS* topics. These are either directly accessed using *ROS* network transparency, or encoded with FEC using the `nimbro_network` developed for Momaro (Schwarz et al., 2017) to increase robustness. This is especially developed to handle communication with low-bandwidth, high latency, and package loss, which is often the case for space missions. For extending the coverage, Centauro can carry a WiFi repeater and drop it at an appropriate location.
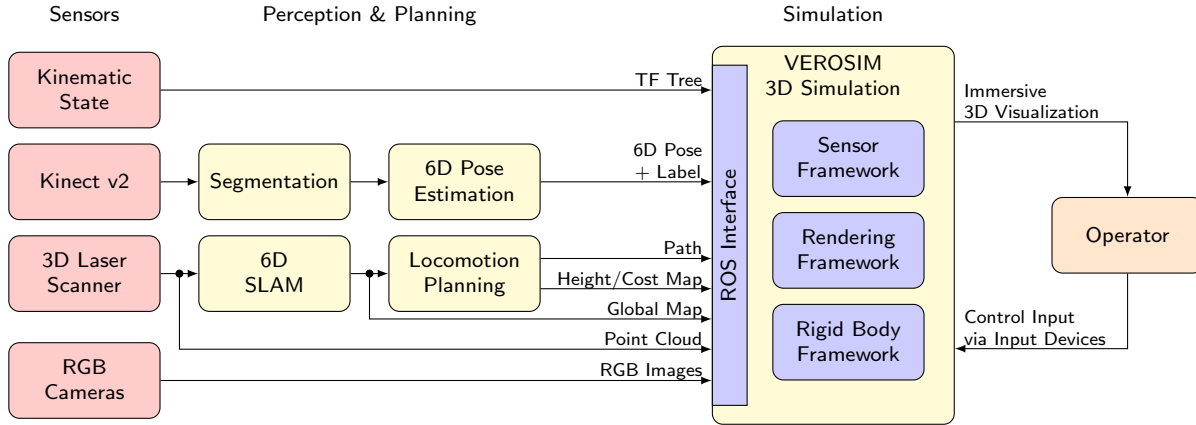
Figure 15: Pipeline for 3D simulation. Sensors are colored red and pipeline components yellow.

6.2 Situation Awareness

A simulation-based approach is used to generate user interfaces which can be displayed in an *HTC Vive* head-mounted display or on arbitrary regular monitors. Thus, we aim at providing an intuitive, semantically enriched visualization of data for the teleoperation of real robots in dynamic scenarios. Based on a digital twin of the real system, we use 3D simulation in-the-loop to visualize the current state of the robot in its environment and add semantic information for the operators, to intuitively interact with the real robotic system. This involves

- the (planar) visualization of incoming data, raw and preprocessed,
- the projection of data onto the place of occurrence,
- the stereoscopic rendering and thus immersion of the operator, and
- a modular combination and selection of all modalities suited for the given application or situation.

For a holistic incorporation of central modules, as well as interfaces to all system soft- and hardware components, we use *VEROSIM* (Virtual Environments and Robotics Simulation System) with its integrated rendering, sensor, and rigid body frameworks as well as its *ROS* interface (see Fig. 15).

Based on the raw sensor data (robot pose, point clouds and images), the digital twin of the robot, as well as perceived environment information can be visualized in simulation (see Fig. 16 l.,c.). Besides the possibility to visualize raw data streams, the 3D simulation can also use preprocessed data from all software modules
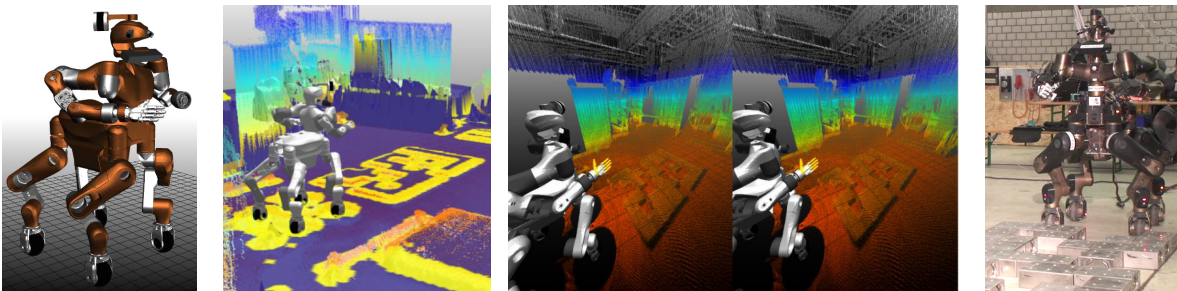

Figure 16: l.: *VEROSIM* provides a digital twin of the Centauro robot encompassing all links, joints, sensors etc., c.: third person view (standard and stereoscopic) on the current scene with a rigid body height map overlaid by a cost map accompanied by color-coded point cloud data, r.: image of the visualized scene, Centauro standing in front of a stepping field.

Figure 17: Overview of visualization possibilities: t.l.: head-up display visualization of internal parameters, b.l.: projection of semantic object information via billboard visualizations, c.: first person operator using the exoskeleton with direct camera view and point cloud visualizations, t.r.: stereoscopic and b.r.: third person view on the scene during a manipulation task.

to generate dynamic environments based on the robot's perception. Based on generated point clouds, rigid body height maps can be created and cost map visualizations can be overlaid (see Fig. 16 c.). Relying on the semantic detection of objects with a given name and pose, the simulation can add the dimension of space to the data by projecting it onto the place of occurrence. Consequently, a template-based model insertion can be used to insert a 3D model of the recognized object into the 3D scene which overcomes point cloud gaps (see Fig. 17 b.l.). Head-up display visualization of data can help each operator to create a unique, personalized and individually optimized user interface (see Fig. 17, Cichon and Roßmann, 2017b). We note that in our system, the environment is rendered on the operator side and thus head movement of the operator can be tracked with very low latency and immediately results in a changed viewpoint. This way, uncomfortable lags are avoided even with high-latency data links to the robot.

Stereoscopic rendering encompasses all of the aforementioned aspects. Incorporating the *SteamVR* API into *VEROSIM* enables the utilization of software modalities such as using stereoscopic headsets (see Fig. 16 t.r.). This enables an operator wearing a head-mounted display to see through the robot's eyes, increasing the immersion and the embodiment in simulation. In addition, it is also possible to optimize the operator's view by either physically walking around in the scene or by moving the camera pose through the 3D scene via computer mouse or to predefined poses in the simulator (see Fig. 17 r.). This leads to a wide set of opportunities to individually change and reposition the view on the scene for each task and to generally mediate and support robotic teleoperation via 3D simulation (see Fig. 17 c.). Further details can be found in Cichon and Roßmann (2017a).

Although this optimized user interface in terms of visualization is the main focus in this contribution, the use of a 3D simulation backend comes along with additional opportunities. During the development process of such complex systems, the simulator is used for most of the system components. The implemented *ROS* interface allows for the combination of internal and external frameworks and the use of the simulation framework as the central integration tool for all system components. We e.g., employed VEROSIM to test the teleoperation with the telepresence suit. The operator was able to grasp simulated objects and receive force feedback (Cichon et al., 2016). Moreover, the digital twin can be used to test actions before they are executed by the real system and to predict action results. Figure 18 shows how VEROSIM is used to simulate a maintenance mission on the Mars with Centauro. Additional optimization, prediction and logging can be accessed during or after operation to analyze the overall system performance (Atorf et al., 2014, 2015).

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

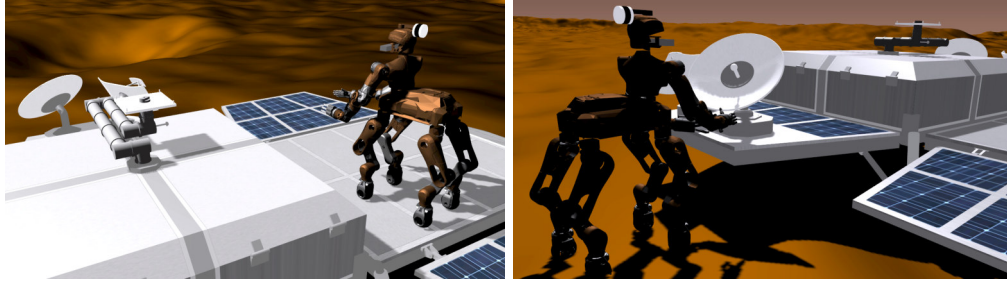- can control the robot via control interfaces other than the telepresence suit,

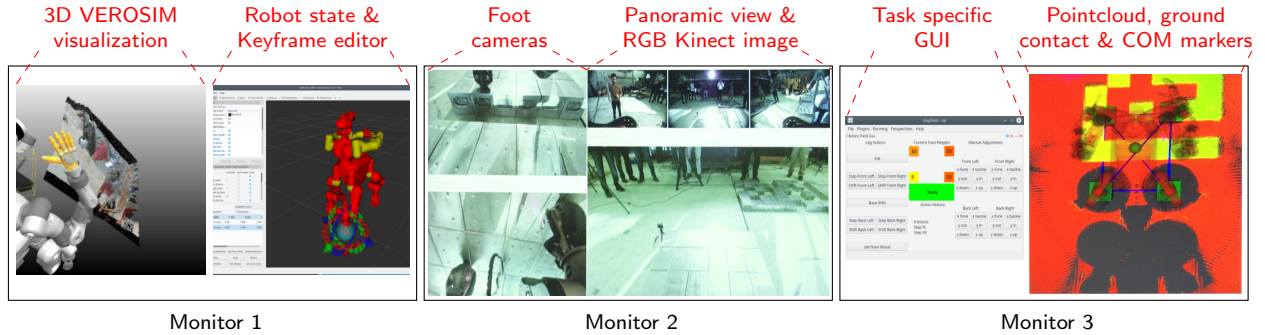Figure 18: Centauro in a simulated Mars mission.

Figure 19: Environment and robot state visualization for the support operators.

- 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. We provide these by displaying processed and unprocessed data from several sensors on multiple monitors, as shown in Fig. 19. A panoramic view from the robot head perspective is helpful for general scene understanding. Two RGB cameras under the robot base provide views which are arranged to enable a detailed assessment for the terrain under the robot base facilitating a save stepping locomotion operation (Fig. 19). Moreover, results of the ground contact estimation are visualized by a colored marker for each foot. 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.

6.3 Locomotion Control

The Centauro lower body provides a wide range of locomotion capabilities which require suitable interfaces to be controlled efficiently. Interfaces for omnidirectional driving control are a joystick and a pedal controller (Section 6.3.1). Leg movements can be controlled by a keyframe editor (Section 6.3.2). A higher degree of autonomy is reached by utilizing a semi-autonomous stepping controller (Section 6.3.3). 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 Section 7.

6.3.1 4D Joystick & 3D Pedal Controller

A *Logitech Extreme 3D Pro* joystick with four axes is employed to control omnidirectional driving. The velocity components of the robot base v_x, v_y and v_θ are mapped to the three corresponding joystick axes.
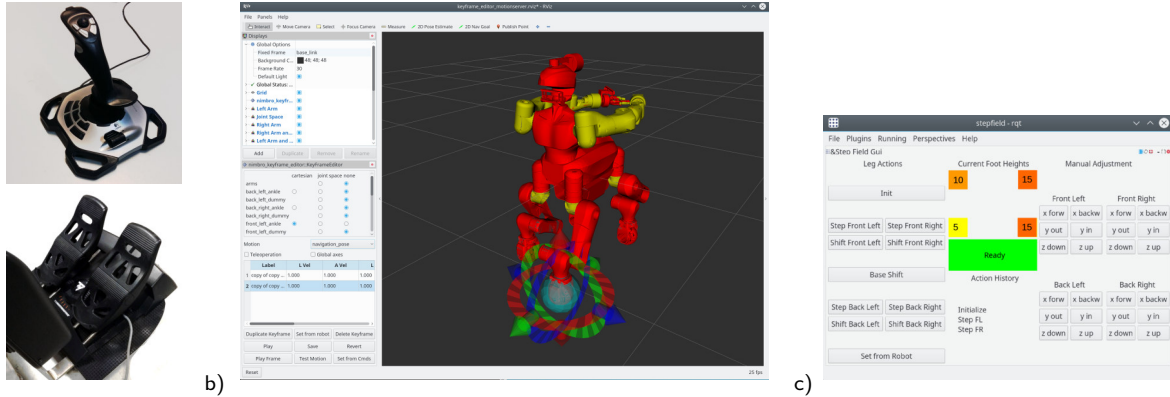
Figure 20: Locomotion control interfaces: a) 4D joystick and 3D pedal controller, b) keyframe editor, c) semi-autonomous stepping controller GUI.

Foot specific velocities and orientations are derived from this robot base velocity as described in Section 7.4. All three velocity components are jointly scaled by the joystick throttle controller.

The same three robot base velocity components are also mapped to a 3-axis *Thrustmaster TFRP Rudder* pedal controller. In addition to individual throttles for each of the two pedals, it also allows for changing the pedal's longitudinal position relative to each other which results in a third DoF. The pedal controller is less intuitive than the joystick but can be used by the teleoperator in the telepresence suit. Both devices are shown in Fig. 20a.

6.3.2 Keyframe Editor

Leg motions can be controlled by a keyframe editor (Schwarz et al., 2016a). 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. 20b). A numerical configuration is also possible. Finally, robot motions are generated by interpolating between given keyframes.

6.3.3 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 (see Section 5.1). Hence, the robot automatically adapts to the ground structure. Motions are represented as sequences of keyframes, as described for Section 6.3.2, and can be executed by the motion player.

The controller is operated through a GUI which provides buttons to trigger the described motions for an individual foot (Fig. 20 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 repetitive motion sequences.
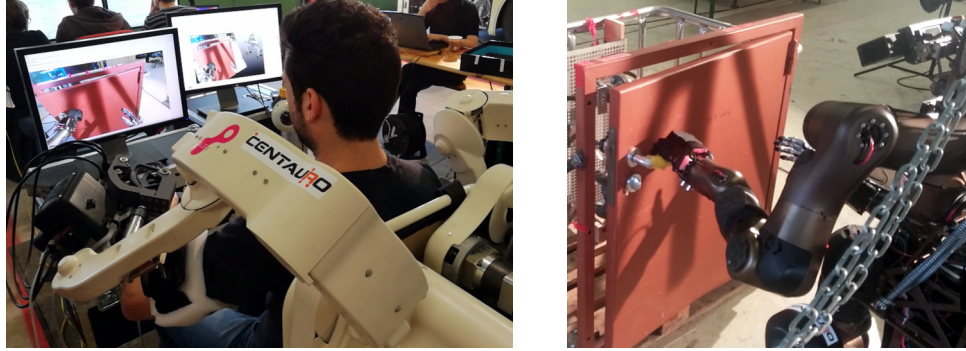
Figure 21: An operator opening a door teleoperating Centauro with the upper-body exoskeleton.

6.3.4 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 (Schwarz et al., 2017). 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.

6.4 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 6.3.2. Furthermore, manipulation can be controlled via the upper-body exoskeleton which mimics the operator's behavior and provides force feedback (Section 6.4.1). Moreover, precise control of the wrist pose is provided by a 6D input device (Section 6.4.2). Finally, we propose an autonomous grasping functionality which is described in Section 8.

6.4.1 Telemanipulation by the Upper-body Exoskeleton

The full-body telepresence station can be used to control the robot through an advanced teleoperation system with force feedback. By means of the bilateral and full upper limb exoskeleton, the operator directly controls the pose and the applied forces of the upper limbs of the robot. Different teleoperation control architectures have been implemented for the arm-wrist segments of the exoskeleton and for the hand grasping.

Regarding teleoperation of the proximal segments of the upper limb, the operator's arm and wrist pose are measured by the correspondent parts of the exoskeleton and a 6D end-effector pose is computed. This pose is then transferred to the robot and applied to the arm using inverse kinematics. A direct mapping of joint configurations from the teleoperator to the robot is not helpful since their kinematic structures are not identical. Force feedback is generated by transferring measured 6D force-torque vectors from the force-torque sensor between right arm and Schunk hand to the exoskeleton and mapping them to joint torques using inverse dynamics. Fig. 21 shows an operator using the upper-body exoskeleton to open a door.

For the teleoperation of grasping motions, the right-hand exoskeleton is connected with the anthropomorphic Schunk hand mounted at the right robot arm. Again, a direct joint mapping of control commands and force feedback is not possible due to differences in the kinematic concept, variable operator hand sizes and underactuation for both the hand exoskeleton and the Schunk hand. To transfer finger and hand motions and applied forces from the operator's hand to the robot hand, we utilize the local admittance control of the hand exoskeleton and the local impedance control embedded in the Schunk hand. The hand exoskeleton
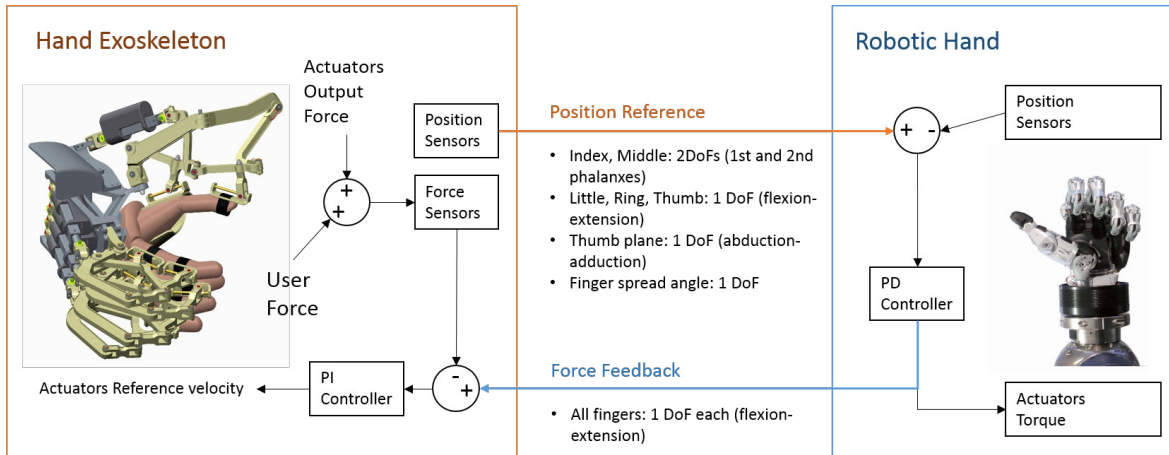
Figure 22: Teleoperation scheme between the Hand Exoskeleton and the Schunk robotic hand.

Figure 23: 6D input device (l.) and corresponding GUI (r.) for dexterous wrist control.

estimates angular position references of the operator's hand. Those are sent to the proportional-derivative (PD) position control system of the Schunk hand. The embedded PD control converts position references into actuator torques proportional to the position error. Hence, once in contact with an object, the operator can increase grasping forces by further closing his or her fingers.

The mapping of positions and forces differs between fingers. Thumb, ring finger and pinkie are operated on the basis of a single DoF each on the operator and robot side, estimating the overall closing/opening of the finger. Index and middle finger are position controlled by the operator using two DoF, matching the rotation of the first and second phalanxes of each finger. The Schunk hand provides this additional DoF for those two fingers which results in a higher grasping precision. In addition, the spread between the four long fingers and the rotation of the thumb's opening/closing plane (abduction/adduction) is controlled.

To provide force feedback, torques applied by the Schunk hand are sent to the local admittance control of the hand exoskeleton and applied to the operator's hand as force feedback. The mapping in this direction is different since the exoskeleton provides less actuators than sensors. Each finger receives a single DoF feedback force which is applied through the adaptive and underactuated mechanism of the hand exoskeleton. For the finger spread, force feedback is not applicable since no correspondent actuator exists in the hand exoskeleton. Fig. 22 visualizes the control architecture between hand exoskeleton and Schunk hand.

To control the 1-DoF SoftHand at the left robot arm, the left side of the exoskeleton is equipped with a lever-shaped 1-DoF grasping controller instead of a hand exoskeleton. Its position is mapped to the SoftHand actuator and the measured force at the robot hand is transferred to the controller as force feedback.

6.4.2 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. 23). The developed interface establishes the connection between the device and the motion player mentioned in Section 6.3.4. The 6D *SpacePilot* joystick movement from the operator 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 can be used to easily adjust the following control parameters: *End-effector* (a wrist for arm control or an ankle for leg control), *Reference frame* (end-effector frame, robot base frame, or a custom frame), *Enabled axes* (each translational and rotational axis can be enabled/disabled so the user input on this axis is considered/ignored), and *End-effector speed*. All control parameter can also be changed using buttons on the input device. This interface is well suited whenever precise motions along certain axes are required. This is the case for manipulation tasks where e.g., an arm needs to be moved along a plane surface or an object needs to be turned around a specified axis.

Note that switching control authority from the exoskeleton to the 6D input device is easily possible. The other direction induces much harder problems, since end-effector location is always directly (and absolutely) coupled to hand position in the exoskeleton.

7   Hybrid Driving-Stepping Locomotion Planning

Autonomous locomotion planning and execution is a promising approach to increase locomotion speed and safety, to lower the operator's cognitive load, to execute locomotion under bad data transmission between the operators and the robot, and to keep the flexibility to control the robot in unknown tasks and unforeseen situations. The only required operator input is a desired robot goal state. The locomotion planning pipeline is visualized in Fig. 24. Laser scanner point clouds (Section 5.2) are processed to cost maps that represent the environment (Section 7.1). A search-based planning approach uses this environment representation to generate plans (Section 7.2). Due to the unique hardware design and the many DoF of the robot platform, this planning approach is novel since it merges planning approaches for both driving-based and stepping-based locomotion and is capable of handling the respective complexity. To handle planning queries for large environments, the approach is extended to plan on multiple levels of abstraction (Section 7.3). Finally, a controller executes these paths and outputs robot control commands (Section 7.4).
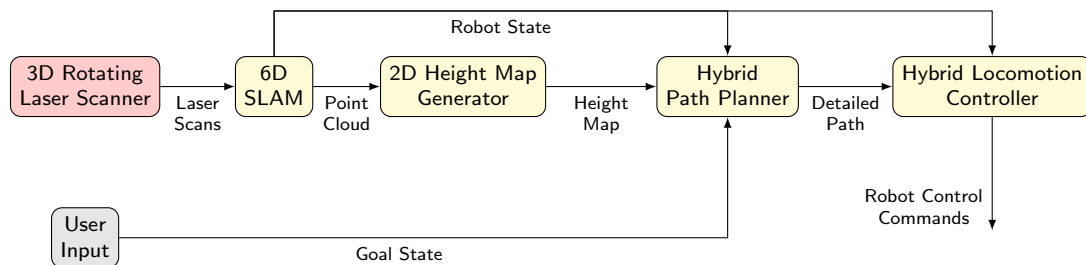

Figure 24: Overview of the pipeline for locomotion planning. Sensors are colored red, pipeline components yellow, and other inputs are colored gray.

7.1   Environment Representation

A 2D height map is generated from the registered point clouds and serves as the environment representation. Height values are processed to foot and body costs. Foot costs describe the cost to place an individual foot at a given position in the map. They include information about the terrain surface and obstacles in the vicinity. Body costs describe the costs to place the robot base in a given configuration in the map. They include information about obstacles and the terrain slope under the robot. Foot costs of all four feet and body costs
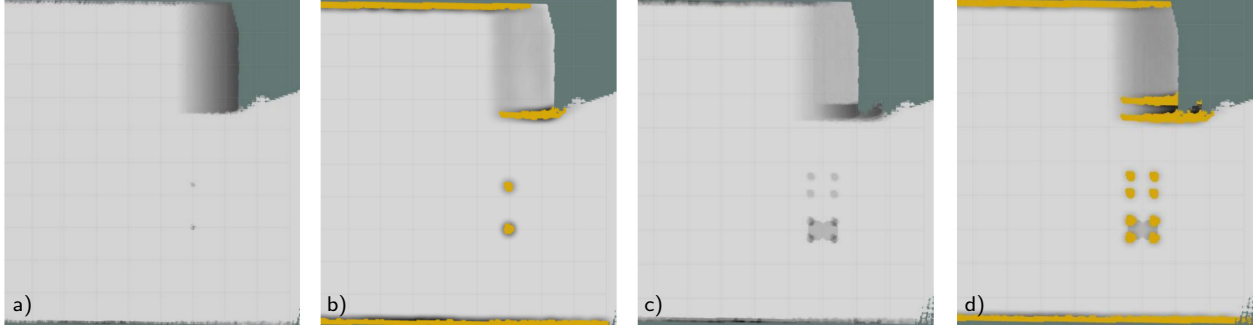
Figure 25: Environment representation: The map shows two walls, a ramp and two poles of different height. a) Heights, b) Foot costs, c) Body costs, d) Pose costs.
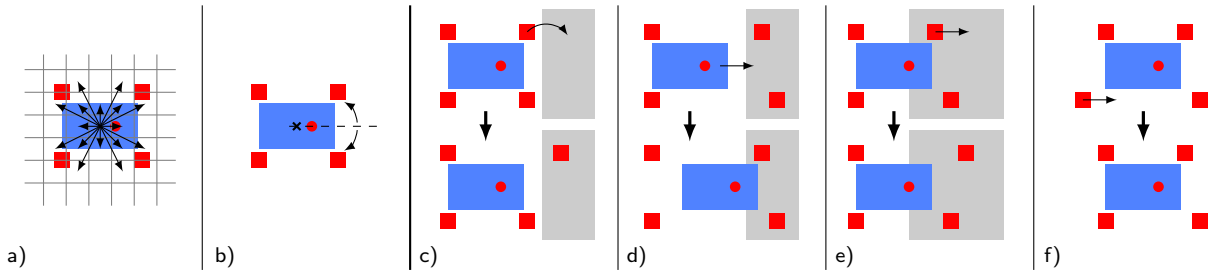

Figure 26: Neighbor states can be reached by driving or stepping related motions: a) Omnidirectional driving with fixed orientation, b) Turning on the spot to the next discrete orientation, c) Abstract step, d) Longitudinal base shift, e) Shift a single foot forward, f) Shift a foot back to its neutral position.

are combined to pose costs which describe the costs to place the whole robot in a given configuration on the map (see Fig. 25).

7.2 Search-based Planning

Path planning is done by a search-based approach on these pose costs. The used algorithm is Anytime Repairing A* (Likhachev et al., 2003). Neighbor states are generated online during the planning. They include omnidirectional driving (see Fig. 26 l.) and stepping motions (see Fig. 26 r.). Since driving shall be the preferred locomotion mode, stepping motions are only considered if several criteria are fulfilled. During path search, steps are represented as abstract maneuvers—the direct transition from a pre-stepping pose to an post-stepping pose. Robot stability and detailed motion sequences are explicitly not considered in this planning layer.

The resulting path is expanded to a motion sequence which can be executed by the robot. During this path expansion, abstract steps are transformed into stable motion sequences. Stable stepping poses are established through roll motions, foot shifts and longitudinal base shifts. Stability computation is limited to static stability since motion execution is sufficiently slow and thus, dynamic effects can be neglected. In addition, leg length information is generated for each pose in the resulting path. Further details about the approach can be found in (Klamt and Behnke, 2017).

7.3 Planning on Multiple Levels of Abstraction

The search-based planning approach is extended to plan on multiple levels of abstraction, as illustrated in Fig. 27 and Fig. 28, which allows for planning for significantly longer path queries while the result quality
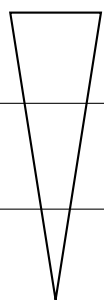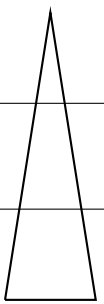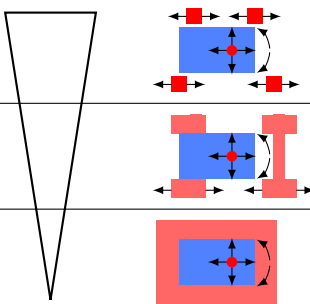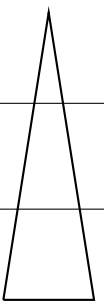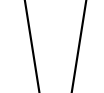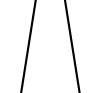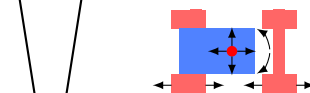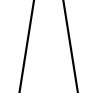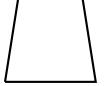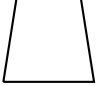
Level	Map Resolution	Map Features	Robot Representation	Action Semantics
1	• 2.5 cm • 64 orient.	• Height		• Individual Foot Actions
2	• 5.0 cm • 32 orient.	• Height • Height Difference		• Foot Pair Actions
3	• 10 cm • 16 orient.	• Height • Height Difference • Terrain Class		• Whole Robot Actions

Figure 27: The planning representation is split into three levels of abstraction. Coarser representations are enriched by additional semantics to compensate the loss of information, caused by abstraction.
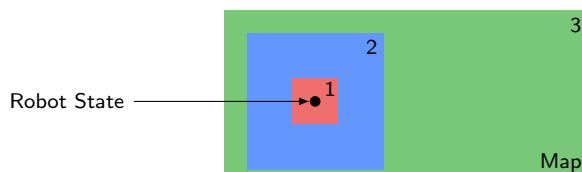

Figure 28: Representation level positioning. A fine planning representation is only provided in the vicinity of the robot. With increasing distance from the robot, the representation becomes more abstract.

stays comparable. A detailed planning representation is only utilized in the vicinity of the robot. This representation is called *Level 1* representation and is described in Section 7.2. With increasing distance from the robot, the planning representation becomes more abstract. This is achieved by representing the environment and possible actions in a coarser resolution and by using a robot representation with less DoF. The loss of information that comes along with such abstraction is compensated by enriching those representations with additional semantics. The environment is represented with additional features and robot action generation accesses these features. All three representation levels are unified in a single planner which is able to handle transitions between these levels. Moreover, during path execution, path segments can be refined from a coarse representation to a finer representation, which decreases the amount of required re-planning steps. Finally, we utilized a heuristic for the planner which is based on precomputed costs in the *Level 3* representation. Further detail can be found in (Klamt and Behnke, 2018).

7.4 Path Execution

Path segments that are represented in *Level 1* and which are expanded to detailed motion sequences are executed by a driving-stepping controller. This is split into the control of omnidirectional driving and the control of leg movements.

To control path segments that require omnidirectional driving, a three-dimensional (x, y, θ) B-Spline (de Boor, 1978) is laid through the next five robot poses and a pose on this B-Spline is chosen as the controller set value $\mathbf{r}_{\mathrm{sv}} = (r_{x,\mathrm{sv}}, r_{y,\mathrm{sv}}, r_{\theta,\mathrm{sv}})$. Extracting this set value from the B-Spline in some distance from the current robot pose $\mathbf{r} = (r_x, r_y, r_\theta)$ leads to a smoother controller output. A distance of half the B-Spline length yields the desired behavior. The velocity command $\mathbf{v} = (v_x, v_y, v_\theta)$ is computed with

$$\mathbf{v} = (\mathbf{r}_{\mathrm{sv}} - \mathbf{r}) \times k, \tag{1}$$

where k is chosen in a way that the linear velocity component norm $\left\| (v_x, v_y)^{\mathrm{T}} \right\|$ is equal to v_{des}, the desired velocity. v_{des} is 0.1 m/s close to obstacles and 0.25 m/s otherwise. With the given relative foot position $\mathbf{f}^{(i)}$,

the individual velocity command

$$
\begin{pmatrix} v_x^{(i)} \\ v_y^{(i)} \\ v_z^{(i)} \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ v_\theta \end{pmatrix} \times \mathbf{f}^{(i)} \tag{2}
$$

can be computed for each of the four wheels. Before moving with the linear velocity $||(v_x^{(i)}, v_y^{(i)})^\mathrm{T}||$ in the desired direction, each wheel needs to rotate to the respective yaw angle $\alpha^{(i)} = \mathrm{atan2}(v_y^{(i)}, v_x^{(i)})$. While driving, the robot continuously adjusts the ankle orientations.

For path segments that require leg movement, actions are expanded to sequences of linear end-effector movements. It is distinguished between movements with wheel rotation (e.g., shift an individual foot on the ground relative to the robot base) and actions without wheel rotation (e.g., longitudinal robot base shift). The transformation from the Cartesian space to joint space is described in Section 6.3.4.

8 Autonomous Manipulation

Autonomous manipulation capabilities can help reducing the load on the main operator and the dependency on a good data connection. For example, repetitive motions such as grasping a tool, opening a door, and placing an object may be automated, allowing the operator to focus on the current higher-level task rather than direct teleoperation of the manipulation actions. To realize autonomous manipulation, several components were integrated (see Fig. 29). Input to the system are RGB-D measurements from the *Kinect V2* sensor and range measurements from the rotating laser scanner at the robot head. In addition to the perception components described in Sections 5.3 and 5.4, we will discuss the grasp generation and trajectory optimization modules here. The pipeline outputs a joint trajectory describing the grasping motion which is executed by low-level control software. A bi-manual extension of this pipeline has been proposed in (Pavlichenko et al., 2018).

8.1 Grasp Planning

Objects belonging to a category often exhibit several similarities in their extrinsic geometry. Based on this observation, we transfer grasping skills from known instances to novel instances belonging to the same category such as drills, hammers, or screwdrivers. Our approach has two stages: a learning stage and an inference stage.

During learning, we build a category-specific linear model of the deformations that a category of objects can undergo. For that, we firstly define a single canonical instance of the category, and then we calculate the deformation fields relating the canonical instance to all other instances of the category using Coherent Point Drift (CPD) (Myronenko and Song, 2010). Next, we find a linear subspace of these deformation fields, which defines the deformation model for the category (Fig. 30).
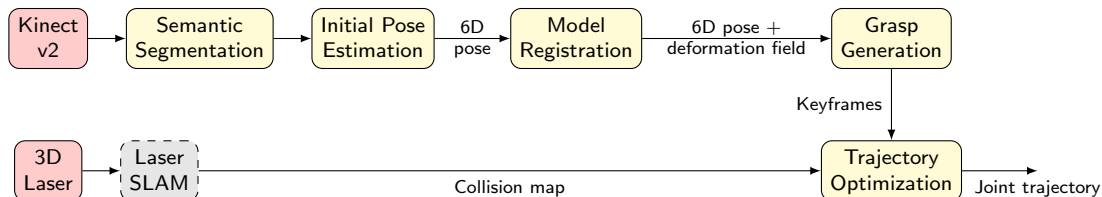

Figure 29: Pipeline for autonomous manipulation. Sensors are colored red, pipeline components yellow, and external modules from other workpackages are colored gray.
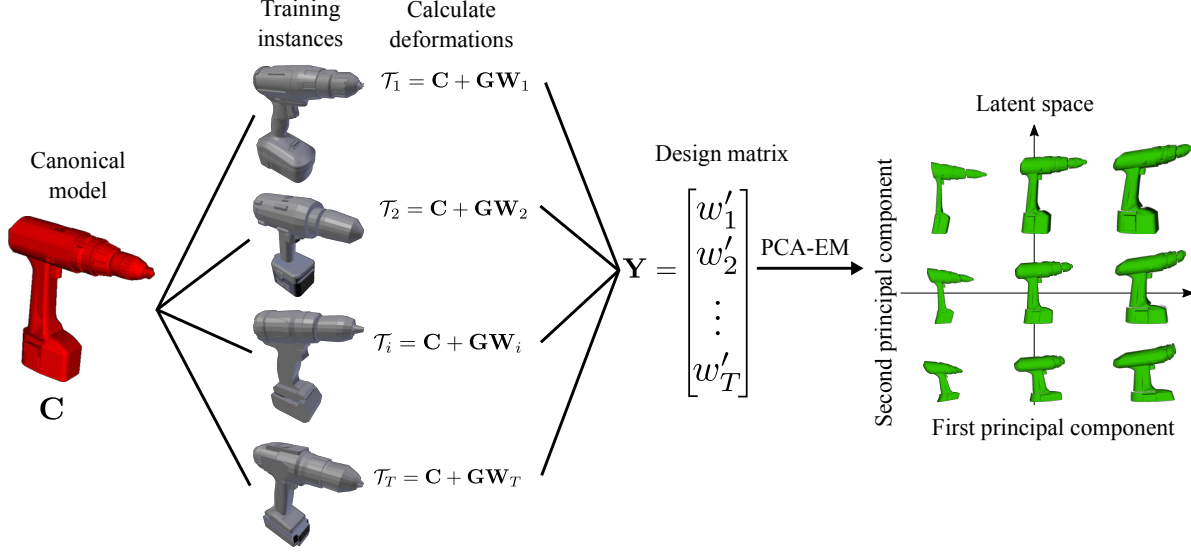
Figure 30: Training phase. Deformations between the canonical model and each instance are calculated by using CPD. The deformations are then assembled into the design matrix $\mathbf{Y}$. Finally, the principal components (latent space) are found by using PCA-EM.

A category is defined as a group of objects with similar extrinsic shape. From the set of objects, we select one to be the canonical model of the class. For each training sample, we find the deformation field that the canonical model has to undergo to transform itself into the training sample. As explained in (Rodriguez et al., 2018), this transformation can be expressed as:

$$\mathcal{T}_i = \mathbf{C} + \mathbf{G}\mathbf{W}_i, \tag{3}$$

where $\mathbf{C}$ refers to the canonical model, $\mathbf{G}$ is a Gaussian kernel matrix, and $\mathbf{W}_i$ is a matrix of kernel weights.

Because $\mathbf{C}$ and $\mathbf{G}$ remain constant across all training samples, the uniqueness of the deformation field is captured only by $\mathbf{W}_i$. Each of the $\mathbf{W}_i$ matrices contains the same number of elements. This allows us to assemble a $\mathbf{Y}$ design matrix containing all deformation fields as column vectors. Finally, we apply Principle Component Analysis Expectation Maximization (PCA-EM) on the design matrix $\mathbf{Y}$ to find a lower-dimensional manifold of deformation fields for this category.

In the inference stage, we formulate the problem as: given a newly observed instance, search this subspace of deformation fields to find the deformation field which best relates the canonical instance to the novel one. Associated control poses used for grasping defined for the canonical model are also transformed to the observed instance and used for the final grasping motion.

The transformation between a novel instance and the canonical model is defined by its latent vector plus an additional rigid transformation. The function of the rigid transformation is to reduce the impact of minor misalignments in the pose between the canonical shape and the target shape. We use gradient descent to simultaneously optimize for pose and shape. In general, we aim for an aligned dense deformation field that when exerted to the canonical shape $\mathbf{C}$ minimizes the following energy function:

$$E(\mathbf{x}, \boldsymbol{\theta}) = \sum_{m=1}^{M} \sum_{n=1}^{N} P \left\| \mathbf{O}_n - \Theta(\mathcal{T}_m(\mathbf{C}_m, \mathcal{W}(\mathbf{x})_m), \boldsymbol{\theta}) \right\|^2, \tag{4}$$

where M is the number of points of the canonical model, N is the number of points of the observed instance $\mathbf{O}$, $\mathbf{x}$ is the latent vector, Θ is a function that applies a rigid transformation with parameters θ, and P
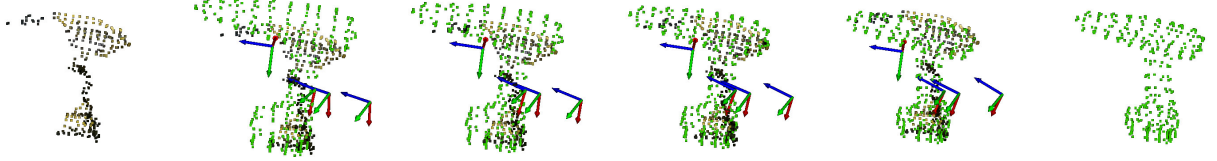
Figure 31: Transferring grasping knowledge to the presented novel instance. The input point cloud is at the leftmost while the inferred shape is at the rightmost.

represents the probability or importance weights between points expressed as:

$$P = \frac{e^{\frac{1}{2\sigma^2}||\mathbf{O}_n - \Theta(\mathcal{T}(\mathbf{C}_m, \mathcal{W}(\mathbf{x})_m), \boldsymbol{\theta})||^2}}{\sum_{k=1}^{M} e^{\frac{1}{2\sigma^2}||\mathbf{O}_n - \Theta(\mathcal{T}_k(\mathbf{C}_k, \mathcal{W}(\mathbf{x})_k), \boldsymbol{\theta})||^2}}. \tag{5}$$

A grasping action is composed of a set of parametrized motion primitives. Poses expressed in the same coordinate system of the shape of the object serve as the parameters of these motion primitives. These poses are defined only for the canonical model. For novel instances, the poses are calculated by warping the poses of the canonical model to the novel instance. We orthonormalize the warped poses because the warping process can violate the orthogonality of the orientation. Fig. 31 shows how the canonical model of a *Drill* category is warped to fit to the observed point cloud (leftmost), the warped grasping poses are also shown. For a complete analysis and discussion about this method, please refer to (Rodriguez et al., 2018) and (Rodriguez and Behnke, 2018).

The output of the grasp generation pipeline is end-effector poses on the object. These are converted into joint-space trajectories using inverse kinematics and our keyframe-based interpolation system (see Section 6.3.4).

8.2 Trajectory Optimization

In an unstructured environment, there may be obstacles which obstruct a direct way to the final pose, which would render the output of the keyframe interpolation step unusable. An additional optimization step can repair the trajectory. Moreover, by performing optimization of trajectory duration and actuator loads, it is possible to decrease the power consumption, which is of high importance during extended missions. In order to reduce the time of the task completion, it is necessary to perform the planning fast. In this subsection, we describe our approach to trajectory optimization which satisfies these criteria.

Our approach (Pavlichenko and Behnke, 2017) extends Stochastic Trajectory Optimization for Motion Planning (STOMP) (Kalakrishnan et al., 2011). An initial trajectory Θ is the input to this method. This trajectory may be very naïve, for example a straight interpolation between the start and the goal. The method outputs a trajectory, optimized with respect to a cost function. Both trajectories are represented as a set of N keyframes $\boldsymbol{\theta}_i \in \mathbb{R}^J$ in joint space. The initial and goal configurations, as well as the number of keyframes N are fixed during the optimization. In order to gradually minimize the cost, the optimization is performed in an iterative manner. The method reliably finds a feasible solution despite the fact that the initial trajectory is far from a valid one.

For collision avoidance, we consider the robot and the unstructured environment. We assume that the robot base does not move during the trajectory execution and that the environment is static. Two signed Euclidean Distance Transforms (EDT) are used for static objects. One of them is used to represent the environment and is precomputed before each optimization task. The other EDT represents the static part of the robot and is computed only if the robot base moves. The moving parts of the robot are approximated by a set of spheres, which allows for performing fast collision checking against EDTs.
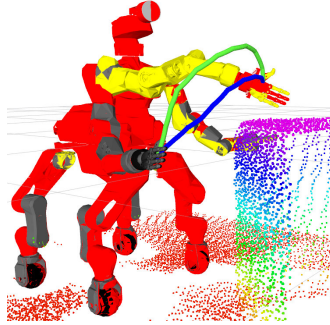
Figure 32: Two qualitatively different trajectories generated by our trajectory optimization: priority on obstacle avoidance (green) and priority on minimizing trajectory duration (blue).

In contrast to the original STOMP, our cost function is defined as a sum of costs of *transitions* between the consequent keyframes instead of the keyframes themselves. The cost estimated by each component is normalized to be within a $[0, 1]$ interval. This allows to attach a weight $\lambda_j \in [0, 1]$ to each cost component $q_j(.,.)$ in order to introduce a prioritized optimization. An example of two qualitatively different trajectories obtained with two different values of weight λ_{obst} for obstacle costs is shown in Fig. 32.

In order to accelerate motion planning, the optimization is split into two phases. During the first phase, we use a simplified cost function, which only includes obstacle, joint limit, and constraint costs. Optimization with the simplified cost function continues until a valid solution is found. In the second phase, we use the original cost function. This approach allows reducing the overall optimization time by eliminating duration and torque costs as long as no valid solution is found due to obstacles. From our observations, the two-phased approach in most cases did not yield qualitatively different trajectories compared to direct full optimization.

9 Evaluation

We evaluated the CENTAURO system with dedicated tests at facilities of the Kerntechnische Hilfsdienst GmbH in Karlsruhe, Germany (KHG). We note that our evaluation intentionally focuses on holistic, system-level testing of the CENTAURO system and thus aims to provide insight into the current state-of-the-art in larger-scale robotic problems. The chosen evaluation tasks are typical for mobile manipulation in the context of construction, maintenance, and exploration in planetary and man-made environments. Some of the designed tasks were inspired by the DARPA robotics challenge (Krotkov et al., 2017). Each task was attempted at least once, with the final count of attempts depending on task execution time and the assessed task difficulty. When failures were encountered, more attempts were added to gain insight into the possible failure modes.

In general, there were three operators with specified responsibilities. One operator was sitting in the telepresence suit. A support operator was responsible for system parameter monitoring, interface management, and supported omnidirectional driving with the 4D joystick. A second support operator controlled robot limb motions through the keyframe editor, the 6D input device, and task-specific GUIs . Depending on the individual task, not all operators were required. Moreover, some tasks required an additional operator who was specialist for the employed interface such as autonomous controllers. All tasks were performed without visual contact such that the operators had to rely on information provided by our interfaces. While the acting operators were generally experienced in the operation of robots though the provided interfaces, the tasks themselves were new to them. The operators were allowed to inspect the task site before execution, but training runs were not allowed for any of the tasks. A video with footage from the experiments is available[9]. The results are summarized in Table 2. Some of the experiments have been previously described in less detail in (Klamt et al., 2018).

---

[9]https://www.ais.uni-bonn.de/videos/JFR_2018_Centauro

Table 2: Tasks evaluated at Kerntechnische Hilfsdienst GmbH.

Locomotion				Manipulation					
Task	Diff.[1]	Success	t [s][2]	Task	Diff.[1]	6D[3]	t_{6D} [s][2]	Exo[3]	t_{Exo} [s][2]
Ramp	2	3/3	83	Surface	3	2/2	32	1/1	30
Regular door	6	3/3	680	Small door	3	-	-	3/3	42
Gap	4	3/4	210	Valve (lever)	4	-	-	2/2	44
Step field	7	2/2	885	Valve (gate)	5	-	-	3/3	80
Stairs	9	0/1	-	Snap hook	6	-	-	3/3	20
Auto locomotion[4]	9	3/3	256	Fire hose	7	-	-	3/3	128
				230 V connector	7	3/3	151	1/5	300
				Cutting tool	8	-	-	3/9	33
				Driller	8	-	-	2/2	64
				Screw driver	10	3/3	613	0/2	-
				Auto grasping	8		7/14	220 s	

[1] Difficulty score estimated by operators before the attempts (1-10).
[2] Times are averaged over successful attempts.
[3] Success rates are shown separately for different control methods (6D mouse and exoskeleton). No inputs were used for the autonomous grasping experiment.
[4] This task was performed in a lab environment after the official evaluation.

In the following, we discuss the performed evaluation tasks, divided into largely-locomotion and largely-manipulation tasks. For component level evaluation we refer to the cited works in Sections 4 to 8.

9.1 Locomotion Tasks

The tested locomotion tasks mainly focused on proving that the robot can be effectively navigated across terrain types of different complexity:

Ramp: The robot was required to drive up and down a ramp of 120 cm length with 20° incline. This was accomplished using joystick teleoperation. The robot faced towards the higher platform for both subtasks. To compensate for the slope, the base pitch was adjusted by predefined motion primitives to keep the robot base parallel to the ground. This task was performed by the two support operators who reported that it posed no challenges and was easy to conduct with the provided interfaces. The mainly used visualization were camera images of which especially the two cameras under the robot base provided valuable perspectives.

Regular door: In the regular door experiment (see Fig. 33), the robot had to open a regular sized door (200 cm high and 90 cm wide) and drive through it. The door had to be opened away from the robot using a standard door handle and was not locked. The manipulation part was successfully tested using the 6D mouse. The door handle was pushed either by moving the left wrist downwards in just one axis, or by lowering the whole robot base. The latter was performed through a parametrizable motion primitive. The joystick was used to let the robot approach the door and drive through it. Again, the two support operators conducted this task. They reported that the robust SoftHand was valuable for this task since they had not to worry about damaging the hand, even when applying high forces to push down the door handle which reached its mechanical limit at some point. Since no force feedback was provided for this interface, it was challenging to assess when this limit was reached. After opening the door, the operators chose a right arm configuration such that the door was automatically pushed open when driving through it such that challenging simultaneous locomotion and manipulation control was not required. Overall, the operators assessed the difficulty of this task to be rather low.
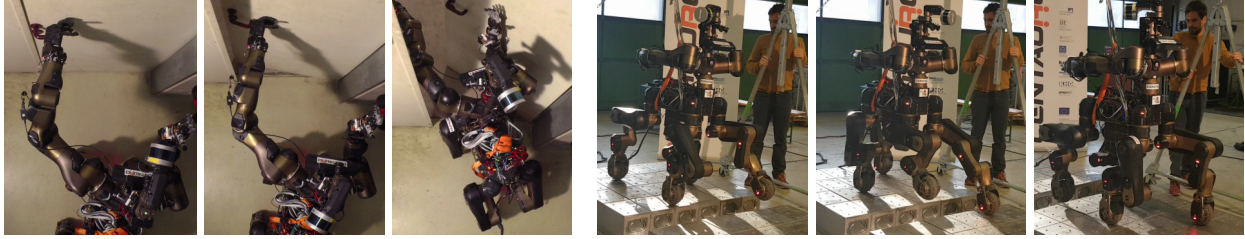
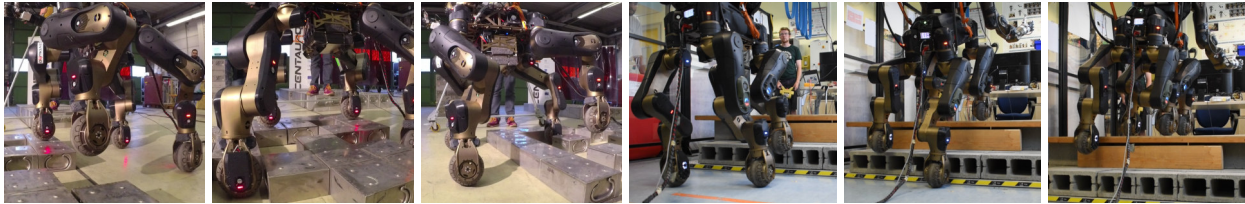Figure 33: Left: Opening the door and driving through it. Right: Overcoming a gap with the Centauro robot.

Figure 34: Complex locomotion tasks. Left: Traversing a step field. Right: Climbing stairs.

Gap: The gap test required the robot to overcome a 30 cm gap, which was accomplished using predesigned motion primitives interleaved with joystick driving commands (see Fig. 33). Motion primitives which incorporated the capability to move individual legs while under load were very helpful to obtain stable stepping configurations and solve this task successfully. In a first attempt the robot lost balance during a stepping motion. Moving the arms further backwards shifted the CoM and solved this problem. Both support operators jointly accomplished this task. Since a precise coordination of motion primitives and joystick commands was required—which were distributed among the operators—it was helpful that both operators sat next to each other and shared the same visualization. This allowed for e.g., easily pointing at details in images facilitating communication.

Step field: Metal blocks of size 20×20×10 cm were arranged in a random grid scheme and built a 200 cm long and 120 cm wide step field (see Fig. 34). The robot started 100 cm in front of the step field and was controlled via the semi-autonomous stepping controller and the respective GUI (see Section 6.3.3) to solve this task. The joystick was used to approach the field and to adjust the robot's position within the task. The latter required very careful maneuvers for which the capability of the joystick to scale the velocity command was very helpful. Accomplishing this task put a high cognitive load on the operators which had two reasons:

1. Since many actions had to be triggered to overcome the whole step field, execution took long resulting in a long time of high concentration. A higher degree of autonomy would have been desirable to accelerate execution and relieve cognitive load from the operators.

2. Button positions and captions in the GUI could be improved to increase intuitiveness. As depicted in Fig. 20 c, buttons were located depending on the robot top view foot position and captions were clear. Nevertheless, especially in the later phase of the experiment, the operator controlling the GUI tended to choose wrong maneuvers or wrong feet for the next maneuver. Operators avoided false decisions by verbally announcing their planned next action and waiting for approval of a second operator before triggering the action.

Nevertheless, the step field was traversed reliably two out of two attempts.

Stairs: The objective was to climb a flight of stairs consisting of three steps with 20 cm height, 30 cm depth, and 120 cm width and ending in a platform of the same width. In a first test, predefined motion primitives were used to climb the steps. In this demanding task, leg actuators failed mainly due to overheating. After improving the fan design for the respective actuators, this test was repeated in a later lab environment using the autonomous locomotion planner (see Section 9.4).
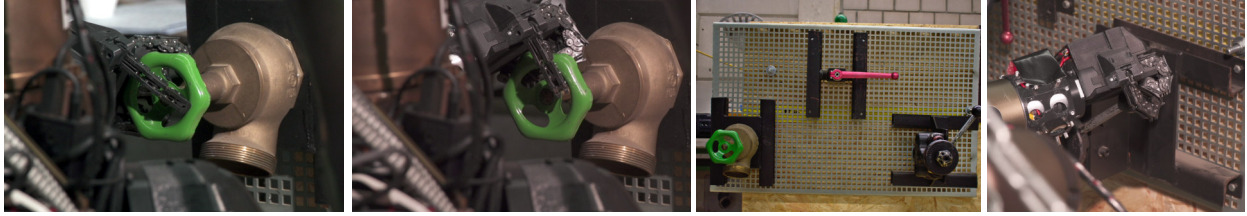
Figure 35: Valve experiments. Left to right: Grasping and turning of the green gate type valve, overview showing gate type and lever type valves, turning a lever type valve.
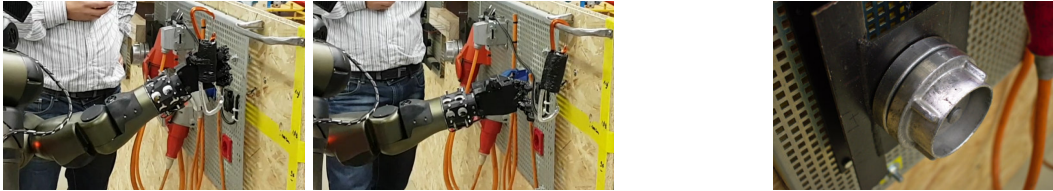

Figure 36: Left: Clipping a snap hook on a fixed metal bar. Right: The fire hose plug to be removed.

Overall, the locomotion capabilities were demonstrated successfully during the evaluation. Especially the more complex tasks would have been impossible to finish in acceptable time without the developed autonomous assistance functions or operator interfaces.

9.2 Telemanipulation Tasks

The telemanipulation tasks were designed to test a large variety of manipulation capabilities. A first set of experiments consists of direct object manipulation task using the robot hands. The second set of experiments was designed to test extended robot manipulation capabilities through tool usage. All manipulation tasks were mainly performed using the exoskeleton (see Section 6.4.1), with the support operators in a monitoring role. As an alternative manipulation interface, the 6D mouse control (see Section 6.4.2) was employed for experiments which required very precise manipulation.

Surface: The first task required the robot to sweep a planar surface with a (dummy) radiation sensor in a distance less than 10 cm without touching the surface. This task was successfully performed both using the exoskeleton and the 6D mouse for wrist control and locomotion via joystick. Especially useful was the ability of the 6D mouse control to constrain hand movement to the horizontal plane, parallel to the surface. As can be seen in Table 2, both control modalities provided a similar fast operation and a 100% success rate.

Small door: In this experiment a small door (90 cm×90 cm) had to be opened towards the robot. This task was successfully performed using the exoskeleton controlling the left arm and hand. The
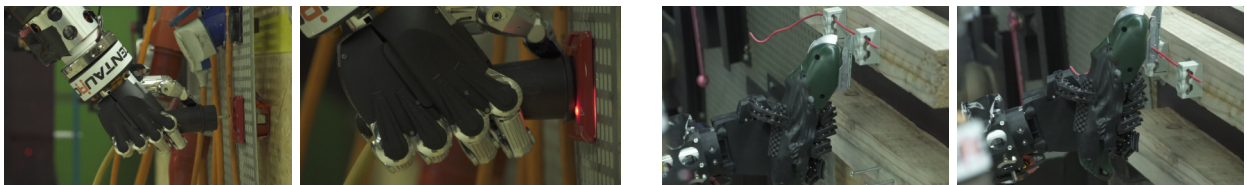

Figure 37: Manipulation experiments. Left: Inserting a power plug. Right: Cutting a fixed wire using a powered cutting tool.

robot started in front of the door and the handle was located roughly 1 m above the ground. The exoskeleton provided fast and successful teleoperation.

Valve: Two different valve types (gate and lever) were successfully manipulated using the exoskeleton control (see Fig. 35). Both valves were mounted in a height of 1 m above the ground. The exoskeleton operator intuitively used the palm of the hand to exert higher forces on the lever type valve. In both cases, the mechanical limit of the valve was detected using force feedback. Since the higher fidelity of the Schunk hand was not required, this experiment was performed with the more robust SoftHand.

Snap hook: In this task, a snap hook was to be clipped onto a fixed metal bar (see Fig. 36). Under the assumption that the robot brought the snap hook with it, we modified it slightly to make it more easily graspable. Again, the exoskeleton was used for sending wrist movement commands to the robot and a 100 % success rate was achieved. As can be seen in Table 2, this task proved to be fairly easy using this interface and was solved in little time.

Fire hose: In a more complex experiment, a fire hose connector was to be disconnected from the wall (Fig. 36). The connector was secured by a lock which required a 90° rotation to connect and disconnect. The robot managed to grasp and disconnect the connector. Additionally, we tried to connect it again, but failed in all three trials that were made due to insufficient precision under exoskeleton control.

230V connector: An electrical plug had to be inserted by the robot (Fig. 37). We attempted both exoskeleton and 6D mouse control. Here, the exoskeleton test suffered from inconvenient wrist poses and insufficient precision for inserting the plug. The 6D mouse control was more suited for this task, since very small adjustments could be made easily. After successful completion of the third attempt, a plastic part in the robot wrist broke due to excessive force—the operators had misjudged the situation slightly and did not have sufficient feedback of the exerted forces.

Cutting tool: In the first experiment with tool usage, the robot had to cut a wire held at both ends using a powered cutting tool (see Fig. 37). The test was performed under exoskeleton control. The tool was not easy to trigger by the operator, which accounts for the large number of failed attempts. After a modification of the tool to enlarge the trigger, operators had much less difficulty with the task and were able to complete it quickly in the last three attempts (see Table 2).

Driller: The next tool experiment used a power drill to drill a hole through wood at a specified point (see Fig. 38). The robot started with the tool in the hand. This task was performed without problems using exoskeleton control.

Screw driver: The most complex telemanipulation task required the robot to fasten a screw into a wooden block (see Fig. 38). The robot used a cordless screw driver for this task, starting with the tool in hand approx. 1 m away from the workspace. The wooden block was approached using joystick locomotion control, mainly guided by camera images and the 3D laser scanner point cloud. After approaching the workspace, the tip of the screw driver was aligned with the screw using 6D mouse control. This was the most difficult part of the task. Since our depth sensors were not able to measure the thin tip, operators were forced to use 2D images for estimating the 6D offset. For gaining an additional perspective, a small webcam was mounted on the other hand, providing a controllable-viewpoint perspective to the operators. After alignment was visually confirmed, the cordless screwdriver was activated using the index finger of the robot. Multiple times, a visually confirmed alignment turned out to be imprecise (i.e., the tip slipped off) and a further alignment attempt was required. During the screwing process, the operators had to ensure that the tool tip was in constant contact with the screw head, which was facilitated using the single-axis mode of the 6D mouse interface. Overall, three out of three task attempts were successful.

In summary, the manipulation tests successfully showcased a wide variety of manipulation capabilities. The exoskeleton allowed for intuitive and robust manipulation, while the 6D mouse control was superior for
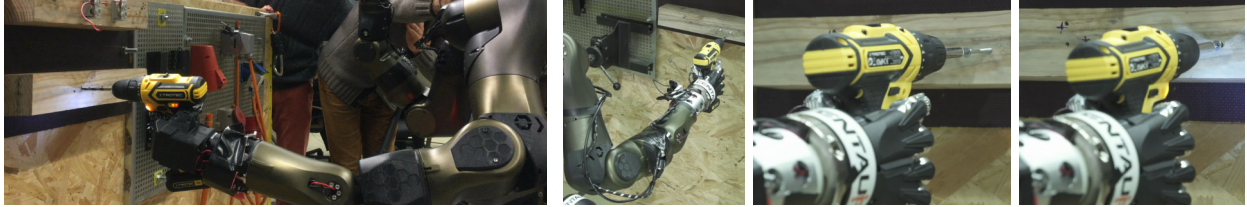
Figure 38: Power tool usage. Left: Drilling a hole into wood. Right: Driving a screw into wood.
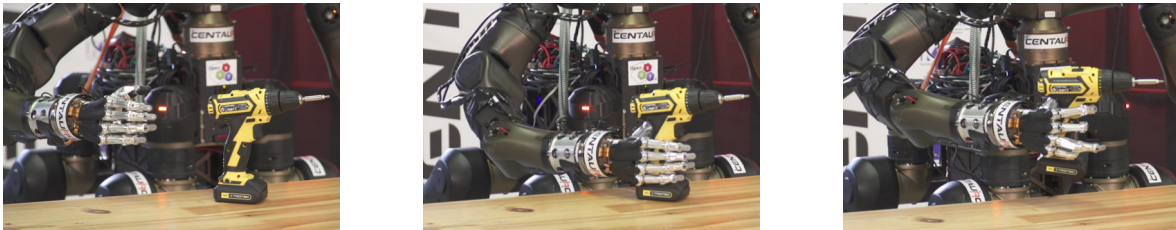

Figure 39: Autonomous grasping: approaching (left), grasping (center), and lifting the drill (right).

tasks which required precise incremental adjustments. VEROSIM (see Section 6.2) was a very helpful tool, increasing the operator's situation awareness. The digital twin of the robot provided views from any required perspectives onto the scene. Additionally, full 3D models of only partial visible 3D objects or tools could be rendered into the scene to compensate for occlusions.

In some cases, the mentioned failures highlighted the deficiencies of each control mode. The exoskeleton operators experienced difficulty performing fine-grained manipulation tasks, especially alignment tasks. While the 6D mouse control could perform small movements and even axis-aligned movements with relative ease, it offers no tactile feedback, leading to a misjudgment of the involved forces. This could be seen during the plug task, where the operators damaged the robot.

Many telemanipulation tasks required cooperation between different operators, e.g. the operator controlling locomotion using the joystick, and exoskeleton or 6D mouse operators. Because an integrated interface including all controls was not available, we could only evaluate this configuration. In our opinion, this split of concerns performed well, although it places special importance on clear communication between the operators. One particular inconvenience is that the exoskeleton operator, while wearing the HMD, can only communicate verbally, while certain manipulation ideas could be much more clearly and quicker communicated by demonstration. In the other direction, it was often first unclear what the main operator could see from his HMD perspective, which then had to be described verbally.

9.3 Autonomous Manipulation

We tested our autonomous grasping pipeline in a dedicated experiment. The objective was to detect, segment, and estimate the pose of a cordless driller in front of the robot (Fig. 39). Since this driller was previously unknown, a grasping pose was to be transferred from a known model to the new instance after pose estimation. Subsequently, the driller was to be grasped. Before any motion was executed, the support operators could assess the planned arm trajectory in detail and could trigger a re-computation if necessary.

Due to the complexity and the number of involved components this task had a higher failure rate such that we performed it multiple times. Failure cases include imprecise segmentation or misregistration, both resulting in missed grasps. In addition, we experienced hardware failures such as overheated actuators in the arm due to longer times of inactivity between attempts. Overall, the success rate improved during testing.

9.4 Autonomous Locomotion

After improvement of the fan design for some of the lower body actuators, we tested the hybrid driving-stepping locomotion planner (see Section 7) in a later lab experiment. The objective was to reach a goal pose on top of a flight of stairs (each step was 20 cm high and 30 cm long) while the starting pose was in some distance to these stairs. After an environment representation was generated based on registered point clouds (see Section 5.2), a support operator defined the desired goal pose. Since this can be easily done by positioning a marker on visualized cost maps, no extensive experience was required. The planner subsequently generated a path. Environment perception, map generation, and path computation took roughly 30 seconds. The generated path was inspected by the operator. For path inspection, the planner result was visualized on the generated cost map which, again, required no extensive experience. Subsequently, the operator triggered execution.

Omnidirectional driving was used to approach the stairs. The robot then used the capability to shift individual feet under load to obtain stable configurations to perform steps (see Fig. 34). In addition, base pitch and roll motions as well as base shift motions were used for stabilization. To facilitate balancing, the robot changed between different arm configurations for steps with the front and rear feet.

As can be seen in Fig. 34, we used concrete blocks to build the staircase. Those possessed holes such that the vertical segments of the staircase were not plane. Unfortunately, when performing a step with the right rear foot, the wheel got slightly stuck in such a hole which affected the robot balance. Since this setup does not represent a realistic step designs, a person at location was allowed to give the robot a slight push to regain balance. However, to increase system robustness, we derived the need for further improvement of the robot localization such that it can position itself on the staircase with a higher precision and avoids such contacts. Overall, three out of three attempts were successful for this experiment.

10   Lessons Learned & Conclusion

As always, the development and testing of such a complex system is a learning experience for all involved parties. We already reported some task-level insights in Section 9. In this section, we will take a step back and discuss the resulting insights on a higher level.

First of all, system integration is challenging, especially in an international project, where site visits between the partners are possible only for short durations. A precise definition of interfaces and interactions between the partner modules is of key importance, but difficult to achieve at the beginning of the project. In particular, interdependencies might not be immediately obvious. For example, a slightly bigger battery than expected resulted in less movement range for the legs.

A second insight is that operators still prefer 2D camera images over 3D visualizations. While the latter are certainly helpful in addition to 2D images, in general, operators tended to rely on camera images. Since we still believe that immersive 3D visualization is a fundamentally better solution, this can mean that the available 3D sensing, aggregation, and fusion with 2D camera images is not good enough yet. Also, more operator training with the 3D visualization could help.

The most appropriate interface for teleoperation is highly dependent on the specific task. Our telepresence suit provides a higher degree of immersion and thus allows more intuitive teleoperation, but for some tasks, other interfaces showed better performance. e.g., operating an electrical screw driver for driving a screw was easier with the 6D mouse interface. Still, as highlighted by the plug task, the 6D mouse interface has its own drawbacks. Careful choice of the right teleoperation interface is thus highly important. Also, further developments of the operator interfaces, e.g. towards hybrid interfaces which can change input modalities on the fly to match the current situation, should improve the ease of use considerably.

The performed system-level evaluation was especially useful for identifying issues and inconveniences that do not turn up during component-level testing. For example, the exoskeleton operators had trouble due to slightly different workspaces of the exoskeleton and the real robot (in particular, wrist positions were sometimes suboptimal for the operator). After the evaluation, a dynamic offset between the operator hand position and the robot end-effector was introduced, which can be adjusted by a support operator and allows the exoskeleton operator to manipulate in a comfortable posture. Furthermore, as evidenced by several experiments, the 6D input device suffers from inexistent force feedback. As it nevertheless proved its worth for intricate manipulation, we will add further feedback mechanisms such as visualizations of measured contact forces and auditory signals at predefined force thresholds.

During the evaluation we observed that communication between the different operators is highly important. While the described arrangement worked sufficiently for solving the tasks, some situations showed potential for improvement, especially communication between the main operator and the support operators. We recommend a) some method for the HMD-wearing main operator to see the other operators (e.g. by rotating the HMD out of the way or by means of an additional camera) and b) providing an additional view of the HMD perspective to the support operators, so that they can better refer to what the main operator sees.

A detailed simulation-based and specification-driven hardware design is not sufficient if the simulations or specifications are imprecise. In our case, the actuators failed in some cases, such as in complex robot configurations on the staircase. A detailed prediction of occurring forces and moments was not possible since the maneuvers were not known during the hardware development but where developed in parallel as part of the autonomous planning components. After identifying particularly stressed actuators during the evaluation, we improved the actuator cooling system with special focus on these points, which already showed highly positive results during the repeated stair experiments.

More general, in the context of space application, data transmission is usually affected by high latency and potential risk of data loss. While interfaces with a high degree of autonomy are immune to such problems, a weakened data link strongly affects e.g., the telepresence suit control or operator visualization. Nevertheless, since, we believe that a set of interfaces with complementary strengths is indispensable for advanced teleoperation, we aim at conducting future research in this direction. A promising idea is to extend the presented pipeline to an architecture in which the operator solely interacts with digital twins of the robot and its environment which are frequently updated from delayed data of the real robot. The same applies to the transmission pipeline from the digital twins to the real robot. While we already demonstrated how 3D visualizations and force feedback are generated from VEROSIM, updates of the digital twins require prediction from the delayed signals to the current time stamp. This is challenging but recent developments in this direction make such ideas promising.

Overall, we demonstrated a versatile, effective, and robust robotic system capable of solving a wide range of complex mobile manipulation tasks under the guidance of remote human operators, e.g., for construction, maintenance, and exploration on other planets as well as disaster response tasks on earth. The robot is accompanied by a set of operator interfaces on different levels of autonomy which have complementary strengths and are capable of addressing a wide range of the provided robot flexibility. All components have been integrated into a holistic remote mobile manipulation system. During the evaluation, the system met all expectations and provided valuable insights for the developments in the final phase of the CENTAURO project, towards providing robotic assistance and removing the need for direct human work in dangerous and inaccessible environments altogether.

References

Abe, Y., Stephens, B., Murphy, M. P., and Rizzi, A. A. (2013). Dynamic whole-body robotic manipulation. *Unmanned Systems Technology XV*, 8741.

Atorf, L., Cichon, T., and Roßmann, J. (2015). Flexible data logging, management, and analysis of simulation results of complex systems for eRobotics applications. In *European Simulation and Modelling Conference (ESM)*.

Atorf, L., Schluse, M., and Rossmann, J. (2014). Simulation-based optimization, reasoning, and control: The eRobotics approach towards intelligent robots. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*.

Baccelliere, L., Kashiri, N., Muratore, L., Laurenzi, A., Kamedula, M., Margan, A., Cordasco, S., Malzahn, J., and Tsagarakis, N. G. (2017). Development of a human size and strength compliant bi-manual platform for realistic heavy manipulation tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Buongiorno, D., Sotgiu, E., Leonardis, D., Marcheschi, S., Solazzi, M., and Frisoli, A. (2018). WRES: a novel 3DoF WRist ExoSkeleton with tendon-driven differential transmission for neuro-rehabilitation and teleoperation. *IEEE Robotics and Automation Letters (RAL)*, 3(3):2152–2159.

Catalano, M. G., Grioli, G., Farnioli, E., Serio, A., Piazza, C., and Bicchi, A. (2014). Adaptive synergies for the design and control of the Pisa/IIT SoftHand. *International Journal of Robotics Research (IJRR)*, 33(5):768–782.

Cichon, T., Loconsole, C., Buongiorno, D., Solazzi, M., Schlette, C., Frisoli, A., and Roßmann, J. (2016). Combining an exoskeleton with 3D simulation in-the-loop. In *International Workshop on Human Friendly Robotics (HFR)*, pages 31–34.

Cichon, T. and Roßmann, J. (2017a). Robotic teleoperation: Mediated and supported by virtual testbeds. In *International Symposium on Safety, Security, and Rescue Robotics (SSRR)*.

Cichon, T. and Roßmann, J. (2017b). Simulation-based user interfaces for digital twins: Pre-, in-, or post-operational analysis and exploration of virtual testbeds. In *European Simulation and Modelling Conference (ESM)*.

de Boor, C. (1978). *A Practical Guide to Splines*. Springer.

Droeschel, D., Schwarz, M., and Behnke, S. (2017). Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. *Robotics and Autonomous Systems*, 88:104 – 115.

Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., and Siegwart, R. (2015). Kinect v2 for mobile robot navigation: Evaluation and modeling. In *International Conference on Advanced Robotics (ICAR)*.

Gabardi, M., Solazzi, M., Leonardis, D., and Frisoli, A. (2018). Design and evaluation of a novel 5 DoF underactuated thumb-exoskeleton. *IEEE Robotics and Automation Letters (RAL)*, 3(3):2322–2329.

Halme, A., Leppänen, I., Suomela, J., Ylönen, S., and Kettunen, I. (2003). WorkPartner: Interactive human-like service robot for outdoor applications. *International Journal of Robotics Research (IJRR)*, 22(7-8):627–640.

Hebert, P., Bajracharya, M., Ma, J., Hudson, N., Aydemir, A., Reid, J., Bergh, C., Borders, J., Frost, M., Hagman, M., et al. (2015). Mobile manipulation and mobility as manipulation—design and algorithms of RoboSimian. *Journal of Field Robotics*, 32(2):255–274.

Hirose, S., Yokota, S., Torii, A., Ogata, M., Suganuma, S., Takita, K., and Kato, K. (2005). Quadruped walking robot centered demining system - development of TITAN-IX and its operation. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Hoffman, E. M., Rocchi, A., Laurenzi, A., and Tsagarakis, N. G. (2017). Robot control for dummies: Insights and examples using opensot. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*.

Johnson, M., Shrewsbury, B., Bertrand, S., Wu, T., Duran, D., Floyd, M., Abeles, P., Stephen, D., Mertins, N., Lesman, A., et al. (2015). Team IHMC's lessons learned from the DARPA robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208.

Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., and Schaal, S. (2011). STOMP: Stochastic trajectory optimization for motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Kamedula, M., Kashiri, N., Caldwell, D. G., and Tsagarakis, N. G. (2016). A compliant actuation dynamics Gazebo-ROS plugin for effective simulation of soft robotics systems: Application to CENTAURO robot. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

Kashiri, N., Ajoudani, A., Caldwell, D. G., and Tsagarakis, N. G. (2016). Evaluation of hip kinematics influence on the performance of a quadrupedal robot leg. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

Kashiri, N., Malzahn, J., and Tsagarakis, N. (2017). On the sensor design of torque controlled actuators: A comparison study of strain gauge and encoder based principles. *IEEE Robotics and Automation Letters (RAL)*, 2(2):1186–1194.

Kashiri, N., Tsagarakis, N. G., Laffranchi, M., and Caldwell, D. G. (2013). On the stiffness design of intrinsic compliant manipulators. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*.

Klamt, T. and Behnke, S. (2017). Anytime hybrid driving-stepping locomotion planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Klamt, T. and Behnke, S. (2018). Planning hybrid driving-stepping locomotion on multiple levels of abstraction. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Klamt, T., Rodriguez, D., Schwarz, M., Lenz, C., Pavlichenko, D., Droeschel, D., and Behnke, S. (2018). Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Krotkov, E., Hackett, D., Jackel, L., Perschbacher, M., Pippine, J., Strauss, J., Pratt, G., and Orlowski, C. (2017). The DARPA robotics challenge finals: Results and perspectives. *Journal of Field Robotics*, 34(2):229–240.

Lawin, F. J., Forssén, P.-E., and Ovrén, H. (2016). Efficient multi-frequency phase unwrapping using kernel density estimation. In *European Conference on Computer Vision (ECCV)*.

Likhachev, M., Gordon, G. J., and Thrun, S. (2003). ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems (NIPS)*.

Lin, G., Milan, A., Shen, C., and Reid, I. (2017). Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mehling, J. S., Strawser, P., Bridgwater, L., Verdeyen, W. K., and Rovekamp, R. (2007). Centaur: NASA's mobile humanoid designed for field work. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Muratore, L., Laurenzi, A., Hoffman, E. M., Rocchi, A., Caldwell, D. G., and Tsagarakis, N. G. (2017). XBotCore: A real-time cross-robot software platform. In *IEEE International Conference on Robotic Computing (IRC)*.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(12):2262–2275.

Pavlichenko, D. and Behnke, S. (2017). Efficient stochastic multicriteria arm trajectory optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Pavlichenko, D., Rodriguez, D., Schwarz, M., Lenz, C., Periyasamy, A. S., and Behnke, S. (2018). Autonomous dual-arm manipulation of familiar objects. In *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*.

Pirondini, E., Coscia, M., Marcheschi, S., Roas, G., Salsedo, F., Frisoli, A., Bergamasco, M., and Micera, S. (2014). 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.

Raibert, M., Blankespoor, K., Nelson, G., and Playter, R. (2008). BigDog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41(2):10822–10825.

Rigano, G. F., Muratore, L., Laurenzi, A., Hoffman, E. M., and Tsagarakis, N. G. (2018). Towards a robot hardware abstraction layer (R-HAL) leveraging the xbot software framework. In *IEEE International Conference on Robotic Computing (IRC)*.

Rodehutskors, T., Schwarz, M., and Behnke, S. (2015). Intuitive bimanual telemanipulation under communication restrictions by immersive 3D visualization and motion tracking. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*.

Rodriguez, D. and Behnke, S. (2018). Transferring category-based functional grasping skills by latent space non-rigid registration. In *IEEE Robotics and Automation Letters (RA-L)*, pages 2662–2669.

Rodriguez, D., Cogswell, C., Koo, S., and Sven, B. (2018). Transferring grasping skills to novel instances by latent space non-rigid registration. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Roennau, A., Heppner, G., Nowicki, M., and Dillmann, R. (2014). LAURON V: A versatile six-legged walking robot with advanced maneuverability. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*.

Roozing, W., Malzahn, J., Kashiri, N., Caldwell, D. G., and Tsagarakis, N. G. (2017). On the stiffness selection for torque-controlled series-elastic actuators. *IEEE Robotics and Automation Letters (RAL)*, 2(4):2255–2262.

Ruehl, S. W., Parlitz, C., Heppner, G., Hermann, A., Roennau, A., and Dillmann, R. (2014). Experimental evaluation of the Schunk 5-finger gripping hand for grasping tasks. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*.

Sarac, M., Solazzi, M., Sotgiu, E., Bergamasco, M., and Frisoli, A. (2017). Design and kinematic optimization of a novel underactuated robotic hand exoskeleton. *Meccanica—International Journal of Theoretical and Applied Mechanics*, 52(3):749–761.

Schwarz, M., Beul, M., Droeschel, D., Schüller, S., Periyasamy, A. S., Lenz, C., Schreiber, M., and Behnke, S. (2016a). Supervised autonomy for exploration and mobile manipulation in rough terrain with a centaur-like robot. *Frontiers in Robotics and AI*, 3:57.

Schwarz, M., Lenz, C., García, G. M., Koo, S., Periyasamy, A. S., Schreiber, M., and Behnke, S. (2018). Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Schwarz, M., Rodehutskors, T., Droeschel, D., Beul, M., Schreiber, M., Araslanov, N., Ivanov, I., Lenz, C., Razlaw, J., Schüller, S., Schwarz, D., Topalidou-Kyniazopoulou, A., and Behnke, S. (2017). NimbRo Rescue: Solving disaster-response tasks with the mobile manipulation robot Momaro. *Journal of Field Robotics*, 34(2):400–425.

Schwarz, M., Rodehutskors, T., Schreiber, M., and Behnke, S. (2016b). Hybrid driving-stepping locomotion with the wheeled-legged robot Momaro. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Semini, C., Tsagarakis, N. G., Guglielmino, E., Focchi, M., Cannella, F., and Caldwell, D. G. (2011). Design of HyQ - a hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(6):831–849.

Stentz, A., Herman, H., Kelly, A., Meyhofer, E., Haynes, G. C., Stager, D., Zajac, B., Bagnell, J. A., Brindza, J., Dellin, C., et al. (2015). CHIMP, the CMU highly intelligent mobile platform. *Journal of Field Robotics*, 32(2):209–228.

Stückler, J., Schwarz, M., Schadler, M., Topalidou-Kyniazopoulou, A., and Behnke, S. (2016). NimbRo Explorer: Semi-autonomous exploration and mobile manipulation in rough terrain. *Journal of Field Robotics*, 33(4):411–430.

Tsagarakis, N. G., Caldwell, D. G., Negrello, F., Choi, W., Baccelliere, L., Loc, V., Noorden, J., Muratore, L., Margan, A., Cardellino, A., et al. (2017). WALK-MAN: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259.

Ur Rehman, B., Focchi, M., Frigerio, M., Goldsmith, J., Caldwell, D. G., and Semini, C. (2015). Design of a hydraulically actuated arm for a quadruped robot. In *International Conference on Assistive Robotics (CLAWAR)*.

Yamauchi, B. M. (2004). PackBot: A versatile platform for military robotics. In *Proceedings of SPIE, Unmanned Ground Vehicle Technology VI*, volume 5422.

Zucker, M., Joo, S., Grey, M. X., Rasmussen, C., Huang, E., Stilman, M., and Bobick, A. (2015). A general-purpose system for teleoperation of the DRC-HUBO humanoid robot. *Journal of Field Robotics*, 32(3):336–351.