Ref. Ares(2016)4900061 - 31/08/2016

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

Deliverable D4.2 - Simulation of CENTAURO Robot and Environment

Dissemination Level: Public

Project acronym:	CENTAURO
Project full title:	Robust Mobility and Dexterous Manipulation in Disaster Response by Fullbody Telepresence in a Centaur-like Robot
Grant agreement no.:	644839
Lead beneficiary:	RWTH – Rheinisch-Westfaelische Technische Hochschule Aachen
	MMI - Institute for Man-Machine Interaction
Authors:	Torben Cichon and Christian Schlette
Work package:	WP4 – Modeling and Simulation
Date of preparation:	2016-08-31
Type:	Demonstrator
Version number:	1.0

Document History

Version	Date	Author	Description
0.1	2015-08-15	Torben Cichon	initial version
0.2	2015-08-15	Torben Cichon	general setup added
0.3	2015-08-15	Torben Cichon	interfaces added
0.4	2015-08-18	Torben Cichon	MMI internal revision
0.5	2015-08-19	Christian Schlette	MMI internal revision
0.6	2015-08-19	Torben Cichon	minor updates
0.7	2015-08-23	Torben Cichon	post UBO and PGX review
0.8	2015-08-24	Christian Schlette	MMI internal revision
0.9	2015-08-30	Christian Schlette	post coordinator review
1.0	2015-08-31	Coordinator	submitted version

Executive Summary

In D4.2 a demonstrator is presented, which encompasses the simulation functionalities of the CENTAURO robot and environment. Here, a prototypical CENTAURO robot can be maneuvered in a disaster scenario by means of direct control, force feedback, and visual (augmented reality) metaphors—using third and first person views. Additionally, two more demonstrators are shown, which represent the interface structures of the simulation. First, the simulation is connected to an exoskeleton simulator (by partner SSSA) exchanging position and force data at high frequencies. Second, the simulation is connected via ROS (Robot Operating System) with the *Momaro* setup (by partner UBO) to ensure a reliable hardware interface to the robot and sensor hardware.
In summary, we developed a comprehensive demonstrators of the 3D simulation-based operator interface, the ROS interface, and the exoskeleton interface as described in T4.2.

Contents

1 Introduction

For a robotic field system such as planned in CENTAURO, 3D simulation can be used prior to the completion of the full system for the design, development and optimization of hardware and software components. Plus, simulation can be used in the field as an additional interface between the operator and the robotic field system, e.g. to plan and verify next actions in simulation first before executing them in reality. Thus, CENTAURO aims at developing novel 3D simulation-based interfaces to ensure an intuitive interaction of operator and robot during system development as well as during system operation.

As described in T4.2 (CENTAURO robot and environment simulation), a major part of work package WP4 (Modeling and Simulation) is to provide a close-to-reality simulation of the CENTAURO robot in various environments. This simulation is the basis for the predictive robot model and the interaction of the operator with the look-ahead simulation (see Grant Agreement [7]):

> The robot's dynamic model consists of rigid bodies and may be updated by using provided CAD files in the development process. For complex shapes, simpler substitutions are used to enable dynamic multi-body simulation in real-time. For actuator simulation, a motor model is needed and the control algorithms from WP2 (T2.3 - T2.5) are included. CENTAURO's actuator control can either be directly integrated into the Virtual Testbed (VTB, see D4.1), or it can run as software in the loop. Sensor simulation is another important part of the robot model. Existing components of the VTB can be configured and used for sensor simulation such as laser scanners, RGB-D sensors, and stereo cameras. Extensions and modifications are carried out where necessary - as well as an appealing visualization of sensor data in real time. Attention also has to be paid to environment simulation. As first step, different scenarios can be modeled statically (again, import from CAD or other 3D files is possible). The resulting virtual worlds provide a testing ground for the simulated CENTAURO robot in different disaster scenarios, possibly with elements such as stairs, doors, or collapsed walls. To generate a realistic impression, more effects and features can be added to the VTB, e.g., advanced lighting simulation with various new materials, or fog, smoke, and dust simulation. Egocentric and third-person views into the simulated worlds are directly observable by the operators.

The core components of work package WP4—also with respect to WP8 (Requirement Specification and Evaluation)—are:

1. Interfaces
   (a) Exoskeleton Interface
   (b) ROS Interface
   (c) Sensor Interface,
2. Simulatable Robot Model, and
3. Simulatable Environment Model.

Thus, D4.2 is not only limited to the robot and environment simulation but moreover to its use and connectivity. The interface of the simulation model to real hardware components and external input or output devices is of paramount importance.

2 Overview

The deliverable D4.2 (Simulation of CENTAURO Robot and Environment) encompasses:

- CENTAURO robot model,
- environment model for disaster scenarios,
- interface to the Robot Operating System (ROS),
  - incorporation of specific ROS data types in simulation
- interface to the exoskeleton, and
  - incorporation of force feedback in simulation
- sensor interface, data processing, and visualization.

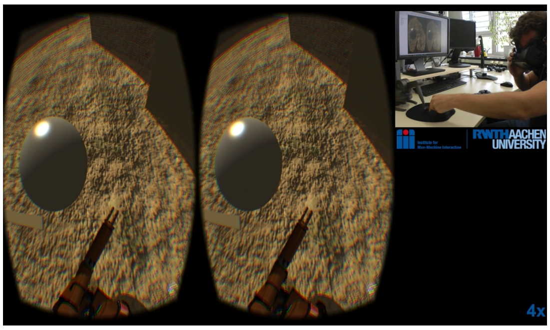
The results of this deliverable are integrated in a demonstrator that is detailed in the next section. In the subsequent sections the different parts of the demonstrator are presented and their status, progress and contribution are described. In addition to the main simulation demonstrator, we also developed one demonstrator for each of the two major interfaces: ROS and the exoskeleton. These demonstrators are important to ensure the upcoming transfer from simulation to real hardware components.

3 Demonstrator

The demonstrator of D4.2 was shown at the *Automatica* fair in Munich (June 2016) and comprises almost all features developed in T4.2. The setup of the demonstrator at Automatica can be seen in Fig. 1. It consists of the *Momaro* robot and the simulation setup. The *Momaro* robot, from a prior UBO project, is accompanied by a video screen, showing results from the DARPA Robotics Challenge. Its rotating laser scanner head is connected to a *Oculus Rift Dev Kit 2* to visualize live sensor data (cf. (a), in the background). The demonstrator setup is placed on a table (cf. (a), in the foreground).

(a) CENTAURO booth at Automatica

(b) Demonstrator setup - Overview of the devices used with the simulation



(c) Demonstrator setup - Stereoscopic view of CENTAURO operators in simulation

Figure 1: Demonstrator setup as presented at the *Automatica* fair in Munich, June 19-22, 2016.

The demonstrator setup is also depicted in Fig. 1(b) and Fig. 2: The CENTAURO robot model is placed in a virtual disaster scenario environment. Most types of movement control are accomplished via a gamepad. This includes driving movements (acceleration, deceleration, and turns), prescripted whole-body movements (body elevation and squad), additional head movements, and the switch for additional camera overlay images. Additionally, a *Geomagic Touch X* haptic force feedback device is connected to the left arm of the robot with force feedback in its tool center point. The third person view onto the scene can be switched into a first person stereoscopic view utilizing the *Oculus Rift Dev Kit 2* (cf. (c)). This overall setup enables the user to operate the virtual CENTAURO robot from the first person view, using also the gamepad for maneuvering and the *Geomagic Touch X* for force feedback. Additionally, it is possible to switch to the support operator, third person view, with the possibility to add additional information in terms of camera overlays for the user.

In short, this demonstrator shows a model of the CENTAURO robot and a sample environment (cf. also Section 4). Additionally, force feedback of the simulation is introduced and displayed with the *Geomagic Touch X* (cf. also Section 6). Simulated sensors are introduced to support the user, accompanied by the possibilities of first person and support operator views.

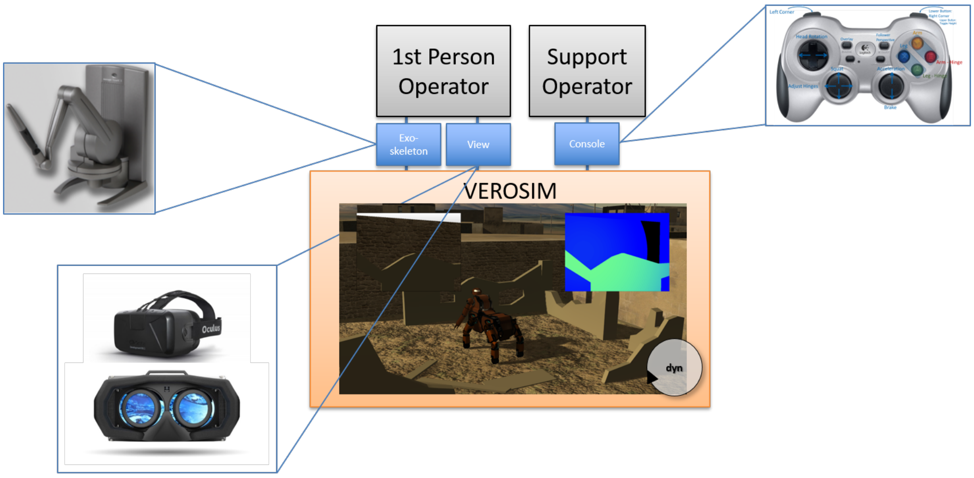


Figure 2: Schematic view of the demonstrator setup.

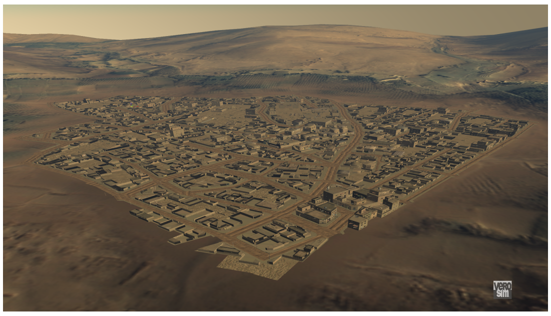
One central aspect, the ROS interface described in Section 5, is currently not integrated in this demonstrator but scheduled for integration as part of the ongoing works in WP4.

4 Robot and Environment Modeling

Figure 3: Robot and environment simulation in the demonstrator.

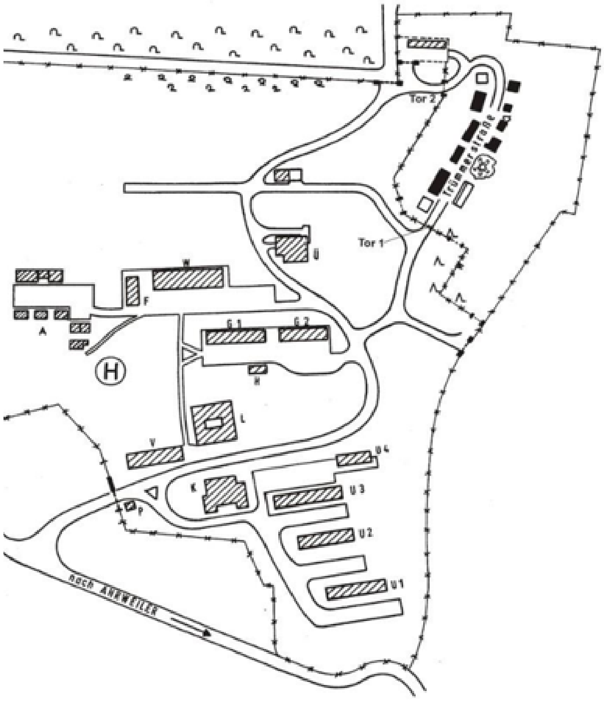
Fig. 3 depicts the rigid body dynamic simulation of a prototypical version of the CENTAURO robot in a disaster scene. The robot is modeled using rigid bodies, which are connected via different joints and controlled via motors. Although the robot model is based on the descriptions of WP2, we are not yet using specific details, but equip the model with standard motors (position or velocity controlled) in the rotational and ball-in-socket joints. The rigid body model is currently using standard shapes (mainly pill- or sphere- shaped objects). Nevertheless, complex geometries can also be included and transferred into rigid body shapes using either oct-spheretrees [5] or a triangulation approach. The robot model is continuously updated by using geometry data from partner IIT, who maintain and provide Gazebo models of their robot. Without using any control algorithms yet, the posture of the robot is modeled with a predefined startup orientation in the position-controlled joint motors. Also pre-scripted motor-based actions are implemented the same way. Sensor information can be gathered with the integrated sensor simulation framework. In the current model, a standard RGB camera and a simulated PMD sensor are used to visualize additional information. These sensors have been selected for the purpose of demonstrator development and presentation, while the details of the sensors and cameras used in the final CENTAURO setup have still been under revision (the setup of sensors and cameras was finally decided at the CENTAURO project meeting May 21-22, 2016, in Stockholm). The parameters of the simulated sensors and cameras will be adapted to represent this final setup as part of the next general update of the simulation, scheduled for Q4/2016.

The environment model is defined as a hierarchical spatial tree of 3D geometries (primitives or CAD data), which can be enhanced with environmental details for rendering (e.g. time of day, weather conditions) as well as automatically generated support for collision detection and rigid-body simulation. For example, the geometries of the environment model depicted in Fig. 3 were first imported from an existing CAD model, then enhanced with modules for rendering and dynamic simulation. Currently, the environment models can seamlessly be exchanged by any given environment, using the same robot and also the same input devices. Some possible environments are shown in Fig. 4.



(a) *VEROSIM* village

(b) Benchmark Arena [7]



(c) Ahrweiler Test Ground

Figure 4: Different types of environment models for *VEROSIM*.

At the moment, the environment model (a) is suitable to show the capabilities of the current version of the CENTAURO robot. In a next step, we plan to provide a catalogue of environment models representing test environments of the final robot as defined in the CENTAURO requirements. These test environments are focussing on specific, well-defined situations and can act as ”building blocks” to setup more complex models for benchmarking step-by-step. As such a more complex benchmark scenario the consortium is considering a ”box”-based benchmark arena (cf. Fig. 4b) or the existing testing ground in Ahrweiler (Fig. 4c). The modeling of the catalogue of test evironments as well as the modeling of the complex benchmark scenario will be feasible based on the definition of environment models developed in T4.2.

Although modeling is the focus of this work package, much effort has also been put on interfaces as modeling of the robot does not only involve a geometric approximation, but has to mimic real behavior in terms of interfaces, too. Therefore, the control interface to the exoskeleton, and the ROS interface to all other hardware parts is of paramount importance in this deliverable. These central interface modules are discussed in the next sections.

5 ROS Interface

We implemented a generic integration of ROS into the used 3D simulation system *VEROSIM* to enable to use the full spectrum of ROS functionalities from within the simulation. Implementing an interface to the communication infrastructure of ROS aims at connecting the message passing system with *roscore* to open up many possibilities regarding other core components of ROS. The milestone to achieve was to be able to resemble the features in the *Momaro* setup [12][8] in order to make use of the knowledge already available from prior ROS setups.

The *Momaro* setup is mainly based on standard ROS data types. Thus, we started the implementation in *VEROSIM* with corresponding standard message types as well as combined message types for the central input/output board of *VEROSIM* (so-called IO Board) [9], which then allows for dynamically connecting internal simulation functionalities with the ROS framework. As a result, simulation scheduling, rendering and the other frameworks can utilize the inputs and outputs of ROS nodes.

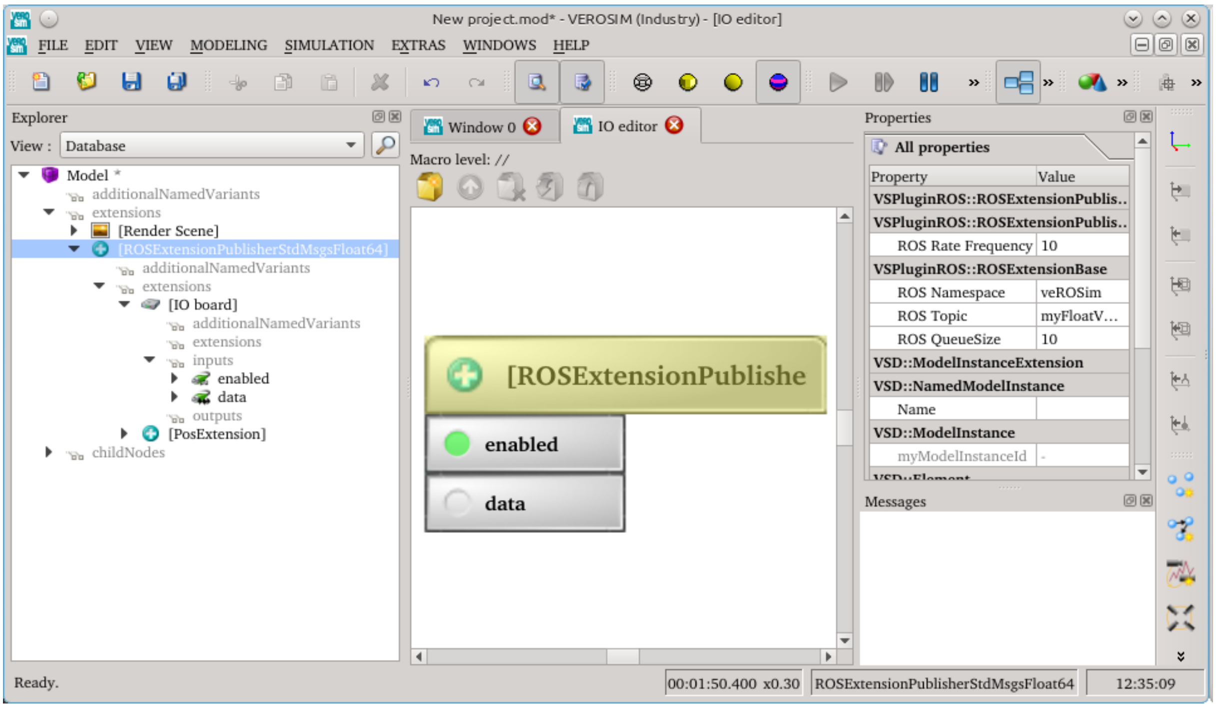


Figure 5: *VEROSIM* IO Board using dynamic input from simulation to publish, and submit to ROS messages via outputs in simulation.

The following implementation scheme is used to continuously add ROS functionalities to *VEROSIM* using the *roscpp* (and *rospy*) API:

1. Implementation of static data type conversions for *std_msg* types,
2. Implementation of static data type conversions for combined *std_msg* and specialized types,
3. Implementation of template-based conversions for arbitrary message types, and
4. Completely dynamic (Python-based) embedding of ROS data types into *VEROSIM*.

Step by step, this led to a continuous integration of more data types. Finally, it will be possible to use a given set of ROS message types from within simulation (input and output). In

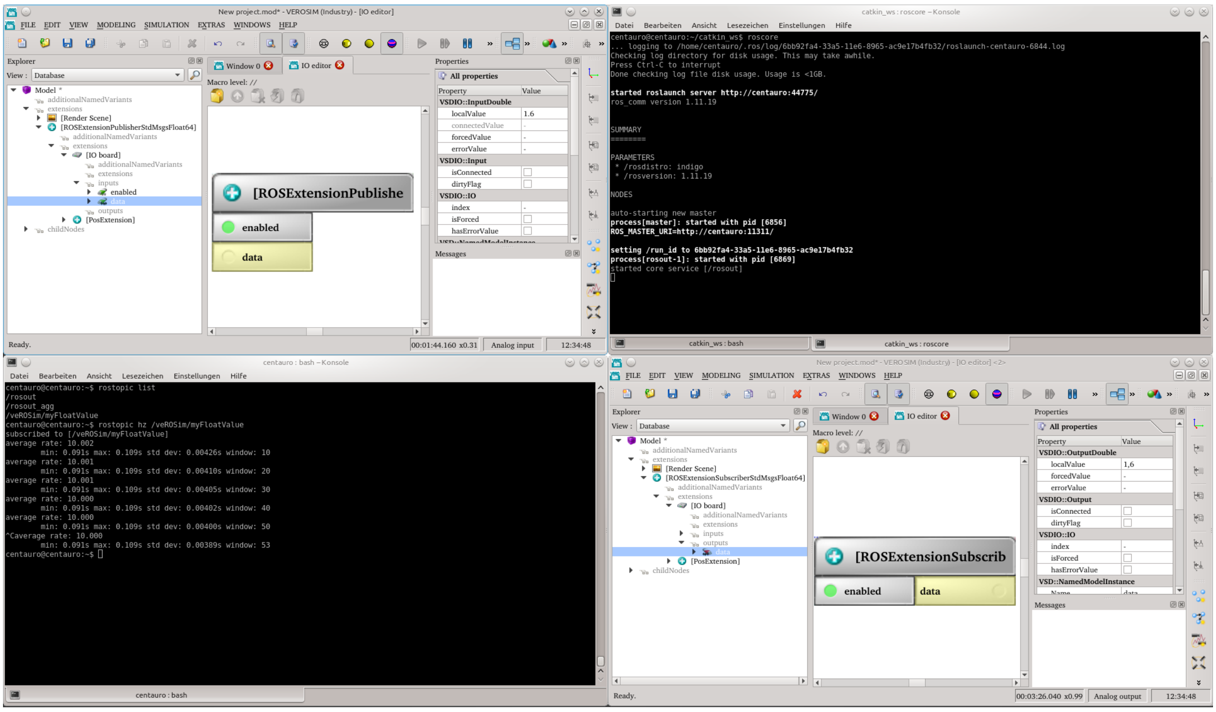


Figure 6: Using ROS with *VEROSIM*: (top/left) *VEROSIM* issuing data using ROS publisher, (top/right) console running *roscore*, (bottom/left) console for inspection with *rostopic*, (bottom/right) second instance of *VEROSIM* reading the data using ROS subscriber.

addition, we will offer an easy way to add new functionalities based on C++ templates, or even use Python for dynamic ROS message implementation (on the fly).

Using this implementation scheme led to the following results: Fig. 5 shows a ROS publisher extension in the *VEROSIM* IO Board to publish a *float64* in the ROS namespace ”veROSim” on the ROS topic ”myFloatValue”. In Fig. 6 this ROS *std_msg float64* publisher and its corresponding subscriber are used to connect two instances of *VEROSIM*. As a result, the input data from the top/left *VEROSIM* is published and thus transferred via the ROS network to the subscriber in the bottom/right *VEROSIM*. Now, it is possible to use the template-based conversion analogously to quickly add new ROS message types communicating the same way as described above. The completely dynamic, Python-based approach is still under development, would allow a more dynamic development approach, but is not strictly needed for further developments with the implemented ROS interface.

In an internal meeting (CENTAURO *VEROSIM* meeting, April 22nd 2016 in Aachen) the definition of this interface lead to a requirement analysis and some core components summarized in Tab. 1

A first proof of concept could already been achieved during this meeting in Aachen, where we connected one joint of the *Momaro* robot with one virtual CENTAURO robot joint via ROS. Since then, all crucial message types (marked ”high” and ”medium” in Tab. 1) have been implemented and tested with the help of the *Momaro* simulation in *Gazebo*, provided by UBO. This can be also seen as one demonstrator of this part of the deliverable D4.2. Implementation of additional message types (marked ”low” in Tab. 1), and their testing and optimization is momentarily in progress. Additionally, are we currently implementing the required sensor message types to be connected to the integrated sensor framework of *VEROSIM* to utilize both, ROS and *VEROSIM* sensor data processing and visualization.

Type	description	Importance
rosgraph_msgs/Clock	Simulation time (2kHz)	high
sensor_msgs/JointState	Joint measurements (125Hz)	high
sensor_msgs/LaserScan	Single Laser Scan (40Hz)	medium
gazebo_msgs/ModelStates	Position and Velocity	low
control_msgs/JointControllerState	Single Joint Feedback	low
std_msgs/Float64	JointAngle	high
dynamic_reconfigure/Reconfigure	PID gains	low

Table 1: ROS interface definition/ requirement analysis.

6 Exoskeleton Interface

The integration of force feedback in 3D simulation environments is not addressed in current research. Most commonly used as three-dimensional input devices for modeling, force feedback devices are only in some rare applications also used in specialized simulation environments, such as surgical simulations, where force feedback is then the main aspect of simulation.

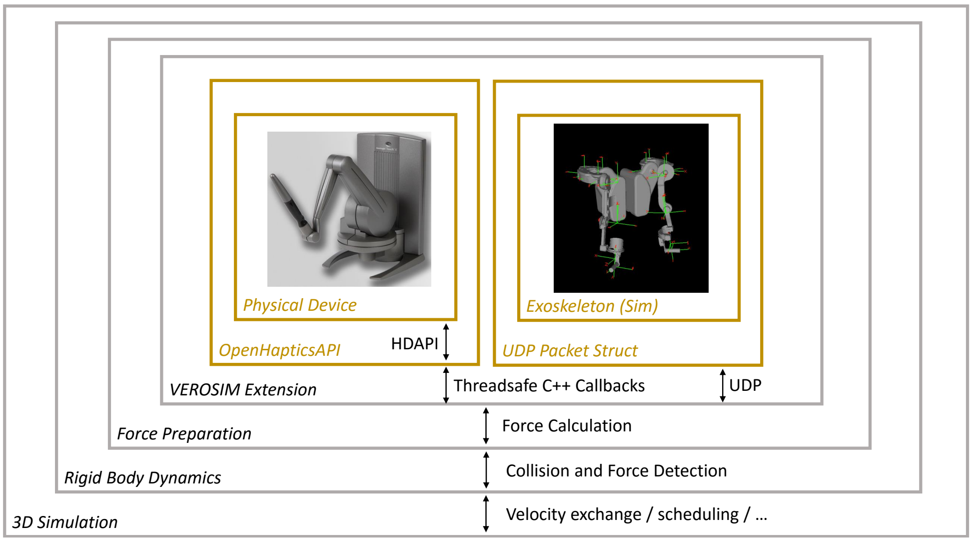


Figure 7: Modular force feedback concept chart (chart idea based on [6]). Using a modular organization, the physical device and its API can be easily exchanged. The connection of simulation scheduling, rigid body dynamics, collision detection and force preparation is carried out in 3D simulation.

We developed a generic interface concept to couple rigid body dynamics based force generation, force reprocessing and specialized driver interfaces for each force feedback device. Devlopment of this interface was started with the *Geomagic Touch X* haptic force feedback device and then extended towards a force feedback ready exoskeleton. As a result, the overall force feedback interface implements three layers:

1. Intertwining of dynamic simulation and events of force feedback calculation at time t_{FF},
2. Generic interface for force feedback devices, calculating a generic force feedback force F_{FF} at the time t_{FF},

3. Specialized driver interfaces for each haptic device,
   (a) *Geomagic Touch X* with *OpenHaptics API*
      - transmit the calculated force F_{FF}^{TouchX}, and
      - provide positional input p^{TouchX} of the tool center point.
   (b) Exoskeleton with UDP/IP connection
      - transmit an exoskeleton device command struct, either in 'force mode' (using joint torques τ_i^{exo} for each joint i) or 'compliant position mode' (using the end effector position p_{out}^{exo}) and
      - provide an exoskeleton device data struct, with positional input of the end effector p_{in}^{exo}.

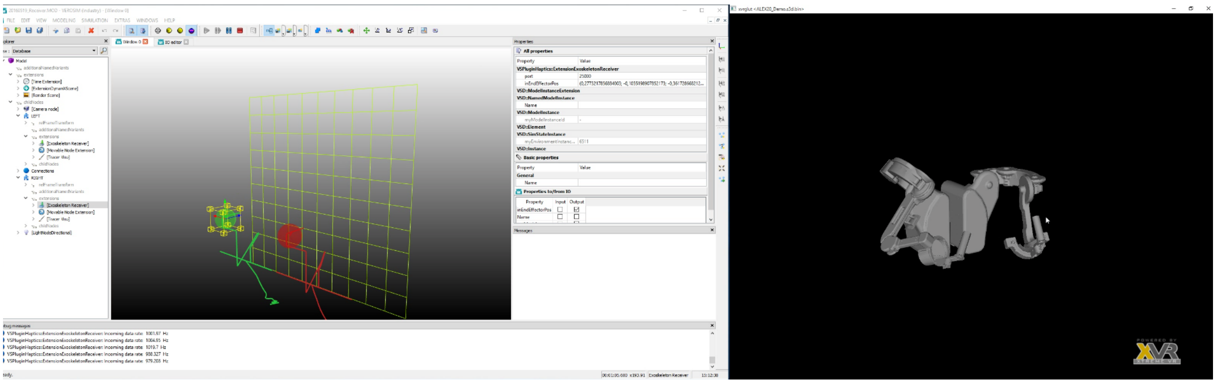
Starting with the *Geomagic Touch X*, we used the freely available *OpenHaptics API* [4] to implement the driver interface, while the deeper layers were achieved in simulation. As one can see in Fig. 7, the API is just used for low level interfacing the physical hardware. Visible for the user in the 3D simulation is just an extension that manages a thread-safe communication channel. On a higher level, the collision and force detection, calculation and scheduling is of paramount importance. We implemented a collision-based determination of each force feedback event ($\rightarrow t_{FF}$). Now, either

- the calculated force on interacting rigid bodies (F_{RB}) can be used as force feedback,
- specific force torque sensors (F_{FT}) (e.g. in the joints) can be used for force feedback, or
- a more general approach, where the virtual coupling is based on a mass-spring-damper system as found in [1][11].

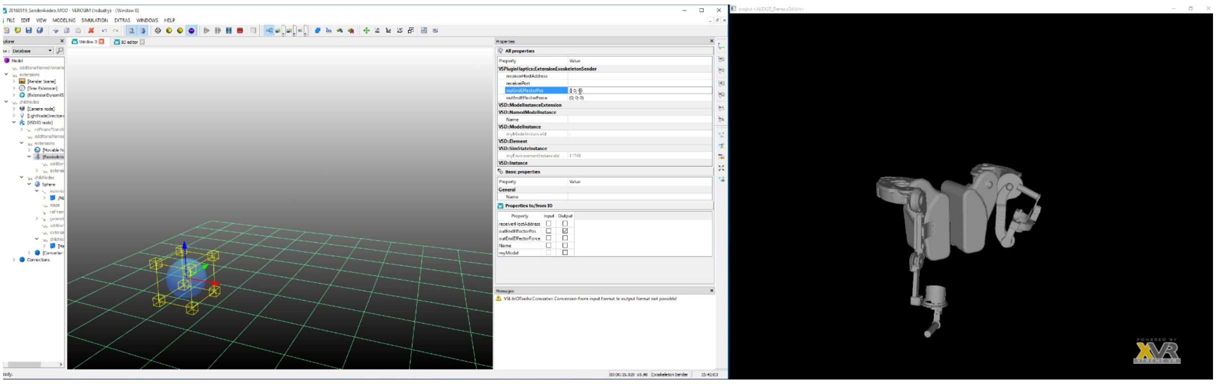
In the third option, a variance analysis of current position and target position is used to calculate a (virtual) spring-damper based force (F_{SD}). This procedure has the advantage of equal force dimensions, irrespective of the two colliding bodies. Otherwise the calculated collision force could become too high or too volatile for the force feedback device. As a result, we use the general option for force direction and magnitude calculation, the integrated dynamic rigid body framework for collision detection, and a separate thread to safely collaborate with the *OpenHaptics API*.

Using this interface, it is also possible to exchange the *Geomagic Touch X* with other force feedback devices, in poarticular the exoskeleton by partner SSSA. During the development of the final exoskeleton, an exoskeleton simulator (by SSSA) is used as a substitute to define, develop, and use the exoskeleton interface in the 3D simulation. This exoskeleton simulator provides the exact same interface design as the final exoskeleton. Therefore, defined exchange information structs (encompassing end-effector position, joint angles, joint force and torques, etc.) can already be received by and send from simulation. Although the communication between simulation and *Geomagic Touch X* is based on a specific API and thus completely different to the UDP- based connection of the exoskeleton, the infrastructure of the force feedback interface already provides all necessary pre-processing of forces. The low level interface layer of the UDP exoskeleton is then added on top of the force feedback fundament.

Using the exoskeleton simulator led to a defined interface concept for simulation and already shows first promising results in terms of the communication protocol and also realtime capable communication. In Fig. 8 the developed demonstrator can be seen for positional in- and output between *VEROSIM* and the exoskeleton simulator. In synchronization with the development and availability of the full exoskeleton setup by SSSA, next steps are scheduled to generate force feedback from simulation for a direct and intuitive sense of immersion with the real exoskeleton.



(a) Exoskeleton input to simulation. Motions of the exoskeleton as commanded with the exoskeleton simulator by SSSA (right) are transferred to *VEROSIM* and mapped to representation of each wrist (left).



(b) Simulation output to exoskeleton. Motions of the wrist representations in *VEROSIM* (left) are transferred to the exoskeleton simulator (right) in order to command the exoskeleton.

Figure 8: Demonstrator of the exoskeleton interface.

7 Sensor Interface, Data Processing and Visualization

The visualization of the sensor data collected from optical and other sensors has to be preprocessed and made available in an intuitively and understandable manner.

Due to the fact that one person will not be capable of supervising the robot alone, the CENTAURO project proposes the introduction of one or more support operators who provide additional, necessary information for the pilot. For the two types of operators, we accomplish two different views on the scene, 1) an immersive first person view using a head mounted stereoscopic display (here the *Oculus Rift Dev Kit 2*[5]), and 2) a third person view onto the whole scene in simulation. This requires the stereoscopic view implementation in simulation which could already be achieved for the *Oculus Rift Dev Kit 2*. Another integration of the *HTC Vive* is under development. The *Oculus Rift Dev Kit 2* was already used for presenting the demonstrator at Automatica.

For the sensor pipeline development, we need sufficient data processing for visualizing the input from various sensors. This data processing and visualization is based on the internal sensor framework (see [10] for further information) at first, but can be modularly extended or replaced by external libraries, particularly ROS sensors and ROS-based data processing algorithms as discussed in Section 5. Thus, based on the developments in T4.2, we will be able to achieve a collaborative composition of internal functionalities and external frameworks with regards to sensor hardware, sensor data communication, sensor data processing and sensor data

[5] `https://www.oculus.com/en-us/rift/`

visualizations as required in CENTAURO.

8 Conclusions

Taking everything into consideration, we could already provide an integrated demonstrator which shows the capabilities of the 3D simulation setup (see Fig. 9 - green). Using a simplified robot model in a dynamic environment, controlled with diverse input devices, visualized in third and first person view already shows the overall system of operator, support and robot (simulation).

In addition, we also showed in two other demonstrators first prototypes to communicate also with real hardware - the exoskeleton and ROS interfaces. We achieved a stable in- and output of positions with the exoskeleton simulator from SSSA (see Fig. 9 - blue). We also used the *Momaro* simulation to test and evaluate our ROS interface (see Fig. 9 - red). These are promising results in the overall scope of the CENTAURO project.

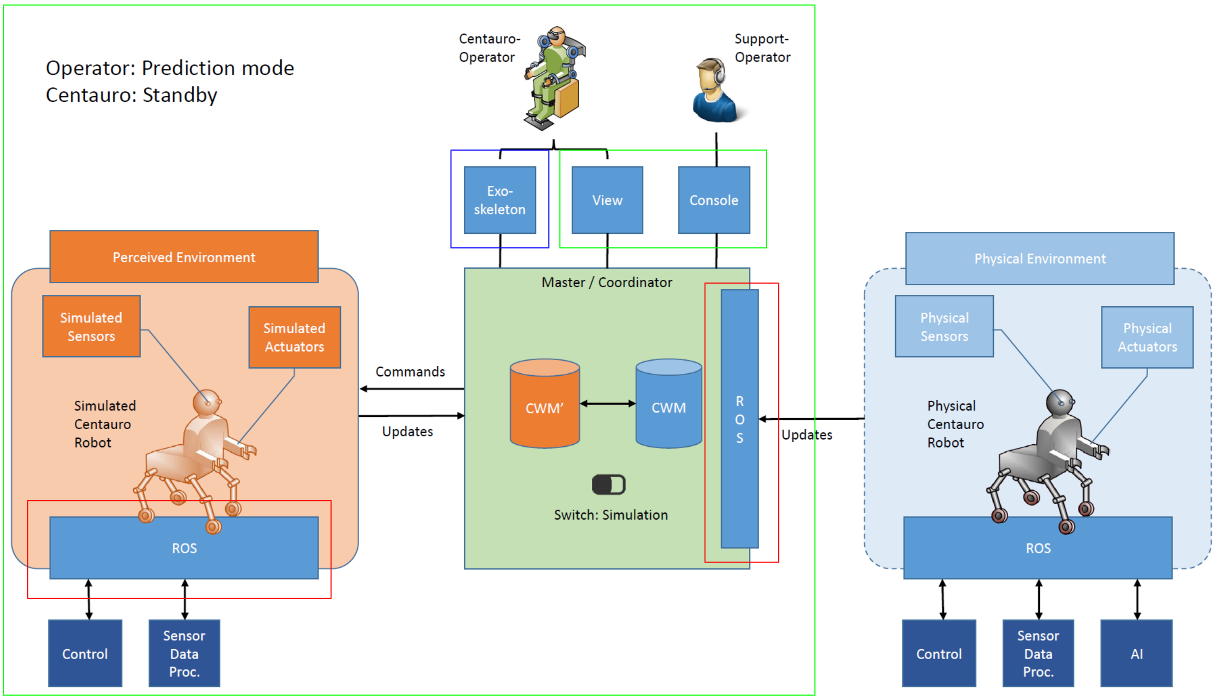


Figure 9: Demonstrators - general simulation system (green), ROS interface (red), and exoskeleton interface (blue).

In addition to the testing and optimization of these prototypical interfaces, further focus has to be put on sensor data, its processing, the automatic environment generation, and other data handling. Here, the implementation of the ROS-based interface to the *Momaro* setup by UBO was an important achievement, as the *Momaro* setup already provides variants of the functionality required in CENTAURO and represents a robotic field system of similar complexity. Still open is the continuous adaption of the robot model and simulated sensors in synchonization with the progress of WP2 as well as the modeling of a catalogue of testing scenarios according to the requirements.

Regarding the state-of-the-art, the individual interfaces to devices such as *Geomagic Touch X* and *Oculus Rift Dev Kit 2* as well as ROS interfaces and the integration of the exoskeleton are based on existing APIs and thus well-known. However, the deliverable D4.2 makes significant progress beyond the state-of-the-art in terms of combining all these individual interfaces in one comprehensive simulation system to support the purpose of a 3D simulation-based operator interface for a robotic field system.

References

[1] Leon Sze-Ho Chan and Kup-Sze Choi. Integrating physx and openhaptics: Efficient force feedback generation using physics engine and haptic devices. In *Pervasive Computing (JCPC), 2009 Joint Conferences on*, pages 853–858. IEEE, 2009.

[2] Torben Cichon, Claudio Loconsole, Domenico Buongiorno, Massimiliano Solazzi, Christian Schlette, Antonio Frisoli, and Jürgen Rossmann. Combining an exoskeleton with 3D simulation in-the-loop. *accepted*. 9th International workshop on Human-Friendly Robotics (HFR) 2016.

[3] Torben Cichon, Christian Schlette, and Jürgen Rossmann. Towards a 3D simulation-based operator interface for teleoperated robots in disaster scenarios. *accepted*. International Symposium on Safety, Security and Rescure Robotics (SSRR) 2016.

[4] Brandon Itkowitz, Josh Handley, and Weihang Zhu. The openhaptics toolkit: A library for adding 3D touch navigation and haptics to graphics applications. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 590–591. IEEE, 2005.

[5] Thomas Josef Jung. *Methoden der Mehrkörperdynamiksimulation als Grundlage realitätsnaher Virtueller Welten*. PhD thesis, RWTH Aachen University, 2011.

[6] Alireza Mohammadi, Mahdi Tavakoli, and Ali Jazayeri. Phansim: a simulink toolkit for the sensable phantom haptic devices. *Proceedings of the 23rd CANCAM, Canada*, 11:787–790, 2011.

[7] European Commission Directorate General for Communications Networks, Content and Technology Components and Systems: Robotics. Grant Agreement 644839: CENTAURO—Robust Mobility and Dexterous Manipulation in Disaster Response by Fullbody Telepresence in a Centaur-like Robot, 2014.

[8] Tobias Rodehutskors, Max Schwarz, and Sven Behnke. Intuitive bimanual telemanipulation under communication restrictions by immersive 3D visualization and motion tracking. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 276–283. IEEE, 2015.

[9] Juergen Rossmann, Thomas Josef Jung, and Malte Rast. Developing virtual testbeds for mobile robotic applications in the woods and on the moon. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4952–4957. IEEE, 2010.

[10] Jürgen Rossmann, Nico Hempe, Markus Emde, and Thomas Steil. A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–6. VDE, 2012.

[11] Kenneth Salisbury, Francois Conti, and Federico Barbagli. Haptic rendering: introductory concepts. *Computer Graphics and Applications, IEEE*, 24(2):24–32, 2004.

[12] Max Schwarz, Tobias Rodehutskors, Michael Schreiber, and Sven Behnke. Hybrid driving-stepping locomotion with the wheeled-legged robot Momaro.

Towards a 3D simulation-based operator interface for teleoperated robots in disaster scenarios

Torben Cichon, Christian Schlette and Jürgen Roßmann[1]

***Abstract*—**

Beyond robot hardware and control, one major element for an efficient, constructive and safe mission of teleoperated robots in disaster scenarios such as Fukushima is the quality of the interface between operator and robot. In this contribution, we present the concept of utilizing 3D simulation as a central interface component for the operator to intuitively collaborate with teleoperated robots. Thus, means of 3D simulation are not only used during the development but also in the deployment of the final field system. Based on this notion, we will discuss operator interfaces with regards a) to direct interaction with the robot, b) communication between control station and real robot and c) the integration of already acquired knowledge and existing libraries in the robotics community.

keywords: operator interface, virtual testbed, 3D simulation, force feedback, ROS

I. INTRODUCTION

Disaster scenarios such as at the Fukushima facility site clearly show that the capabilities of current disaster-response robot systems are hardly sufficient for providing the desperately needed support to reconnoiter and secure the situation – especially in the first critical hours.

Based on the state-of-the-art today, the operation of autonomous mobile robots in such highly unpredictable scenarios is not feasible in terms of algorithmic robustness as well as skillfulness of autonomous mobility and manipulation. Thus, the most realistic choice currently is the combination of the cognitive capabilities of a human operator with a highly mobile and dexterous teleoperated robot.

In such a robotic field system, 3D simulation can be utilized as a central component: Simulation can be used prior to the completion of the full system for the design, development and optimization of hardware and software components. Plus, simulation can be used in the field as well, as an additional interface between the operator and the robotic field system, e.g. to plan and verify next actions in simulation first before executing them in reality.

II. MOTIVATION

In todays mobile robots one main task is the optimization of direct control possibilities of the robot. As one could see at for example the DARPA Robotics Challenge (DRC) a huge amount of operators were necessary each responsible for one single task, like hand movement, sensor data pre-processing, or leg movements for instance. Thus, new interfaces are needed to ensure an intuitive interaction of operator and robot with less man power needed. We propose the use of only one main operator, responsible for all interaction tasks, and one support operator utilizing additional information of the robot's sensor data to assist the main operator. Special focus is put onto using 3D simulation in-the-loop of the final operator to ensure a stable, reliable, and easy to use man-robot interface.

The CENTAURO[2] project aims at the development of a novel teleoperated Centaur-like robot with whole-body telepresence of the human operator to allow for making elaborate decisions during the mission (see Fig. 1). In this context also new man-machine interfaces are in focus of the research which is why it has become the framework for this research. We will establish a safe cooperation where the operator is immersively present at the site of emergency, supported by situation-aware interpretations based on multi-modal information collected with the robot sensors as well as a-priori knowledge from other sources, e.g. 2D maps. At the Institute for Man-Machine Interaction (MMI), we develop such specialized operator interface based on 3D simulation technologies.

Fig. 1: Basic idea of the CENTAURO project – the joint hardware/software development of a teleoperated robotic field system for disaster-response.

Starting in section III, we will motivate the underlying concept of this approach by related work in the fields of teleoperation, operator interfaces, force feedback and ROS. In section IV we give more details on the goals and requirements of our approach in the context of the CENTAURO project. The resulting requirements for the 3D simulation-based operator interface are presented in section V. In order to meet the requirements we develop a concept (see

[1]All authors are with the Institute for Man-Machine Interaction (MMI), RWTH Aachen University, 52074 Aachen, Germany
{cichon,schlette,rossmann}@mmi.rwth-aachen.de

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

section VI) and an implementation plan (see section VII) to identify and integrate the crucial components in one central simulation framework. Finally, we conclude our findings and show prospects for further developments in teleoperated robotics in combination with 3D simulation.

III. RELATED WORK

To give a comprehensive overview of related work we discuss teleoperation, its use in disaster scenarios, 3D simulation technology, operator interfaces, force feedback and the Robot Operating System (ROS) as the preferred communication platform in joint research projects in robotics.

A. Teleoperation

Teleoperation – in general the operation of a machine from a distance – is commonly associated with robotics. In our application, we aim for teleoperation combined with telepresence: The operator remotely controls a semi-autonomous robot and is additionally supported by multi-modal, immersive information from the scene. An overview of bilateral teleoperation is given in [1], where the two main goals of teleoperation are specified as "stability" in terms of system control and "telepresence" regarding the transparency of the robotic system, the environment and the operator. [2] gives an overview on the general history of teleoperation interfaces for mobile robots and vehicles and summarizes their central aspects: "[...] teleoperation interfaces provide tools and displays to perceive the remote environment, to make decisions and to generate commands."

B. Teleoperated robots in disaster scenarios

The ability to project the expertise of a given operator to another location and the ability to scale movements and other actions have already yielded many applications of teleoperation, ranging from surgery to robot platforms for various environments, e.g. underwater and in the air. However, disaster scenarios such as Fukushima in 2011 newly underpinned the need for teleoperated robots to safely act in highly hazardous and contagious environments. This demand forstered several highly funded challenges, in particular the DARPA Robotics Challenge (DRC), which focussed on the development of robot technologies for disaster-response tasks. In order to come up with stable system control, the most successful teams in the DRC concentrated not only on the development of reliable robot hardware but also of human-in-the-loop control schemes where the control stations are a few hundred of meters away from the robot on the course [3][4][5].

One standard setup of teleoperated robots and their operators in such missions is shown in Fig. 2. According to the setup in Fig. 2, the robot and the human operators have to work in concert to perform difficult tasks. Besides challenges for the robot hardware, this shows the need for optimized human-robot interaction, particularly optimized, intuitive user interfaces for teleoperation. The system setup has to withstand and to recover from loss of communication to the robotic field system due to difficult environments. To overcome communication loss or instabilities is one major aspect in todays research in mobile rescue robotics, which are mainly related to the robotic hardware.

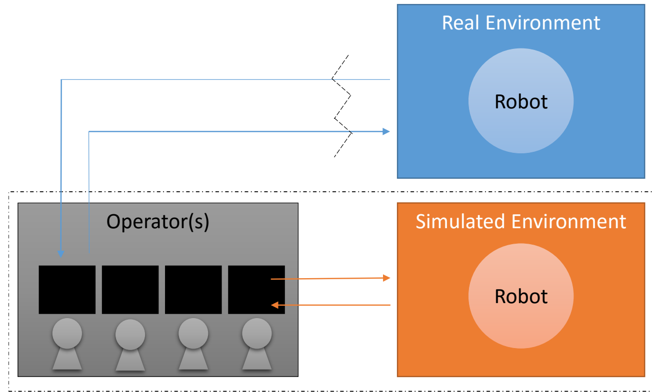


Fig. 2: Teleoperated robot setup: Spatially separated from the teleoperated robot, one or more operators interpret the incoming data in order to issue commands to the robotic field system. Simulation may be applied for basic planning etc.

C. 3D Simulation Technology

As depicted in Fig. 2, simulation plays a role in control schemes for teleoperated robotic field systems only in some rare and rather limited cases. Simulation in teleoperation is mostly used for testing and validation of individual modules or algorithms during development. These simulation tools are focussing on individual aspects or specific application areas, e.g. ROS Gazebo [6], which is also applied in the aforementioned DRC. A more holistic approach to 3D simulation in robotics is provided by the eRobotics methodology [7][8][9] and so-called Virtual Testbeds, where complex technical systems and their interaction with prospective working environments are first designed, programmed, controlled and optimized in 3D simulation, before commissioning the real system.

D. Operator Interfaces

Fong et al [2] define four major interface types for (vehicle) teleoperation: "Direct" (low level control), "Multimodal/Multisensoral" (telepresence), "Supervisory Control" (high level control) and "Novel" (including Virtual Reality and haptic interfaces). An introduction to operator interfaces utilizing Augmented Reality and Virtual Reality is given in [10], where they are applied in search-and-rescue applications. Sheridan summarizes the state-of-the-art of human robot interaction in [11] by reviewing current challenges ranging from supervisory control of robots for routine industrial tasks to teleoperated vehicles and planes and human-robot social interaction. He concludes that, "[w]e need to revisit the discussions of where humans best fit into systems as compared with AI and computer control". As a central method in modern robot control Sheridan sees much potential in so-called mental models as "[...] built-in models of what is going on in the environment that are continually updated, much as what humans seem to do." Using simulation on combination with mental models is motivated in [12] as an conceptual extension of Virtual Testbeds towards simulation-based control and simulation-based support.

E. Force Feedback

Force feedback is one element of haptic feedback, which compromises force feedback, tactile feedback, and proprioceptive feedback [13]. Although teleoperation is mainly based on audio-visual feedback today, also force feedback starts to have more and more applications. Several force feedback devices are commercially available, in particular the 6 DOF *Geomagic Touch X* (formerly Phantom Device) [3] as the most common one. A general overview about history, complexity and benefits of haptic interfaces in simulation is given in [14]. From a technical point of view, the interface between simulation and (any) force feedback device should be the same: "Force feedback interfaces can be viewed as computer extensions that apply physical forces and torques on the user." [13].

F. Robotic Operating System (ROS)

The preferred platform for system development in (mobile) robotics currently is the Robot Operating System (ROS). With standardized formats for message and service deployment, ROS has been established as a standard for intra- and interprocess communication between hardware and softare components. ROS was developed as a flexible framework for writing robot software [15] . In terms of infrastructure, ROS can be categorized into three layers, a) file system, b) computation graph and c) community. On the other hand, in technical terms, the framework can be subdivided into a) communication infrastructure, b) robot-specific features and c) tools. At its lowest infrastructural level, ROS offers interprocess communication via passing messages. In addition, ROS provides a specific robot description language (URDF), powerful development tools (e.g. *rviz* and *rqt*) as well as a huge set of reference implementations of important methods and algorithms in robotics. Especially in robotics research, ROS is very common as middleware to link up arbitrary software components in a network for interoperability. Hence, it is beneficial for any software developement in robotics to be able to connect to at least the message interface.

IV. CENTAURO PROJECT

In the context of the CENTAURO project, we develop novel operator interface for teleoperated robotic field systems in disaster scenarios by extending the existing approaches described in section III resp. Fig. 2. We aim for enabling the operator to effectively combine the strengths of direct robot control in real-time with simulation-based support to develop elaborate decisions. Thus, the operator interface ideally allows for mixing multi-modal information from exteroceptive and proprioceptive sensors with semi-autonomous functionalities as means of supervisory control. Plus, the operator interface should enable the inclusion of one or more support operators to relieve the main operator from the huge amount of information coming in from telepresence

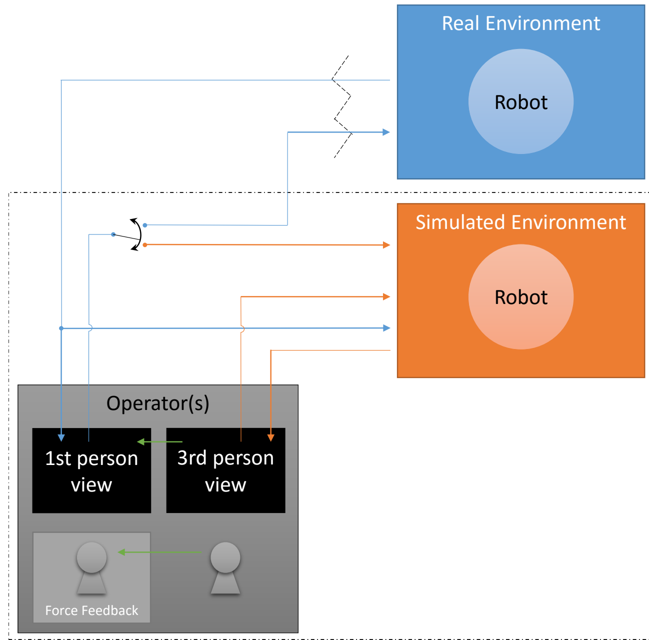


Fig. 3: Teleoperated robot setup enhanced with 3D simulation-based support of additional operators. Based on the Virtual Testbed, the operator can switch between controlling the simulated or the real robotic field system, e.g. to program and test manipulation actions in simulation first before commanding the action in reality. Additionally, the second operator can give hints or send visual AR overlays to the main operator.

In our previous work we utilized Virtual Testbeds as integrated development and simulation platforms, which compromise system models as well as environment models and connect them with simulation methods and algorithms, e.g. for perception and control. As Virtual Testbeds are designed to represent complex technical systems during development, they enable the management of interfaces which can be switched from simulation to reality, in order to maintain rapid prototyping techniques such as hardware-in-the-loop resp. software-in-the-loop. In addition, selected parts of Virtual Testbeds can be run under real-time conditions to directly interface the real target system by means of simulation-based control.

Based on the concepts and findings in [12], we thus decide to utilize a Virtual Testbed to meet the aims and requirements of the operator interface in the CENTAURO project. They allow us to use the same input devices, control algorithms, sensor data processing etc. in 3D simulation as well as in reality and thus to interface with the simulated CENTAURO robot equivalently as to the real robot. Following this approach, 3D simulation is available during the development of the robot, and more importantly also as the central system for providing the operator interface for field missions. It is then

[3] http://www.geomagic.com/en/products/phantom-desktop

fit to safely test and verify actions in simulation first, before executing them in the real world with adequate guidance and support for the operator.

This characteristic is often referred to as ”digital twin” of the real system. There are specific requirements for the simulation framework to accomplish this behavior. Next to real-time performance, the central aspects are **integration** and **interfaces**: All necessary functionality for the simulation and simulation-based control of a robot in its environment has to be integrated ideally in a modular, complementary structure. Additional interfaces are needed to connect to the state-of-the-art in robot control software frameworks (here ROS) to incorporate developments by our project partners into the resulting robotic field system.

The simulation-based operator interface then encompasses

- the direct control of the final robotic field system,
- the possibility to test and validate actions of the robot in a virtual environment, which is built from a-priori knowledge as well as real sensor data,
- the possibility to incorporate the state-of-the-art in terms of control algorithms etc., utilizing ROS as a commonly accepted communication link.

The final setup addressing these points is drafted in Fig. 3, whereas Fig. 4 represents the direct control of the real robot by a first person operator. Fig. 5 shows the equivalent simulated digital twin supervised by a support operator with a third-person view on the robot and its environment. In Fig. 3 the seamless switch of operating the real robot or its digital twin is shown, supported by a second operator utilizing the 3D simulation.

To overcome the complexity of the robotic system, giving all necessary data to the user and leading to a complete system of interoperable robots, we propose the 3D simulation-based interface for the operator. Thus, the simulation serves as an interface to combine a defined degree of robotic automation with software-based mental models extends semi-autonomous robots with the decision possibilities of a human operator ”present” in the scene.

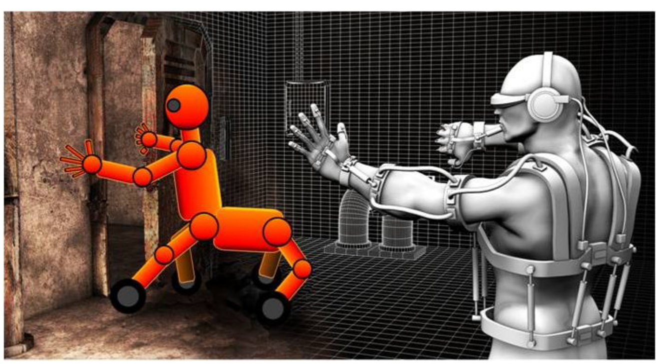


Fig. 4: Conceptual drawing – Direct control of the main operator using force feedback.

V. REQUIREMENTS

The application of teleoperated robots in disaster scenarios leads to the following requirements for the operator interface,

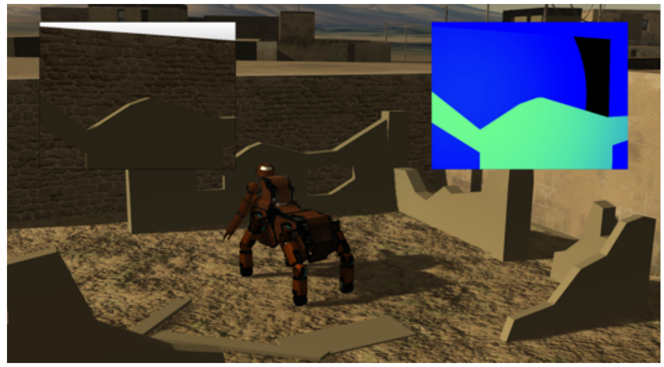


Fig. 5: Actual result – Third person view of the support operator based on 3D simulation.

it should

- reduce the complexity of teleoperation,
- overcome the lack of mobility and manipulation skills of robots,
- ”stabilize” the communication link between robot and operator station,
- transfer huge amounts of sensor data from and to the robot (data storage, data preprocessing),
- reduce enormous workload for the operator(s) to reduce the risk of hazardous decisions.

Therefore, besides robust but dexterous and versatile robot hardware, a comprehensive operating system allows for intuitively controlling a highly complex teleoperated robot as well as for representing resp. visualizing the necessary data for the operator to achieve an immersive control and supervision of the robot. As described in section IV, we aim for addressing these points based on a Virtual Testbed in combination with commonly used (haptic) input devices and ROS. Additionally, simulation could be ”used” to stabilize the communication link: In case of communication loss, the real robot would rest in a predefined state, while the operator could already plan the next steps in simulation. Such unstable communication areas could be indicated in simulation, so that the operator could consider this knowledge in path planning.

VI. CONCEPT

Regarding our concept, the key aspect is the availability of a rigid body based physics simulation of the robot and its environment throughout the development process as well as during field operation. A human operator will control the robot intuitively using a full-body telepresence suit including force feedback. He will be supported by a second operator from a third-person perspective. Control tasks are mainly executed by the main operator, whereas the support operator can use the simulation in parallel to give hints and push visual assistance in the focus of the first operator. The simulation itself serves as an additional interface to the operator, so the complexity of robot hardware is hidden and the workload of the operator is reduced. As a result, the concept of the operator interface comprises the holistic interface between user and robotic field system.

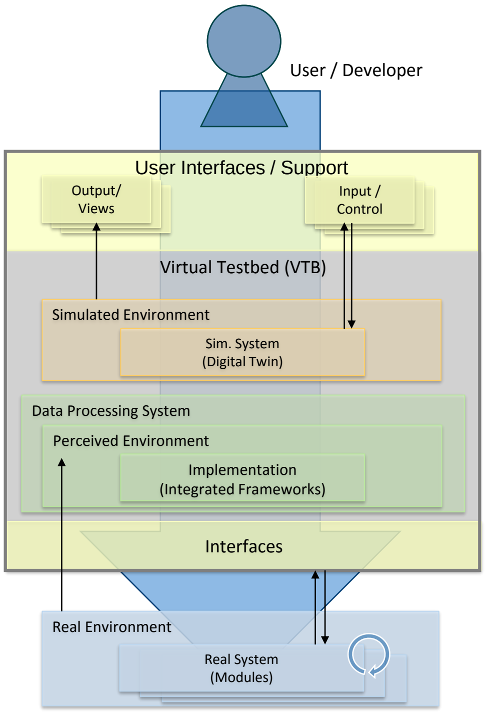


Fig. 6: Concept of the 3D simulation-based operator interface utilizing a Virtual Testbed.

As depicted in Fig. 6, our concept involves a modular 3D simulation system – resp. the Virtual Testbed (VTB) – as the central integration platform which integrates various modules for data processing and visualization. Additional interfaces are now on the one hand the final operator interface for the user with an immersive force feedback interface, and on the other hand the direct interface to ROS. This results in a (simplified) three-layered structure of interfaces, integrated functionalities in the VTB and interfaces to other systems. This structure is what we then call **3D simulation-based operator interface**, which allows operators as well as developers to easily and directly connect to the real robotic field system.

Using simulation as an additional abstraction layer this 3D simulation-based operator interface involves:

- the modular integration of additional needed functionality in simulation,
- the interface of user and simulation,
- the interface of simulation and real hardware,
- and the interface of simulation with existing external software libraries.

This concept defines the necessary background to develop, test and optimize a virtual setup of a real robot in different virtual environments. Furthermore, it enables the operator in field missions to use the simulation in the loop for testing and evaluating proposed tasks and to effectively evaluate sensor data from the robot.

VII. SYSTEM IMPLEMENTATION

The following section describes the implementation of the proposed concept. In terms of feasibility one has to distinguish between integration and interfaces: While the integration of functionalities in a holistic simulation environment is often favorable in terms of real time requirements and interoperability, it might also be reasonable in other cases to establish interfaces to commonly used and accepted other software (frameworks). Thus, we use a Virtual Testbed as a basis for integrative developments as well as extended interoperability with other frameworks to create an overall system of robot hardware, software, simulation and most importantly the operator.

A. Modular 3D Simulation System

In CENTAURO we develop a Virtual Testbed which comprises the relevant system components and enables early integration, testing and evaluation of system modules from our project partners. Core aspect is the development of a physical simulation of the robot in interaction with its environment as well as establishing a Central World Model (CWM), which can be updated from the percepts and actions of the robot. A predictive model for the robot-environment interaction will support the operators by enabling them to estimate the future behavior of the robot in order to evaluate alternative actions during missions.

For our approach we use the *VEROSIM*[4] system which we co-develop at MMI. The modularity of *VEROSIM* enables us to easily integrate additional functionality, as interfaces to and from *VEROSIM* have to be established to communicate with other (given) frameworks and assimilate available prior developments, knowledge or modules.

B. Data Processing and Visualization

We have seen in section III that although system development in robotics often focusses on robot hardware and control software, the operator interface is of paramount importance to enable modes of teleoperation and telepresence. In particular the visualization of the sensor data collected from optical and other sensors has to be pre-processed and made available in an intuitively and understandable manner.

Due to the fact that one person will not be capable of supervising the robot alone, we propose the introduction of one or more support operators who provide additional, necessary information for the pilot. For the two types of operators, we accomplish two different views on the scene, 1) an immersive first person view using a head mounted stereoscopic display (here the *Oculus Rift*[5]), and 2) a third person view onto the whole scene in simulation.

For this development we need sufficient data processing for visualizing the input from various sensors. This data processing and visulization is based on the internal sensor framework (see [16] for further information) at first, but can be modularly extended or replaced by external libraries,

[4]https://www.youtube.com/user/VEROSIMSimulations
[5]https://www.oculus.com/en-us/rift/

particularly ROS sensors and ROS-based data processing algorithms as discussed in section VII-D.

In addition, the *Oculus Rift* stereoscopic view has to be implemented in simulation. In the end, we achieve a collaborative composition of internal functionalities and external frameworks with regards to sensor hardware, sensor data communication, sensor data processing and sensor data visualizations.

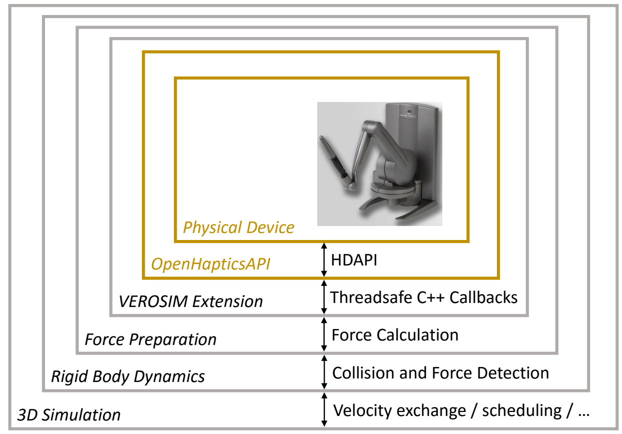
C. Force Feedback Interface

The integration of force feedback in 3D simulation environments is not quite common in current research. As stated before, force feedback devices such as the *Geomagic Touch X* depicted in Fig. 7b are most commonly used as three-dimensional input devices for modeling. Only in some rare applications, the devices are also used in specialized simulation environments, such as surgical simulations, where force feedback is then the main aspect of simulation. Integrating force feedback into a rigid body based simulation framework is therefore an advancement of the given technology.

Using the *Geomagic Touch X* as a sample force feedback input device, we developed a generic interface to couple rigid body dynamics based force generation, force reprocessing and specialized driver interfaces for each force feedback device As a result, the overall force feedback interface implements three layers:

1) Intertwining of dynamic simulation and events of force feedback calculation at time t_{FF},
2) Generic interface for force feedback devices, calculating a generic force feedback force F_{FF} at the time t_{FF},
3) Specialized driver interfaces for each haptic device, e.g. Phantom Device, which
    a) transmit the calculated force F_{FF}^{TouchX},
    b) and provide positional input p^{TouchX}.

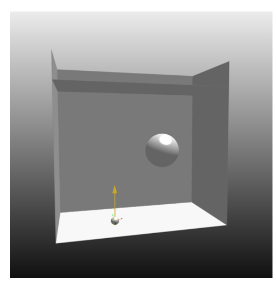
Starting with the *Geomagic Touch X*, we used the freely available *OpenHaptics API*[18] to implement the driver interface to the *Geomagic Touch X*, while the deeper layers were achieved in *VEROSIM*. The modularity of *VEROSIM*, and thus the associated independence of individual modules, requires a systemic management of force feedback in time, space and magnitude. As one can see in Fig. 7a, the *OpenHaptics API* is just used for low level interfacing the physical hardware. Visible for the user in the 3D simulation is just an extension that manages a thread-safe communication channel between *VEROSIM* and the *OpenHaptics API*. On a higher level, the collision and force detection, calculation and scheduling is of paramount importance. We implemented and validated a collision-based determination of each force feedback event ($\rightarrow t_{FF}$). The setup can be seen in Fig. 7d) and Fig. 7e where a simple 3D model is used to test the collision detection in *VEROSIM* and the force vectors are analyzed in Matlab. Now, either a) the calculated force on interacting rigid bodies F_{RB} can be used as force feedback, b) the force of specific force torque sensors (F_{FT}) e.g. in the joints – or c) a more general approach, where the virtual coupling is based on a mass-spring-damper system as found in [19][20]. In c), a variance analysis of current position and target position is used to calculate a (virtual) spring-damper based force (F_{SD}) (see Fig. 7e). This procedure has the advantage of equal force dimensions, irrespective of the two colliding bodies. Otherwise the calculated collision force could become too high or too volatile for the force feedback device. As a result, we use c) for force direction and magnitude calculation, the integrated dynamic rigid body framework for collision detection and a separate thread to safely collaborate with the *OpenHaptics API*. This interface, using the *TouchX*, is implemented, coupled with the rigid body dynamics, and is already in practical use. Experiements,



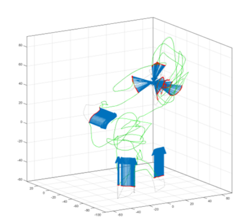
(a) Modular force feedback concept chart (chart idea based on [17]). Using a modular organization, the physical device and its API can be easily exchanged. The connection of simulation scheduling, rigid body dynamics, collision detection and force preparation is carried out in *VEROSIM*.

(b) *Geomagic Touch X*

(c) 3D model of *Touch X*



(d) Force feedback from simulation in *VEROSIM*



(e) Force feedback analyzed in MATLAB

Fig. 7: The force feedback interface couples collision detection, spring-damper dynamics and the haptic device.

evaluation and optimization is still done with regards to feedback force calculation and preparation in general.

Of course, using haptic devices for positional input also leaves room for interpretations. We assume either a direct positional input of a frame to move a given rigid body in time and space, or we use the transfered joint-angles directly, which are then mapped to an equivalent virtual model, e.g. of the device (see Fig. 7c). Also with respect to other input devices, like an exoskeleton, it will always be possible to transmit either end-effector position or a given set of joint angles or torques. Device specific characteristics can be managed independently of the generic interface in the manufacturer-specific driver interfaces resp. in specialized *VEROSIM* extensions.

D. *ROS Interface*

We implemented a generic integration of ROS into *VEROSIM* to enable to use the full spectrum of ROS functionalities from within the 3D simulation and thus the operator interface. As we stated before, implementing an interface to the communication infrastructure of ROS aims at connecting the message passing system with *roscore* to open up many possibilities regarding other core components of ROS. In order to make use of the knowledge already available from prior ROS setups, we use the Momaro setup [4][21], a working robotic setup based on ROS. The milestone to achive was to be able to resemble the features in the Momaro setup to test and verify the developed interface.

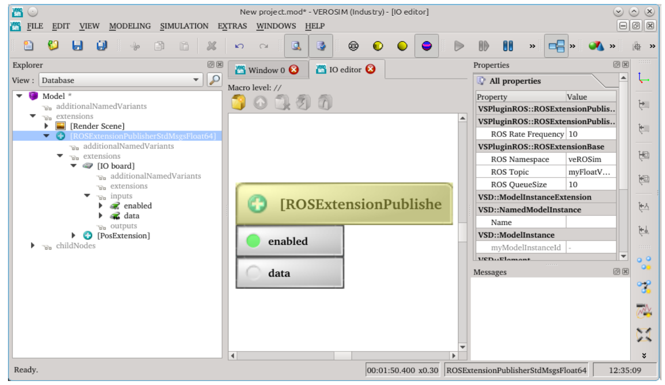
The Momaro setup is mainly based on standard ROS data types. Thus, we started the implementation in *VEROSIM* with according standard message types as well as combined message types for the central input/output board (IO Board) [7] which then allows for dynamically connecting internal functionalities with the ROS framework. As a result, internal scheduling, rendering and the other frameworks can utilize the inputs and outputs of ROS nodes from within *VEROSIM*.

The following implementation scheme is used to continuously add ROS functionalities to *VEROSIM* using the *roscpp* (and *rospy*) API:

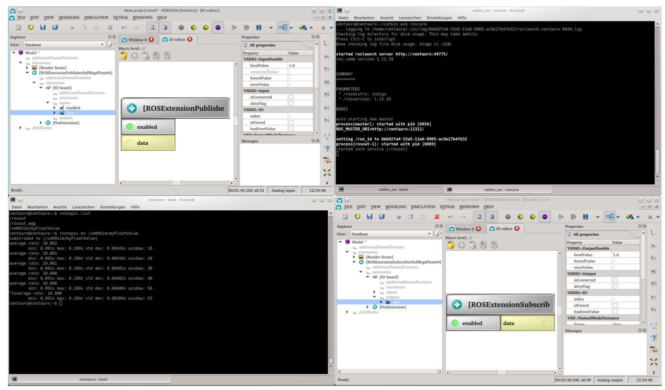
1) Implementation of static data type conversions for *std_msg* types
2) Implementation of static data type conversions for combined *std_msg* and specialized types
3) Implementation of template-based conversions for arbitrary message types
4) Completely dynamic (Python-based) embedding of ROS data types into *VEROSIM*

Step by step, this will lead to a continuous integration of more data types according to the requirement analysis. Finally, it will be possible to use a given set of ROS message types from within simulation (input and output). In addition, we will offer an easy way to add new functionalities based on C++ templates, or even use Python for dynamic ROS message implementation.

Using this implementation scheme already led to first results, visualized in Fig. 8. Fig. 8a shows a ROS publisher extension in the *VEROSIM* IO Board to publish a *float64* in



(a) *VEROSIM* IO Board using dynamic input from simulation to publish, and submit to ROS messages via outputs in simulation.



(b) Using ROS with *VEROSIM*: (top/left) *VEROSIM* issuing data using ROS publisher, (top/right) console running *roscore*, (bottom/left) console for inspection with *rostopic*, (bottom/right) second instance of *VEROSIM* reading the data using ROS subscriber.

Fig. 8: ROS Interface to *VEROSIM*.

the ROS namespace ”veROSim” on the ROS topic ”myFloatValue”. In Fig. 8b this ROS *std_msg float64* publisher and its corresponding subscriber are used to connect two instances of *VEROSIM*. As a result, the input data from the top/left *VEROSIM* is published and thus transferred via the ROS network to the subscriber in the bottom/right *VEROSIM*.

We are already using the implemented ROS nodes and are currently implementing according sensor message types to be connected to the integrated sensor framework of *VEROSIM* to utilize both, ROS and *VEROSIM* sensor data processing and visualization.

E. *Complete System*

Our concept intertwines integrated functionalities from a 3D simulation system with external libraries, mainly from the ROS context. This leads to a modular, flexible and robust overall setup, enabling an optimized 3D simulation-based operator interface. The overall setup as shown in Fig. 9 includes an optimized operator interface for controlling robots in disaster scenarios by means of immersive and intuitive control from a first person perspective with the *Oculus Rift* and force feedback using the *Geomagic Touch X* device. Accompanied by a supporting third person view, overlays for sensor data visualization and a direct interface to ROS,

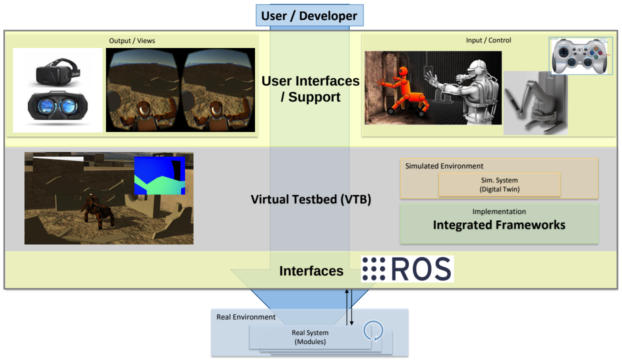


Fig. 9: Final setup variant with respect to Fig. 6.

the operator interface is able to effectively support the first operator in his decisions.

VIII. CONCLUSIONS

We presented the concept of a 3D simulation-based operator interface which comprises the development of the simulation technology necessary to setup a high fidelity operator interface consisting of simulatable models of the robot and its environment as well as means for intuitive interaction and visualization to safely operate and supervise the remote robot with multiple operators using various devices. The concept provides a fully operable virtual robot including mechanics, actuators, sensors as well as control algorithms which can act in various virtual environments.

In addition, the operator interface enables the direct access to the common ROS middleware. Thus the interface is prepares for connecting the simulation with customized ROS nodes and the reuse of existing ROS modules. This holistic setup opens many possibilities regarding the development process of teleoperated robots and also during the final mission.

The use of force feedback devices supports the operator in his mission by means of intuitive control and the positive effects of immersion, and hence being telepresent at the site of operation accompanied by simulation and supported by pre-processed data. The integration of force feedback in simulation in general opens up prospect to a huge amount of applications to dive into virtual realities prior to the real setup.

In summary, the 3D simulation-based operator interface for teleoperated robots in disaster scenarios applies 3D simulation to generate new integration and interface options, to establish an intuitive and effective combination of robotic field system, data exchange and processing algorithms and to the coordination of human control and supervision.

ACKNOWLEDGEMENT

This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 644839.

REFERENCES

[1] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[2] T. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Autonomous robots*, vol. 11, no. 1, pp. 9–18, 2001.

[3] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padır, S. Feng *et al.*, "Human-in-the-loop control of a humanoid robot for disaster response: A report from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 275–292, 2015.

[4] M. Schwarz, T. Rodehutskors, M. Schreiber, and S. Behnke, "Hybrid driving-stepping locomotion with the wheeled-legged robot momaro," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 1, pp. 49–55, 2016.

[5] J. Lim, I. Shim, O. Sim, H. Joe, I. Kim, J. Lee, and J.-H. Oh, "Robotic software system for the disaster circumstances: System of team kaist in the darpa robotics challenge finals," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 1161–1166.

[6] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2149–2154.

[7] J. Rossmann, T. J. Jung, and M. Rast, "Developing virtual testbeds for mobile robotic applications in the woods and on the moon," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 4952–4957.

[8] J. Rossmann, M. Schluse, B. Sondermann, M. Emde, and M. Rast, "Advanced mobile robot engineering with virtual testbeds," in *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*. VDE, 2012, pp. 1–6.

[9] J. Rossmann, E. G. Kaigom, L. Atorf, M. Rast, and C. Schlette, "A virtual testbed for human-robot interaction," in *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*. IEEE, 2013, pp. 277–282.

[10] J. Rossmann, A. Kupetz, and R. Wischnewski, "A universal approach for the intuitive control of mobile robots using an ar/vr-based interface," *Proc. World Acadamy of Science, Engineering and Technology (WASET)*, no. 67, 2010.

[11] T. B. Sheridan, "Human–robot interaction status and challenges," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 58, no. 4, pp. 525–532, 2016.

[12] T. Cichon, M. Priggemeyer, and J. Roßmann, "Simulation-based control and simulation-based support in erobotics applications." *Applied Mechanics & Materials*, vol. 840, 2016.

[13] G. C. Burdea, G. C. Burdea, and C. Burdea, *Force and touch feedback for virtual reality*. Wiley New York, 1996.

[14] G. C. Burdea, "Haptic feedback for virtual reality," in *Proceedings of the Virtual Reality and Prototyping Workshop*. Citeseer, 1999.

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

[16] J. Rossmann, N. Hempe, M. Emde, and T. Steil, "A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications," in *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*. VDE, 2012, pp. 1–6.

[17] A. Mohammadi, M. Tavakoli, and A. Jazayeri, "Phansim: a simulink toolkit for the sensable phantom haptic devices," *Proceedings of the 23rd CANCAM, Canada*, vol. 11, pp. 787–790, 2011.

[18] B. Itkowitz, J. Handley, and W. Zhu, "The openhaptics toolkit: a library for adding 3d touch navigation and haptics to graphics applications," in *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*. IEEE, 2005, pp. 590–591.

[19] L. S.-H. Chan and K.-S. Choi, "Integrating physx and openhaptics: Efficient force feedback generation using physics engine and haptic devices," in *Pervasive Computing (JCPC), 2009 Joint Conferences on*. IEEE, 2009, pp. 853–858.

[20] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *Computer Graphics and Applications, IEEE*, vol. 24, no. 2, pp. 24–32, 2004.

[21] T. Rodehutskors, M. Schwarz, and S. Behnke, "Intuitive bimanual telemanipulation under communication restrictions by immersive 3d visualization and motion tracking," in *Humanoid Robots (Humanoids),*

2015 IEEE-RAS 15th International Conference on. IEEE, 2015, pp. 276–283.

Combining an exoskeleton with 3D simulation in-the-loop

Torben Cichon[1], Claudio Loconsole[2], Domenico Buongiorno[2], Massimiliano Solazzi[2]
Christian Schlette[1], Antonio Frisoli[2], and Jürgen Roßmann[1]

***Abstract*—**

Beyond robot hardware and control, one major element for an efficient, constructive and safe mission of teleoperated robots in disaster scenarios such as Fukushima is the quality of the connection between operator and robot. In this contribution, we present the concept of using an exoskeleton and utilizing 3D simulation as a central interface component for the operator to intuitively collaborate with mobile teleoperated robots.

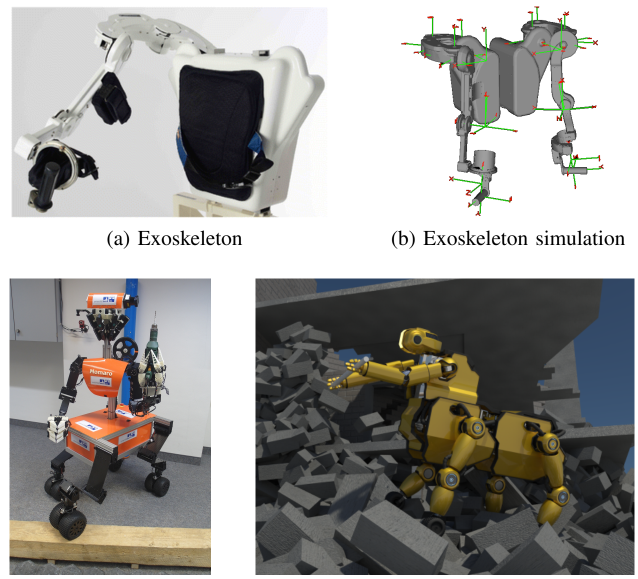
keywords: 3D simulation, exoskeleton, force feedback, operator interface

I. INTRODUCTION

Disaster scenarios such as at the Fukushima facility site clearly show that the capabilities of current disaster-response robot systems are hardly sufficient for providing the desperately needed support to reconnoiter and secure the situation – especially in the first critical hours.

The CENTAURO[3] project aims at the development of a novel teleoperated Centaur-like robot with whole-body telepresence of the human operator supported by 3D simulation in-the-loop, to allow for making elaborate decisions during the mission. Hence, the project will establish a safe cooperation where the operator is immersively present at the site of emergency, supported by situation-aware interpretations based on multi-modal information collected with the robot sensors as well as a-priori knowledge from other sources, e.g. 2D maps. The exoskeleton and a specialized exoskeleton simulator, used during the implementation, are developed at SSSA. At the MMI, a specialized force feedback interface for this exoskeleton based on 3D simulation technologies is developed.

The overall CENTAURO setup is shown in Figure 1. Based on prior knowledge in developing mobile robots, like the Momaro robot ((c), [1]), a holistic setup is developed consisting of a new Centaur-like robot, an exoskeleton for control (a), and 3D simulation for support (d). During the development process, special focus is put on the 3D simulation system and also an exoskeleton simulation (cf. (b)) to develop necessary interface structures used also in the final setup. The operator can use the information gathered from simulation and additionally switch seamlessly between real world interaction and its virtual counterpart. This feature will be used in risky situations to evaluate movements or actions in the virtual world first, before executing them in the real hazardous environment.



(a) Exoskeleton (b) Exoskeleton simulation

(c) Real Centaur-like mobile robot[3] (d) 3D simulation of robot and environment

Fig. 1: Using an exoskeleton with force feedback for robotic teleoperation, utilizing 3D simulation

II. RELATED WORK

A. Exoskeleton

The robotic interfaces for physical human-robot interaction represent an important aspect of tele-existence cockpits [2]. The exoskeleton represents the robotic system where the highest physical symbiosis with the human operator is achieved. Active exoskeleton systems are robotic devices that can be worn on the user's body, implying that they should satisfy requirements of safety and better compliance. Exoskeletons built for rehabilitation and human power augmentation make use of different actuation solutions, such as geared solutions, tendon drives, hybrid solutions (screw and cable actuators) or variable-impedance actuators [3], [4], [5], [6], [7], [8], [9]. Based on the adopted actuation, active exoskeletons can be classified as impedance based design (open-loop impedance control and impedance control with force feedback) or admittance-based design (admittance control with position feedback).

[1]Authors are with the Institute for Man-Machine Interaction (MMI), at the RWTH Aachen University, 52074 Aachen, Germany
cichon@mmi.rwth-aachen.de

[2]Authors are with the Perceptual Robotics Laboratory (PECRO), at the Scuola Superiore Sant' Anna (SSSA), 56127 Pisa, Italy
c.loconsole@sssup.it

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

B. 3D Simulation Technology

Normally, simulation does not really plays a role in control schemes for teleoperated robotic field systems only in some rare and rather limited cases. It is mostly used for testing and validation of individual modules or algorithms during development. A more holistic approach to 3D simulation in robotics is provided by the *eRobotics* methodology [10][11][12][13] and so-called *Virtual Testbeds*. Complex technical systems and their interaction with prospective working environments are first designed, programmed, controlled and optimized in 3D simulation, before commissioning the real system. In our previous work we utilized 3D simulation already as integrated development and simulation platforms, which compromise system models as well as environment models and connect them with simulation methods and algorithms, e.g. for perception and control. Now, the simulation is used during the development process of robot and the exoskeleton, but more importantly will it also serve as the central system for providing the operator interface during field missions.

C. Force Feedback in 3D Simulation

Although, force feedback and corresponding devices are not new, their use in simulation is quite limited. Only specialized applications can be found where force feedback is used as one central compartment of simulation. Several force feedback devices are commercially available, in particular the 6 DoF *Geomagic Touch X*[4] (formerly Phantom Device) as the most common one. A general overview about history, complexity and benefits of haptic interfaces in simulation is given in [14]. From a technical point of view, the interface between simulation and (any) force feedback device should be the same and "can be viewed as computer extensions that apply physical forces and torques on the user." [15].

III. RESULTS

The following section describes the results in terms of combining an exoskeleton, force feedback and 3D simulation. On the one hand, the development of the exoskeleton and corresponding exoskeleton simulator is described. On the other hand, the required force feedback integration in 3D simulation and its interface to the exoskeleton (simulator) is presented.

A. Exoskeleton and Exoskeleton Simulator

The exoskeleton designed within the framework of the CENTAURO project (see Figure 2) is based on ALEx robot [5], a 12 DoFs (6 DoFs$\times$2 upper limbs) mechanically compliant exoskeleton for the human upper limb: 4 DoFs per arm are sensorized and actuated (shoulder abduction, rotation, and flexion; elbow flexion), and 2 DoFs per arm are sensorized and passive (forearm prono-supination and wrist flexion). However, the CENTAURO Master exoskeleton will substitute passive DoFs and will include additional DoFs for wrist and hand actuation to allow also the manipulation of objects through the teleoperated Centaur-like robot. More in detail, there will be 3 DoFs for each wrist and 17 underactuated DoFs (actually 5 DoFs) for each hand. The entire CENTAURO Master exoskeleton can reach about 90 % of the natural workspace of the human arm without singularities, covering an extended range of motion for each DoF. Moreover, the exoskeleton can be operated either in force mode, providing desired input forces to the EE or joint torques to each joint, or in compliant position mode, providing desired trajectories with the associated stiffness to the EE or to the joints.

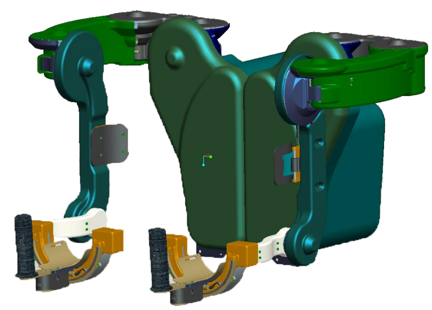


Fig. 2: The ALEx exoskeleton for upper limb.

A simulator of the CENTAURO Master exoskeleton has been designed for preliminary interaction with 3D simulation of the disaster scenario. The simulator includes the kinematic and dynamic models of the exoskeleton and relies on a physical model engine. The communication with the simulator is based on UDP/IP communication and integrates four channels: two for the device data (one for left and one for right arm) and two for the device command (one for left and one for right arm). The device data packet includes all the data related to the exoskeleton status, such as joint position, speed and torque, and end-effector position, speed and force. On the other hand, the device command packet includes several control strategies for piloting the exoskeleton, such as the desired end-effector force, the desired end-effector position, the desired joint torque, the desired joint pose or the desired joint impedance.

B. Using 3D simulation in-the-loop

The final Operation with a 3D simulation as a support system in parallel to the direct control of the real system which can be 'switched' seamlessly enhances the immersion into the teleoperated robot and its operability. Therefore, the force feedback has to be incorporated in the 3D simulation, too. Using a modular integrating approach, the underlying concept can be extended easily. First, the rigid body simulation within the 3D simulation is modified to enable a collision-based force feedback. Secondly, a simple force feedback device—the Geomagic Touch X [16] (formerly known as Phantom Device)—is used as an input device for simulation, testing and optimizing the force feedback capabilities. In the end, the full body exoskeleton can be used

[4]http://www.geomagic.com/en/products/phantom-desktop/overview

to interface a fully tested simulation environment including force feedback to the different joints.

C. Force Feedback Integration in 3D Simulation

The integration of force feedback in 3D simulation environments is not quite common in current research. Most commonly used as three-dimensional input devices for modeling, force feedback devices are only in some rare applications also used in specialized simulation environments, such as surgical simulations, where force feedback is then the main aspect of simulation. Integrating force feedback into a rigid body based simulation framework is therefore an advancement of the given technology.

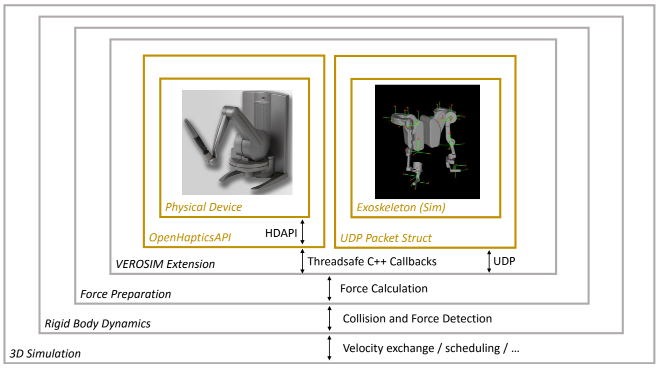


Fig. 3: Modular force feedback concept chart (chart idea based on [17]). Using a modular organization, the physical device and its API can be easily exchanged. The connection of simulation scheduling, rigid body dynamics, collision detection and force preparation is carried out in 3D simulation.

We developed a generic interface to couple rigid body dynamics based force generation, force reprocessing and specialized driver interfaces for each force feedback device. This interface is initialized with the *Touch X* and then extended towards a force feedback ready exoskeleton. As a result, the overall force feedback interface implements three layers:

1) Intertwining of dynamic simulation and events of force feedback calculation at time t_{FF},
2) Generic interface for force feedback devices, calculating a generic force feedback force F_{FF} at the time t_{FF},
3) Specialized driver interfaces for each haptic device,
   a) *Touch X* with *OpenHaptics API*
      - transmit the calculated force F_{FF}^{TouchX},
      - and provide positional input p^{TouchX} of the tool center point.
   b) Exoskeleton with UDP/IP connection
      - transmit an exoskeleton device command struct, either in 'force mode' (using joint torques τ_i^{exo} for each joint i) or 'compliant position mode' (using the end effector position p_{out}^{exo})
      - and provide an exoskeleton device data struct, with positional input of the end effector p_{in}^{exo}.

Starting with the *Touch X*, we used the freely available *OpenHaptics API* [18] to implement the driver interface, while the deeper layers were achieved in simulation. As one can see in Fig. 3, the *API* is just used for low level interfacing the physical hardware. Visible for the user in the 3D simulation is just an extension that manages a thread-safe communication channel. On a higher level, the collision and force detection, calculation and scheduling is of paramount importance. We implemented a collision-based determination of each force feedback event ($\rightarrow t_{FF}$). Now, either a) the calculated force on interacting rigid bodies (F_{RB}) can be used as force feedback, b) specific force torque sensors (F_{FT}) e.g. in the joints, or c) a more general approach, where the virtual coupling is based on a mass-spring-damper system as found in [19][20]. In c), a variance analysis of current position and target position is used to calculate a (virtual) spring-damper based force (F_{SD}). This procedure has the advantage of equal force dimensions, irrespective of the two colliding bodies. Otherwise the calculated collision force could become too high or too volatile for the force feedback device. As a result, we use c) for force direction and magnitude calculation, the integrated dynamic rigid body framework for collision detection, and a separate thread to safely collaborate with the *OpenHaptics API*.

Using this interface, it is also possible to exchange the *Touch X* with other force feedback devices, like the exoskeleton. During the development of the final exoskeleton an exoskeleton simulator is used as a substitute to define, develop, and use the exoskeleton interface in the 3D simulation. This exoskeleton simulator provides the exact same interface design as the final exoskeleton. Therefore, defined exchange information structs (encompassing endeffector position, joint angles, joint force and torques, etc.) can already be received by and send from simulation. Although the communication between simulation and *Touch X* is based on a specific API and thus completely different to the UDP- based connection of the exoskeleton, the infrastructure of the force feedback interface already provides all necessary pre-processing of forces. The low level interface layer of the UDP exoskeleton is then added on top of the force feedback fundament.

Using the exoskeleton simulator led to a defined interface concept for simulation and already shows first promising results in terms of the communication protocol and also realtime capable communication. More effort has to put on optimizing feasible force feedback generation from simulation for a direct and more intuitive sense of immersion.

IV. CONCLUSIONS

In the final operation of the CENTAURO project, this robot will be directly controlled by a first person operator, using an exoskeleton (with force feedback) for control and 3D simulation in-the-loop, supporting the operator. The use of a force feedback exoskeleton supports the operator in his mission by means of intuitive control and the positive effects of immersion, and hence being telepresent at the site of operation accompanied by simulation. The development of an exoskeleton for teleoperating mobile robots is continuously evolving and refined, accompanied by the exoskeleton simulator which is already of paramount importance in terms of interface definition and developments. We could already achieve first results in coupling dynamic simulation, force reprocessing, and interfacing multiple force feedback devices. The integration of force feedback in simulation in general also opens up prospect to a huge amount of applications to dive into virtual realities prior to the completion of the real setup or also in parallel to the real mission.

ACKNOWLEDGEMENT

This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 644839.

REFERENCES

[1] M. Schwarz, M. Beul, D. Droeschel, S. Schüller, A. S. Periyasamy, C. Lenz, M. Schreiber, and S. Behnke, "Supervised autonomy for exploration and mobile manipulation in rough terrain with a centaur-like robot," *Frontiers in Robotics and AI*, vol. 3, p. 28, 2016.

[2] M. Bergamasco, A. Frisoli, and C. A. Avizzano, "Exoskeletons as man-machine interface systems for teleoperation and interaction in virtual environments," in *Advances in Telerobotics*. Springer, 2007, pp. 61–76.

[3] A. Frisoli, F. Rocchi, S. Marcheschi, A. Dettori, F. Salsedo, and M. Bergamasco, "A new force-feedback arm exoskeleton for haptic interaction in virtual environments," in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. IEEE, 2005, pp. 195–201.

[4] R. Vertechy, A. Frisoli, A. Dettori, M. Solazzi, and M. Bergamasco, "Development of a new exoskeleton for upper limb rehabilitation," in *2009 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2009, pp. 188–193.

[5] E. Pirondini, M. Coscia, S. Marcheschi, G. Roas, F. Salsedo, A. Frisoli, M. Bergamasco, and S. Micera, "Evaluation of the effects of the arm light exoskeleton on movement execution and muscle activities: a pilot study on healthy subjects," *Journal of neuroengineering and rehabilitation*, vol. 13, no. 1, p. 1, 2016.

[6] J. C. Perry, J. Rosen, and S. Burns, "Upper-limb powered exoskeleton design," *IEEE/ASME transactions on mechatronics*, vol. 12, no. 4, p. 408, 2007.

[7] J. Klein, S. Spencer, J. Allington, J. E. Bobrow, and D. J. Reinkensmeyer, "Optimization of a parallel shoulder mechanism to achieve a high-force, low-mass, robotic-arm exoskeleton," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 710–715, 2010.

[8] P. Garrec, J. Friconneau, Y. Measson, and Y. Perrot, "Able, an innovative transparent exoskeleton for the upper-limb," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1483–1488.

[9] Y. Mao and S. K. Agrawal, "Design of a cable-driven arm exoskeleton (carex) for neural rehabilitation," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 922–931, 2012.

[10] J. Rossmann, T. J. Jung, and M. Rast, "Developing virtual testbeds for mobile robotic applications in the woods and on the moon," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 4952–4957.

[11] J. Rossmann, M. Schluse, B. Sondermann, M. Emde, and M. Rast, "Advanced mobile robot engineering with virtual testbeds," in *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*. VDE, 2012, pp. 1–6.

[12] J. Rossmann, E. G. Kaigom, L. Atorf, M. Rast, and C. Schlette, "A virtual testbed for human-robot interaction," in *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*. IEEE, 2013, pp. 277–282.

[13] T. Cichon, M. Priggemeyer, and J. Roßmann, "Simulation-based control and simulation-based support in erobotics applications." *Applied Mechanics & Materials*, vol. 840, 2016.

[14] G. C. Burdea, "Haptic feedback for virtual reality," 1999.

[15] G. C. Burdea, G. C. Burdea, and C. Burdea, *Force and touch feedback for virtual reality*. Wiley New York, 1996.

[16] T. H. Massie and J. K. Salisbury, "The phantom haptic interface: A device for probing virtual objects," in *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, vol. 55, no. 1. Chicago, IL, 1994, pp. 295–300.

[17] A. Mohammadi, M. Tavakoli, and A. Jazayeri, "Phansim: a simulink toolkit for the sensable phantom haptic devices," *Proceedings of the 23rd CANCAM, Canada*, vol. 11, pp. 787–790, 2011.

[18] B. Itkowitz, J. Handley, and W. Zhu, "The openhaptics toolkit: a library for adding 3d touch navigation and haptics to graphics applications," in *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*. IEEE, 2005, pp. 590–591.

[19] L. S.-H. Chan and K.-S. Choi, "Integrating physx and openhaptics: Efficient force feedback generation using physics engine and haptic devices," in *Pervasive Computing (JCPC), 2009 Joint Conferences on*. IEEE, 2009, pp. 853–858.

[20] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *Computer Graphics and Applications, IEEE*, vol. 24, no. 2, pp. 24–32, 2004.