Multi-Agent Mental Models for Cooperation of Heterogeneous Robots in a Space Scenario

Adrian S. Bauer[1], Sant Brinkman[2], Anne Köpken[1], Nesrine Batti[1], Timo Bachmann[1], Jörg Butterfaß[1], Tristan Ehlert[1], Emiel Den Exter[2], Werner Friedl[1], Thomas Gumpert[1], Philipp G. Knestel[1], Florian S. Lay[1], Xiaozhou Luo[1], Ajithkumar N. Manaparampil[1], Luisa Mayershofer[1], Antonin Raffin[1], Anne Reichert[1], Annika Schmidt[1], Florian Schmidt[1], Lioba Schürmann[1], Daniel Seidel[1], Nicole Roshardt[2], Michael Beetz[3], Rute Luz[2], Peter Schmaus[1], Daniel Leidner[1,3], Thomas Krüger[2], Neal Y. Lii[1]

[1]German Aerospace Center (DLR)
Institute of Robotics and Mechatronics
82234 Weßling, Germany
adrian.bauer@dlr.de
firstname.lastname@dlr.de

[2]European Space Agency (ESA)
Human Robot Interaction Lab
2201 AZ Noordwijk, Netherlands
thomas.krueger@esa.int
firstname.lastname@ext.esa.int

[3]University Bremen
Institute for Artificial Intelligence
28359 Bremen, Germany
mbeetz@uni-bremen.de

Abstract—**With each space robotic mission, a team of specialists on ground is usually required to support the robot throughout its mission. When scaling from single robots to fleets of robots, this will no longer be a viable way of conducting missions. Instead, in order to operate more autonomously in complex scenarios and adapt to dynamically changing environments without relying on a team of human experts, robots must be able to create a mental model of their surroundings. The relevance of this requirement becomes especially salient in the Surface Avatar ISS-to-ground telerobotic technology demonstration mission, where astronauts onboard the International Space Station command a team of heterogeneous robots in our lab. This work focuses on the final Surface Avatar session with NASA astronaut Jonny Kim where he was tasked with commanding the robots in a collaborative manner in order to complete simulated science and exploration tasks. To equip a complex robot such as DLR's Rollin' Justin with the ability to collaborate and coordinate with other robots, we deploy a combination of models and heuristics that allow it to create a mental model of its environment including its surrounding agents. Rollin' Justin also shares the estimated states generated in the mental model with the other robots. In this paper, we describe the application of this concept to our space experiment, present the models used to create the mental model, and evaluate it based on data collected in the final Surface Avatar experiment.**

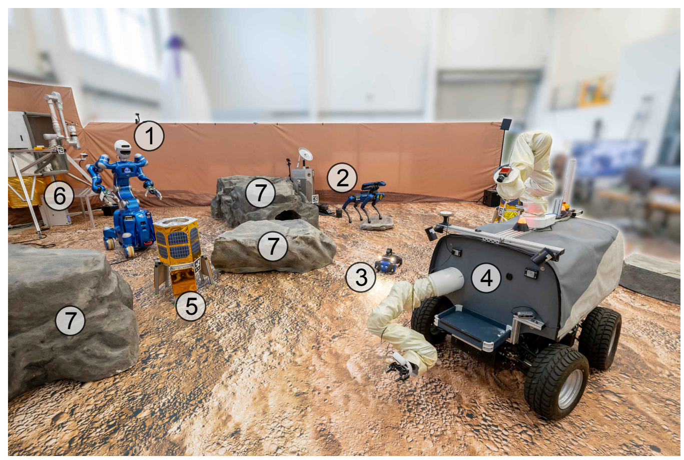


Figure 1: Ground segment setup for the final Surface Avatar experiment. The setup consists of (1) Rollin' Justin, (2) Spot, (3) BERT, (4) Interact, (5) a handover station, (6) a robotized lander, and (7) multiple boulders separating the area.

TABLE OF CONTENTS

979-8-3315-7360-7/26/$31.00 ©2026 IEEE

1. INTRODUCTION

When humanity will launch large numbers of robots to remote celestial bodies such as Moon, Mars, or beyond, the robots are going to be, at least initially, there on their own. While we expect that humans will command and supervise these robotic assets remotely, constant intervention is impractical first due to the need for one-on-one operator-robot mapping and second because of significant communication delays ranging from seconds to minutes. Due to the limited communication capabilities with human experts, the robots are required to independently make decisions on their course of action based on environmental states. Therefore, the ability for robots to understand their environment is a fundamental requirement to act autonomously.

In the Surface Avatar *International Space Station* (ISS)-to-ground technology demonstration mission *German Aerospace Center* (DLR) and *European Space Agency* (ESA) partner to research how astronauts can efficiently teleoperate a heterogeneous team of robots on a remote celestial body [1]. In order to increase the efficiency of teleoperation in this scenario, the astronauts are able to command high-level actions to the robots which the robots execute autonomously. To reduce the risk of overwhelming the astronaut with irrelevant command options, the robots filter commands based on whether they are possible in the current context [2]. Thus, in order to act autonomously and to support the astronaut in selecting meaningful commands, robots must be aware of the state of their environment.

While there might not be any humans present in proximity of the robots, the robots may share their environment with other robotic agents. To be situationally aware, robots must understand other active agents in the environments, their actions, and the resulting changes they induce in the environment. It is postulated that humans interpret their environment and predict its evolution through the use of *mental models* [3]. Mental models are simplified models of entities or processes in the environment used by humans to predict the behavior and evolution of their environment.

In the field of robotics, it is common practice to use simulations to predict the evolution of the world the robot acts in. While simulations are not usually labeled as such, they can still be seen as mental models [4, 5]. However, the drawbacks of typical simulations, such as physics simulations, become apparent with increasing complexity of the environment or variety of tasks to be simulated. With increasing size or complexity of the environment, simulation becomes computationally more and more expensive. At a certain level of complexity it gets infeasible to simulate the whole environment in high fidelity, especially considering the limitations of space hardware and limited power budgets of space robots. Furthermore, simulations are often fine tuned to certain types of interactions to simulate. A typical observation in this process is that by fine tuning a simulation for a certain type of interaction, its accuracy in other interactions drops, which is consistent with the infamous *no free lunch theorem* [6]. Beyond the cannibalistic effects of fine tuning of simulations, there is the fundamental problem that different types of simulations exist, that cover different domains, such as rigid body simulations, soft body simulations, fluid simulations, behavioral simulations, etc.

To tackle these issues, we introduced the *Multi-Agent Heterogeneous Digital Twin* (MAHDT) framework [7], that combines heterogeneous executable models, such as simulations, into a meta simulation framework. While each simulation – or model – represents a mental model, the resulting meta-model serves as a digital twin of the robot and its environment. Within this concept, heterogeneous simulations simulate distinct subsets of the environment. In this framework, the mapping between simulations and the environment subsets can change dynamically based on the context of the environment. The underlying assumption is that complex "intelligent" behavior can emerge from a combination of – potentially simple – models.

In this work, we describe an implementation of the MAHDT framework for a space scenario. In concrete terms, we show how we apply the concept of combining simple heuristics into a multi-agent heterogeneous digital twin to the final DLR-ESA Surface Avatar telerobotic technology demonstration mission in which astronauts on the ISS command a heterogeneous team of robots in our Mars analog environment at DLR in Oberpfaffenhofen, depicted in Fig. 1. The experiment session consists of multiple protocols in which the astronaut have to fulfill different goals through commanding the robotic team. This work focuses on the sample container retrieval protocol in which the astronaut has to command Rollin' Justin and Spot to jointly retrieve and stow away sample containers. Thus, the MAHDT covers both agents, Rollin' Justin and Spot. However, as the MAHDT represent the mental model from Rollin' Justin's perspective it runs as a single instance on Justin. As Rollin' Justin and Spot collaborate, Justin makes use of the framework to reason about its environment and communicates the inferred states to Spot. The contributions of our paper are the demonstration of a successful application of the MAHDT framework in a space robotics experiment, the description of the models used in our experiment to allow coordination between the robots, and an analysis of the resulting system in a space experiment. Through our work, we aim to showcase the feasibility and effectiveness of the framework in addressing the problem of building the mental model of another agent and making use of the information to enable informed decision making for robots.

In the remainder of this work we first introduce related work in Section 2 before describing the experimental scenario in Section 3 followed by the implementation of the models for the MAHDT in Section 4. Next, we present the results of our system in Section 5 and discuss them in Section 6. The paper concludes with outlook and conclusion in Section 7.

2. Related Work

Autonomous robots must continuously determine what to do. Common approaches to enable robots to select their course of action include reinforcement learning [8], behavior trees [9], vision-language-action models [10, 11], state machines/state flows, and planning-based approaches [12]. Due to the requirement of deterministic and predictable behavior in space scenarios in combination with the flexibility of planners, we believe that service robots for future space missions will employ the latter approach.

However, planning purely on a geometric level does not scale well for longer tasks due to the continuous and high-dimensional configuration spaces of robots, that serve as search spaces for solutions. A common approach to reduce the search space is by combining geometric planning in configuration space with task-level planning in a discrete symbolic state. The combination of task-level planning and geometric planning is called integrated *Task-and-Motion Planning* (TAMP) [13, 14] or *hybrid planning*. An example for using integrated TAMP in space experiments is presented in [15] where a TAMP approach based on Action Templates [16, 17] is employed.

The speed up of hybrid planning is based on a symbolic representation of the world state and comes at the cost of having to provide an accurate symbolic state for the planner. While some systems rely mostly on visual data to update the belief state of the robot like RoboSherlock [18], it is beneficial to create dynamically evolving world states to be able to estimate the state of objects outside the current field of view of the cameras and to predict their change over time.

The MAHDT framework [7] employs multiple heterogeneous models to create a combined model of the robot's environment. The rational behind this concept stems from the observation that simulations, that are often used to model the interaction between robots and their environment, are usually tuned to specific interactions and to modeling certain phenomenons, such as for example rigid body simulations versus soft body simulations versus fluid simulations. The authors propose to combine models in a meta-model, called the Multi-Agent Heterogeneous Digital Twin, that serves as a mental model that the robot creates of its environment. The MAHDT framework is employed to create standardized interfaces between heterogeneous models (such as e.g. physic simulations, state machines, surrogate models, etc.) and orchestrates them to create a combined representation of the environment. A key aspect of the MAHDT approach is that the *responsibility* of simulating an aspect of the environment can change dynamically based on the context. For example the position of a ball being grasped by a robot and dropped into a bowl of water would first be simulated by a general physics simulation. When being grasped by the robot the responsibility could be transferred to a simulation that is specialized to simulate the multi-contact grasp by the robot hand. As the robot opens the hand and drops the ball, the ball's position is simulated again by a general physics simulation until it gets in contact with the water, when the responsibility is transferred to a fluid simulation.

In this regard, the MAHDT is similar to the concept of federate simulations which are e.g. used in NASA's Artemis program [19] or the integration of Systems Modeling Language models into an executable meta model [20, 21]. The main difference between these approaches and the MAHDT lies in the dynamic responsibility transfer. This makes the MAHDT suited for dynamically changing scenarios such as those faced by service robots in space.

Mental models are a concept from psychology [3] that are (simplified) models of entities or processes in a person's environment. Mental models serve as an abstraction of the real world by representing selected concepts and relationships between them [22]. There has been the suggestion to use mental models or mental simulations as a tool for reasoning and scene understanding in robotics [4, 5]. Generally, these concepts employ physical simulations to reason about the mechanical properties of an assembly [4] or the stability of a stack of objects [5]. In the MAHDT all mental models are runnable, meaning that they can be executed and they update their state based on the inputs they receive. In turn, they can produce output that is utilized by other models.

3. Experimental Scenario

The presented work was developed in the context of the final Surface Avatar [1] ISS-to-Earth telerobotic experiment session, as a means to enable cooperation between two heterogeneous robots in one of the experiment protocols. In this section we will provide an overview of the final Surface Avatar experiment session with focus on the protocol that is linked to this work.

Overall Scenario

In Surface Avatar we investigate how astronauts in orbit can efficiently command a heterogeneous team of robots on the surface of a celestial body. The topics in focus of the mission range from user interaction [23], telepresence in time-delay conditions [24], handling of failures [25, 26], to robot autonomy [27]. The setup consists of two sites: the remote site on the ISS containing an astronaut in front of the *Robot Command Terminal* (RCT) and the experiment site on ground with the robots in an analog Mars scenario [1].

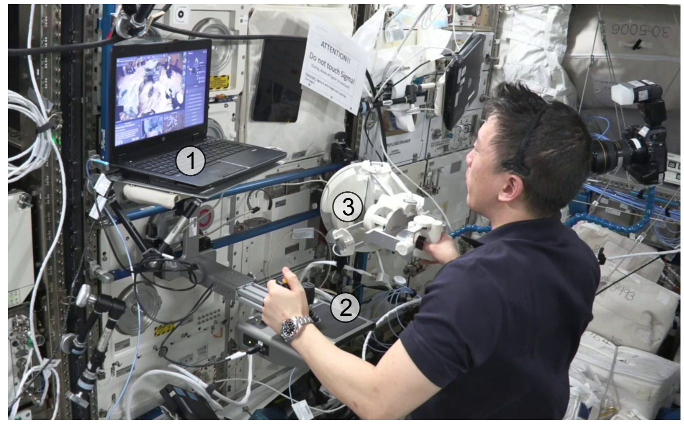


Figure 2: Space segment containing the remote teleoperation station with the Robot Command Terminal (RCT). The image shows NASA astronaut Jonny Kim in the final Surface Avatar session aboard the ISS in front of (1) the Surface Avatar GUI running on a computer, (2) a joystick, and (3) a sigma.7 haptic input device. Image Credit: ESA/NASA

The remote teleoperation station is shown in Fig. 2 and consists of a laptop, a 7-Degrees-of-Freedom (DoF) haptic input device (sigma.7[2]), and a 3 DoF joystick with 7 buttons. These devices are used to command the robotic team either by issuing high-level commands in supervised autonomy via the Graphical User Interface (GUI) on the laptop, or by teleoperating the robots via the sigma.7 or the joystick. While the sigma.7 is used to teleoperate the robotic arms with force feedback, the joystick is used to move the cameras of the robots or drive the robots through the environment. When opting to use supervised autonomy, the astronauts select a command from the graphical user interface, parameterize it if necessary, and send it to the robot. The robot then executes the command autonomously [23].

Fig. 1 shows the simulated Martian habitat on Earth with multiple robots and environmental assets. The robotic team in our experiment consists of DLR's humanoid robot Rollin' Justin [28], ESA's Interact rover [29], DLR's quadruped Bert [30], and an ESA customized Boston Dynamics Spot[3] robot with an arm. Further assets in the environment are a lander mockup and a watchtower that provide additional camera views on the scene, multiple rocks dividing the scene essentially in two parts, a handover station in the intersection of the two parts of the scene, a cave in one of the rocks, and multiple sample containers distributed in the area shared by Spot, Interact, and Bert. Due to the partition of the scene with big boulders, the robots are required to use the handover station to transfer objects between the areas.

Sample Container Retrieval Protocol

While experimental sessions of the Surface Avatar mission consist of multiple protocols with tasks that the astronauts have to achieve through commanding their robotic team, this work focuses on the methodology that supports autonomous behavior in the sample container retrieval scenario of the final Surface Avatar session with NASA astronaut Jonny Kim in July 2025. In this scenario the astronaut was tasked to pick

[2] `https://www.forcedimension.com/products/sigma` last accessed 03.10.2025

[3] `https://bostondynamics.com/products/spot/` last accessed 03.10.2025

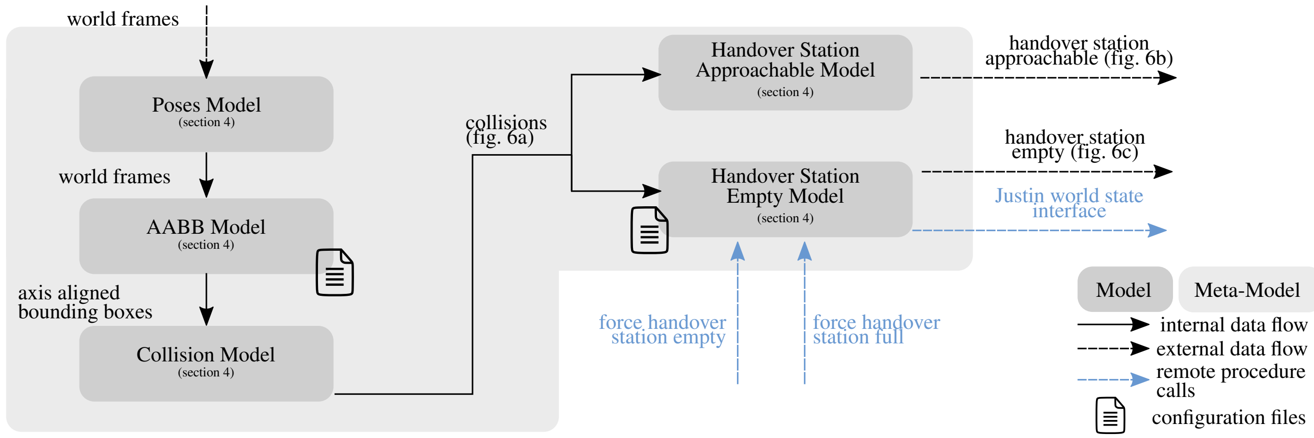


Figure 3: Overview of the data flows of the resulting meta-model (light gray). Solid lines represent data flow between models (gray) inside the meta-model, dashed lines represent external data flow. Light blue dashed arrows represent event driven remote procedure calls. Based on world frames received as input, the meta-model infers whether the handover station is safe to approach and whether it contains a sample container. The computation is achieved through the interaction of an ensemble of mental models. The model executes remote procedure calls to modify the world state of Rollin' Justin and offers services to set specific states in case of an error of the meta-model.

up sample containers and stow them on the tray of the lander. As the sample containers were distributed in the area shared by Spot, Interact, and Bert and the lander was in the area occupied by Justin, the astronaut had to pick up the sample container with Spot or Interact, place them in the handover station, pick them up with Justin from the handover station, and finally place them with Justin on the tray of the lander. Initially the handover station was covered with a lid that had to be removed.

To achieve their goal, the astronauts have the option to use pure supervised autonomy or a combination of supervised autonomy and direct teleoperation. Initially, Spot offers a command to remove the lid from the handover station autonomously. Furthermore, Spot provides an autonomous action of searching multiple sample containers, picking them up, and placing them in the handover station, one after the other. This action is, from the astronauts' perspective, an atomic action, meaning that once the astronauts command it, it is executed autonomously by Spot without any need for interaction. After having placed a container in the handover station, Spot goes back to its charging station and waits for the handover station to be empty to continue the sequence. In an optimal case, the astronaut only has to initiate this action once and then Spot executes it continuously and repeatedly. Intervention of the astronaut becomes necessary if Spot does not find a sample container autonomously. If that happens, the astronaut is able to take over control and use the joystick to move Spot to the vicinity of a sample container. The astronaut can then instruct Spot to resume the autonomous sequence of searching and picking up sample containers, allowing it to operate autonomously again.

Once a sample container is in the handover station, Justin can be commanded to navigate to the handover station, pick up the sample container, navigate to the lander, and finally place the sample container on the lander tray. Theoretically, all of Justin's actions can also be executed in direct teleoperation, however, execution in direct teleoperation usually requires more time than executing autonomous actions [25] and is limited as only the right arm of the robot can be teleoperated. Furthermore, the time delay of $\sim$800 ms between operator and robot and the limited 3D perception on the computer screen render teleoperation difficult.

Challenges of the Experiment Protocol with Respect to Robot Cooperation

Rollin' Justin's hybrid TAMP approach [16] requires a correct symbolic representation of the environment states to plan actions. One of these states in *Planning Domain Definition Language* (PDDL) syntax [31] is `in sample_container handover_station` which describes that an object of type `sample_container` is in an object of type `handover_station`. These states are connected to objects in the robot's world representation which in turn are connected to object poses. This combined information allows the robot to plan actions both on a symbolic (chaining multiple actions and selecting parameters) and a geometric level (finding feasible trajectories while avoiding obstacles). Once Spot places a sample container in the handover station, Rollin' Justin must update its world state accordingly.

Furthermore, the aforementioned experiment protocol requires communication between the robots involved in order to keep a safe distance between the robots and to coordinate their actions. The focus of this work lies on interactions between Rollin' Justin and Spot. Core information to be communicated between the robots include whether the handover station is safe to approach and whether it contains a sample container. The handover station is considered to be safe to approach, if no other agent is in its vicinity. This information is necessary as only one robot is allowed to be in the proximity of the handover station in order to avoid collisions, especially when using multiple robots in parallel. Additionally, the experiment was set up so that no more than one sample container was supposed to be in the handover station at any given time. Therefore Spot needs to know whether a sample container is in the handover station to avoid placing multiple ones in it.

One of the challenges in multi-robot collaboration in our experiment is therefore the detection of state changes concerning the handover station (being available and empty/filled) and the communication of these changes between robots. In the following we will present how we solved this by making use of the MAHDT framework and implementing the necessary models for our scenario.

4. Setup of the Multi-Agent Heterogeneous Digital Twin

The idea behind the MAHDT framework is to combine multiple models into a bigger meta simulation. The concept of a model in this framework is defined as the representation of at least one property of at least one entity, like a physical object or a virtual concept, in the world [7]. A model can also represent multiple properties of multiple entities simultaneously. This section describes the models that were used in our experiment to infer whether the handover station was safe to approach and whether it was filled with a sample container.

While models can be as complex as for example a fluid simulation to infer whether a fluid spills from a container carried by the robot, the guiding principle for the development of models is to keep them as simple as possible and as complex as necessary to save on computational resources in resource scarce scenarios. Following this principle allows to i) demonstrate that combinations of simple models can lead to "intelligent" behavior, ii) be able to run all computations on a limited power budget onboard the robot, iii) allow for easy extensibility and maintainability of the models. An example of more complex models is provided in [7].

Models

Our setup contains a model that receives object poses, published by our middleware *Links and Nodes* (LN) [32], and makes them available for the framework. Another model uses these poses to compute *Axis Aligned Bounding Boxes* (AABBs) [33] for the objects in the scene. Based on this information a third model computes collisions between the AABBs which represent objects/robots being in close vicinity. Given these general models, two additional models are used to infer whether the handover station is available for approach and whether it is filled. In the following we describe these models in detail before describing the resulting meta-model that is also visualized in Fig. 3.

Poses Model—The *Poses Model* is responsible for modeling the pose of objects in the world and providing this data to other models in the framework. It subscribes to data published by the robots about their current location and the location of other environmental assets. As we focus on the interaction between Rollin' Justin, Spot, and the handover station, the poses of these three assets are handled by the model. Despite its simplicity, we consider this to be a model as it represents the poses of entities in the world and "predicts" the next pose by updating the information from the topic.

AABB Model— Based on the output of the Poses Model, the *AABB Model* computes AABBs of Rollin' Justin, Spot, and the handover station. The model is configured with low-polygonal convex hulls of the objects and uses them to compute the AABBs. As we are interested in knowing when the robots are in *proximity* of the handover station, we bloat the convex hulls such that they overlap when the robots interact with the handover station.

Collision Model— This model consumes the AABBs and checks for collisions between the different entities. It uses the AABB collision algorithm [33] to detect overlaps between the AABBs and provides them for the framework.

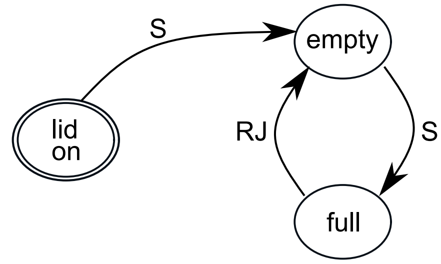
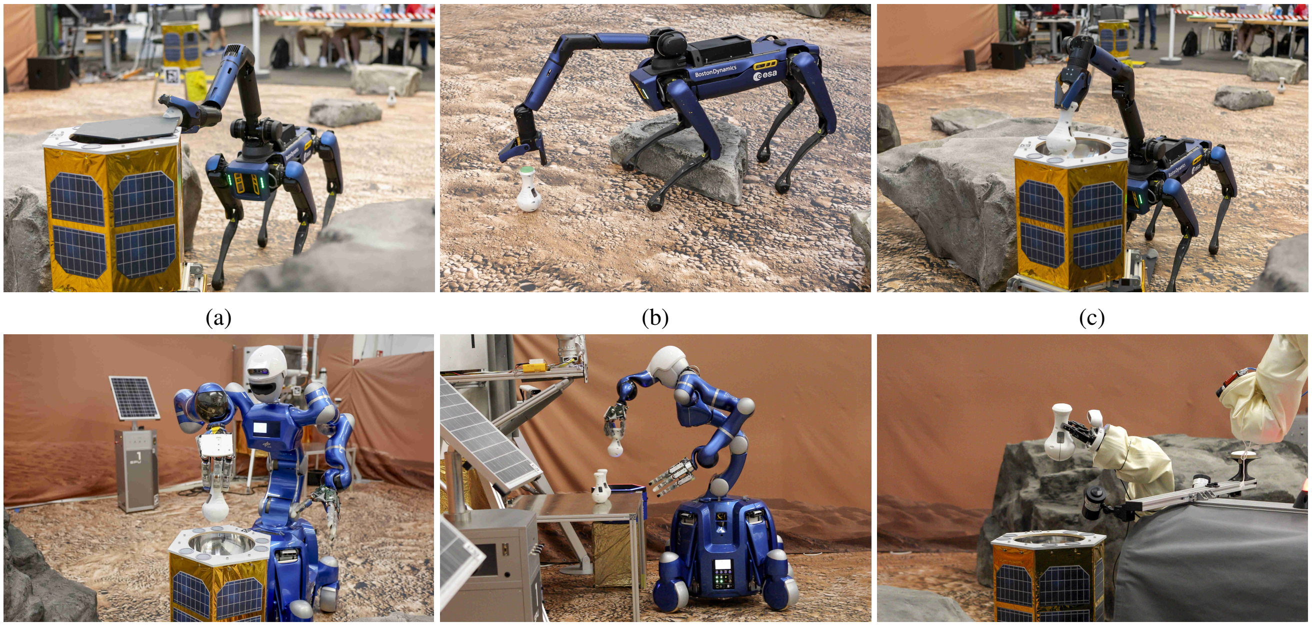


Figure 4: Depiction of the state machine to model whether the handover station is covered by a lid (*lid on*) a sample container is in the handover station (*full*) or not (*empty*). Transitions between states happen through interactions between Spot and the handover station (*S*) or Rollin' Justin and the handover station (*RJ*). Initially the state machine starts in the *lid on* state.

Handover Station Approachable Model— While the previously described models are rather general, the *Handover Station Approachable Model* is specific to this experiment. It consumes the collision information between the handover station and each of the robots to provide a custom `approachable` state for the handover station. This state is exposed to the robots who utilize it to determine whether it is safe to approach the handover station or not. The goal of this module is to prevent possible damage to the robots due to collisions with each other by marking the handover station as `not approachable` when a robot is in its vicinity.

Handover Station Empty Model—This is another experiment-specific model that is tuned to the experiment protocol to infer whether the handover station contains a sample container or whether it is empty. The model is set up as a state machine consisting of three states (see Fig. 4). The first state represents the handover station being covered by a lid which is the initial state of the experiment. The second and third state represent the handover station being empty or filled with a sample container respectively. The transitions of the state machine are depicted in Fig. 4 and are based on a heuristic of interactions between the robots and the handover station. This representation is inspired by how we imagine a human to reason about such a situation. We assume that a human who is tasked to take out the sample container from the handover station whenever a colleague has placed one would check the handover station once they see their colleague interacting with it. While a human might have a more detailed model of an interaction, e.g. observing the motions of their colleague to identify whether they place a sample container in the handover station, our model is only able to distinguish between by-passing and a longer interaction.

The transitions of the state machine are defined as follows: State *lid on* transitions to state *empty* when Spot interacts with the handover station for the first time. State *empty* transitions to state *filled* when Spot interacts with the handover station (again) and state *filled* transitions to state *empty* when Justin interacts with the handover station. This is a strongly simplified heuristic based on the a priori knowledge of the mission of the robots. The model registers an interaction of a robot with the handover station when the robot leaves collision with the handover station after having been in collision with it for a time t_{coll} greater than a predefined threshold $t_{coll} \geq t_{thr}$. The thresholds are provided as a configuration to the model.



(a) (b) (c)

(d) (e) (f)

Figure 5: Images taken during the final Surface Avatar experiment session. (a) shows Spot removing the lid from the handover station, (b) shows Spot picking up a sample container, and (c) shows Spot placing the sample container in the handover station. In the second row, (d) shows Justin picking up the sample container from the handover station and (e) placing it on the tray of the lander. (f) shows Interact being teleoperated to place the sample container in the handover station.

In order to influence the behavior of the robots, this model provides its information to the robots in two different ways. On the one hand, the model publishes the current state of the handover station as a boolean variable in a topic of the middleware of choice. In our setup, the intra-robot communication is realized via *Data Distribution Service* (DDS)[4] topics. This information is used by Spot to determine whether to go searching for a new sample container (if the handover station is empty) or stay in the charging station to charge (when the handover station is filled). While charging, Spot subscribes to the DDS topic and waits until it is being notified that the state of the handover station changes to *empty*. Once it receives this notification, Spot goes searching for another sample container without any interaction required from the astronaut.

On the other hand, as the MAHDT serves as a mental model for Rollin' Justin, it is connected to Justin's world state representation and is able to change the robot's current belief state. Therefore it spawns a sample container inside the handover station when the handover station changes the state from *empty* to *full*. At this step, the pose of the sample container is only known approximately as it is known that it is in the handover station but not, where exactly and whether it is tilted. Thus, the model sets the state `localized` of the sample container to `false`. As a result, the robot is aware that a sample container is in the handover station but requires more precise localization in order to interact with it, e.g. in order to pick it up. It is necessary to manipulate the world state of Rollin' Justin in this manner, as the state change is induced by an external agent, here Spot. When the world state changes due to actions of Rollin' Justin it does not need to be adapted as Rollin' Justin is able to keep track of changes to the world state induced through its own actions. After a change of the world state, Rollin' Justin updates the list of actions that are advertised to the astronaut as described in [27].

Because this model is used in a space experiment we can not afford a failure. Thus, we do not use learned models in this scenario as we require a deterministic behavior of the model. Furthermore, it is necessary that the operators are able to overwrite the inferred state manually. Therefore, the model exposes two services that can be triggered externally, enforcing the model to either switch to the *empty* or *full* state.

Resulting Meta Model

Justin's mental model continuously estimates the state of the handover station based on the status published by Spot and data from Justin's sensors. The meta-model resulting from the connection of the models presented above can be visualized as a network of models connected by data and is visualized in Fig. 3. It requires information about the frames of objects in the world as well as configurations for the AABB model and the Handover Station Empty model. The information provided by the meta-model is whether the handover station is safe to approach and whether it is filled. In addition, the model sets the correct world state when the handover station changes from empty to filled and exposes remote procedure hooks for forcing the Handover Station Empty model to take on either the *empty* or *full* state.

5. RESULTS

The data in this section is taken from the final experiment of the Surface Avatar mission. During the ISS-to-ground experiment the protocol was split up in two parts. First the astronaut had to remove the lid from the handover station (Fig. 5a) and collect and stow away two sample containers (Fig. 5b - Fig. 5e). Then the protocol was interrupted to task

[4] `https://www.omg.org/spec/DDS/`, last accessed 03.10.2025

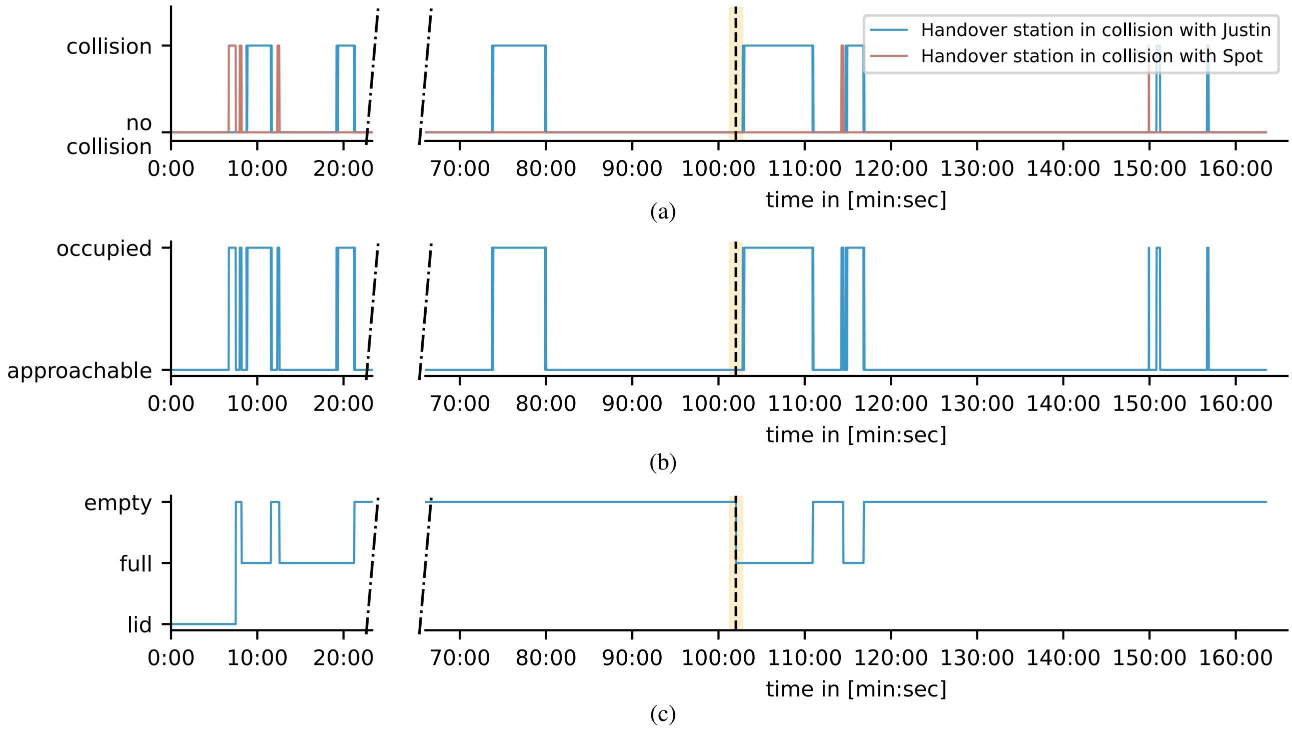


Figure 6: Results from the experiment. (a) shows the result from the collision model indicating whether a collision is estimated between the handover station and Justin (blue) and Spot (red). (b) depicts the output of the Handover Station Approachable model, indicating whether the handover station is available for a robot to navigate there or whether it is already occupied by another robot. (c) shows the state of the state machine used in the Handover Station Empty model. All plots are cut between 23:30min and 66:40min because the astronaut performed a different protocol during that time. The yellow area indicates when a manual call was issued to set the state of the handover station to *full*.

the astronaut with another protocol, before he was tasked to collect and stow two more sample containers. The MAHDT meta-model consisting of the models described in section 4 was running continuously throughout all protocols of the session.

The states estimated by the framework are depicted in Fig. 6. It shows the inferred states of the *Collision* model and the *Handover Station Approachable* model. Furthermore, it shows the active state of the state machine used in the *Handover Station Empty* model.

It can be seen that the state of the Handover Station Empty model changes on falling edges of the *Collision* model, which is because of how the model is implemented. Furthermore, it is visible, that at time $102{:}41.1\,\mathrm{min}$ the state of the Handover Station changes from empty to full without any change in the *Collision* model. At this moment, the ground operators of Rollin' Justin triggered a remote procedure call to the `force handover station full` service. This happened when Spot was unable to retrieve the next sample container that was close to Interact. Thus, the astronaut decided to teleoperate the arm of Interact to pick up the sample container and place it in the handover station (see Fig. 5f). As this was unforeseen and the MAHDT meta-model was not prepared for this situation, the ground operators took the initiative to set the correct state.

At $t \approx 150\,\mathrm{min}$, Fig. 6a shows multiple short collisions with both Spot and Justin. While the handover station is marked as not-available during these collisions, the state of the Handover Station Empty model does not change. As the initial collision with Spot is too short, the model does not consider it an interaction. Afterwards, even though the handover station gets occupied by Justin for longer times, the state of the Handover Station Empty model does not change, as it is already *empty* and an interaction with Justin can only transition it from *full* to *empty* (see fig. 4).

The MAHDT was running with a configurable rate of $10\,\mathrm{Hz}$ during the experiment onboard the application PC of Rollin' Justin. While a key feature of the framework is that the single models can run at individual rates independently from the rate of the MAHDT, in this setup all models share a rate of $10\,\mathrm{Hz}$. On average it used $\approx$1.2% of one CPU core on an Intel® Core™ i7-7820EQ.

6. Discussion

The data collected during the experiment shows that the digital twin framework worked as desired. Overall, the models were able to generate the expected outputs. Even the short collisions between the handover station and the robots at $t \approx 150{:}00\,\mathrm{min}$ were classified correctly as being too short for an interaction. During that instance, the robots passed by the handover station multiple times without interacting with it. This supports the claim, that a combination of very simple models and heuristics can be used to enable a robot to reason about its environment. The combination served as a mental model that Rollin' Justin constructed of itself, the environment, and other agents. The reasoning capabilities are, however, clearly restricted to the properties of the environment that are governed by the models. As such, models always represent the perspective of the developer that created the model and encode their understanding of the phenomenon that is modeled.

This became visible in our experiment when the astronaut selected an alternative strategy to overcome Interact blocking Spot's access to a sample container. Contrary to the developers' expectation, the astronaut chose to pick the sample container with Interact instead of relocating Interact and using Spot. This was unforeseen, as picking up the sample container in direct teleoperation with Interact is more time-consuming and labor-intensive than issuing an autonomous pick-up command to Spot. As a result, the meta-model was not equipped to handle this situation and would have needed an additional model for the interactions between Interact and the handover station. Even though it would have been little effort to add another model for Interact, this was omitted during preparation of the experiment as the designers did not anticipate this usage of Interact. This is an example for the aforementioned restriction of the mental models to the perspective of the developer. In a real space scenario we would expect most assets in the environment to contain sensors. Similarly to how the *Poses Model* makes position information available, sensors can be included in the framework to make their readings available for other models.

While it might be argued that our models are over-simplifying their underlying phenomenon, we are convinced that every model is a simplification of a real existing phenomenon. Therefore the question should not be how accurate a model is but whether it is *accurate enough* to be useful in a given scenario. George EP Box phrased this as *"Essentially, all models are wrong, but some are useful."* [34]. The MAHDT framework employed in this work allows for mixing complex and simple models. In this context, and generally in space scenarios, we opt for simple models to allow the ground operators to understand the state of the models and monitor them, if necessary. Nevertheless, we acknowledge that in space missions more sophisticated models will be required which will potentially be learned by the robot and included in the MAHDT framework. In this context, our work serves as an initial proof-of-concept for the applicability of our approach in the space domain.

In our experiment, the models were configured via hand-written configuration files. While this approach was sufficient for the scope of the experiment, the manual effort and complexity may grow rapidly with bigger scopes of properties to model. Consequently, we believe that it would be beneficial to learn the parameters of the system in a data-driven manner.

While Rollin' Justin and Spot were cooperating in this experiment, the framework is also suited for uncooperative agents as, from Rollin' Justin's perspective, it is sufficient to observe the other agent. If the robots were even more cooperative, they could also share a channel to announce their status and their next actions. This information could enrich their mental model of the respective other, taking the intentions of another into account in their decision making process.

7. Conclusion and Outlook

In this work we presented how we used the MAHDT framework in an ISS-to-ground technology demonstration mission to generate a mental model for the robot Rollin' Justin that includes modeling another agent. We showed, that an implementation of simple heuristics, including a behavioral model of another agent, enables a robot to reason about state changes in its environment that are induced by another agent and thus supports autonomy and teleoperation of the robot. To do so, the robot creates a mental model of the environment and other agents, consisting of a combinations of multiple models.

The implementation enabled Rollin' Justin to estimate whether a shared handover area was safe to visit and whether it contained a sample container to pick up. Rollin' Justin shared this information with other robots in the environment. This allowed Spot to know when the handover station was empty and when it could start to search for a new sample container to place it in the handover station. Being able to generate this mental model of the scenario and acting based on it, the robots supported the commanding astronaut who previously had to generate a mental model of the remote scene and keep it up to date to successfully fulfill his task. The astronaut was, consequently, freed from reasoning about e.g. whether it was safe to command Justin to the handover station while Spot was in its vicinity as Justin ensured, based on the state of the mental models, to offer solely safe actions to the astronaut.

During the experiments, the ground operators supervised the state of the mental models and triggered a state change when they deviated from the state as observed by the ground operators. Due to the short time available for training the astronaut on our experiment, we decided against shifting the responsibility for supervising the state of the models to the astronaut. However, in future missions this shift would be required to enable the astronaut to make the best use of the robot.

In future work we aim to investigate how robots are able to generate models of their environment on their own. In terms of user interface implementations for space (tele)robotic missions it is important to communicate the models and their state to the user and allow the user to influence both. Understanding how this can be achieved in a general manner will be of future interest as well.

We will, furthermore, continue testing this approach in more complex scenarios to investigate the potential and the limits for scaling it.

Acknowledgments

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), NASA, ESA, and NASA astronaut Jonny Kim. We thank them for the support during experiment preparation, testing, and astronaut training.

References

[1] N. Y. Lii, P. Schmaus, D. Leidner, T. Krueger, J. Grenouilleau, A. Pereira, A. Giuliano, A. S. Bauer,

A. Köpken, F. S. Lay, M. Sewtz, N. Bechtel, S. Bustamante Gomez, M. Denninger, W. Friedl, J. Butterfass, E. Ferreira, A. Gherghescu, T. Chupin, E. den Exter, L. Gerdes, M. Panzirsch, H. Singh, R. Balachandrand, T. Hulin, T. Gumpert, A. Schmidt, D. Seidel, M. Hermann, M. Maier, R. Burger, F. Schmidt, B. Weber, R. Bayer, B. Pleintinger, R. Holderried, P. H. Pavelski, A. Wedler, S. von Dombrowski, H. Maurer, M. Görner, T. Wüsthoff, S. Bertone, T. Müller, G. Söllner, C. Ehrhardt, L. Brunetti, L. Holl, M. Bévan, R. Muehlbauer, G. Visentin, and A. Albu-Schäffer, "Introduction to Surface Avatar: The First Heterogeneous Robotic Team to be Commanded with Scalable Autonomy from the ISS," in *Proc. Int. Astronaut. Congr. IAC*, vol. IAC-22. Paris, France: International Astronautical Federation, IAF, Sep. 2022.

[2] D. Leidner, P. Birkenkampf, and N. Y. Lii, "Context-aware Mission Control for Astronaut-Robot Collaboration," in *Proc 14th Symp Adv. Space Technol. Robot. Autom. ASTRA*. Leiden, The Netherlands: European Space Agency (ESA), Jun. 2017.

[3] K. Craik, *The Nature of Explanation*. Cambridge, UK: Cambridge Univ Press, 1943.

[4] M. Hegarty, "Mechanical Reasoning by Mental Simulation," *Trends in Cognitive Sciences*, vol. 8, no. 6, pp. 280–285, Jun. 2004.

[5] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an Engine of Physical Scene Understanding," *Proc. Natl. Acad. Sci.*, vol. 110, no. 45, pp. 18 327–18 332, 2013.

[6] D. Wolpert and W. Macready, "No Free Lunch Theorems for Optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[7] A. S. Bauer, A. Köpken, and D. Leidner, "Multi-Agent Heterogeneous Digital Twin Framework with Dynamic Responsibility Allocation for Complex Task Simulation," in *Proc. 21st Int. Conf. Auton. Agents Multiagent Syst.*, ser. AAMAS '22. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2022, pp. 53–61.

[8] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement Learning in Robotics: A Survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.

[9] P. Ögren and C. I. Sprague, "Behavior Trees in Robot Control Systems," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, no. Volume 5, 2022, pp. 81–107, May 2022.

[10] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," in *Proc. 7th Conf. Robot Learn.* PMLR, Dec. 2023, pp. 2165–2183.

[11] J. Liu, M. Liu, Z. Wang, P. An, X. Li, K. Zhou, S. Yang, R. Zhang, Y. Guo, and S. Zhang, "RoboMamba: Efficient Vision-Language-Action Model for Robotic Reasoning and Manipulation," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 40 085–40 110, Dec. 2024.

[12] M. Ghallab, D. S. Nau, and P. Traverso, *Automated Planning and Acting*, 1st ed. New York, NY, USA: Cambridge University Press, 2016.

[13] L. P. Kaelbling and T. Lozano-Pérez, "Integrated Task and Motion Planning in Belief Space," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, Aug. 2013.

[14] F. Gravot, S. Cambon, and R. Alami, "aSyMov: A Planner That Deals with Intricate Symbolic and Geometric Problems," in *Robot. Res. Elev. Int. Symp.*, B. Siciliano, O. Khatib, P. Dario, and R. Chatila, Eds., vol. 15. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 100–110.

[15] 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 Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3836–3843, Oct. 2018.

[16] D. Leidner, C. Borst, and G. Hirzinger, "Things are Made for What They are: Solving Manipulation Tasks by Using Functional Object Classes," in *Proc. 2012 IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2012, pp. 429–435.

[17] D. S. Leidner, *Cognitive Reasoning for Compliant Robot Manipulation*, 1st ed., ser. Springer Tracts in Advanced Robotics. Cham: Springer International Publishing, 2019, vol. 127.

[18] M. Beetz, F. Balint-Benczedi, N. Blodow, D. Nyga, T. Wiedemeyer, and Z.-C. Marton, "RoboSherlock: Unstructured Information Processing for Robot Perception," in *2015 IEEE Int. Conf. Robot. Autom. ICRA*. Seattle, WA, USA: IEEE, May 2015, pp. 1549–1556.

[19] P. Whittington, K. Dodd, E. Z. Crues, J. Gentile, and N. Trevino, "Using Distributed Simulation Capabilities to Support Studies and Mission Operations for NASA Artemis Program," in *2025 IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2025, pp. 1–11.

[20] L. Jones-Wilson, F. Karagoz, M. Vierlboeck, T. Boyer-Chammard, V. Gehlot, and H. Grip, "Mars Sample Recovery Helicopter Aerial Mode Commander Design Using Executable Models," in *2025 IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2025, pp. 1–17.

[21] F. Karagoz, D. Liu, P. Xaypraseuth, and C. Delp, "Application of Model-Based Approaches for Earth-Observation Missions," in *2025 IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2025, pp. 1–10.

[22] J. W. Forrester, "Counterintuitive Behavior of Social Systems," *Technological Forecasting and Social Change*, vol. 3, pp. 1–22, Jan. 1971.

[23] P. Schmaus, N. Batti, A. Bauer, J. Beck, T. Chupin, E. Den Exter, N. Grabner, A. Köpken, F. Lay, M. Sewtz, D. Leidner, T. Krüger, and N. Y. Lii, "Toward Multi User Knowledge Driven Teleoperation of a Robotic Team with Scalable Autonomy," in *2023 IEEE Int. Conf. Syst. Man Cybern. SMC*. Honolulu, Oahu, HI, USA: IEEE, Oct. 2023, pp. 867–873.

[24] M. Panzirsch, H. Singh, X. Wu, M. Iskandar, A. Koepken, R. Luz, N. Batti, F. S. Lay, A. N. Manaparampil, L. Mayershofer, X. Luo, R. Burger, S. Bustamante-Gomez, J. Butterfass, E. Den Exter, W. Friedl,

T. Gumpert, P. Pavelski, G. Quere, F. Schmidt, A. Albu-Schaeffer, A. S. Bauer, D. Leidner, P. Schmaus, A. Hagengruber, T. Krueger, J. Vogel, and N. Y. Lii, “Virtual Elasto-Plastic Robot Compliance to Active Environments,” *Sci. Robot.*, vol. 10, no. 99, p. eadq1703, Feb. 2025.

[25] A. Köpken, N. Batti, A. S. Bauer, J. ButterfaB, T. Ehlert, W. Friedl, T. Gumpert, F. S. Lay, X. Luo, A. N. Manaparampil, L. Mayershofer, A. Raffin, F. Schmidt, D. Seidel, E. den Exter, R. Luz, A. Schmidt, P. Schmaus, D. Leidner, T. Krueger, and N. Y. Lii, “Toward Robust Task Execution through Telerobotic Failure Recovery in Space Operations,” in *2025 IEEE Aerosp. Conf.*, Mar. 2025, pp. 1–13.

[26] N. Batti, L. Mayershofer, A. Köpken, A. S. Bauer, F. S. Lay, T. Ehlert, T. Gumpert, X. Luo, A. N. Manaparampil, A. Raffin, D. Seidel, E. Den Exter, R. Luz, A. Schmidt, P. Schmaus, D. Leidner, T. Krüeger, and N. Y. Lii, “Toward Intuitive Robot-to-Human Error Reporting to Enhance User Awareness in Space (Tele)Operation,” in *2025 IEEE Aerosp. Conf.*, Mar. 2025, pp. 1–13.

[27] A. S. Bauer, A. Köpken, N. Batti, J. Butterfaß, T. Ehlert, W. Friedl, T. Gumpert, F. S. Lay, X. Luo, A. N. Manaparampil, L. Mayershofer, A. Raffin, F. Schmidt, D. Seidel, E. Den Exter, R. Luz, A. Schmidt, P. Schmaus, D. Leidner, T. Krüger, and N. Y. Lii, “System Architecture and Design Considerations for the Humanoid Robot Rollin’ Justin in Context of the Surface Avatar Mission,” in *2025 IEEE Aerosp. Conf.* Big Sky, MT, USA: IEEE, Mar. 2025, pp. 1–14.

[28] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schaffer, and G. Hirzinger, “Rollin’ Justin - Mobile Platform with Variable Base,” in *Proc. 2009 IEEE Int. Conf. Robot. Autom. ICRA*. Kobe, Japan: IEEE, May 2009, pp. 1597–1598.

[29] T. Krueger, E. Ferreira, A. Gherghescu, L. Hann, E. den Exter, F. P. van der Hulst, L. Gerdes, A. Pereira, H. Singh, and M. Panzirsch, “Designing and Testing a Robotic Avatar for Space-to-Ground Teleoperation: The Developers’ Insights,” in *71st Int. Astronaut. Congr. IAC 2020*. International Astronautical Federation, 2020.

[30] D. Seidel, A. Schmidt, X. Luo, A. Raffin, L. Mayershofer, T. Ehlert, D. Calzolari, M. Hermann, T. Gumpert, F. Löffl, E. Den Exter, A. Köpken, R. Luz, A. Bauer, N. Batti, F. Lay, A. Manaparampil, A. Albu-Schäffer, D. Leidner, P. Schmaus, T. Krüger, and N. Y. Lii, “Toward Space Exploration on Legs: ISS-to-Earth Teleoperation Experiments with a Quadruped Robot,” in *2024 IEEE Conf. Telepresence*, Nov. 2024, pp. 10–15.

[31] M. Ghallab, A. Howe, D. Christianson, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins, “PDDL - The Planning Domain Definition Language,” *Fourth Int Artif. Intell. Plan. Syst. AIPS98 Plan. Comm.*, vol. 78, no. 4, pp. 1–27, Aug. 1998.

[32] F. Schmidt and R. Burger, “How we Deal with Software Complexity in Robotics:‘Links and Nodes’ and the ‘Robotkernel’,” in *14th IEEE-RAS Int. Conf. Humanoid Robots Humanoids*, 2014.

[33] C. Ericson, *Real-Time Collision Detection*, 1st ed. Boca Raton: CRC Press, Dec. 2004.

[34] G. E. P. Box and N. R. Draper, *Empirical Model-Building and Response Surfaces*, 7th ed., ser. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics. New York: Wiley, 1987.

BIOGRAPHY

Adrian S. Bauer *received a Bachelor in Mechanical Engineering in 2012, a Bachelor in Cognitive Sciences in 2015, and a Master in “Robotics, Cognition, Intelligence” from the Technical University of Munich (TUM) in 2018. Currently he is pursuing a PhD in robotics at the German Aerospace Center (DLR). His interest is in enabling robotics to generate meaningful symbolic plans in presence of epistemic uncertainty.*

Sant Brinkman *received her MSc in Mechanical Engineering from the Technical University of Delft, the Netherlands. She then worked for two years as a robotics software engineer specializing in ROVs (underwater robotics). Currently, she is a member of the Human-Robot Interaction Lab at ESA.*

Anne Köpken *received a Bachelor in Electrical Engineering from the Technical University of Munich (TUM) in 2019, and a Master in “Robotics, Cognition, Intelligence” from the Technical University of Munich (TUM) in 2021. She spent one semester at the JSK Laboratory at the University of Tokyo in 2019/20. Currently she is pursuing a PhD in robotics at the German Aerospace Center (DLR) in Oberpfaffenhofen near Munich. She is interested in enabling robots to cope with unexpected situations and finding ways to prevent and recover from failures.*

Nesrine Batti *received her ”Diplôme d’Ingénieur” from the National Institute of Applied Science and Technology, University of Carthage, Tunisia in 2022. Since then she is pursuing a PhD in robotics at the German Aerospace Center (DLR) in Oberpfaffenhofen, Germany. She investigates how robots can learn from past errors and transfer this acquired experience to other robots to enhance the collective performance.*

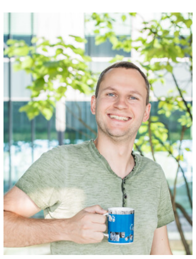
Timo Bachmann *received his B.Sc. degree in Computer Science from the Baden-Wuerttemberg Cooperative State University (DHBW) and graduated with a M.Sc. in “Robotics, Cognition, Intelligence” from the Technical University of Munich (TUM) in 2025. Since 2018 he is working at the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR). His main interests range* *from the kinematic analysis for robotic manipulators to modularized and data-driven software developement focusing on knowledge transferability and explainability.*

Jörg Butterfaß *received his diploma degree from the Technical University of Darmstadt in 1993 and his doctoral degree in 1999 respectively. He joined the German Aerospace Center (DLR) Institute of Robotics and Mechatronics in 1993 where he was involved in the design of various robotic hands. A further working field is the design of robotic sensors for space applications.*

Tristan Ehlert *holds a B.Sc. in General Engineering Science from the Hamburg University of Technology (TUHH) and an M.Sc. in Robotics and Mechatronics from the Technical University of Munich (TUM). Since 2023 he works at the German Aerospace Center (DLR), where he investigates and develops elastic robots for efficient locomotion.*

Emiel den Exter *works as Industrial Design and Human Factors Engineer at the ESA's Human Robot Interaction Laboratory and ATG-Europe VirtualLab. He is specialized in developing novel user interfaces for Robotics and XR Engineering solutions within the Space sector.*

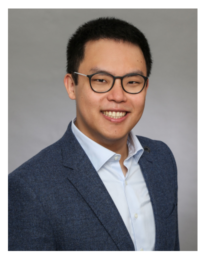
Werner Friedl *received his Dipl.-Ing. (FH) in Mechatronics at the University of Applied Sciences in Munich and started at the German Aerospace Center (DLR) in 2004. In 2006 he developed the torso of DLR's humanoid Justin. In the DLR Hand-Arm-project he developed the forearm of the AWIWI hand and AWIWI II. Since 2015 he is responsible for the mechanical hand development at DLR. His main research focus includes variable stiffness actuation, tendon driven hands and grasping.*



Thomas Gumpert *received his B.Eng. in Mechatronics 2012 and in 2018 his M.Sc. in Applied Research on Mechatronic Systems from the University of Applied Sciences Augsburg. He joined the German Aerospace Center (DLR) Institute of Robotics and Mechatronics in 2008 with focus on electrical drives. Since 2015 he is heading the Drive Technology Lab. As part of different teams he was involved in the development of the lightweight robot SARA and space qualified robotic arm CAESAR as well as the space qualified force feedback joystick used in Kontur-2. Currently he is responsible for the robotic hardware of the humanoid robots Rollin' and Agile Justin, and he coordinates the development of DLR's quadruped robot Bert with his main focus on actuator and sensor electronics.*

Philipp G. Knestel *received a B.Sc. in Information Systems from the Technical University of Munich (TUM) and an M.Sc. in "Robotics, Cognition, Intelligence" from the Technical University of Munich (TUM). Since 2024, he is pursuing a PhD at the Center for Robotics and Mechatronics of the German Aerospace Center (DLR). His research interests lie in robotic intelligent systems, with a focus on foundation models for robotic systems and their applications.*

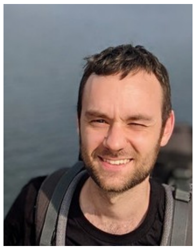
Florian S. Lay *received his B.Sc. degree in Engineering Science in 2018, and his M.Sc. in "Robotics, Cognition, Intelligence" in 2020 both from the Technical University of Munich (TUM). Since 2020 he is pursuing a PhD in robotics at the German Aerospace Center (DLR). His interests range from symbol grounding and emergence for task and motion planning to multi-robot world representations.*



Xiaozhou Luo *received his B.Sc. in Mechanical Engineering in 2019, and his M.Sc. in Aerospace Engineering in 2023, both at the Technical University of Munich (TUM). Following his graduation, he started as a research fellow at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). Pursuing a PhD, his research focuses on SLAM and the creation of semantic 3D maps for navigation purposes in dynamic indoor environments.*

Ajithkumar N. Manaparampil *received his B.E. in Mechanical Engineering from the University of Pune, India and his M.Sc. in "Robotic Systems Engineering" from RWTH Aachen University. He is a Research Associate at the Center for Robotics and Mechatronics of the German Aerospace Center (DLR) since November 2023. His research interests include teleoperation with a focus on assistance in manipulation tasks.*

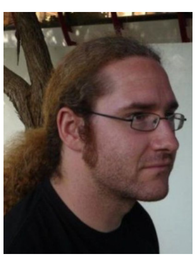
Luisa Mayershofer *received a B.Eng. in Electrical Engineering from the University of Applied Sciences in Augsburg and a M.Sc. in "Robotics, Cognition, Intelligence" from the Technical University of Munich (TUM). Since November 2023, she is pursuing a PhD at the Center for Robotics and Mechatronics of the German Aerospace Center (DLR). Her research interests lie in Human-Robot-Interaction, with a focus on teleoperation and task model estimation, as well as multi-modal user interfaces.*



Antonin Raffin *is a research engineer at the German Aerospace Center (DLR) who specializes in reinforcement learning (RL). He is the lead developer of Stable-Baselines3 (SB3), an open-source library that implements Deep RL algorithms. His main focus is on learning controllers directly on real robots and improving the reproducibility of RL.*

Anne Reichert *is a computer vision researcher in the department of Perception and Cognition at the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR). Her research area is semantic aware robotic manipulation with special interest in 6D pose tracking of (articulated) objects based on visual input and kinematic considerations. Anne received her Master's degree in "Robotics, Cognition, Intelligence" from the Technical University of Munich (TUM) and a Bachelor's degree in Computer Science from Baden-Wuerttemberg Cooperative State University (DHBW).*

Annika Schmidt *holds a B.S. degree in Biomimikry and received her M.S. in Mechanical Engineering from the Technical University of Delft. Since then she has been working jointly at the Technical University of Munich (TUM) and the German Aerospace Center (DLR) to obtain her PhD in bio-inspired control mechanisms for robots. In this, she combines Neuroscience elements with human user studies and control theory for soft robots.*



Florian Schmidt *received his Master of Science degree from the Munich University of Applied Sciences in 2007 (computer graphics and digital image processing). Since then he is with the German Aerospace Center's Institute of Robotics and Mechatronics (DLR-RM). In his master thesis he developed a planning system to solve the rubik's cube with the humanoid robot Justin. His research interests include task and motion planning, real-time capable software architectures, and novel robot programming interfaces.*

Lioba Schürmann *received a B.Sc. in Informatics from the Technical University of Munich (TUM) and a M.Sc. in Neural Systems and Computation from ETH Zurich and the University of Zurich. She joined the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR) in 2024 to pursue a PhD in cognitive robotics.*

Daniel Seidel *received his M.Sc. degree in "Intelligent Systems" from Bielefeld University, Germany, in 2014. Afterwards he joined the Chair for Sensor Based Robotic Systems and Intelligent Assistance Systems of Prof. Albu-Schäffer at the Technical University of Munich (TUM). Furthermore, he is an on-going researcher at the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR) where he is involved in the development of the DLR quadruped robot Bert. His main interests lie in motion generation and planning for elastically actuated legged robots and machine learning on hardware systems.*

Nicole Roshardt *received her B.Sc. in Computer Science in 2020 and her M.Sc. in Computer Science in 2023, both at ETH Zürich. Her master's thesis, conducted in cooperation with the European Space Agency (ESA), focused on routing scalability in space delay- and disruption-tolerant networks. She is currently a Young Graduate Trainee (YGT) at (ESA), working in development of data management infrastructure for the International Space Station (ISS), and supporting in the Human Robot Interaction Laboratory.*

Rute Luz *received her B.Sc. and M.Sc. degrees in aerospace engineering from Instituto Superior Técnico, Lisbon, Portugal, where she is currently pursuing her PhD. Since 2018 she is a researcher at the Institute for Systems and Robotics, and at the Interactive Technologies Institute. In 2024, she also integrated the team at the Human Robot Interaction Laboratory at the European Space Agency (ESA). Her research interests involve human-robot interaction, haptic interfaces, and robotics and enabling more effective teleoperation.*

Peter Schmaus *received his M.Sc. degree in "Robotics, Cognition, Intelligence" from Technical University of Munich (TUM), Germany, in 2013. He joined the German Aerospace Center (DLR) Institute of Robotics and Mechatronics in 2011 where he was involved in the ISS-to-ground telerobotics projects Kontur-2, METERON SUPVIS Justin, and became Co-Investigator of the Surface Avatar experiment suite. His main interests lie in Shared Autonomy and effective Human-Robot Interaction.*

Daniel Leidner *received his diploma degree in communications engineering in 2010, and his M.Sc. degree in information technology in 2011 with distinction from the Mannheim University of Applied Sciences, Germany. In 2017 he received the PhD degree in artificial intelligence from the University of Bremen, Germany. His dissertation was honored with the Georges Giralt PhD Award as well as the Helmholtz Doctoral Prize. He is a Professor at the Institute of Artificial Intelligence at the University of Bremen and leads the department of Autonomy and Teleoperation at the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Weßling, Germany. As Co-Investigator of the Surface Avatar experiments, he investigates artificial intelligence in the context of astronaut-robot collaboration.*

Thomas Krüger *is head of the Human Robot Interaction Laboratory at the European Space Agency (ESA). Involved with ESA's haptic experiments since the beginning, Thomas headed the technology development for the ANALOG-1 experiments and is Co-Investigator on Surface Avatar. He received his PhD in Lab Automation from the University of Rostock and worked on wind turbines before joining ESA.*

Neal Y. Lii *is the domain head of Space Robotic Assistance, and the co-founding head of the Modular Dexterous (Modex) Robotics Laboratory at the German Aerospace Center (DLR). Neal received his BS, MS, and PhD degrees from Purdue University, Stanford University, and University of Cambridge, respectively. He has served as the principal investigator of the ISS-to-Earth telerobotic experiments, Surface Avatar, and METERON SUPVIS Justin. Neal is primarily interested in the use of telerobotics in both space and terrestrial applications.*