Uncertainty-Aware Task Parameterization for Telerobotic Space Missions

Anne Köpken[1], Adrian S. Bauer[1], Nesrine Batti[1],
Timo Bachmann[1], Sant Brinkman[2], 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], Nicole Roshardt[2], Annika Schmidt[1],
Florian Schmidt[1], Daniel Seidel[1], Lioba Schürmann[1],
Rute Luz[2], Peter Schmaus[1], Daniel Leidner[1,3], Thomas Krüger[2], Neal Y. Lii[1]

***Abstract*— Robots are crucial for exploring distant celestial bodies. The Surface Avatar ISS-to-Earth Telerobotic Technology Demonstration Mission investigates how to command a heterogeneous team of robots from orbit using scalable autonomy. In this experiment series astronauts aboard the International Space Station (ISS) command a team of robots located on Earth. One challenge identified in these experiments is that the robot's decision-making process is affected by environmental uncertainty especially for the robot's pose estimation and navigation. Traditional planning algorithms assume perfect knowledge of the robot's surroundings, neglecting the role of imperfect sensing. This paper addresses this limitation by developing an uncertainty-aware planning method and demonstrating its application to address perception inaccuracies in navigation, paving the way for its broader adoption in other domains. The proposed uncertainty aware planning framework is tested for localization and navigation by the humanoid robot Rollin' Justin during a space-to-ground telerobotic experiment as part of the Surface Avatar mission.**

I. INTRODUCTION

Robots are essential for the exploration of distant celestial bodies. Due to limitations in communication, such as limited bandwidth and time delays, commanding the robots from Earth may be challenging. The DLR-ESA Surface Avatar ISS-to-Earth Telerobotic Technology Demonstration Mission investigates how to command a heterogeneous team of robots on a distant celestial body from its orbit. To simulate this situation, astronauts aboard the ISS command a team of robots located in our lab at the German Aerospace Center (DLR), Oberpfaffenhofen Germany. The astronauts can use different input methods ranging from immersive direct teleoperation with force reflection to task level commanding of the robots similar to coworkers. Fig. 1 shows NASA astronaut Jonny Kim aboard the ISS commanding our robotic team on Earth with the aforementioned user interface. The developed technology can potentially be used to command robots on Moon from the Gateway, a future orbiter around the Moon [1].

One difficulty for autonomous robots is the decision-making process on how to interact with the environment to

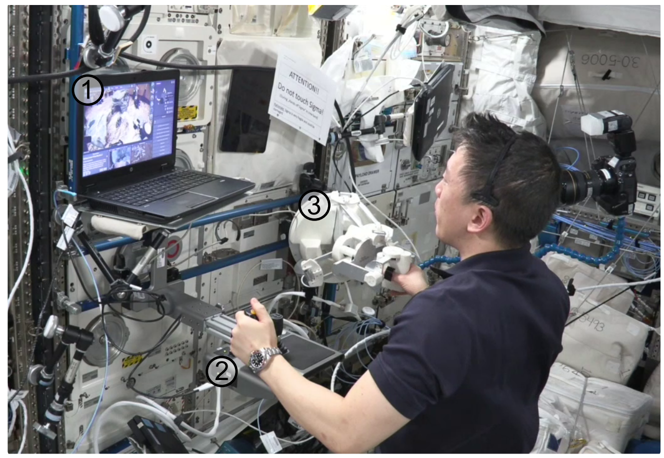


Fig. 1: NASA astronaut Jonny Kim performing a sample collection task, by operating a heterogeneous team of robots from the ISS. He is using a Graphical User Interface (GUI) (1), a joystick (2) and a 7 Degrees-of-Freedom input device with force reflection (3) to command the robots. Image credit: ESA/NASA

reach certain goals. Planning algorithms typically assume perfect knowledge of the robot's surroundings. However, a robot's understanding of its environment is inherently imperfect and uncertain. This uncertainty depends highly on the methods used to infer the robot's belief state. For instance, depending on various factors, localization accuracy to one specific object might be more accurate with the use of AprilTags [2] in comparison to camera-based Simultaneous Localization and Mapping (SLAM) [3] techniques, which in turn usually outperform wheel odometry. The accuracy also depends on the robot's surroundings, if a satellite-based navigation system is available it might be very accurate on the open field, but not usable inside a cave. It's essential for informed decision-making to account for these uncertainties and plan accordingly, as the tolerable level of uncertainty may vary depending on the task at hand.

Depending on the uncertainty of the robot's belief state, it may be necessary to restrict certain actions on a symbolic level and modify the geometric plan accordingly. For example, while a rough localization might be sufficient for driving with a safe distance from obstacles, a more precise

[1] German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany. {firstname.lastname}@dlr.de
[2] European Space Agency (ESA), Human Robotics Interaction Lab (HRI), Keplerlaan 1, 2201 AZ Noordwijk, Netherlands
[3] University Bremen, Institute for Artificial Intelligence, Am Fallturm 1, 28359 Bremen, Germany.

 DOI: 10.1109/iSpaRo66239.2025.11436623

localization is required for direct interaction with objects and movements near them.

One challenge identified in the Surface Avatar mission, is the different accuracy of localization via SLAM vs. AprilTags. While we incorporate uncertainties into task and motion planning with a focus on solving the problem of inaccuracies in localization and navigation, we generalize our approach to allow for application to other domains in the future.

Our contributions are (i) the development of an uncertainty-aware task and motion planning method, which selects and parameterizes actions based on uncertainties, and (ii) the application of the proposed method for perception inaccuracies in the context of navigation in the Surface Avatar missions. The remainder of this paper has the following structure: we begin with an overview of related work in section II, followed by an introduction to the Surface Avatar project in section III. After that we introduce our method of uncertainty-aware task and motion planning in section IV which is then demonstrated in the final Surface Avatar experiment session in section V. Section VI discusses our work and section VII concludes this paper.

II. Related Work

Given the large search space in a continuous domain, robot action planning can quickly grow infeasible. One approach to handle this is to use integrated task and motion planning (TAMP) [4]. TAMP splits the planning problem into two domains: The task domain and the motion planning domain. In the discrete task domain the planner chains together a series of actions that lead to a defined symbolic goal state. The motion planning then grounds this semantic plan with a trajectory that is executable by the robot. While classical TAMP has proven effective for generating motions to reach a symbolic goal states, it does not account for uncertainty. Common TAMP approaches assume perfect knowledge of the state, but in reality, symbolic predicates may be more accurately described as having a certainty level under which they are considered True or False.

To handle uncertainties in the environment, there are several methods that extend TAMP with probabilistic models. Shah et al. [5] develop TAMP plans for stochastic environments using probabilistic planning. Each generated plan functions as a policy with various contingencies for different outcomes at runtime. Similarly, Curtis et al. [6] propose an extension of TAMP called TAMPURA, which incorporates Uncertainty and Risk Awareness (URA). Bauer et el. [7] predict action success probabilities. While these approaches offer more advanced planning in the environment, they require a reliable model of probability distributions and are computationally expensive. It can be hard or even impossible to reliably model the required probability distributions and computational power is limited in space applications. Thus, we propose an approach which is less computationally expensive and does not require the modeling of entire probability distributions.

In the field of space robotics uncertainties are important to be considered, especially as the safety of robot operation has highest priority as failures can be costly or even irreversible. The Mars Rover Perseverance employs odometry and AutoNav, an autonomous navigation software for a significant portion of its navigation [8]. For long autonomous drives, human rover planners designate safe (Keep in) and dangerous (Keep out) areas by marking boundaries on orbital maps. However, as the rover navigates, its position uncertainty increases. To maintain safety, the onboard ”Keep in” (and ”Keep out”) zones adjust based on current uncertainty levels. This increased uncertainty can potentially cause drives to end prematurely if the expanding ”Keep out” zones block narrow passages [9] requiring human intervention. Our approach of modeling the positional uncertainty is similar to the approach used on Perseverance. However, our focus is on incorporating uncertainties into the robot's TAMP system and actively planning perception steps if necessary to reduce the uncertainty, while the Mars rover asks ground operators for support.

III. Surface Avatar

The proposed method is developed for and demonstrated in the Surface Avatar experiment series. Surface Avatar builds upon the METERON SUPVIS Justin [10] and Analog-1 [11], [12] telerobotic missions. In METERON SUPVIS Justin astronauts aboard the ISS were commanding the humanoid robot Rollin' Justin located in an analog Mars environment [13] at DLR in Germany. In multiple experiments the astronauts used supervised autonomy to command the robot similar to a coworker by specifying a desired goal state and the robot planned and executed a task sequence that reached the commanded state autonomously. The robot employed TAMP in the form of Action Templates (ATs) to generate autonomous action sequences [14]. The user interface showed augmented reality (AR) overlays on top of the robot's first person camera view. The overlays could be selected to show the available commands for an object but also to verify the robot's localization with respect to the object, by comparing its overlay position to the real position. Analog-1 added direct commanding via joystick and a 7 Degrees-of-Freedom (DoF) force reflection input device. In this experiment series the Interact rover, stationed in a simlunar test environment in the Netherlands, was commanded from the ISS.

A. Overview

Surface Avatar advances from commanding a single robot to commanding a team of heterogeneous robots. This allows for more efficient usage of the astronaut's time as multiple robots can be commanded in parallel. During the experiments, the astronauts are tasked to perform maintenance, exploration, and scientific protocols that require them to command the robots collaboratively.

Fig. 2 shows the experimental area at DLR in Oberpfaffenhofen Germany with our four robotic assets: (1) Rollin' Justin [15], a humanoid robot with dexterous manipulation

Fig. 2: The astronaut has the option to command any one of four robotic assets: Rollin' Justin (1), Spot (2), Bert (3) or the Interact Rover (4). The environment includes a handover station (H) and a lander (L).

Fig. 3: The GUI displays information about the robots, such as the robots' known objects and their positions in the form of augmented reality overlays (1) and the robots' positions in the map (2). It can also be used to give task level commands (3) to the robots.

capabilities, (2) Spot[1] a quadruped robot used for exploration, (3) Bert [16] a small quadruped especially useful to explore tight spaces and (4) Interact [17] a rover equipped with two arms, good at traversing long distances and carrying larger payloads.

Fig. 1 shows NASA astronaut Jonny Kim on board of the ISS commanding the robotic team. The user interface consists of a GUI (1), a 3-DoF joystick (2) and a 7-DoF input device (sigma.7)[2] with force reflection (3). For direct teleoperation, the astronaut can use the joystick to move the robot's base or head and the sigma.7 to move the arm while receiving force feedback from the robot for increased situational awareness. Teleoperation with force feedback and time delay is possible through the time domain passivity approach (TDPA) [18].

Fig. 3 shows the GUI, that is used for task level commanding of the robotic team. In this screenshot Justin is currently selected and the main part of the GUI shows its camera view. The camera view is augmented with overlays (in blue) of the objects known by the robot. The blue overlay of the handover station (1) matches the object position roughly, but it is visible, that the believed position does not match the actual one perfectly. The robots' positions in the map view (2) are also updated according to their belief state. Furthermore, depending on the belief state, the available commands that are shown on the right (3) are updated dynamically such that only commands which are currently safe for the robot to execute are available. A more detailed description of the GUI can be found in [19].

B. Localization and Navigation in METERON SUPVIS Justin and Surface Avatar

Despite the robot's ability to generate long task sequences in METERON SUPVIS Justin, the robot was never allowed to plan past a localization step, ensuring that the human supervisor could verify the localization before the robot interacted with an object. This part is crucial, as failure should be prevented at any cost in space context. To localize itself in its environment the robot relied on the detection of AprilTags [2] as reference points at predefined positions. When the robot was localized with one landmark it was allowed to navigate to the close proximity of another landmark, relying on the assumption of a perfect detection and a perfect model for both the marker positioning and navigation execution.

However, in the Surface Avatar project, we encounter new challenges when astronauts command teams of multiple robots. With increasing experimental area size it becomes more challenging to keep an accurate world model. Additionally, longer task sequences are desired to increase autonomy times of the robots, freeing the astronauts from having to switch between different robots in the GUI at a high rate. Furthermore, when being moved manually by the joystick, the robots need to be able to constantly update their position. This is why we integrate SLAM [20] as an additional localization method besides relying on AprilTags. The main advantage is that SLAM provides continuous localization throughout the experimental area, even if no marker is in the robot's sight. When the astronaut moves the robot manually, the robot's position in the map and the overlays on the camera view are continuously updated on the GUI (see fig. 3 (1) and (2)). However, our SLAM-based localization introduces greater uncertainty compared to traditional marker-based approaches.

IV. Uncertainty-Aware Task and Motion Planning

Generating plans for the robot Rollin' Justin has been accomplished in METERON SUPVIS Justin [10] and Surface Avatar [21] through the use of Action Templates (ATs) [14]. ATs build up on the assumption of a perfectly known world state and perfect knowledge of the effects of action execution. Thus, the evolution of the world when executing actions can

[1] `https://bostondynamics.com/products/spot/`, last accessed 17.10.2025

[2] `https://www.forcedimension.com/products/sigma`, last accessed 17.10.2025

be simulated without worrying about uncertainties. In the Surface Avatar experiment series this assumption is being challenged calling for a relaxation of the assumption of perfect world knowledge.

The integration of SLAM as a second localization method, besides localization with AprilTags, poses two challenges, namely how to update the robot's belief state with conflicting information from different localization methods and how to take the different localization accuracies into account while planning.

In the following we describe how we solved these two challenges for the localization and navigation in the Surface Avatar experiment series and how this can be generalized for other tasks in the future.

A. Uncertainty-Based Belief State Updates

To address the first challenge – dealing with conflicting information – we developed a mechanism that prioritizes the most certain information to update the robot's belief state. The process works in a "winner-takes-it-all" fashion which means that the robot's belief state is being updated with the most certain state estimate while the other estimate is being discarded.

For localization this means that the robot's belief state remains constantly updated with the current SLAM position if it is not localized to any landmarks via AprilTags. However, once a landmark localization is available and the robot is in standstill, it updates its state with the landmark localization and pauses updating its position through SLAM. As soon as the robot detects a movement of its platform, its localization uncertainty increases. Once the uncertainty surpasses the SLAM uncertainty, the robot's position is again updated based on SLAM data as this becomes the most certain state estimate.

B. Uncertainty-Aware Task Planning

To address the second challenge of integrating localization uncertainties into planning, it is necessary to represent uncertainties in the task planning domain. For this, we introduce an additional symbolic state variable to the robot's state.

A classical task planning problem $\mathcal{P}$ is defined as $\mathcal{P} = \{\mathbf{s}_0, \mathcal{S}_*, \mathcal{T}\}$, consisting of the initial state $\mathbf{s}_0 \in \mathcal{S}$ where $\mathcal{S}$ is the set of states or state space, the set of goal states $\mathcal{S}_* \subset \mathcal{S}$ that define the desired end of the task, and the state transitions $\mathcal{T}$ [4]. A transition moving the system from state $\mathbf{s}$ to another state $\mathbf{s}'$ is defined as a pair $t = \langle \mathbf{s}, \mathbf{s}' \rangle \in \mathcal{T}$ with the set of all possible transitions between states being $\mathcal{T} \subset \mathcal{S} \times \mathcal{S}$.

A single state representation $\mathbf{s}$ can be decomposed into M distinct state variables, $v \in \mathcal{V}$, each having a finite domain $\mathcal{X}_v$. An individual state is represented as an assignment of values $x_v \in \mathcal{X}_v$ to each variable v. This structure results in a state space $\mathcal{S} = \mathcal{X}_1 \times ... \times \mathcal{X}_M$, where the size of the state space grows exponentially with the number of variables.

A common way to represent transitions t is by using preconditions and effects. This approach proves particularly useful when a transition affects only a small subset of the state variables. The set of effects $eff : \{v_1 \leftarrow c_{e,1}, ..., v_k \leftarrow c_{e,k}\}$ with $c_{e,i} \in \mathcal{X}_i$ modifies the specified state variables $(v_1, ..., v_k)$ to their respective values $(c_{e,1}, ..., c_{e,k})$. Preconditions define testable logic conditions based on the space of state variables $\mathcal{V}$ that are used to test whether a transition t can be executed in state s. Different languages can be used to describe these planning problems. However, the most popular formalism is the Planning Domain Definition Language (PDDL)[22].

While other approaches require changes to the whole planning pipeline to incorporate uncertainties into task planning, we propose to add uncertainties through additional state variables. Each additional state variable $v_{M+i} = \sigma_{v_i}$ represents the uncertainty σ_{v_i} of state variable v_i. This allows to use the well-tested planning pipeline from previous ISS-to-Earth missions, keeping the changes minimal and with that minimizing the risk of failure due to the extension.

Extending the state $\mathbf{s}$ by the corresponding uncertainties $\mathbf{s}_\sigma$ results in the new state $\hat{\mathbf{s}}$:

$$\hat{\mathbf{s}} = \begin{pmatrix} \mathbf{s} \\ \mathbf{s}_\sigma \end{pmatrix} = \begin{pmatrix} v_1 \\ ... \\ v_M \\ \sigma_{v_1} \\ ... \\ \sigma_{v_M} \end{pmatrix} \tag{1}$$

The new state variable v_{M+i} can take any of the values $(x_{M+i,1}, ...x_{M+i,N})$ with $x_{M+i,j} \in \mathcal{X}_{M+i}$ describing N different certainty intervals for the state variable v_i.

As there is no order to the uncertainties σ_{v_i}, the developer of the domain has to put manual effort into ensuring that preconditions and effects of actions in the domain match the uncertainties. Let us assume the developer specifies the uncertainties $\mathcal{X}_{M+i} = \{\textit{low}, \textit{medium}, \textit{high}\}$ for state variable v_i. An action that can be executed with *high* uncertainty can also be executed with *medium* or *low* uncertainty. Thus, in the preconditions of this action, the developer has to check whether the uncertainty is *low* OR *medium* OR *high*.

This approach is impractical if many different uncertainty intervals are required due to the high manual effort linked to it. One approach to solve this problem of growing complexity is by assigning numeric values to the uncertainty state variable v_{M+i}, transforming the domain $\mathcal{X}_{M+i}$ from a discrete domain with predefined values into a continuous one. This can for example be implemented with numeric fluents in PDDL. Numeric fluents are an extension to classical planning allowing for real-valued state variables. In addition to assigning numeric values to variables, it also supports simple mathematical operations like addition, subtraction and multiplication. Furthermore preconditions support comparisons using inequalities. Each action can then use a certain upper bound for the uncertainty in the precondition and set an uncertainty in the effect. With this extension the developer only needs to specify one inequality in the precondition and does not need to manually define the possible values of the domain $\mathcal{X}_{M+i}$. Revisiting the example from above, the developer could specify the maximum uncertainty for an action as e.g. $v_{M+1} \leq c_{high}$ in the precondition of the action. If the uncertainties are defined as $c_{low} < c_{medium} < c_{high}$ it entails that the action can also be executed when $v_{M+1} \leq c_{low}$ or $v_{M+1} \leq c_{medium}$.

To solve the challenges with uncertainties in perception and navigation for the Surface Avatar mission, we add a state variable describing *how accurate* the robot is localized in addition to *whether* it is localized. To reduce the complexity we consider the uncertainty to be a scalar value similar to the approach on Perseverance [9]. The uncertainty σ_l describes the radius of the circle around the estimated position of the robot in which we expect it to be with a certain confidence.

The uncertainty of the robot's current position stems from two factors: the perception uncertainty and the uncertainty of the robot's knowledge of where objects are located in the map. As both factors potentially impede the robots ability to safely traverse in close vicinity of objects in the environment, we combine both factors in one state variable. The new state variable can be assigned either a value that represents that the robot is localized via SLAM, or the name of the object that contains the AprilTag the robot is localized to. The planner interprets the information about the object, with respect to which the current localization happened, as an accuracy of the localization. If the object the robot plans to navigate to closely is the one it is localized to, the localization accuracy is assumed to be good, otherwise it is assumed to be on par with the accuracy of SLAM. Thus, the uncertainty of the localization σ_l can be described as

$$\sigma_l = \begin{cases} \sigma_{april} & if\ o_l == o_a \\ \sigma_{SLAM} & else \end{cases} \tag{2}$$

where σ_{SLAM} is the localization uncertainty of SLAM, σ_{april} is the localization uncertainty of localizing with April-Tags, o_l is the object the robot is localized to and o_a is the object that the robot plans to navigate to. Using the new state, we are able to define actions that define upper bounds on the maximally allowed uncertainty in their preconditions.

While the focus of this section is on the usage of uncertainties in localization, the same approach can be applied to other use cases. For example manipulation tasks might require different accuracies depending on how delicate they are. If the robot pours granulate into a huge container it does not need to know the location of itself and the container as well as if it is tasked to pour into a narrow opening. In conjunction with adding uncertainty information of object locations to the world state, an action precondition can define limits on how accurate the positions need to be known in order to execute a pouring action.

Furthermore, the benefit of an added uncertainty predicate is not limited to positions, but also applies to symbolic states. Staying in the example of pouring, it is crucial to know whether a container is *filled* or *empty*. Therefore, also the information about how accurate this knowledge is, is important. If the robot plans to pour from container A into container B, the information that container B is *empty* needs to be accurate as it not being empty might lead it to overflow. Conversely, inaccuracies regarding the fill level of container A do not pose significant risks, making it less critical to have precise information in this case.

C. Uncertainty-Based Motion Planning Parameterization

Beyond task planning we also incorporate uncertainties into our motion planning process. To mitigate potential collisions, we parameterize motion planning with an obstacle avoidance margin based on the localization uncertainty. This allows the robot to maintain a safe distance from obstacles in its environment, taking into account the limitations of its localization accuracy. By maintaining this buffer zone, the robot avoids potentially hazardous situations that may arise due to errors or inaccuracies in its localization capabilities, thereby enhancing overall safety.

So far, our implementation of this strategy is being applied primarily to navigation tasks, where the robot navigates with a safe distance from obstacles. However, extending this approach to motion planning for the entire body is straightforward. One common use case in Surface Avatar experiments involves instructing the robot to reach a specific predefined joint configuration. Currently, we apply a uniform obstacle avoidance margin independent of *how* the robot is localized. However, by adjusting the margin based on the degree of localization uncertainty, more nuanced decision-making is possible.

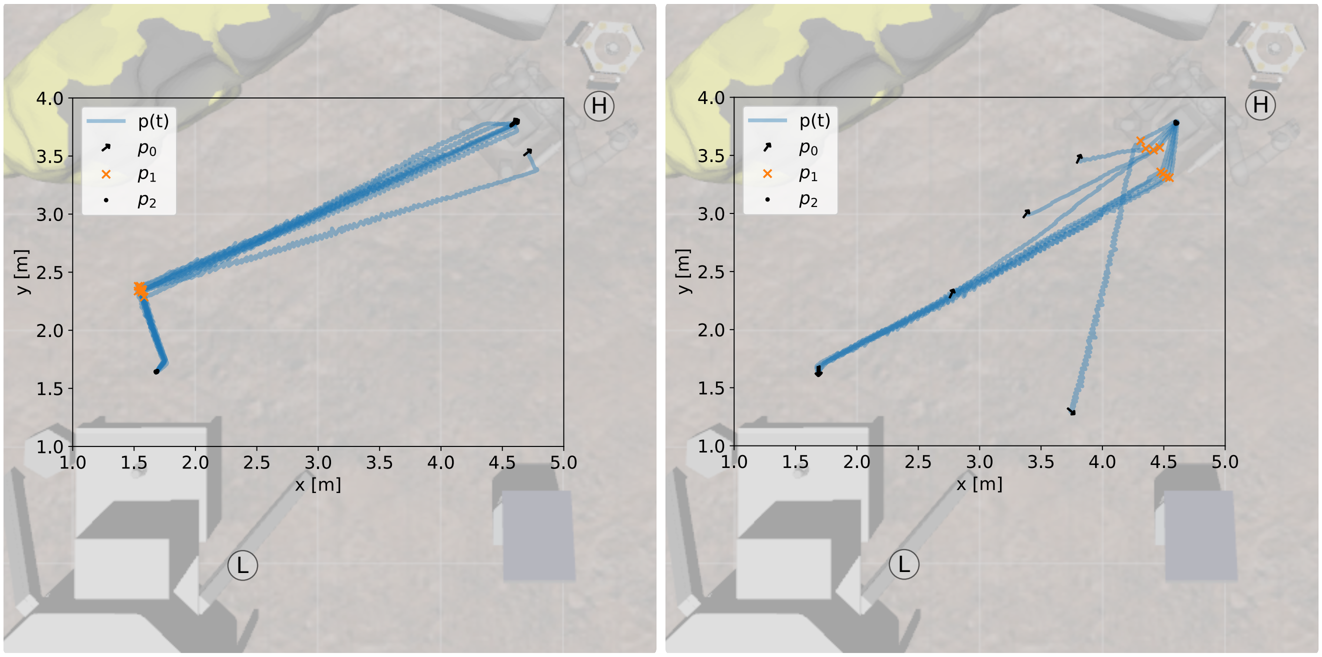
Another use case for the localization uncertainty is to aid the selection of an appropriate grasp. By taking into account the uncertainty of the objects location, the robot may choose between different types of grasps. Specifically, selecting a more stable power grasp when uncertainty is higher or selecting a delicate pinch grasp when confident in its positioning.

Uncertainty is not limited to localization; it also plays a crucial role in decision-making for other aspects of the robot's operation. For instance, when considering acceleration and speed, uncertainty related to the current battery status can significantly impact the robot's choices. Even when the actual battery level is still relatively high, uncertainty about its status can influence the robot to create an energy-conserving motion plan, thereby reducing the risk to run out of battery before reaching the charger.

Moreover, uncertainties in boolean predicates – such as whether a container in the robot's manipulator is empty – can also influence the motion planning. In case of doubt about the container's state, the robot may chose to maintain the container upright during its execution, thereby avoiding potential spillage.

V. On-orbit Demonstration

In July 2025, we demonstrated the integration of uncertainties in TAMP on the Rollin' Justin system during an ISS-to-earth telerobotic experiment session. Fig. 2 shows the experimental area. Among other tasks, one objective for the astronaut was to collect sample containers from the handover station (H) and place them on the lander tray (L) utilizing autonomous actions of Justin (1).



(a)

(b)

Fig. 4: When the robot performs navigation actions starting from the start pose p_0, its initial navigation is inaccurate (p_1) due to initial localization uncertainties and uncertainty in the navigation action. After a more precise localization at (p_1) and a shorter navigation towards (p_2) the second navigation point (p_2) is reached more precisely. (a) shows the recorded positions of eight navigation actions to the lander (L). (b) shows the recorded positions of eight navigation actions to the handover station (H).

To achieve this, the astronaut commanded the robot to pick up a sample container from the handover station (H), navigate from the handover station to the lander (L), place the sample container, return to the handover station, and repeat the process with the next sample container. Due to the uncertainties in the robot's position in the environment and the exact positions of environmental assets, such as handover station or lander, the astronaut could not command the robot to directly navigate from one object to another. Doing so could result in collisions between the robot and the environment.

Without integration of uncertainties, the solution was to manually move the robot in sight of the AprilTag on the target object, localize the robot with respect to the target object, and finally approach it. The integration of uncertainties and intermediate perception steps allow the robot to drive completely autonomously from the lander to the handover station and vice versa.

In order to implement the uncertainties, the predicate *localized_inmap* is used to define *whether* the robot is currently localized in the map while the predicate *localized* describes *how* the robot is currently localized. In the hybrid planning process the uncertainty is empirically determined to be 0.05m when the robot is localized to the AprilTag corresponding to the object it aims to interact with. When the robot is localized by *SLAM* or by AprilTag to any other object the uncertainty is empirically determined to be 0.2m, which combines both, the uncertainty of the localization approach as well as the uncertainty in the alignment of the landmarks. Using this information, the planner chooses one of two navigation poses, either a close pose (p_2) or a far pose (p_1) that were both previously specified by the developers.

With this extension to the existing architecture, the robot is able to navigate towards the vicinity of a landmark with no intermediate human intervention required, even if the robot is localized by SLAM or by AprilTag to another object. We furthermore improve the localization accuracy by centering the expected AprilTag position in the robot's camera image.

The uncertainty in the navigation procedure itself poses an additional challenge. The further the robot navigates, the more it deviates from the planned path at execution time. To account for this uncertainty, the obstacle avoidance margins are increased when the robot plans a navigation to a position that is far away. Due to the dynamic obstacle avoidance margins, the robot is only able to plan to poses that are far enough away from environment objects. Once it reaches the pose further away from the object and re-localizes itself, it is able to navigate closer. If the initial position is in collision with an object due to the selected margins, the robot resolves the collision and continues with the increased margins once sufficient distance to the object is reached.

Fig. 4 visualizes the uncertainty in the navigation action. Fig. 4a shows the recorded position data from eight executions of the *navigate to lander* action. In this action, the robot starts at any point p_0, first navigates to a point further away from the lander (p_1), re-localizes itself with respect to the AprilTag mounted on the lander and then navigates closer (p_2). In these executions the robot always starts from a position (p_0) near the handover station (H) in the upper right corner. The spread at p_1 is 0.105m, while the spread at p_2 is 0.0158m. This shows that the resulting position at the final point (p_2) is reached with higher precision than p_1.

Fig. 4b shows the logged position data from eight executions of the *navigate to handover station* action. The spread of p_1 is 0.394m, while the spread of the second point (p_2) is 0.0177m. The spread of p_1 is higher than the spread of p_1 in fig. 4a. This is likely due to the fact that the robot starts from different initial positions (p_0) that contain more uncertainty, especially in the rotation. A rotational difference in the starting position accumulates to a positional difference with the distance traveled. The second navigation point (p_2) is again more precise.

Without our extension and awareness of uncertainties, the robot would not have had a means to reason that it needed a two-step approach to navigate to an object if it was not well enough localized to it. Thus, the final position of the robot would have shown the spread visible in p_1 in Fig. 4. In the worst case this could have lead to collisions with the objects while approaching them. By considering uncertainties, we were able to compensate for the localization uncertainties and conduct the 2.5 hour experiment session without the robot failing to accomplish any task or colliding with the environment.

In addition to improving navigation tasks, we also used uncertainties on the task level for other tasks. While delicate interactions with objects, such as picking up the sample container from the handover station, are only allowed if the robot is localized to that concrete object, other actions tolerate higher uncertainties. One example for such an action is moving to the idle configuration, a predefined joint configuration, which is also allowed when localized with SLAM.

VI. Discussion

We presented a method to integrate uncertainty information into a TAMP planner and demonstrated the implementation in the Surface Avatar telerobotic experiment. All in all, our implementation allowed the humanoid robot Rollin' Justin to autonomously traverse the experimental area in a safe manner during the final Surface Avatar session.

To achieve this result, we made some simplifications to our approach. Firstly, uncertainties are reduced to a scalar value instead of representing them as a probability distribution. The intuition behind this simplification is that the uncertainty scalar represents an expected upper bound on the delta between believed state and the real state, such that the delta is smaller than the uncertainty value with a predefined probability. This uncertainty can be adapted for the use case at hand, allowing for more conservative or more "aggressive" planning. If the uncertainty is assumed to be higher than it is in reality, the robot is unnecessarily "cautious" and unable to perform tasks that would have been safe. If the uncertainties are assumed to be lower than in reality the robot might generate plans that collide with the environment. A more complex representation of the uncertainty should only be used if necessary, as it makes the planning process more complex, increasing failure risks and computational demand. We foresee to change our model in the future if needed for a specific application. For example for fine manipulation there might be a limitation of using one scalar for the robot's positional uncertainty, as a small rotational offset will lead to a big offset in the robot's manipulator position if the arm is stretched. This would call for using a separate uncertainty for rotational uncertainties and translational uncertainties. In addition it would probably be beneficial to add a more fine-granular differentiation of different uncertainty levels as proposed in section IV.

Secondly, we currently employ a winner-takes-it-all selection of the most accurate information source in case of conflicting information sources. While this simplified version was sufficient for our application, it might make sense to employ a more advanced sensor fusion approach in the future. We expect that this would result in better state estimates, if detailed knowledge of the uncertainties of each approach is available. Therefore, incorporating perception modules, which can estimate the uncertainty for each individual localization instead of having one fixed uncertainty assumption per method, would be beneficial.

Fig. 4 shows that our approach successfully reduced navigation inaccuracy at the end of a long navigation task, compared to the navigation accuracy observed in the intermediate step p1. Without our extension of the pre-existing planning approach, we would expect to see the same variance in positions in p2 as can be seen in p1.

While we showed the application of the proposed method for navigation tasks, the same approach can also be used for other tasks such as manipulation, as described in section IV.

As the presented approach only required a minimal change to the approach that proved to be stable in previous missions, we reduced the risk of failures during the experiment. Furthermore, we could reuse existing task definitions with minor adaptations, reducing the additional effort for considering uncertainties in the tasks.

Previously the astronaut had to reason about the accuracy and decide if certain commands are safe to execute. Given our proposed solution, the astronauts are freed from this responsibility while still being able to overrule the robot's autonomous decisions.

VII. Conclusion and Outlook

We demonstrated how a mission under space conditions can benefit from the integration of uncertainties into TAMP. While we implemented uncertainties for localization and navigation tasks, we proposed a mechanism that can be applied to other predicates and tasks in the future. We foresee that these methods will be needed, as the environments of the robots grow and the tasks become more complex. In the future it would also be interesting to apply our approach to robots in other domains.

In our work, we manually estimated the uncertainties of our perception methods and the required certainties for different tasks. In the future, the uncertainty intervals could be learned from collected data. The required certainty in the precondition of tasks could also be re-used for reactivity at run-time similar to the approach in [23].

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), NASA, ESA, and NASA astronaut Jonny Kim. We thank them for the support through out the experiment planning, preparation, testing, astronaut training, and on-orbit operations.

Funded by the European Union (ERC, RECOVER.ME, 101116620). Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] S. Fuller, E. Lehnhardt, C. Zaid, and K. Halloran, “Gateway program status and overview,” *Journal of Space Safety Engineering*, vol. 9, no. 4, pp. 625–628, Dec. 2022.

[2] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3400–3407.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2017.

[4] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, “Integrated Task and Motion Planning,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 265–293, May 2021.

[5] N. Shah, D. Kala Vasudevan, K. Kumar, P. Kamojjhala, and S. Srivastava, “Anytime Integrated Task and Motion Policies for Stochastic Environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 9285–9291.

[6] A. Curtis, G. Matheos, N. Gothoskar, V. Mansinghka, J. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling, “Partially observable task and motion planning with uncertainty and risk awareness,” *arXiv preprint arXiv:2403.10454*, 2024.

[7] A. S. Bauer, P. Schmaus, F. Stulp, and D. Leidner, “Probabilistic Effect Prediction through Semantic Augmentation and Physical Simulation,” in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation*. Paris, France: IEEE, May 2020, pp. 9278–9284.

[8] V. Verma, M. W. Maimone, D. M. Gaines, R. Francis, T. A. Estlin, S. R. Kuhn, G. R. Rabideau, S. A. Chien, M. M. McHenry, E. J. Graser, A. L. Rankin, and E. R. Thiel, “Autonomous robotics is driving Perseverance rover’s progress on Mars,” *Science Robotics*, vol. 8, no. 80, p. eadi3099, Jul. 2023.

[9] V. Verma, J. Nash, L. Saldyt, Q. Dwight, H. Wang, S. Myint, J. Biesiadecki, M. Maimone, A. Tumbar, A. Ansar, G. Kubiak, and R. Hogg, “Enabling Long & Precise Drives for The Perseverance Mars Rover via Onboard Global Localization,” in *2024 IEEE Aerospace Conference*, Mar. 2024, pp. 1–18.

[10] N. Lii, D. Leidner, P. Birkenkampf, B. Pleintinger, R. Bayer, and T. Krueger, “Toward scalable intuitive telecommand of robots for space deployment with METERON SUPVIS Justin,” Jun. 2017.

[11] T. Krueger, F. van der Hulst, E. Ferreira, K. Wormnes, E. den Exter, A. Gherghescu, L. Gerdes, L. Hann, A. Pereira, N. Y. Lii, and F. Conti, “A newcomer’s guide to the challenges of a complex space-to-ground experiment, with lessons from analog-1,” in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS) 2020*, Oct. 2020.

[12] K. Wormnes, W. Carey, T. Krueger, L. Cencetti, E. den Exter, S. Ennis, E. Ferreira, A. Fortunato, L. Gerdes, L. Hann, C. Lombardi, E. Luzzi, S. Martin, M. Massironi, S. Payler, A. Pereira, A. P. Rossi, R. Pozzobon, F. Sauro, P. Schoonejans, F. P. J. van der Hulst, and J. Grenouilleau, “ANALOG-1 ISS - The first part of an analogue mission to guide ESA’s robotic moon exploration efforts,” in *Global Space Exploration Conference (GLEX 2021)*, Jun. 2021.

[13] N. Y. Lii, D. Leidner, A. Schiele, P. Birkenkampf, R. Bayer, B. Pleintinger, A. Meissner, and A. Balzer, “Simulating an extraterrestrial environment for robotic space exploration: The METERON SUPVIS-Justin telerobotic experiment and the solex proving ground,” 2015.

[14] D. Leidner, C. Borst, and G. Hirzinger, “Things are made for what they are: Solving manipulation tasks by using functional object classes,” in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Nov. 2012, pp. 429–435.

[15] 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 Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2025, pp. 1–14.

[16] D. Lakatos, K. F. Ploeger, F. C. Loeffl, D. Seidel, F. Schmidt, T. Gumpert, F. John, T. Bertram, and A. Albu-Schäffer, “Dynamic locomotion gaits of a compliantly actuated quadruped with SLIP-like articulated legs embodied in the mechanical design,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3908–3915, Jul. 2018.

[17] T. Krueger, E. Ferreira, A. Gherghescu, L. Hann, E. den Exter, F. P. J. van der Hulst, L. Gerdes, A. Pereira, H. Singh, M. Panzirsch, T. Hulin, R. Balachandran, B. Weber, and N. Y. Lii, “Designing and testing a robotic avatar for space-to-ground teleoperation: The developers’ insights,” in *71st International Astronautical Congress, IAC 2020*. International Astronautical Federation, Oct. 2020.

[18] M. Panzirsch, A. Pereira, H. Singh, B. Weber, E. Ferreira, A. Gherghescu, L. Hann, E. den Exter, F. P. J. van der Hulst, L. Gerdes, L. Cencetti, K. Wormnes, J. Grenouilleau, W. Carey, R. Balachandran, T. Hulin, C. Ott, D. Leidner, A. O. Albu-Schäffer, N. Y.-S. Lii, and T. Krüger, “Exploring planet geology through force-feedback telemanipulation from orbit,” *Science Robotics*, vol. 7, no. 65, pp. 1–11, Apr. 2022.

[19] P. Schmaus, A. Bauer, N. Bechtel, M. Denninger, A. Köpken, 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 *2023 IEEE Aerospace Conference*, 2023, pp. 1–10.

[20] M. Sewtz, X. Luo, J. Landgraf, T. Bodenmüller, and R. Triebel, “Robust Approaches for Localization on Multi-camera Systems in Dynamic Environments,” in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, Feb. 2021, pp. 211–215.

[21] 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.

[22] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. Sri, A. Barrett, D. Christianson *et al.*, “Pddl—the planning domain definition language,” *Technical Report, Tech. Rep.*, 1998.

[23] A. Kopken, 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. Kruger, and N. Y. Lii, “Toward Robust Task Execution through Telerobotic Failure Recovery in Space Operations,” in *2025 IEEE Aerospace Conference*, Mar. 2025, pp. 1–13.