Toward Multi User Knowledge Driven Teleoperation of a Robotic Team with Scalable Autonomy

P. Schmaus[1], N. Batti[1], A. Bauer[1], J. Beck[2], T. Chupin[2], E. Den Exter[2], N. Grabner[1], A. Köpken[1], F. Lay[1], M. Sewtz[1], D. Leidner[1], T. Krüger[2], and N. Y. Lii[1]

***Abstract*— This paper proposes a knowledge-driven teleoperation framework that enables multiple operators to command a team of robots to execute complex tasks in an efficient and intuitive manner. The framework leverages a shared knowledge base that captures domain-specific information and procedural knowledge about the task at hand. This knowledge base is used by a hybrid planner to generate context-specifically relevant commands for supervised autonomy robot command as well as direct teleoperation modes. By filtering the available commands, the operators are guided in their decision-making towards efficient task completion. This paper further extends our knowledge driven approach to address the switching between multiple operators and robotic assets, with the aim to be able scale up human-robot team for space exploration. Overall, this work represents a step towards more intelligent and collaborative teleoperation systems. The described system will be used in the Surface Avatar ISS-to-ground experiments slated for July 2023.**

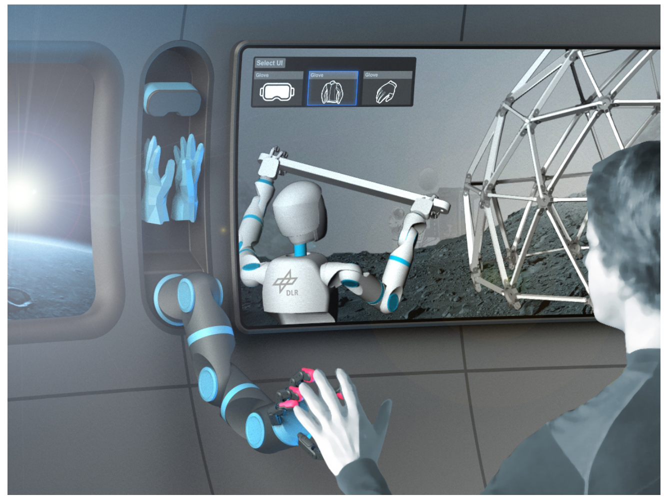

Fig. 1. Concept of a multimodal console for scalable autonomy teleoperation, utilizing various inputs along with task-level commands to facilitate commanding a team of robots based on task-specific requirements [2]

I. Introduction

As humankind ventures back to the Moon and sets its sights on Mars, space agencies around the world are working to develop new technologies and strategies to enable these missions. The *European Space Agency (ESA)* has laid out its goals in the Terrae Novae 2030+ strategy roadmap, which aims to "lead Europe's human journey into the Solar System using robots as precursors and scouts, and to return the benefits of exploration back to society" [1].

To achieve these goals, ESA plans to use robots for accessing the surfaces of celestial bodies, including the Moon and Mars. These robots will assist astronauts in numerous tasks, including exploration, scientific experiments, and infrastructure setup and maintenance as depicted in Figure 1. However, as communication delays increase with distance from Earth, it becomes increasingly difficult to directly teleoperate the robots in a traditional way. This means that autonomous capabilities of the robots become more important.

To reduce communication delays and enable more efficient robotic operation, the robots can be commanded from astronauts on board an orbiting spacecraft. However, the difficulties of the microgravity environment and the mental load of operating a spacecraft make it important to limit the astronaut's utilization for robot commanding. In addition, communication links to the surface robots may be hindered by limited bandwidth, delay, and jitter.

To address these challenges, intelligent robotic co-workers are proposed to autonomously handle tasks on celestial surfaces. In cases where the robot's autonomy is insufficient, astronauts can remotely operate the system as avatars using teleoperation methods that are tolerant to challenging communication channel characteristics. This requires the ability to scale the autonomy of the robot in order to account for the current situation and personal preferences of the astronaut.

The work described in this paper builds on prior work on a knowledge-driven approach for effective teleoperation of an intelligent service robot [3], exploring planet geology through force-feedback telemanipulation from an orbiting spacecraft [4], and extending the knowledge-driven approach for direct teleoperation of a robotic avatar [5]. The contribution of this paper is an extension of the previously described system for the use with multiple operators with varying level of expertise and connectivity to the robotic team. In addition to this, a system for generating high-level commands through analyzing command sequences is proposed and discussed. The extended approach enables efficient commanding of a heterogeneous robot team through a team of operators, which is critical for the success of planetary expeditions.

II. Related Work

Over the past several years, there have been significant advancements in the field of space telerobotics, with numerous projects and experiments being conducted by space agencies around the world. This section summarizes the space-to-ground teleoperation projects lead by DLR and ESA of the

[1]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany peter.schmaus@dlr.de
[2]Human Robot Interaction Lab, European Space Agency (ESA), Noordwijk, The Netherlands

 DOI: 10.1109/SMC53992.2023.10394365

past eight years in order to provide context for this work.

From 2015 to 2017, cosmonauts in orbit commanded robots on Earth in the Kontur-2 project [6] [7]. A 2-*Degree-Of-Freedom (DOF)* force-reflection joystick was used in the Russian Svezda module of the ISS to command robots located in Germany and Russia through a direct station-to-ground communication link with low latency. This allowed for direct telepresent robot command from a microgravity environment, enabling the crew to interact with unmodelled rigid objects in a remote environment using a robotic avatar.

In the *Multi-Purpose End-To-End Robotic Operation Network (METERON)* project, ESA, DLR, NASA, and Roscosmos investigated the operation and relevant technology of space telerobotics [8]. The METERON HAPTICS experiments, deployed a 1-DOF force-feedback joystick together with a tablet computer inside the Columbus module of the ISS, and investigated the astronaut's perception of force-feedback in a microgravity environment [9] [10]. The METERON Interact experiment supplemented teleoperation with semi-autonomous navigation capabilities of the ground robot, enabling the astronaut to execute a sub-millimeter precision peg-in-hole task using a robotic rover located at the *European Space research and TEchnology Centre (ESTEC)* [11].

In the METERON SUPVIS-E and SUPVIS-M experiments, ESA investigated the use of supervisory robot command [12] for optimizing the workload balance between the robot and astronaut [13], while in the METERON SUPVIS Justin experiment, the focus shifted to treating the robot as a coworker of the astronaut [14] [15]. DLR's Rollin' Justin robot provided intelligent features such as autonomous object detection, reasoning, and action execution [16] [17]. A *Graphical User Interface (GUI)* installed on the tablet computer upmassed to the ISS allowed the astronaut to select robot actions which Justin would autonomously execute [18] [3]. The METERON ANALOG-1 experiment expanded investigations of robot command user interface by introducing the *Robot Command Terminal (RCT)*, consisting of a laptop computer, a 3-DOF joystick with buttons, and a 7-DOF force-feedback haptic input device. The RCT allowed for open-loop teleoperation and force-feedback teleoperation of a rover at ESTEC for navigation and sampling tasks [4].

III. Knowledge Driven Teleoperation of Space Assistance Robots

The limited time of astronauts in space should be utilized in the most effective way possible. Astronauts are highly trained professionals with specific scientific and technical expertise, and their cognitive abilities are invaluable to conducting complex research in space. Repetitive and cumbersome robotic operations can be time-consuming and may not require the full cognitive abilities of astronauts. Therefore, it is essential to automate and delegate routine tasks to robotic systems, allowing astronauts to focus on more intellectually challenging and scientifically significant research activities.

Due to the various challenges of space communication links, such as signal delay, jitter, limited bandwidth, and packet loss, it can be difficult to remotely command robots. Ground control may be inefficient and slow for time-sensitive decisions. Therefore, autonomous robots with the ability to make decisions based on their sensors and pre-programmed algorithms are crucial for mission success.

Another important aspect of autonomous robots in space is their ability to reason about the outcome of their actions in order to ensure safety. In space, the upmassing of components and repairing systems can be impossible or highly challenging, making it crucial for autonomous systems to operate safely and avoid potential hazards. By incorporating reasoning and decision-making capabilities, robots can assess the consequences of their actions and adjust their behavior to avoid potential hazards or unwanted outcomes.

The operation of such autonomous robots would only require intervention from astronauts in the event of a failure of the robot's autonomy. Therefore, the commands sent to the robot can be considered as a side task for the astronaut, and they can be executed during periods when the astronaut is not occupied with other activities. This allows the astronaut to maximize their time in space and focus on more challenging tasks that require human intervention, while the robot performs routine operations independently.

A. How to represent the knowledge?

In the context of autonomous space robots, the organization and management of their knowledge base is crucial for successful mission operations. While autonomously learning systems show impressive adaptability, their unpredictable output can pose risks to mission safety. Therefore, careful control and management of the robot's knowledge base are essential. Previous research has demonstrated the benefits of organizing the knowledge of the robot in an object-centered context, which encompasses advantages for both the operation of the robot and supervisory command. This organizes information related to objects and manipulation instructions alongside the objects. By utilizing an inheritance mechanism, the object-centered approach allows the knowledge of parent objects to be reused by their children. This not only reduces the workload required to populate the knowledge base but also facilitates the process of introducing similar objects into the system. To optimize the possibility of success in a particular mission, the knowledge base of the robot should be populated by mission or robotics experts to ensure that the knowledge aligns with mission requirements.

One important aspect of knowledge representation for autonomous space robots is the ability to manage and update information in case an unforeseen situation occurs. This includes handling changes in the environment, such as the introduction of new objects or alterations to the existing ones. It is also essential to ensure that the knowledge base is kept up-to-date with the latest mission requirements and objectives. The object-centered approach achieves this by employing a modular design for the knowledge base, where each module contains information related to a specific aspect of the specific object, such as perception, navigation, or manipulation.

B. How to share the knowledge?

The object-centered knowledge base enables robots to reason about objects in their environment and their properties. In order to make more informed decisions and achieve their goals more efficiently, robots can share information about objects with other agents in the system. There are several ways in which robots can share information with each other:

First, the robots can directly exchange information about objects they encounter in the environment. For example, when one robot detects an object, it can transmit its properties and location to other robots in the vicinity. This way, all robots in the area can update their knowledge representation of the environment and objects within it.

Second, the robots can use a centralized server or database to share information about objects. Each robot can update the central database with its observations, and other robots can then access this information. This approach is particularly useful in scenarios where robots are distributed over a large area and cannot directly communicate with each other.

Finally, robots can also use a hybrid approach where they combine direct communication with a centralized database. For instance, each robot can maintain a local cache of object information it has encountered, and periodically synchronize this cache with a central database. This approach can be particularly useful in space scenarios where robots may need to leave areas with continuous network connectivity in order in order to reach scientifically interesting sites.

In all of these ways, the object-centered knowledge base enables robots to share information about objects and their properties. By sharing information, robots can make more informed decisions, avoid collisions and conflicts, and ultimately achieve the goals more quickly and reliably.

C. How to represent autonomous behavior?

To represent how a robot interacts with objects in its environment, our approach involves Action Templates (ATs). This method separates the knowledge about object handling in a robot-independent manner, with a symbolic header and a geometric body [15]. The symbolic header specifies the action in Planning Domain Definition Language (PDDL), detailing the parameters, preconditions, and effects of each action. A symbolic planner then uses this information to determine a possible sequence of ATs to reach a desired goal state. The geometric body defines the process model for interacting with the object, enabling the intended action to be grounded to the physical robot. As a result, a sequence of operations is defined that describes the robot's movement as it executes the AT. These operations are robot-agnostic, allowing any robot implementing all operations in a geometric body of an AT to execute the underlying manipulation.

A hybrid planning approach is used for defining the actual interactions of the robot with its environment. The hybrid planner starts with the symbolic planner, which generates a sequence of ATs based on the symbolic information in the headers of each action. The sequence of ATs is designed to reach a desired goal state. Afterwards the geometric planner takes over and generates robot-specific execution plans for each AT in the sequence. These plans specify how the robot will execute the actions and interact with the environment. If one of the planners fails, geometric and symbolic backtracking allows to generate alternative solutions to the planning problem.

In other words, if the initial plan fails, the hybrid planning system will work to find a new sequence of ATs that will allow the robot to achieve its goal. The backtracking process involves revisiting the symbolic and geometric information and attempting to find a new sequence of ATs that will lead to a successful plan. This process continues until a valid plan is found or until it is determined that a valid plan cannot be generated.

The described approach can be highly effective in achieving a desired goal state in a predictable and systematic way. However, it has its limitations when it comes to dynamic environments or situations where anomalies occur during execution, such as sensor failures or unexpected obstacles. In such cases, the transition of symbolic properties can no longer be guaranteed, and the planned sequence of action templates may no longer be valid. One possible solution is to manually update the properties by an expert, who can identify and correct the deviations from the expected behavior. Another approach is to incorporate an autonomous detection and correction mechanism that can identify and correct deviations automatically. An intermediate step could involve incorporating the astronaut operator, who could validate any critical state changes before allowing the robot to proceed with the next action.

The deterministic planning process, based on an expert-populated knowledge base, provides a key advantage of the approach for space operations by ensuring that the autonomous behavior of the robot is always explainable. Furthermore, the sequential execution of action templates enables online and offline monitoring and modification of sequences. In addition, this approach allows for the incorporation of safety checks and error handling mechanisms, which can ensure that the robot operates within predefined safety limits. These checks can include monitoring the robot's internal state, such as its battery level or temperature, as well as external factors, such as obstacles or changes in the environment. This makes the system robust to unexpected events and allows more reliable operation in space missions.

The METERON SUPVIS Justin experiment demonstrated that the described approach is appropriate for supervising the supervised autonomy command of a robot with limited autonomy, as long as an operator is involved in managing and assessing the execution of actions and the robot's perception.

D. How to integrate direct command?

The object-centered knowledge representation approach is well-suited for autonomous interaction with objects in the robot's environment. However, when it comes to interacting with unknown or unmodelled objects or commanding robot functions, this approach is not sufficient. By extending the knowledge driven approach in order to allow commanding the robot on a lower level of autonomy by including direct robot commands, a scalable autonomy system is realized. The challenge lies in finding a way to represent these direct robot commands in the object-centered domain without compromising the autonomy of the robot.

The integration of robot-centric functions into the object-centered domain is achieved by treating the robot as an object in the knowledge management system. To allow for knowledge-driven teleoperation, an AT is created for each robotic skill or function needed for operating the robot on lower autonomy levels. The symbolic header of the AT is used to ensure that the robot is in a symbolically safe state for action execution. Symbolic properties of the objects in the environment that may be changed by the directly commanded robot actions are invalidated to trigger re-evaluation before continuing autonomous operation. The geometric body of the AT is used to plan the execution of the robot-centric functions, and can be accompanied by safeguarding mechanisms to ensure robust execution. The current operation mode of the robot, such as controller mode and localization accuracy, is tracked as symbolic properties of the robot object. This information is then used to transition autonomously between different autonomy layers of the scalable autonomy system by sequencing the ATs accordingly. This system enables the seamless integration of traditional skill-centered systems into the object-centered domain.

The integration of teleoperation modes is a critical focus of scalable autonomy. The ability to remotely control robots in a range of scenarios is essential for a variety of applications, including exploration, construction, and rescue missions. Three teleoperation modes cover most use cases of today's systems: discrete, open loop, and closed loop teleoperation, each with its own unique benefits and challenges.

In the discrete mode, the operator specifies a single target configuration or pose for the robot to reach. This target is a parameter of the underlying AT, and the hybrid reasoning system is used to plan the required robot movements to reach the target safely. This mode is already covered by the system needed for autonomous command and execution, so no further infrastructure work is needed.

Open loop teleoperation is often used for velocity or position command where the operator specifies and continuously updates a target. This mode adds a data channel for streaming the commands to the robot. An additional visual feedback channel is often provided if the operator has no visual of the robot, such as in cases where the robot is at a great distance. In the teleoperation system, the operator can use various input devices to control the robot's movements. These input devices are transmitted as parameters of the underlying AT. By using a generic controller, the input devices can be used robot- and mode-independent by reconfiguration through the currently active AT. This decouples the teleoperation controller of the robot from the *User Interface (UI)* providing flexibility to the system and allows for easy integration of different types of input devices and robots, without the need to develop specific teleoperation controllers for each device.

Closed loop teleoperation is a mode where the operator provides and continuously updates a target while receiving a continuous feedback from the robot. A typical application of this mode is force-feedback teleoperation, where the operator needs to feel the forces and torques applied by the robot. To enable this, closed loop teleoperation adds another data channel for streaming feedback information, such as forces, torques, and distances, to the operator. The operator can then use this feedback to adjust their inputs and improve the precision of the robot's movements. Despite the added complexity, the handling of input devices in closed loop teleoperation is similar to that of open loop teleoperation.

E. How to generate operator commands?

The use of a symbolic planning algorithm enables the determination of all feasible actions based on the current symbolic state of the environment and the capabilities of the robot. This planning algorithm can generate an extensive amount of commands for achieving arbitrary symbolic goal states. However, in order to prevent cognitive overload for the operator, the generated commands are filtered. This is achieved through a Mission Control utility, which is managed by mission and task specialists. The Mission Control utility allows for the definition and updating of filters, which are specific to the context of the mission. For instance, filters can be used to remove scientific sample taking commands when the operator's mission is to repair a planetary asset. By using this approach, the number of generated commands available to the operator is reduced to a manageable amount.

Moreover, the filters applied to the generated commands can be adjusted dynamically based on the changing needs of the mission. For example, if the operator needs to perform a complex manipulation task, then the filters can be adjusted to allow for the additional commands needed to complete the task. The Mission Control utility also provides a means for monitoring the filtered commands and their execution, thus allowing the mission and task specialists to evaluate the performance of the robot during the mission. This approach ensures that the generated commands are always relevant to the current mission requirements, and that the operator's cognitive load is minimized, enabling them to focus on the critical aspects of the mission.

Another approach for offloading low-level robot commanding work from the operator is based on a generating high-level commands by analyzing previous command sequences. This approach involves analyzing the patterns of commands generated by the symbolic planning algorithm, and grouping them into high-level commands that can be executed with a single input from the operator. For example, if the operator frequently uses a series of commands to service a planetary asset, these commands can be grouped into a high-level command such as "service asset." This high-level command can then be added to the list of available commands. The generation of the high-level command names can be done using a Large Language Model, e.g. ChatGPT, using the underlying AT sequence or the PDDL output of the symbolic planner as input. The describe approach for generating high-level commands can also include machine learning techniques to predict which high-level commands are likely to be required in the future based on previous command sequences. This can further reduce the cognitive load on the operator, as the system can proactively suggest high-level commands that are relevant to the current mission. However, it is important to note that this approach requires a significant amount of data to be effective, and the accuracy of the predictions will depend on the quality and quantity of the data available.

F. How to share the command between multiple operators?

Multiple operators can command a team of robots simultaneously. To achieve this, variety of techniques to manage the inputs from different operators can be employed to ensure that the team operates effectively.

A key element is the operator-specific filtering of command options. This means that the available commands for each operator are tailored to their specific mission, training, capabilities, and user interface. This ensures that each operator can focus on their assigned tasks and only see the commands that are relevant to them. Additionally, the communication channel used by each operator can also be taken into account when filtering commands. For example, an operator with high communication delay may be given a more restricted set of commands without teleoperation modes.

Another feature is the ability to lock the command of a robot. When a command is issued by an operator, the corresponding robot is locked and cannot be commanded by other operators. This ensures that the team operates efficiently and minimizes the potential for conflicting commands. However, it's also possible for operators with higher privileges to override or cancel locks set by other operators. This provides a mechanism for coordinating the team's activities and ensuring that the mission objectives are achieved in a timely and efficient manner.

Finally, the system allows for multiple operators to simultaneously monitor each robot's activities. This allows for real-time coordination between operators and ensures that any issues can be addressed quickly. Additionally, the system provides a centralized view of the robot team's activities, which allows for effective overview management of the mission as a whole.

IV. Knowledge Driven User Interface

The UI for the knowledge-driven teleoperation makes use of augmented reality technology to provide the astronaut with a clear understanding of the environment the robot is operating in. The live video feed from the robot's cameras is augmented with information on the current world state of the robot. This information is displayed as highlighted objects in the live video realized by CAD renderings as can be seen in Figure 2. This approach allows the astronaut to easily assess the quality of localization and sensor calibration. By visually highlighting the objects in the robot's environment, the system provides the astronaut with a clearer understanding of the robot's location, as well as the locations and positions of objects within the robot's operating area. This augmented reality video feed, with its ability to highlight and provide additional information on objects in the robot's environment, serves as an important tool for the astronaut when commanding the robot.

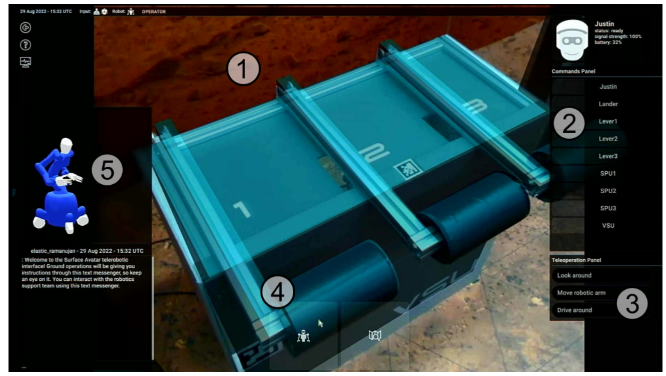

Fig. 2. Graphical User Interface for the knowledge driven approach including (1) video stream of the robots' camera, (2) available object-centered commands, (3) available teleoperation modes, (4) virtual object overlays, and (5) virtual robot viewer [5]

The available commands are determined by the respective robot using symbolic reasoning. All of the available commands originate from the object-centered knowledge knowledge base and can be bound to the object instances they manipulate. By selecting the highlight of an object the astronaut wants the robot to interact with in the GUI, the list of bounded commands is shown. This approach limits the information displayed to the astronaut to context-specific relevant information. By default, the available commands for the current robot object are shown to allow the astronauts direct access to skill-like commands, such as "look around" or "drive around". The system thus makes it easier and more intuitive for astronauts to interact with the robots and perform their tasks efficiently.

The knowledge-driven UI is enhanced by the integration of different telepresence modes which allows for intuitive teleoperation. This integration enables the use of the robot's intelligence to context-specifically tune teleoperation properties. The system has a feature that allows for the enabling and disabling of autonomous robot commands whenever a teleoperation mode is active, which ensures the prevention of interference between autonomous and teleoperated operation. Additionally, the teleoperation command channel is automatically configured based on the available user interface devices and the selected command mode. This allows for easy switching between different input devices and modes.

By using this approach, a single UI can be used to command multiple robots. The team of robots that can be commanded will be modified in the UI as new robots enter or leave the area of connectivity to the UI. This robot-agnostic access eliminates the need for custom interfaces for each individual robot and allows for a more streamlined approach to robot command. The knowledge driven approach is highly scalable, making it an excellent choice for commanding multiple robots in complex environments.

The knowledge driven teleoperation approach has been im-

Fig. 3. The Robot Command Terminal set up with a preliminary version of the User Interface and used to command Rollin' Justin in a simulated Martian environment at DLR during development

plemented for the use in the Surface Avatar space telerobotics experiment. In the experiment, various astronauts on board the ISS command a team of heterogeneous robots in Germany which are placed in a simulated Martian environment. The astronaut commands the robots using the RCT that has been upmassed for METERON ANALOG-1 [4] as depcited in Figure 3.

V. Discussion

The use of ATs for robot-object interaction has various benefits, such as a robot-independent representation of interactions and a symbolic header for action specification, as well as a geometric body for defining the process model that grounds the intended action to the physical world. However, one potential challenge in using this approach is the creation and maintenance of numerous ATs for complex interactions. For complex robotic systems with a large number of objects and interactions, the creation and maintenance of ATs can be time-consuming and challenging. This may limit the scalability and applicability of the approach, particularly in real-world applications where the environment is constantly changing and evolving. However, especially space applications benefit from the explainable and deterministic approach of ATs that are provided by human experts. For future system extensions, this human input can also be leveraged to improve machine learning algorithms. By analyzing the ATs developed by experts, machine learning models can learn from these examples and improve their own performance.

Extending the knowledge-driven approach to lower levels of autonomy is an important step towards creating a scalable autonomy system, as it enables robots to operate more autonomously while still allowing for human oversight and intervention when necessary. This can lead to increased efficiency, flexibility, and adaptability in robot operation. However, integrating the different layers of autonomy and ensuring that they work together effectively can be a complex and challenging task. Another major problem is the consistency of the symbolic properties of the objects being manipulated during teleoperation. The ATs involved in teleoperation cannot update the symbolic properties of objects they manipulate, as they lack knowledge of the objects being manipulated. This creates a risk of inconsistencies. To address this issue, observer processes can be introduced that leverage the robot's perception capabilities and physics simulations to identify potential changes in symbolic properties.

The generation of high-level commands through command sequence analysis and machine learning can greatly improve the usability of teleoperated robotic systems, making them more efficient and user-friendly. By reducing the cognitive load on the operator, the system can enable more complex and demanding missions to be performed with greater ease and accuracy. However, the need for large amounts of data to train the machine learning algorithms limits the applicability of this approach especially for space applications where bandwidth and operations time is limited. Additionally, the algorithms may not always perform accurately or reliably, especially in situations that are different from those encountered during training. Also adding another level of autonomy for the operator to choose from increases the complexity of the overall commanding approach. Still, the possible labor relief for the astronauts and thus freeing of valuable astronaut time makes it important to further investigate the possibility of high-level commands. A possible intermediate step could involve having mission experts identify and specify important command sequences that are added to the operator's available commands through the use of a Mission Control utility.

While individual robot control provides a complete understanding of each asset and a consistent interface for the astronaut, there may be other approaches to commanding a robotic team that could further reduce the astronaut's workload. Rather than issuing detailed commands for each robot, the operator could simply provide objectives for the team to accomplish. The robots could then collaborate and coordinate their actions based on their capabilities, usage, and the current environmental state. This could result in reduced cognitive load for the operator and more efficient team performance. A new mechanism for coordination among the robotic teammates needs to be investigated in order to implement this approach.

The ability for multiple operators to command a team of robots simultaneously has many positive aspects. It allows for a more efficient use of resources, as multiple operators can work together to accomplish the mission objectives. Additionally, operator-specific filtering of command options ensures that each operator can focus on their assigned tasks and only see the commands that are relevant to them. This can lead to a more streamlined and efficient mission. However, the use of locks to prevent conflicting commands can lead to delays in the mission, as operators may need to wait for a robot to be unlocked before issuing their own commands. Another potential drawback is the increased complexity of the system. Managing the inputs from multiple operators requires sophisticated filtering algorithms and communication protocols. There is also a risk of miscommunication between operators, which could lead to conflicting commands and confusion. Furthermore, allowing operators with higher privileges to override or cancel locks set by other operators may lead to power struggles or conflicts that could negatively impact the effectiveness of the team. Still, the ability to have multiple operators command a team of robots simultaneously has many potential benefits, but it also requires careful planning and management to ensure that the team operates effectively and efficiently. Proper training, communication, and coordination are essential to minimize the risk of conflicts and maximize the potential benefits of this approach.

In conclusion, the knowledge driven teleoperation has various positive aspects, such as the use of ATs and the hybrid planning approach, but also presents challenges, such as complexity and potential limitations on scalability and applicability. The success of the METERON SUPVIS Justin experiment, the preliminary sessions of Surface Avatar and the versatility of the teleoperation modes demonstrate the potential of the approach for various applications. Further research is needed to address the challenges and optimize the approach for different scenarios.

VI. Future Work

The presented knowledge-driven teleoperation approach is set to be used in the Surface Avatar orbit-to-ground experiment sessions aboard the ISS beginning in July 2023. To prepare for the final experiment session in H2/2024, the next steps involve implementing and assessing the components for sharing knowledge among robots, generating high-level commands, and applying multi-operator filters.

Acknowledgment

Realizing the Surface Avatar experiments would not have been possible without the support of the German Space Operations Center (GSOC), the Columbus Control Centre (Col-CC), and the European Astronaut Training Centre (EAC). We thank them for the support during experiment preparation, testing, and astronaut training.

References

[1] European Space Agency (ESA), "Terrae Novae 2030+ Strategy Roadmap," https://esamultimedia.esa.int/docs/HRE/Terrae_Novae_2030+strategy_roadmap.pdf, April 2023.

[2] N. Y.-S. L. etal., "Introduction to Surface Avatar: the First Heterogeneous Robotic Team to be Commanded with Scalable Autonomy from the ISS," in *73rd International Astronautical Congress (IAC)*, vol. IAC-22. International Astronautical Federation, IAF, September 2022. [Online]. Available: https://elib.dlr.de/189618/

[3] P. Schmaus, D. Leidner, T. Krüger, R. Bayer, B. Pleintinger, A. Schiele, and N. Y. Lii, "Knowledge Driven Orbit-to-Ground Teleoperation of a Robot Coworker," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 143–150, 2020.

[4] M. Panzirsch, A. Pereira, H. Singh, B. Weber, E. Ferreira, A. Gherghescu, L. Hann, E. den Exter, F. van der Hulst, L. Gerdes, L. Cencetti, K. Wormnes, J. Grenouilleau, W. Carey, R. Balachandran, T. Hulin, C. Ott, D. Leidner, A. Albu-Schäffer, N. Y. Lii, and T. Krüger, "Exploring planet geology through force-feedback telemanipulation from orbit," *Science Robotics*, vol. 7, no. 65, p. eabl6307, 2022.

[5] P. Schmaus, A. Bauer, N. B. an Maximilian Denninger, A. Koepken, F. Lay, F. Schmidt, M. Sewtz, T. Krüger, D. Leidner, A. Pereira, and N. Y. Lii, "Extending the knowledge driven approach for scalable autonomy teleoperation of a robotic avatar," in *Proceedings of the IEEE Aerospace Conference*. IEEE, March 2023.

[6] 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 International Conference on Robotics and Automation (ICRA)*, 2016.

[7] M. Stelzer, B. M. Steinmetz, P. Birkenkampf, J. Vogel, B. Brunner, and S. Kühne, "Software Architecture and Design of the Kontur-2 Mission," in *IEEE Aerospace Conference*, 2017, pp. 1–17.

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

[9] 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 *Proc. of the International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 2016, pp. 13–22.

[10] A. Schiele, T. Krüger, 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 International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016.

[11] 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.

[12] M. Sarkarati, M. Merri, K. Nergaard, and P. Steele, "How to plug-in your rover into a space mission to moon or mars," in *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*. IEEE, 2015, pp. 318–324.

[13] M. Cardone, C. Laroque, M. Sarkarati, K. Nergaard, P. Steele, and S. Martin, "MOE: A System Infrastructure for Robotic Experiments," in *Space Operations: Contributions from the Global Community*. Springer, 2017, pp. 27–52.

[14] N. Y. Lii, A. Schiele, D. Leidner, P. Birkenkampf, R. Bayer, B. Pleintinger, A. Meissner, and B. Andreas, "Simulating an Extraterrestrial Environment for Robotic Space Exploration: the METERON SUPVIS-Justin Telerobotic Experiment and The Solex Proving Ground," in *13th Symposium on Advanced Space Technologies for Robotics and Automation*, Noordwijk, The Netherlands, 2015.

[15] N. Y. Lii, D. Leidner, P. Birkenkampf, B. Pleintinger, B. Bayer, and T. Krueger, "Toward Scalable Intuitive Telecommand of Robots for Space Deployment with METERON SUPVIS Justin," in *Symposium on Advanced Space Technologies for Robotics and Automation*, 2017.

[16] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp *et al.*, "Rollin'Justin - Mobile Platform with Variable Base," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009.

[17] D. S. Leidner, *Cognitive Reasoning for Compliant Robot Manipulation*. Springer, 2019.

[18] P. Schmaus, D. Leidner, T. Krüger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, "Preliminary Insights From the METERON SUPVIS Justin Space-Robotics Experiment," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3836–3843, 2018.