CONTEXT-AWARE MISSION CONTROL FOR ASTRONAUT-ROBOT COLLABORATION

Daniel Leidner, Peter Birkenkampf, and Neal Y. Lii

German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Germany, 82234 Weßling,
daniel.leidner@dlr.de, peter.birkenkampf@dlr.de, neal.lii@dlr.de

ABSTRACT

Space robot assistants are envisaged as semi-autonomous co-workers deployed to lighten the workload of astronauts in cumbersome and dangerous situations. In view of this, this work considers the prospects on the technology requirements for future space robot operations, by presenting a novel mission control concept for close astronaut-robot collaboration. A decentralized approach is proposed, in which an astronaut is put in charge of commanding the robot, and a mission control center on Earth maintains a list of authorized robot actions by applying symbolic, geometric, and context-specific filters. The concept is applied to actual space robot operations within the METERON SUPVIS Justin experiment. In particular, it is shown how the concept is utilized to guide an astronaut aboard the ISS in its mission to survey and maintain a solar panel farm in a simulated Mars environment.

Key words: Space Robot Operations; Robotic Space Exploration; Robot Mission Control; Astronaut-Robot Collaboration; Supervised Autonomy.

1. INTRODUCTION

Future space operations are tending increasingly toward close collaborations with robots. Astronaut-robot teams are tasked to maintain orbital structures [1], explore celestial bodies [2], and setup planetary infrastructure for scientific inhabitation [3]. Intuitive user interfaces are to be utilized to command these robots. Among others, smartphones/smartwatches, gesture commands, and speech recognition are envisaged as control modalities. The deployed robots shall incorporate advanced reasoning mechanisms and dexterous manipulation capabilities to allow them to solve complex manipulation tasks autonomously. Although the status of the robot may be monitored by a mission control center on Earth, the decisions on which task should be fulfilled by the robot must remain with the astronaut. Particularly for missions beyond Earth orbit such as the manned exploration of Mars (see Fig. 1), high communication delays deny direct intervention. Nevertheless, it is still necessary to guide the astronaut in its decisions in order to assist him/her during the unnatural working conditions.

Figure 1. Rollin' Justin maintaining a solar panel in the METERON SUPVIS Justin experiment.

This work presents a novel, context-aware mission control approach for robot operations in collaboration with human astronauts. The proposed approach transfers the executive authority from a mission control center located on Earth, directly to the astronaut that works next to the robot. The astronaut selects the actions for the robot to perform in order to realize a given task. The main principle relies on high-level task commands that are easy to understand by the human operator. The robot interprets these commands and schedules the necessary actions and operations to fulfill the task. This transforms the robot from a tool extension on ground to an intelligent coworker. One major issue with this approach is the limitation on the information an astronaut is able to process at once. This effect is amplified by the fact that future service robots will be designed to execute diverse tasks. With increasingly complex robots and tasks, the number of possible commands is likely to exceed the astronauts capabilities by far. For example, it may be impossible for an astronaut to remember all possible voice commands to control a robot verbally. Gesture commands can only communicate basic functionality, such as “follow me”, “come over“, or “stop”. Furthermore, traditional human-machine interfaces, such as computer monitors are limited in visualization space, such that robot capabilities have to be accessed via nested menu structures. However, the majority of the available commands may be irrelevant to solve a given task. Therefore, a mission control concept is proposed that is tailored to reduce the cognitive load for astronaut-robot collaboration based on the current mission context.

Our contributions rely on a supervised autonomy approach, where an astronaut is presented with possible robot commands, provided by a mission control facility on Earth in the form of high-level action definitions, as visualized in Fig. 2. We propose to use the task context as the basis to prune the available actions for the astronaut automatically. First, we evaluate the internal world state of the robot in order to identify robot actions that contribute to the overall mission objectives. Actions that are not immediately feasible are not presented to the user. Our approach combines symbolic state information and geometric features to filter the actions accordingly. The required knowledge is arranged in a semantic map that incorporates geometric information such as object positions, CAD data, and action specific transformations, as well as symbolic information such as the current status of objects, the relation between objects, and their meaning to the robot. In order to limit the list of actions even further w. r. t. a specific task context, we propose a method to apply mission specific filters. An operator on ground may support the astronaut by authorizing additional actions if they become necessary for the progress of the mission. The operator can also revoke certain actions to protect the astronaut and the robot in case of critical environmental conditions. In conclusion, we argue that our contributions enable the use of small-scale input devices as intuitive user interfaces for future space exploration missions. This hypothesis is evaluated under realistic conditions within the *Multi-Purpose End-To-End Robotic Operation Network (METERON)* [4]. In particular, the concepts are implemented for the *SUPVIS Justin* experiment, where an astronaut aboard the *International Space Station (ISS)* controls the humanoid robot Rollin' Justin by means of a tablet computer to maintain a solar panel farm.

The reminder of this work is structured as follows. First, the state of the art on robot mission control concepts for space exploration and other domains is investigated in Sect. 2. Based on this, we emphasize the need for a context-specific mission control framework as it is presented in Sect. 3. Finally, we outline the objectives of the METERON SUPVIS Justin experiment in Sect. 4, where special attention is given to the role of the proposed framework. The insights and benefits, as well as remaining limitations emerging with the proposed concept are discussed within Sect. 5

2. RELATED WORK

The literature on mission control concepts for space robot operations is mainly concerned with concepts that are tailored for remote operations. The most common scenario describes a semi-autonomous robot or rover on a distant planet and a mission control facility located on Earth, controlling the system. Several concepts exist that propose a solution to this issue. A comprehensive survey on the topic is conducted in [5]. One of the most prominent examples for this concept is given with the NASA Mars Exploration Program, as it is for example practiced with the Curiosity rover [2]. The rover is controlled via the *Open Mission Control Technologies (OpenMCT)* software, which provides the means to schedule rover operations under consideration of energy consumption, elapsed surface time, data usage profiles, and other aspects [6]. The software provides a modular framework to design, monitor, and execute robot operations through freely composable plug-ins. This concept is highly flexible. However, the achievable level of autonomy is typically rather limited such that an operator has to monitor the system constantly.

An alternative approach is presented with the *RMC Advanced Flow Control (RAFCON)* tool [7]. RAFCON describes a software framework that allows a user to design complex robot tasks by means of visual programming. The system allows for fully autonomous operations with multiple objectives. For example, it may be used to explore an unknown environment, while scientific instruments are deployed. RAFCON has been efficiently utilized to solve all tasks in the DLR SpaceBotCamp [8]. Furthermore, it is used to explore the surface of Mount Etna in the ROBEX analogue mission [9]. Each mission is represented as a dedicated flow chart. This leaves the user with limited interaction possibilities, as the robot operates fully autonomous.

The previously introduced concepts assume that a robot cannot timely communicate with a crew member nor the ground control segment. However, future space robot operations require a more flexible mission control approach. An effective way to communicate varying task information between agents in a network is given by the TaskMan framework [10]. Depending on the capabilities of the robot and the applied control modalities, the level of abstraction is dynamically assigned. Particular methods for dynamic communication in robotic space operations are investigated in the METERON project [4], where a scenario is envisaged in which an astronaut controls a surface robot from an orbiting spacecraft. This scenario is relevant for the construction of future habitats and infrastructure on celestial bodies. The control modalities in this scenario range from direct control [11], over discrete commands [12], to supervised autonomy [13, 14]. This scenario is taken one step further in [15], where an astronaut collaborates with a crew support robot during surface operations. The authors propose a mixture of gesture commands and touchscreen inputs to command a rover. A similar approach is addressed in [16], where tablet computers and smartwatches are utilized to instrument a collaborative robot. The main problem arising with these scenarios is the limited possibility for the ground segment to support the astronaut, as the delay from Earth to the surface robot is too high. Accordingly, the astronaut has to decide on its own how to proceed to fulfill a mission.

Recently, we have proposed an approach in which a user can command a service robot by means of a tablet computer [17]. The application builds on the task knowledge of the robot and provides only the interaction possibilities to the user that are currently useful. To achieve this, the application explores the symbolic state space of

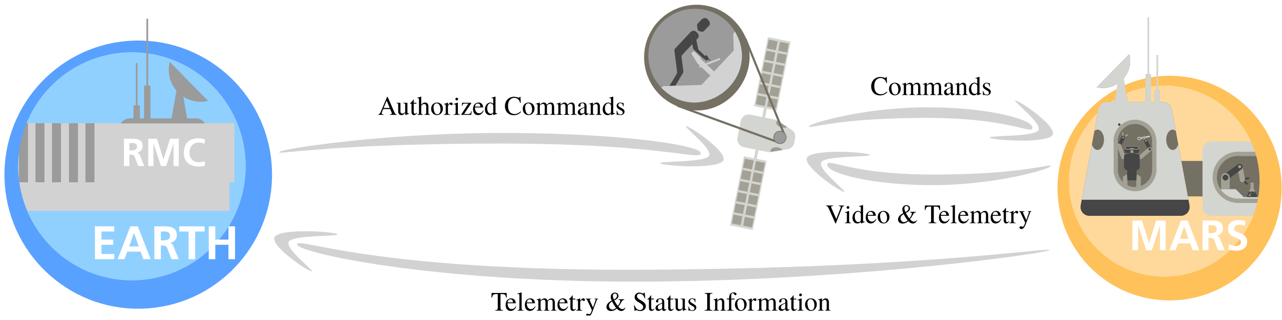

Figure 2. Overview of the proposed mission control concept.

the available objects and the actions they afford. An intuitive user interface allows to navigate through this reduced list of actions in order to guide the user in its decisions. This approach is adapted for space operations and utilized in the METERON SUPVIS Justin experiment [18], in which an asstronaut aboard the ISS shall command the humanoid robot Rollin' Justin [19] to survey and maintain a simulated Martian solar panel farm located on Earth. However, the astronaut is still confronted with numerous actions, that may be irrelevant for the current task. Accordingly, this paper shall present a novel, distributed mission control approach, in which the astronaut is in charge of commanding the robot and a third party facility (i. e. a mission control center) maintains the list of authorized robot actions by applying symbolic, geometric, and context-specific filters.

3. CONTEXT-AWARE MISSION CONTROL

A general overview of the proposed concept is illustrated in Fig. 2. It shows the three participants in a potential Mars habitation scenario, i. e. the mission control center on Earth (left), the space craft in the Martian orbit (center), and the robotic systems on Mars.

The robotic systems on Mars are deployed in order to construct and maintain habitats and infrastructure for future human scientist. The robots provide advanced cognition and manipulation capabilities in order to solve even complex tasks semi-autonomously. Among others, these capabilities may include autonomous planning of navigation and manipulation activities [20], cognition-enabled compliant manipulation strategies [21], as well as failure detection and recovery up to a certain level of complexity [22]. The robot actions are implemented by means of so called *Action Templates* [23], which constitute an object-centric, hybrid representation of robot capabilities, integrating symbolic and geometric features. That is, each object in the world state of the robot affords a set of available actions to the robot, providing the main building blocks for the reasoning apparatus of the robot.

Action Templates are automatically parsed to create a list of possible robot commands that is made available to the astronaut in orbit. Based on this list, the astronaut has to decide on how to instrument the robot in order to solve a given task or proceed w. r. t. to a higher level mission objective. To do so, the astronaut may be in charge of the robots capabilities to survey the environment (i. e. the possibility to alter the camera view), as well as to relocate the robot in the remote environment (i. e. navigate toward a target destination). Once the robot is able to reach a particular object, the astronaut may decide to manipulate it based on the list of available interaction possibilities. The astronaut may thereby utilize an intuitive tablet computer application as it is presented in [16].

While the astronaut may be presented with a basic set of robot commands at the beginning of a mission, an extended list of commands is not made available until after the ground control segment on Earth authorizes it. That is, the third party may decide to prune the list of available robot commands w. r. t. to the overall mission objectives, the current situation, and the experience of the astronaut. The robot telemetry and status information is thereby utilized to filter the command list.

Algorithm 1: Resolve parameters and apply filters.

Input: The object of interest $\mathcal{O}$.
Output: The list of authorized actions $\boldsymbol{\mathcal{A}}$.

$\boldsymbol{\mathcal{A}} \leftarrow$ List()
foreach $\alpha \in$ *GetAllActionTemplates*($\mathcal{O}$) **do**
  foreach $\alpha_r \in$ *RecursivelyResolveParameters*(α) **do**
    $\boldsymbol{p} \leftarrow$ GetPreconditions(α_r)
    $\boldsymbol{e} \leftarrow$ GetEffects(α_r)
    if $\boldsymbol{e} \subset \boldsymbol{\omega}$ **then**
      $\gamma \leftarrow 0$
    else if $\boldsymbol{p} \subset \boldsymbol{\omega}$ **then**
      $\gamma \leftarrow 1$
    else
      $\boldsymbol{\mathcal{T}}_{sym} \leftarrow$ SymbolicPlan($\boldsymbol{e}$)
      $\gamma \leftarrow$ Length($\boldsymbol{\mathcal{T}}_{sym}$)
    if *ApplySymbolicFilters*(α_r, γ) **then**
      continue
    if *ApplyGeometricFilters*(α_r) **then**
      continue
    if *ApplyContextFilters*(α_r) **then**
      continue
    $\boldsymbol{\mathcal{A}}^\frown \langle \alpha_\mathrm{r}, \gamma \rangle$
return $\boldsymbol{\mathcal{A}}$

Algorithm 1 is considered with the filter procedure for one particular object of interest. It reviews all Action Templates α provided by the class of the selected object $\mathcal{O}$ and its parent classes. To cover all action possibilities, the available objects in the world state of the robot are recursively parsed for all possible parameter combinations by the *RecursivelyResolveParameters* function. It substitutes all action parameters with matching objects currently in the world state. The resulting list of available actions α_{r} is validated symbolically, geometrically, and w. r. t. mission specific contexts. The three types of filters are detailed in the following sections.

3.1. Symbolic filters

Symbolic filters are applied based on the symbolic description of Action Templates that is implemented in terms of the *Planning Domain Definition Language (PDDL)* [24]. The definition of actions in PDDL allows for a natural filter based on the currently available world state. Accordingly, the robot is only providing symbolically meaningful actions as a basis for all filters. As an illustration, the robot can only be commanded to deactivate a solar panel if it is currently active. The preconditions for a particular action are therefore evaluated to check if it is already achieved, directly executable by means of one action, or only available after a set of prerequisites is fulfilled, i. e. a sequence of actions with the right effects is executed. Symbolic filters mainly consider the length of the anticipated action sequences denoted as γ. If the effects $\boldsymbol{e}$ for an action are already a subset of the predicates in the world state $\boldsymbol{\omega}$, the action is obsolete and therefore not presented to the astronaut ($\gamma \leftarrow 0$). If the preconditions $\boldsymbol{p}$ are immediately reachable as subset in the world state $\boldsymbol{\omega}$, it requires only one action to solve the problem ($\gamma \leftarrow 1$). Otherwise, a symbolic planner has to calculate the shortest available sequence of actions. Depending on the objectives, mission control may refuse to authorize action sequences of a certain length, since failure likelihood increases with the length of autonomous robot activities where no human is in the loop.

3.2. Geometric filters

The second level of filters is concerned with the geometric relations of the objects in the environment and the position of the robot and its manipulators respectively. As such, geometric filters may refuse actions that are too far away which forces the astronaut to explore the environment. In other words, if the robot is not within the close vicinity of an object, the astronaut should not be allowed to execute the afforded actions. For example, if a robot is too far away from a solar panel, it should not provide the possibility to connect to it, clean it, nor interact with the power switch. The astronaut is therefore directed to navigate the robot to the solar panel first. While this could be incorporated within the PDDL description of each particular action, it is not recommended as the generality of the actions decreases with this.

3.3. Context-specific filters

On a third level, context-specific filters apply. These filters can not be represented based on purely symbolic information, neither by exploiting geometric information alone. These filters may be defined w. r. t. a particular mission scenario, or based on common sense reasoning. In the first case, mission control may decide to authorize particular robot commands based on the protocol procedure for a certain experiment. As an example, the astronaut may first start to survey the surrounding, before the astronaut is authorized to navigate the robot toward a certain goal. To give an example for a common sense reasoning filter, a robot may refuse to receive commands during a space walk if the oxygen of the collaborating astronaut decreases below a critical level.

All actions that are not filtered are concatenated in a list ($\boldsymbol{\mathcal{A}} ^\frown \langle \alpha_{\mathrm{r}}, \gamma \rangle$) and provided to the astronaut. All together the astronaut is guided in its decisions by means of a multi-level filter cascade. This concept is utilized in the METERON SUPVIS Justin experiment, demonstrating supervisory control robot operations from the ISS, as it will be detailed in the following section.

4. IMPLEMENTATION

The proposed mission control concept is realized as part of the ground segment software for the METERON SUPVIS Justin experiment. In this experiment, an astronaut aboard the ISS shall command the humanoid robot Rollin' Justin to maintain a solar panel farm in a simulated extraterrestrial environment. The experiment aims to study an astronauts ability to remote control a robot avatar, in order to accomplish a complex task under realistic conditions of human space flight. In addition to the technology demonstration conducted in METERON, we aim to validate the concepts of the Lunar Exploration Vision 2030, envisaging the remote construction of celestial infrastructure from a space craft orbiting Moon. In both scenarios, an astronaut is charged to control one or multiple robots on the surface of the planet/moon.

Figure 3. The manipulation view of the SUPVIS Justin tablet application.

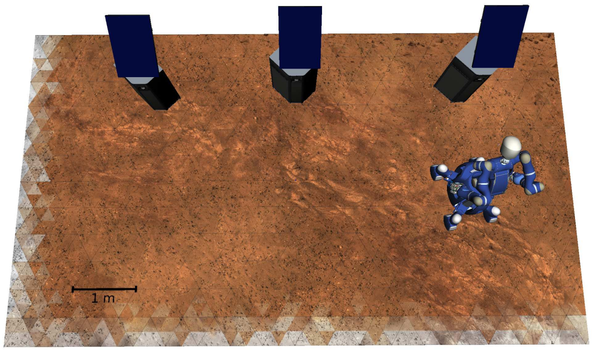

Figure 4. The semantic map forming the basis to reason about any type of command filter.

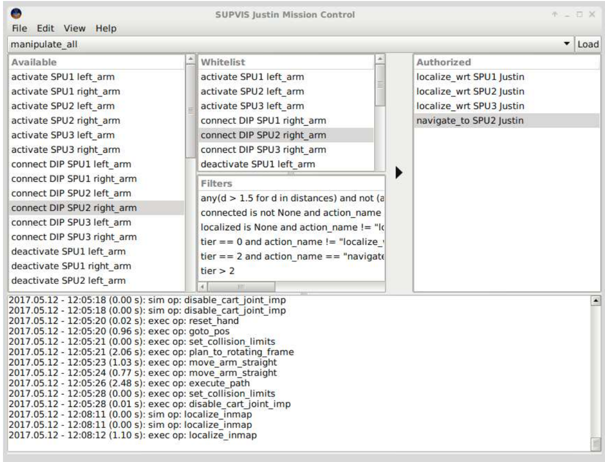

Figure 5. The mission control interface for the operator at the ground control segment on Earth. It shows all available actions (left), the whitelist and filters (center), the authorized actions (right), and a history (bottom).

The METERON project investigates possible command modalities, ranging from haptic teleoperation, to discrete commands, and supervised autonomy. The latter one is the basis for the mission control concept implemented in this section. The principle idea is that of a tablet application that provides high-level robot commands to an astronaut. The application is optimized w. r. t. the intensified cognitive load during human space flight. The application is shown in Fig. 3. The main canvas shows a video stream of the head mounted camera of the robot. The video stream is augmented with CAD information of the detected objects. By selecting an object, the astronaut is enabled to interact with it through the robot on site. A list of related robot commands is presented on the right. Once a command is selected, it is send to the robot which performs the associated actions autonomously.

This list of robot commands is the result of the filter procedure encoded in Algorithm 1. The reasoning is conducted based on the internal world state of the robot. An example world state is visualized by means of a semantic map in Fig. 4. In the illustrated state, the robot is too far away to interact with any of the Solar Panel Units (SPU), forcing the astronaut to navigate toward a target. From a symbolic point of view, the robot is able to activate and deactivate the panels, while it holds a Data Interface Probe (DIP) to connect to the solar panels (highlighted blue in Fig. 3).

A mission control software deployed at the ground segment realizes the individual filters. Fig. 5 shows the software in the status that corresponds to the world state shown in Fig. 4. Initially, all robot actions are explored and listed on the left. A whitelist (center, top) provides the first filter step to select the appropriate actions according to a certain mission or protocol. In the example at hand, the basic capabilities to manipulate the three solar panels is provided. This list is further pruned based on the symbolic, geometric and context specific filters listed in the center. The resulting list of authorized robot commands is listed on the right. To maintain an overview on the mission procedure, a list of recently executed robot actions is listed on the bottom. The list of authorized commands, the whitelist, as well as the list of active filters is forwarded to the robot, such that the actual filters apply locally in order to cope with the signal delay. This way, the list of authorized actions can be updated in real-time, every time the robot executes an action. Once a new set of filters is sent to the robot, the authorized actions are synchronized and the orbiting astronaut receives the new set of commands.

Several different filters are implemented for the SUPVIS Justin experiment. At first, a whitelist is implemented to cope with the geometry of the SPUs. That is, the robot is only allowed to operate the power switch with the left hand, while it should use the right hand to connect the DIP. While this poses an artificial limitation that is not mandatory for the success of the mission, it reliefs the astronaut from selecting the proper manipulator for a given task. From a symbolic point of view, the astronaut is limited to commands that result in action sequences that are no longer than two steps in total ($\gamma > 2$). Additionally, all actions without new effects are hidden ($\gamma = 0$). However, an additional filter describes an exception to this rule, such that the astronaut is always able to localize the robot in the world, no matter if it is actually necessary or not. From a geometric point of view, any action related to objects beyond a distance of 1.5 m is removed. This forces the astronaut to navigate carefully toward a desired target object. One particular context specific filter in the SUPVIS Justin experiment is the fact that the astronaut is not allowed to navigate with the robot while the DIP is connected to a SPU.

The software is developed during the preparations for the SUPVIS Justin experiment. It was noticed that novice users would either tend to underestimate the capabilities of the robot and act too careful, or overestimate it, which resulted in several critical and unforeseen situations. The software was successfully deployed before the first astronaut training in May 2017, providing a safe and reliable way to guide the participating astronaut in its task.

5. CONCLUSIONS

This work describes a mission control concept for space robot operations in close collaboration with human astronauts. It shares the workload between the robot, the astronaut commanding it, and a mission control facility guiding the astronaut in its decisions. It is proposed to prune the list of available robot actions, in order to provide the astronaut only with relevant commands. As such, the cognitive load of the astronaut can be reduced, while his/her mission can be accomplished in a more goal-oriented fashion. It is distinguished between three levels of abstraction to filter the available robot actions, i. e. symbolic filters, geometric filters, and context-specific filters. The proposed concept is implemented for the METERON SUPVIS Justin experiment, to guide an astronaut aboard the ISS during the maintenance and repair of a solar panel farm in a simulated Mars environment.

The concept describes a flexible method to support an astronaut during space robot operations with long communication times. It provides an astronaut with a relevant set of robot commands to solve a wide range from simple to complex tasks by means of supervised autonomy. This concept is most beneficial as the ground segment would be unable to react in time on unforeseen situations, while the astronaut is equipped with the means to react timely based on the provided list of actions.Despite high latencies, the concept allows to enable additional commands within a predictable time frame. However, long communication times make it impossible to revoke a command immediately, such that the astronaut has to stay aware of the situation at all times. Nevertheless, the conducted experiments so far have shown that the proposed approach is inevitable to be able to robustly, and safely, teleoperate robots by means of high-level interfaces.

An additional insight is that the goal-oriented reduction of the robot commands allows to reduce the instructions for the astronaut. Typically, astronauts are provided with detailed procedures that guide them through every single step of an experiment. However, these kind of procedures are not applicable for missions to distant planets as the crew might be confronted with unforeseen events, where no concrete mission objectives are available. In this regard, the proposed approach was rated beneficial by the trained astronaut, as he/she was always able to maintain a good overview on how to achieve the desired goal based on the provided commands.

ACKNOWLEDGMENTS

This work was conducted within the scope of the METERON project. We would like to thank the ESA Telerobotics & Haptics Laboratory for the cooperation within the SUPVIS Justin experiment. Furthermore we would like to thank GSOC, Col-CC, EAC, and DAMEC for their support during experiment preparations, testing, and astronaut training.

REFERENCES

[1] J. N. Pelton, "Current Space Debris Remediation and On-Orbit Servicing Initiatives," in *New Solutions for the Space Debris Problem*, pp. 11–29, Springer, 2015.

[2] D. F. Blake, R. V. Morris, G. Kocurek, S. Morrison, R. T. Downs, D. Bish, D. Ming, K. Edgett, D. Rubin, W. Goetz, *et al.*, "Curiosity at Gale Crater, Mars: Characterization and Analysis of the Rocknest Sand Shadow," *Science*, vol. 341, no. 6153, p. 1239505, 2013.

[3] H. Benaroya, L. Bernold, and K. M. Chua, "Engineering, Design and Construction of Lunar Bases," *Journal of Aerospace Engineering*, vol. 15, no. 2, pp. 33–45, 2002.

[4] A. Schiele, "METERON - Validating Orbit-to-Ground Telerobotics Operations Technologies," in *Proc. of the 11th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2011.

[5] V. Verma, A. Jónsson, R. Simmons, T. Estlin, and R. Levinson, "Survey of Command Execution Systems for NASA Spacecraft and Robots," in *Plan Execution: A Reality Check Workshop at the International Conference on Automated Planning and Scheduling (ICAPS)*, 2005.

[6] J. Trimble, "Reconfigurable Software for Mission Operations," in *13th International Conference on Space Operations (SpaceOps)*, p. 1832, 2014.

[7] S. G. Brunner, F. Steinmetz, R. Belder, and A. Dömel, "RAFCON: A Graphical Tool for Engineering Complex, Robotic Tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3283–3290, 2016.

[8] M. J. Schuster, C. Brand, S. G. Brunner, P. Lehner, J. Reill, S. Riedel, T. Bodenmüller, K. Bussmann, S. Büttner, A. Dömel, *et al.*, "The LRU Rover for Autonomous Planetary Exploration and its Success in the SpaceBotCamp Challenge," in *International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 7–14, 2016.

[9] A. Wedler, M. Hellerer, B. Rebele, H. Gmeiner, B. Vodermayer, T. Bellmann, S. Barthelmes, R. Rosta, C. Lange, L. Witte, *et al.*, "ROBEX–Components and Methods for the Planetary Exploration Demonstration Mission," in *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2015.

[10] N. Y. Lii, Z. Chen, M. A. Roa, M. Annika, B. Pleintinger, and C. Borst, "Toward a task space framework for gesture commanded telemanipulation," in *21st IEEE International Symposium on Robot and Human Interactive Communication - Ro-Man 2012*, pp. 925–932, 2012.

[11] T. Krueger and A. Schiele, "Preparations for the Haptics-2 Space Experiment On-board the International Space Station," in *Proc. of the 13th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2015.

[12] A. Schiele, "Towards the Interact Space Experiment: Controlling an Outdoor Robot on Earth's Surface from Space," in *Proc. of the 13th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2015.

[13] N. Y. Lii, D. Leidner, A. Schiele, P. Birkenkampf, R. Bayer, B. Pleintinger, A. Meissner, and A. Balzer, "Simulating an Extraterrestrial Environment for Robotic Space Exploration: The METERON SUPVIS Justin Telerobotic Experiment and the SOLEX Proving Ground," in *Proc. of the 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2015.

[14] N. Y. Lii, D. Leidner, P. Birkenkampf, B. Pleintinger, R. Bayer, and T. Krueger, "Toward Scalable Intuitive Teleoperation of Robots for Space Deployment with the METERON SUPVIS Justin Experiment," in *Proc. of the 14th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2017.

[15] J. Schwendner, M. Hoeckelmann, M. Schroer, T. Voegele, P. Weiss, B. Gardette, V. Taillebot, T. Gobert, A. Nolte, M. Roberts, B. Imhof, W. Hoheneder, S. Ransom, R. Davenport, R. Waclavicek, D. Urbina, T. Hoppenbrouwers, K. Robert Fossum, V. Parro Garc, and O. Prieto, "Surface Exploration Analogue Simulations with a Crew Support Robot," in *IAA Space Exploration Conference*, 2014.

[16] P. Birkenkampf, D. Leidner, and N. Y. Lii, "Ubiquitous User Interface Design for Space Robotic Operation," in *Proc. of the 14th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2017.

[17] P. Birkenkampf, D. Leidner, and C. Borst, "A Knowledge-Driven Shared Autonomy Human-Robot Interface for Tablet Computers," in *in Proc. of the IEEE/RAS International Conference on Humanoid Robots (ICHR)*, pp. 152–159, 2014.

[18] D. Leidner, P. Birkenkampf, N. Y. Lii, and C. Borst, "Enhancing Supervised Autonomy for Extraterrestrial Applications by Sharing Knowledge between Humans and Robots," in *Proc. of the Workshop on How to Make Best Use of a Human Supervisor for Semi-Autonomous Humanoid Operation at IEEE-RAS International Conference on Humanoid Robots (ICHR)*, 2014.

[19] C. Borst, T. Wimböck, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, *et al.*, "Rollin'Justin - Mobile Platform with Variable Base," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1597–1598, 2009.

[20] D. Leidner, A. Dietrich, F. Schmidt, C. Borst, and A. Albu-Schäffer, "Object-Centered Hybrid Reasoning for Whole-Body Mobile Manipulation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1828–1835, 2014.

[21] D. Leidner, A. Dietrich, M. Beetz, and A. Albu-Schäffer, "Knowledge-enabled Parameterization of Whole-Body Control Strategies for Compliant Service Robots," *Autonomous Robots (AURO): Special Issue on Whole-Body Control of Contacts and Dynamics for Humanoid Robots*, vol. 40, no. 3, pp. 519–536, 2016.

[22] D. Leidner and M. Beetz, "Inferring the Effects of Wiping Motions based on Haptic Perception," in *Proc. of the IEEE/RAS International Conference on Humanoid Robots (ICHR)*, pp. 461–468, 2016.

[23] D. Leidner, C. Borst, and G. Hirzinger, "Things Are Made for What They Are: Solving Manipulation Tasks by Using Functional Object Classes," in *Proc. of the IEEE/RAS International Conference on Humanoid Robots (ICHR)*, pp. 429–435, 2012.

[24] M. Ghallab, A. Howe, D. Christianson, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "PDDL - The Planning Domain Definition Language," *AIPS98 Planning Committee*, vol. 78, no. 4, pp. 1–27, 1998.