71^{st} International Astronautical Congress (IAC) – The CyberSpace Edition, 12-14 October 2020.

IAC–20–D1.6.4

Designing and Testing a Robotic Avatar for Space-to-Ground Teleoperation: the Developers' Insights

Dr. Thomas Krueger * [a], Mr. Edmundo Ferreira[a], Mr. Andrei Gherghescu[a], Mr. Lukas Hann[a], Mr. Emiel den Exter[a], Mr. Frank van der Hulst[b], Mr. Levin Gerdes[a], Mr. Leonardo Cencetti[a], Dr. Aaron Pereira[c], Dr. Harsimran Singh[c], Dr. Michael Panzirsch[c], Dr. Thomas Hulin[c], Mr. Ribin Balachandran[c], Dr. Bernhard Weber[c], Dr. Neal Y. Lii[c]

[a] *Human Robot Interaction Laboratory, European Space Agency, Keplerlaan 1, Noordwijk 2201 AZ, Netherlands. thomas.krueger@esa.int*
[b] *MF Robotics, Hugo van Woerdenstraat 53, 2332 PG Leiden, Netherlands*
[c] *Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Münchener Straße, 82234 Weßling, Germany*
[*] *Corresponding author.*

Abstract

In late 2019, astronaut Luca Parmitano remotely controlled a rover equipped with a robotic manipulator, performing geology tasks on a moon-analog site from the ISS. One year and 7 months later, in July 2021, he will control the same rover in a more realistic moon-analog environment: a field of volcanic rock and regolith on mount Etna, Italy. These experiments constitute the Analog-1 campaign in the frame of ESA's METERON project. As payload developers, we want to create an interface for astronauts to intuitively operate robotic systems on a planetary or lunar surface: how can we maximise task efficiency and sense of immersion/transparency? At the same time, how can we minimise operator fatigue, and physical and mental effort? And how do we do this while constrained in the framework of human spaceflight, with upmass and software requirements, with delayed, low-bandwidth and unreliable communications? We show how we created a telerobotic system featuring an intuitive graphical and haptic user interface. This included a force feedback device and custom joystick, controlling a mobile robotic platform. The robotic platform consisted of an all-terrain chassis and two 7-DOF robotic arms with torque sensing. One arm was mounted on the front of the rover and used for manipulation; the other was mounted on top and used to reposition a camera. With this system, the astronaut was fully in control of the robot to collect rock samples. The only external input was from a ground team of scientists over voice-loop and text-messenger, concerning the choice of geological samples. Full, stable 6-DOF force feedback for the manipulation arm was provided via a sigma.7 haptic input device. This meant that the astronaut could feel (for the first time from space) not only full-DOF contact with the planet surface from orbit, but also the weight of the rocks they grasped. System status feedback was visually and intuitively presented on the user interface – running on a laptop on board the ISS – as well as views from two cameras. During development we continuously integrated requirements from various stakeholders and feedback from astronauts and astronaut trainers to improve the user interface. The analog tests delivered valuable insights about how to design a telepresence system to control robots on a planet's surface from orbit. We expect these insights to be useful for future development of teleoperated planetary robotics as well as terrestrial applications in similar scenarios.
Keywords: (maximum 6 keywords) Teleoperation, Robotics, Low-Bandwidth, Haptics, Real-Time, Latency

Acronyms/Abbreviations

Col-CC	Columbus Control Centre
DDS	Data Distribution Service
DOF	Degree(s) of Freedom
EAC	European Astronaut Centre
EICL	European IP Communication Laptop
ESA	European Space Agency
ESR	Experiment Science Requirements Document
GoEPIC	Golang ESA Provision Image Controller
GUI	Graphical User Interface
HRI	Human Robot Interaction
IMU	Inertial Measurement Unit
IP	Internet Protocol
ISS	International Space Station

LAN	Local Area Network
LOS	Loss of Signal
METERON	Multi-Purpose End-To-End Robotic Operation Network
MPCC	Multi Purpose Computer and Communication
NAT	Network address translation
QoS	Quality of Service
RTPS	Real Time Publish Subscribe version 2.X
SFTP	SSH File Transfer Protocol
TCP	Transmission Control Protocol
TDPC	Time Domain Passivity Control
TDRSS	Tracking and Data Relay Satellite System
TUI	Text-based User Interface
UDP	User Datagram Protocol

1. Introduction

Many tasks in exploration and infrastructure development on the surface of heavenly bodies can be carried out by robots, controlled by humans from a distance. Since robots are expendable and have fewer needs on the planet surface than humans (e.g. food, shelter or oxygen), and many robots can be in theory controlled by a single operator, this has the potential to greatly improve efficiency.

Several questions for developers arise here: what tasks can be performed by robots and what will the workflow look like for the given task? What robotic technology is capable of the necessary demands or how can existing technology be modified? And, of course, how does the operator efficiently and intuitively control these robots at different levels of autonomy, and what communications networks are necessary for them to do so?

The Multi-Purpose End-To-End Robotic Operation Network (METERON) series of experiments attempted to address the third question. Between 2014 to 2019, several experiments investigated ground to earth teleoperation. In [1, 2, 3], haptic teleoperation with reduced DOF systems were trialled between space to ground, and it was shown that sensory perception, including proprioception (force sensing), is degraded in space [4, 5]. Between 2017 and 2018 three ISS-to-ground teleoperation sessions were completed as part of the experiment METERON SUPVIS-Justin, exploring commanding the humanoid robot Justin from an intuitive tablet interface. In a semi-structured mock-Mars environment, the robot was commanded in a high-level way, i.e. in Supervised Autonomy [6, 7].

As a contrast, the Analog-1 mission attempted low-level, telepresence commanding for the task of geological sampling. The astronaut was to drive a mobile robotic platform (rover) to various geological sampling sites, and pick up samples in communication with geologists. This scenario reflects the analog situation of a robot on the lunar surface and an astronaut in the lunar gateway. In unstructured environments such as a geological sampling site, with open-ended tasks such as exploration and sampling, the feasibility and usefulness of automation are limited. Hence the focus was on low-level, immersive telecommanding, where the operator "sees" through the rover's cameras, moves the rovers manipulator with their hand and "feels" the forces that the robot feels.

For developers, this poses a real challenge. Large latencies (500 ms - 1 s), packet loss and low bandwidth characterise communications from ground to orbit. How can the operator stably, intuitively and efficiently control the robot? Video streaming must be compressed, yet still of adequate resolution and frame rate to allow the operator to drive the rover. Telemanipulation should allow the operator to perceive contact, e.g. through force feedback, but must remain stable even under variable time delay.

Added to this, the rover must be operated from space. Sending equipment to space is difficult, because there are a number of design and safety constraints for hardware, which we explain in the next section. Under the constraint of carrying out all of this development in a small team in a short timescale, the problem becomes quite complex – but in the end, feasible.

The first experiment was performed under controlled conditions for the rover, inside a temperature-controlled hangar on a level surface. Following this, a more realistic (but still not space-condition) demonstration is planned on Mount Etna, where the challenges of dust-proofing, localisation and uneven, sloping terrain will be tackled. This experiment was planned for July 2020 but had to be postponed to 2021 due to the Coronavirus pandemic.

In the next section, we detail the experiment scenario and the specific requirements this scenario puts on our payload development. In Sec. 3, we discuss the solutions we came up with. In Sec. 4, we detail our observations, both those that confirmed our expectations as well as those we did not expect. We then detail improvements we will make to the system for the follow-up experiment in July 2021 in Sec. 5 and conclude in Sec. 6.

2. System specification

We detail the broad details of the experiment to be carried out in the following subsection, and the specific consequences these have for us as developers in Sec. 2.2.

2.1 Description of analog experiment scenario

The task for the astronaut was to drive the rover along a marked trajectory to three geological sampling sites. In this scenario, the sites would have been already predetermined from satellite images and previous exploration data. Since it is absolutely new terrain the rover is driven manually (this offers possibilities for automation discussed in Sec. 5.3). At each sampling site, in communication with geologists on the ground, the astronaut (also with basic geological knowledge) surveys the site, identifies desired rock samples, and collects them with the robotic gripper.

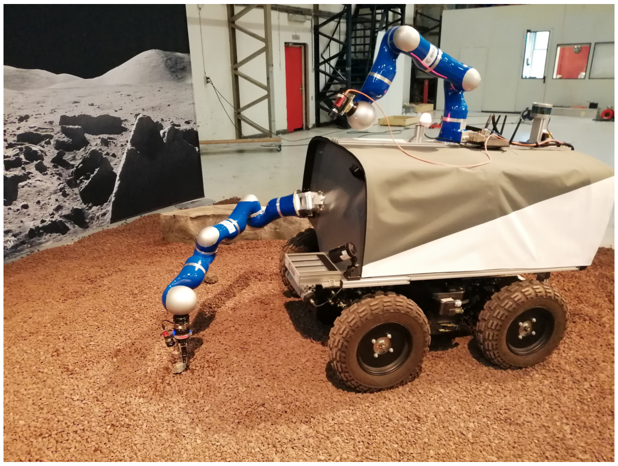


Fig. 1: Rover picking up a geological sample at an analog sampling site.

This setup already has 4 major operations sites: (1) the hangar where the rover is driven, (2) the astronaut on board the space station, (3) operation teams on ground, and (4) scientists in another ground location. All those entities need to be orchestrated.

One question that probably arises is, why would a rover on a planetary surface not be controlled from ground? The main reason is the delay. To travel the 400 000 km to the moon, as well as passing through the several communications relays on ground, will result in a round-trip delay of up to 5 seconds. With this amount of delay, direct telecommanding as described in the introduction is in the best case impractical, in the worst infeasible.

2.2 Payload-specific practical requirements

The aforementioned scenario results in practical requirements for us as payload developers, in terms of Software (Sec. 2.2.1), Communications (Sec. 2.2.2) and Hardware, both on board and on ground (Sec. 2.2.3).

2.2.1 Software

When uploading software to a payload on board the International Space Station (ISS), the ESA standard procedure is to fly a physical device (USB or laptop) containing the required software for the mission. This implies that the software needs to be finalized and will remain unchanged up to several months before the launch.

This experiment was done on an extremely short timescale, with approximately 18 months from Experiment Science Requirements Document (ESR) acceptance to experiment. Due to the timing constraints, it was decided to aim for a flexible approach using over-the-air updates and bare-metal provisioning. This consisted of reusing a device already on the ISS (laptop zBook) and installing on it the operating system and applications to be used in the mission.

Using this approach, software development could continue close to the date of the ISS experiment. Software issues discovered during testing can be addressed and fixed via over the air updates, directly applied to the mission software, increasing the probability of a successful experiment. The software updates had to conform to very specific requirements from ESA's ISS Colombus module network infrastructure.

We present details of the implementation in Sec. 3.4; more details on the specific requirements and solution can be found in [8, Sec. 4.2].

2.2.2 Communications

To make this experiment work we required a communication link from the ISS to the robot and the other experiment sites. Communication systems and solutions on the ISS depend on the module used, the type of payload operated, and ISS partners involved (JAXA, NASA, ESA). Here we are a European (ESA) payload operated in the Columbus module. It is a "class 3 payload" (meaning not racks or infrastructure, and therefore less stringent safety requirements) with connection to the on-board Local Area Network (LAN). This LAN has the possibility to connect to ground via the Multi Purpose Computer and Communication (MPCC) [9].

Under the hood this system uses the Tracking and Data Relay Satellite System (TDRSS) to communicate to NASA's White Sands Complex in the USA, and from there via a transatlantic cable to Europe. Just one way the distance is more than 90 000 km. In addition, the signal needs to pass to many servers and routers and therefore has a round-trip delay of about 850 ms, with packet loss. Furthermore, the communication link is severely bandwidth limited. While the ISS connection to ground can handle 25 Mbps up and 300 Mbps down this does not mean that this can be utilised by a payload. The TDRSS also carries voice communications, onboard video, and telemetry from other payloads and thus needs to be requested and organised. The maximum we could request was 4 Mbps up and down. With the given bandwidth, and with an underlying Internet Protocol (IP) based network we face a high bandwidth delay product which makes the use of Transmission Control Protocol (TCP) and other protocols ineffective [10].

In this experiment we are streaming two video feeds as well as real-time robot control. The video quality needs to be good enough to control the rover, and for the science team to identify interesting rocks for the astronaut to sample, and the video must be shown without significant delay compared to the robot control/feedback.

For the haptic feedback when controlling the manipulators, we need high-frequency communication (in the order of 100s of Hz), but not necessarily high throughput. We also need control architectures that can deal with delay and packet loss, since these necessarily lead to instability in robot control [11].

To avoid security vulnerabilities of all systems and to protect sensitive and all communication needs to adhere to security standards and has to be encrypted.

2.2.3 Hardware

It is costly to bring hardware on board the ISS, and once there, it should be absolutely safe, posing no risk to crew on station. The hardware of the control station must comply with ISS material and safety standards COLRIBRE-SPE-0164, Issue 2A [12], and this must pass an internal safety board review. The GUI must comply with SSP50313, revision F. [13], the Display and Graphics Commonality Standard for the ISS.

The control station for the haptics should provide the operator with a means to stabilise themself when force feedback is applied. On earth, the user gets reaction forces from the ground or a chair, in micrograpity, this is not the case. The hardware also needs a fast setup and teardown time, because astronaut time is precious.

3. Solutions

Here we talk about how we translated the specific requirements from Sec. 2 into engineering solutions.

The rover control station on the ISS was the interface for the astronaut to view and control the rover's movements on the ground. It consisted of a custom-made joystick control station, described in Sec. 3.1, an adapted commercial haptic input device for control of the robot manipulator, see Sec. 3.2, and a laptop which was already on board the ISS. This ran the Graphical User Interface (GUI), described in Sec. 3.3 but also the real-time control for the entire control station, and communication with ground. In order to run our custom software reliably on a generic laptop on the ISS, we employed bare-metal provisioning of our own custom version of Linux, described in Sec. 3.4.

On the ground, we also had a mix of off-the-shelf and custom-made hardware. We detail the design of the ground assets in Sec. 3.5

As described in the previous section, communication is high-latency, low bandwidth and unreliable. Our control algorithms for telemanipulation accommodate this while staying stable (Sec. 3.6).

For reliable communication, we use RTI connext Data Distribution Service (DDS) * configured with a Quality of Service (QoS) that ensures that all packets are delivered. The networking between the ISS and the Ground Systems is non-trivial, and is described in Sec. 3.7. How we sent two video streams of a high-enough quality to drive and manipulate is discussed in Sec. 3.8.

Finally, the hardware and software had to be made robust to misuse, because "everything that can go wrong, will go wrong". In environments with frequent loss of connection and where every second counts, robustness is key. This is described in Sec. 3.9

3.1 Custom-made joystick, mount and handle

Fig. 2: Custom joystick control station with stabilisation handle and enable button (left), 3-axis joystick, and buttons.

Where bespoke requirements are in play, and where the hardware is not too technically complex, it can be easier to develop a custom solution than to adapt commercial hardware for space use. From previous experience performing Haptic experiments in space [1, 2] it is known that force feedback can be applied to the operator in microgravity under the condition that they are able to ground themselves through feet and arm restraints to avoid being pushed away by the haptic device, thus a grounding functionality had to be afforded by the control station.

---

[*] `rti.com/products/dds-standard`, retrieved 2.10.2020

Because the experiment planned required a 3DOF joystick with buttons for driving & state switching it was decided to develop a bespoke joystick that combines the required input components with a structurally robust arm restraint. With the addition of an on-board laptop and bespoke mounting brackets for the sigma.7 & Joystick, the haptic workstation was completed. The use of off-the-shelf camera mounting claps in the design of the brackets allowed for quick set up and teardown of the experiment.

Fig. 3: Mounting of sigma.7 and joystick control station on the ISS. The laptop is above them, out of the picture.

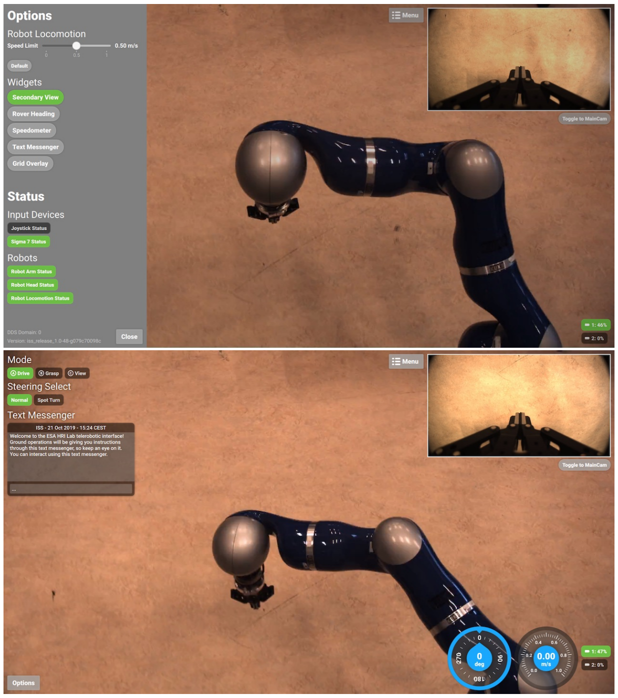
3.2 *Adapting commercial hardware for on-board use*

For real-time control of the robotic manipulator on the rover the sigma.7 haptic input device from Force Dimension was chosen. It is a commercially available device that allows the user to move freely, and receive high-quality force feedback, in 7 DOF (3 translational, 3 rotational, and a 1-DOF gripper).

In addition, its relatively compact form factor and off-the-shelf availability made it the prime candidate for this experiment. The device has also been used over many years in the Human Robot Interaction (HRI) Lab and has proved to be a reliable and easy to work with device. The gravity compensation on earth is assisted with springs, which were removed. In order to comply with ISS payload safety requirements the device was modified by replacing some of the casing with space compliant materials, replacing non-compliant internal electrical components, replacing the power connector with a pre-flown component and adding protection for the operator against pinch-points.

3.3 *Graphical user interface*

For the experiment a custom GUI was developed to be used by the astronaut while operating the rover. This can be seen in Fig. 4.



Speed Limit 0.50 m/s

DDS Domain: 0
Version: iss_release_1.0-48-g079c70098c

ISS - 21 Oct 2019 - 15:24 CEST
Welcome to the ESA HRI Lab telerobotic interface! Ground operations will be giving you instructions through this text messenger, so keep an eye on it. You can interact using this text messenger.

Fig. 4: The Astronaut GUI. *Above:* with the hideable sidebar, the astronaut can set the rover speed limit and select and deselect widgets. They can also see which input devices and robots are connected. In the image, only the secondary view is selected, and only the Joystick controls station is offline. *Below:* the view during operation. The chat, speedometer, and bearing widgets are enabled.

The hideable sidebar on the left (see Fig. 4, above) allows the user to set the maximum speed of the locomotion platform, see the status of input devices and robots, and enable/disable widgets. Several widgets are available including secondary camera view, rover heading display, speedometer, a text-messenger chat for communication with the science team on ground and a grid overlay on the image.

During operation (Fig. 4, below), the operator is provided with two camera feeds from the rover (head-cam view and tool-cam view) and can toggle the inset secondary view with the main view (default is head-cam on main view, tool-cam inset). On the left of the screen, they can also see the mode in which the robot is in: Drive, Grasp, or View mode, to drive the platform, Manipulate, or change the view of the head-cam with the joystick, respectively. These can be selected with the A, B, and C buttons on the joystick control station. In each mode, submodes are available, e.g. in the figure, the operator is in drive mode and can select "normal" driving or "spot turn" submodes. In grasp mode, the submodes are preset arm movements (return to start position, and stow rock in container) and in view mode, they reset the view to preset positions. The operator can also see the locomotion platform's battery levels on the bottom-right.

Besides the operator screen, the program has separate screens including a Help guide that explains the user interface and a Questionnaire page. The approach in designing the GUI was to implement well established design patterns of modern interfaces, while still adhering as to SSP50313 - Rev. F [13], the Display and Graphics Commonality Standard for ISS payloads. The GUI design was kept to the standard through careful choice of terms, typeface and font, button shapes, and specific use of colours (yellow and red being used only in critical situations).

In order to gauge the user experience, the astronaut filled in a questionnaire after the procedure, some elements of which had to do with the user interface.

3.4 Bare-metal provisioning

As previously explained, an existing laptop on the ISS had to be provisioned in order to run the OS and control necessary for the experiment. The practical implementation aspects of this bare-metal provisioning included:

A Custom Linux provision image uploaded to the NASA(missing NASA ground ops name) ISS laptop.

B An existing USB stick onboard the ISS, was flashed with the provision Linux image.

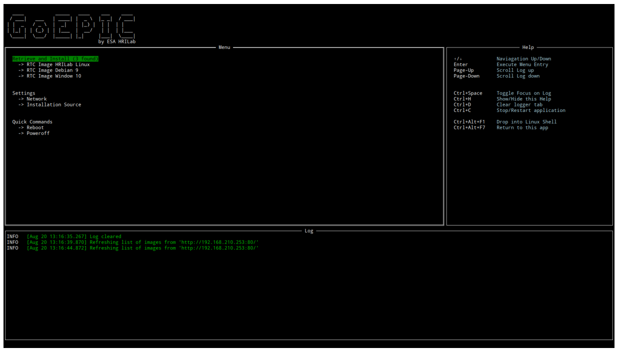
NOTE: Steps A. and B. would not be strictly necessary if the Columbus module had a DHCP and TFTP server configured on the EICL.

This provisioning USB stick could now be used for installing images on any desired device onboard. The steps for the provisioning were as follows:

1. An existing laptop is booted into the provision Linux image on the USB stick.
2. Software application (GoEPIC) is used by the astronaut to connect to the European IP Communication Laptop (EICL) SSH File Transfer Protocol (SFTP) server where operating system images were previously uploaded by Columbus Control Centre (Col-CC) ground operations via MPCC.
3. On user/astronaut input, the latest available image was automatically installed to the laptop's hard disk, with the experiments operating system and software.
4. On reboot, the laptop would boot into the newly installed `systemd` and the user/astronaut could conduct the experiment with the latest software.

The application Golang ESA Provision Image Controller (GoEPIC) mentioned in points 3. and 4., was developed by the HRI Lab. A thorough description is to be found in [8]. It is a Text-based User Interface (TUI) application that allows the user to select images to be installed, but also to configure aspects of the installation source repository and protocol as well as network configuration of the running system.

The mechanism of provisioning is based on the A/B system updates principle used for cloud-infrastructure by Android and DevOps [14]. It is possible to deploy a new image and and revert back to a previous one quickly and with ease.



```
INFO   [Aug 20 13:16:35.267] Log cleared
INFO   [Aug 20 13:16:39.870] Refreshing list of images from 'http://192.168.210.253:80/'
INFO   [Aug 20 13:16:44.872] Refreshing list of images from 'http://192.168.210.253:80/'
```

Fig. 5: Golang ESA Provision Image Controller (GoEPIC) start screen

3.5 Design of the on-ground system

Fig. 6: Interact Rover in hangar.

The Interact Rover (Fig. 6) is the on-ground robotic system to be controlled by the astronaut from space. This is an ongoing development platform for robotics designed and assembled in-house at the HRI Lab. Using modular structural components it combines many off-the-shelf systems into one a highly modular and robust system. Locomotion is provided by the AMBOT GRP-4400 system, providing four-wheel drive and independent steering. On top of the platform a structure is mounted, built from modular 30mm aluminium profile struts. The structure provides 32U of 19" rack space to mount internal components for power distribution, target computers and devices for control, and networking devices. To provide modular space for smaller devices, DIN rail mounted to 19" rack plates was used in many places. A bespoke fabric cover provides weather and dust proofing for the internal components. Two KUKA Light Weight Robot (LWR) manipulators are mounted to the system. For outdoor weather conditions the robots can be dressed with bespoke water and dust proof sleeves.

3.6 Cartesian impedance control with time-domain passivity adaptation

The operator operates the manipulator arm and gripper through the sigma.7 device and the joystick. As long as the operator is holding the stabilising handle and pressing the yellow "enable" button on the top of the handle, velocities are transmitted from the operator's hand to the end effector of the manipulator robot, the gripper open-close position is transmitted from the open-close position user switch on the sigma.7 and forces are transmitted back from the robot to the operator. When the switch is released, the operator can freely move the sigma.7 without transmitting anything. In this way, we reconcile the large workspace of the robot arm with the smaller workspace of the sigma.7 – if the sigma.7 reaches its workspace limits during large movements of the arm, it can be easily repositioned with the "enable" button released.

The mapping from the sigma.7 to the robot tool is meant to be intuitive, and is dependent on the camera view that is selected as the main view in the GUI. If the Tool cam is selected, moving the sigma.7 away from the operator moves the tool along its long axis, regardless of the tool's orientation. If the head cam is selected, moving the sigma.7 away from the operator moves the tool away from the mobile platform in the forward direction of driving. See Fig. 7.

For the task at hand - grasping and unknown objects (rocks) in a highly unstructured environment – full automation is not practical and low-level telemanipulation is likely to yield best results. Force feedback can also aid in this because the operator can feel when they are in contact with the rocks even under poor lighting conditions or when the manipulator is (partially) obscured from view.

However, under high latencies as in our case (800 to 1100 ms) and in combination with packet loss, keeping the system stable is nontrivial [11]. We control the manipulator in Cartesian impedance control [15] using Time Domain Passivity Control (TDPC) to ensure the system remains passive and hence stable. Our algorithm is based on [16]; details are to follow in a separate publication.

The choice of Cartesian impedance control was to prevent damage to the manipulator. Pure position control could be dangerous during contact, as there is no way to moderate the contact forces when the commanded position is inside a rigid object. In Cartesian impedance control, the position of the tool on the remote robot, commanded through the sigma.7 haptic device, is considered to be attached to a virtual spring-damper system, so that the forces from the spring-damper system act on the tool, and these can be limited to safe values.

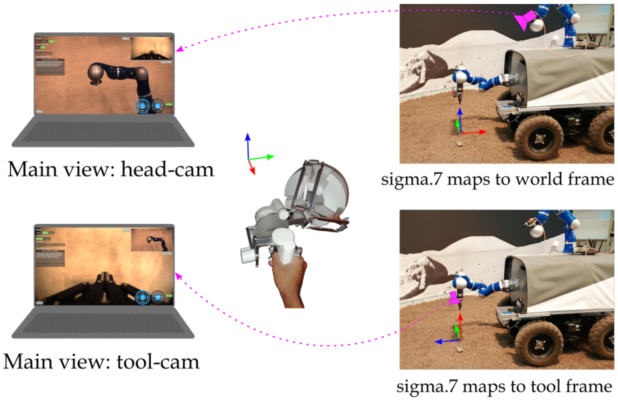


Fig. 7: When the main view selected is from the head-cam, sigma.7 velocity maps to a movement of the tool in the world frame. When the main view is the tool-cam, it maps to a tool movement in the tool frame.

The TDPC modulates the force-feedback applied to the user as well as the commanded Cartesian position at the remote side, ensuring that there is never less energy put into the system than comes out of it. In practice, this means that, while contact forces are perceived with delay, the contact forces at the remote side during this period of delay are kept low, meaning that no force is applied on the remote side which is not intended by the user. The force that the user perceives is also filtered with a low-pass filter, removing possibly confusing vibrations.

3.7 Network

The communication network from ISS to the various ground centres had bandwidth limitations to 4 MB/s, a round trip time of 850 ms, a resulting bandwidth delay product and additional requirements of distributing the data. These required a lot of design steps ranging from architectural planning, technology selection and security certification.

Our solution was to use a User Datagram Protocol (UDP) based protocol, the Real Time Publish Subscribe version 2.X (RTPS). This protocol is used by DDS tunneled through OpenVPN. Beside the pure communication performance, security is also very important, since all communication has to be encrypted. While DDS supports encryption, there is another challenge. During the transfer through all the systems there is extensive Network address translation (NAT) ongoing and the ports need to be defined. DDS with RTPS requires multiple ports while OpenVPN only needs one. This was the reason to have a trade off between performance, security and maintainability.

After the ISS signal arrives at our user center at European Astronaut Centre (EAC) via White Sands and Col-CC, the data needs to be distributed. The receivers (besides the robot which executes the commands) are the operations team, the scientists and the rover team in the hangar. In addition, to replicate our scenario – controlling a rover on the moon – the signal needed to be artificially delayed for the operations team. The distribution has been mainly achieved by special DDS routing services that can sort, split and duplicate data without the risk of multiple accessing and providing observer only capabilities. For the delay we applied standard FreeBSD firewall rules.

3.8 Video compression and bandwidth-saving

The bandwidth is limited both on the up- and down link to 4 MB/s. On the uplink, the bandwidth is shared between two video streams, as well as sensor feedback from the robotic arm essential for the real-time haptic control. The downlink consists mainly of the robot control commands.

Both video streams from the rover are handled by identical but separate GStreamer pipelines. On the rover, the cameras stream video into an application that implements an appSink, which encodes the video into the H.264 format, and sends encoded video frames as DDS messages. On the ISS laptop, the GUI (described in section 3.1) receives the DDS messages and puts the encoded video frames into another GStreamer pipeline which decodes the video and streams it to the laptop screen using a GStreamer plugin for Qt.

The bandwidth required by this system depends on the content of the video: slowly-changing or non-changing scenarios (e.g. surveying a sampling site) require relatively little bandwidth, whereas fast-moving video (e.g. driving or grasping) requires more bandwidth. During the experiment, the video streams used between 0.5 and 1 Mbps combined.

While the video stream entailed sending large messages (video frames) at a low frequency (24 Hz), the real-time robot control including TDPC requires small messages (sensor feedback) to be sent at a high frequency. With these messages, limiting bandwidth is a question of reducing the frequency to stay within the allocated bandwidth, while keeping the update loop of the controller fast enough to ensure that the system works well. Thus, while local control ran at 1 kHz, messages were sent between station and ground at 167 Hz (i.e. every 6 control cycles).

3.9 Improving robustness to misuse

If software crashes or needs to be restarted because connection is lost to a device, the astronaut's valuable time is wasted while they wait for the system to become operational again. Therefore, it is important that the system is robust to the startup order of the various systems, as well as a temporary loss of connection to any given system.

To achieve this, every sub-system is implemented as a state machine. The wheeled platform, as well as the two robot arms all operate independently of each other on the ground. On the ISS the GUI, control software, joystick, and sigma.7 also operate independently of each other. Once all the systems are operational, the astronaut can operate the rover. However, if a subset of the systems fails, the astronaut can still operate everything that does not depend on any of the failed systems. Finally, if the connection to any system is lost (e.g. Loss of Signal (LOS) on the space station, cable accidentally unplugged, etc.), operation can be resumed as soon as the connection is re-established.

4. Observations from experiments and discussion

In this section, we detail some of the biggest findings related to all areas of the experiment, covering operational hardware, human machine interfaces, communication tools, and organisational aspects.

Throughout the experiment's preparatory dry-runs, the actual experiment, and the closing briefings, we have gained many valuable, practical insights. While some of these findings confirmed our expectations, some others were not as intuitive and provided a new perspective, shedding light on the importance of frequent testing with different users. We start by stating the confirmation cases and then go into the more surprising cases.

4.1 Insights confirming our expectations

The following insights were more or less in line with what we had expected to observe

4.1.1 User interface

The astronaut strongly agreed with the statement "It was easy to find what I was looking for on screen", and reported that it took little effort to scan the information displays. We noticed that the astronaut made a lot of use of the ability to toggle the camera which was displayed as the main view, suggesting that this was intuitive. He also rated the tool cam as extremely important for solving the task.

Originally, we had wanted to upmass a touchscreen, however, the decision to limit the amount to be upmassed resulted in us re-using a laptop that was on board. We believe that this would improve the user experience. The Astronaut indicated in the questionnaire that a touch screen would have made the interaction with the GUI neither easier nor more difficult, however, recent developments such as the use of a touchscreen in the SpaceX Crew Dragon launch [17] cemented our view that a touchscreen is essential in order to optimise user experience.

The grid overlay on the camera image facilitated communication between science team and astronauts during sample selection; a shared cursor or highlighting tool could speed this up even further.

4.1.2 Provisioning

The provisioning worked well, our developed tool GoEPIC can be used to provision any custom OS even under challenging restrictions.

4.1.3 Voice loops

Teams that are performing field tests, potentially even with external test users, appreciate the importance of proper communication links between the team members (usually via radio, walkie talkies, VOIP or similar). If the teams are relatively small or if the teams are working on a single experiment at a time, one might not put much effort into organizing a deep hierarchy. Not only is this not necessary for small teams, but command chains may also slow down information flow. Owing to the experience gained in our previous ISS experiments Haptics-1, Haptics-2, and Interact, we could however appreciate/anticipate that some hierarchy has to be implemented and respected in order to even enable experiments with so many institutions and operations unrolling at the same time.

In this environment where many topics have to be treated at the same time and some information needs to be shared across experiments and institutions, it is important to limit the chatter and connections to a minimum basis, since otherwise important information might end up being buried in the flood of information.

There are experts working in this profession that helped set up and test-play/simulate different arrangements. Different roles may need to monitor different channels and only forward the necessary information to other people. Others should only hear a limited amount of channels so as not to be overwhelmed with communication and be able to focus on other work as much as possible.

Additionally, if as much information as possible can be shared visually, i.e. without blocking voice channels, this speeds up the overall interaction. For example, the rover ground team benefited heavily from seeing a video stream of the astronaut (the operator) and having a relay person in the Eurocom command room. The more insight the Rover/Ground team has into what is happening in the operations team and on the ISS, the better they can anticipate and prepare for upcoming actions and prepare.

4.2 Unexpected insights:

The following insights were not entirely expected or even surprising to us as developers.

4.2.1 Importance of technical training or access to a trained operator for complex systems

During the proficiency run, the payload developer was in the voice loop with the astronaut and was able to guide him through the use of the robot. However, during the experiment, only the geologists were on the voice loop with the astronaut, who did not have full knowledge of the system. The astronaut subsequently forgot the pan-tilt functionality of the camera on the "head" robot arm. This is evidenced by feedback in the questionnaire after the experiment, where the astronaut asks for a way to look around without having to move the whole robot. It should be mentioned that there were several months between when the astronaut underwent ground training on the robot, and that the control of the rover changed in the meantime (also in line with his recommendations).

It could be argued that a more intuitive or accessible user interface could avoid functionality being overlooked. However, as complexity of the system increases, there is only so far that an intuitive interface can help. In this case solutions might be either a trained operator on voice loop, a help menu for the operator, or an interactive suggestion function that monitors the user's intention and then highlights functionality that might help, either explicitly via text (like Microsoft's "Clippy") or through visual cues on the GUI.

4.2.2 Communications with the Science team

We noticed that chat functionality was neglected. In the questionnaire, the astronaut reported that it was slightly helpful to have the sampling goals presented in the chat rather than in a separate written document (rating 5 on a scale from 1: very unhelpful to 7: very helpful), but did not find the chat so important for solving the task (rating 3 on a scale from 1: not at all important to 7: extremely important) However, it may be that for time delays larger than between the Earth and the Moon, for example to Mars, chat would be preferable to highly delayed voice loop.

When communicating with the Science team, it was noted the astronaut would use any visual cue on screen (even shadows) when discussing with the science observation team.

4.2.3 Operation patterns for the sigma.7

Unexpected observations were also that during operation of the sigma.7 the operator did not always use the correct finger positions on the grasping joint. This could be corrected by better instruction prior to use. During driving, the operator would also often use their left hand to to drive the rover, keeping their right hand on the sigma.7.

This could be due to ergonomics in lower gravity, and cramped spaces, or difficulty inserting/removing the hand from the sigma.7 in microgravity, since this is not the most efficient way to do this on earth.

It also seemed from analysis of the video from the ISS that swapping from Joystick use to sigma.7 use to laptop button and mouse takes up a lot of time and is taxing on the operator's focus. Related to this, we saw that the operator used the tool-cam to look around the sampling site, controlled by the sigma.7, possibly related to the fact that he was temporarily not aware of the function of the head-cam to pan and tilt. We had not observed this in any previous tests. It is difficult, when exploring with the tool-cam, to preserve a sense of orientation (up-down). Perhaps astronauts are intuitively better at this from their time in microgravity, or it does not disturb them to lose this sense of orientation. This opens up many different possibilities for commanding the movement of cameras during microgravity teleoperation.

5. Improvements and enhancements in progress

For the follow-up experiment, Astronaut Luca Parmitano will operate the rover as part of the ARCHES campaign on Mt. Etna in summer 2021. Additional functionality – mostly for the ground assets – is planned. Some of this is necessary for the environment – on a volcano presents other challenges than in a hangar, including water- and dust-proofing (Sec. 5.1) and dealing with slopes (Sec. 5.2). Additional features which are important for a 21st century mobile platform are localisation and semi-autonomous navigation, which is expected to be implemented as autonomous driving between user-specified waypoints (Sec. 5.3)

5.1 *Water and dust-proofing*

In the expected outdoor environments near Mt Etna, water and dust pose a serious threat to the delicate robotic systems on the rover. While the internal components are relatively protected within the chassis and fabric shell, the external devices including the robotic arms are still exposed. For the arms a custom sleeving system is designed to protect from dust and moisture ingress. A waterproof, highly elastic fabric sleeve (PUL) is mounted to a 3D printed structure design to prevent fabric stuck between joints.

5.2 *Online-updating gravity compensation*

Etna is also significantly different from the Hangar environment in that the slopes on which the mobile platform is driven are no longer small ($< 5°$). As mentioned in [18], slopes in the areas on which the platform would be tested are up to 18°. A difference between the actual gravity vector and that used to compute the gravity compensation results in a body force on the robot equivalent to the differences in these vectors.

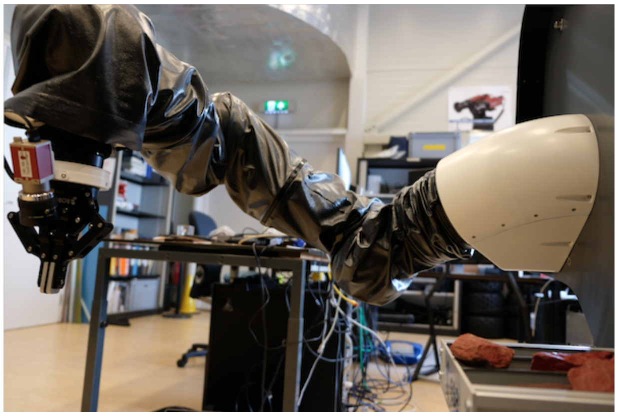


Fig. 8: Arm in sleeve.

For this reason, it is necessary to update the gravity vector. The roll and tilt of the mobile platform is estimated with a state estimator using a Kalman filter and Inertial Measurement Unit (IMU) and gyroscope data from a Realsense T265 camera attached to the base of the manipulator arm. Preliminary tests on slopes up to 20° have been promising.

5.3 *Localisation and semi-autonomous navigation*

Self-localisation was trialled during the Analog-1 experiment in November 2019 using an Intel Realsense D435 depth and RGB camera, and prior to this, some tests were made with Lidar. The performance was not reliable enough to be used for practical purposes such as automated driving.

Implementing accurate and reliable localisation (indoor and outdoor) needs to be fit to specifics of the platform. Trials with lab-developed model using GPS and wheel odometry as inputs have so far been promising.

Using this model, and possibly sensor fusion from cameras and a map of the environment, point-to-point automated navigation can be attempted, where an operator indicates a position either on a map or possibly on the camera view, and the robot navigates there.

In general, more (and more reliable) automation can reduce operator workload during certain tasks, and may speed up task completion. For example, the stowing of the grasped rock was performed automatically since the container location is known, and this is much faster and safer than direct telemanipulation. Reducing operator workload improves overall efficiency and safety, especially during multi-hour, multi-robot operations.

6. Conclusions

The payload development for the Analog-1 experiment was done in a short time by a small team. Nevertheless, a complex set of requirements and restrictions were integrated leading to a successful experiment, from which valuable insights for robotic teleoperation systems in the context of planetary exploration could be drawn. These insights will feed into our upcoming mission in July 2021 on Mount Etna.

Future work in the project MARC-II will focus on scaling up and down the level of autonomy of the remote-side robot and accordingly the command modalities of the operator. Commanding a team of heterogeneous robots in a variety of environments with differing levels of structuredness will also be investigated.

Acknowledgements

We would like to thank the many partners of the Analog-1 project, for a successful collaboration. It would be too much to name them all but especially we like to thank for the passionate and dedicated support from Norbert Schwier of the Airbus Engineering Team in Bremen together with Axel Harms from their EMC test facility, Force Dimension with Francois Conti for relentless support in all matters and Alessandro Gily from Thales Alenia Italy for travelling last minute before launch between Italy and Switzerland for a repair job.

References

[1] A. Schiele, M. Aiple, T. Krueger, F. van der Hulst, S. Kimmer, J. Smisek, and E. den Exter, "Haptics-1: Preliminary results from the first stiffness jnd identification experiment in space," in *Haptics: Perception, Devices, Control, and Applications*, F. Bello, H. Kajimoto, and Y. Visell, Eds. Springer International Publishing, 2016, pp. 13–22.

[2] A. Schiele, T. Krueger, S. Kimmer, M. Aiple, J. Rebelo, J. Smisek, E. den Exter, E. Mattheson, A. Hernandez, and F. van der Hulst, "Haptics-2 — a system for bilateral control experiments from space to ground via geosynchronous satellites," in *IEEE Int. Conf. Systems, Man, and Cybernetics*, 2016, pp. 000 892–000 897.

[3] J. Artigas, R. Balachandran, C. Riecke, M. Stelzer, B. Weber, J.-H. Ryu, and A. Albu-Schaeffer, "Kontur-2: force-feedback teleoperation from the international space station," in *IEEE Int. Conf. Robotics and Automation*, 2016, pp. 1166–1173.

[4] B. Weber, S. Schätzle, C. Riecke, B. Brunner, S. Tarassenko, J. Artigas, R. Balachandran, and A. Albu-Schäffer, "Weight and weightlessness effects on sensorimotor performance during manual tracking," in *Haptics: Perception, Devices, Control, and Applications*, F. Bello, H. Kajimoto, and Y. Visell, Eds. Springer International Publishing, 2016, pp. 111–121.

[5] B. Weber, R. Balachandran, C. Riecke, F. Stulp, and M. Stelzer, "Teleoperating robots from the international space station: Microgravity effects on performance with force feedback," in *IEEE/RSJ Int. Conf Intelligent Robots and Systems (IROS)*, 2019, pp. 8144–8150.

[6] P. Birkenkampf, D. Leidner, and C. Borst, "A knowledge-driven shared autonomy human-robot interface for tablet computers," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 152–159.

[7] N. Y. Lii, D. Leidner, A. Schiele, P. Birkenkampf, B. Pleintinger, and R. Bayer, "Command robots from orbit with supervised autonomy: An introduction to the meteron supvis-justin experiment," in *Proc. Int. Conf. Human-Robot Interaction Extended Abstracts*, ser. HRI'15 Extended Abstracts, 2015, p. 53–54.

[8] E. Ferreira, A. Pereira, A. Gherghescu, L. Gerdes, L. Hann, and T. Krueger, "Slim robotics: Robotics in a small team with space requirements," in *Proc. i-SAIRAS Conf.*, 2020.

[9] A. Schlerf, D. Sabath, G. Söllner, and I. Verzola, "Implementation of an additional command system, pathing the way for new tasks at Col-CC," in *Proc. International Astronautical Congress*, 2017.

[10] T. V. Lakshman and U. Madhow, "The performance of tcp/ip for networks with high bandwidth-delay products and random loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, 1997.

[11] T. Hulin, A. Albu-Schäffer, and G. Hirzinger, "Passivity and stability boundaries for haptic systems with time delay," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1297–1309, 2014.

[12] "COL-RIBRE-SPE-0164: Columbus pressurized payloads interface requirements document," 2013, issue 2A.

[13] "SSP-50313 display and graphics commonality standard," 2018, rev. F.

[14] "source.android.com/devices/tech/ota/ab," website, retrieved 2.10.20.

[15] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint

robots," *Int. J. Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[16] M. Panzirsch, H. Singh, T. Krueger, C. Ott, and A. Albu-Schäffer, "Safe interactions and kinesthetic feedback in high performance earth-to-moon teleoperation," in *Proc. IEEE Aerospace Conference*, 2020, pp. 1–10.

[17] "www.cnet.com/news/nasa-and-spacex-are-about-to-fly-into-space-with-a-touchscreen-demo-2-crew-dragon/," web news article, retrieved 2.10.20.

[18] K. Bussmann, L. Meyer, F. Steidle, and A. Wedler, "Slip modeling and estimation for a planetary exploration rover: Experimental results from mt. etna," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2018, pp. 2449–2456.