AUTONOMOUS COOPERATION OF INTELLIGENT HETEROGENEOUS ROBOTS IN REALISTIC PLANETARY AND LUNAR EXPLORATION SCENARIOS

Tristan Schnell[1]

Tim Wöldecke[1]

David Oberacker[1]

Felix Exner[1]

Lennart Puck[1]

Marvin Grosse Besselmann[1]

Niklas Spielbauer[1]

Arne Roennau[1]

Rüdiger Dillmann[1]*

ABSTRACT

Several hugely successful robotic missions have been performed in planetary and lunar exploration scenarios. While mission goals concerning travelled distance and robot lifetime were greatly exceeded, the utilized degree of autonomy in these systems was chosen to be very low. With the increasing push towards especially the moon in recent years and the associated need for greater long-term numbers of active robots for exploration, prospecting, and other tasks, though, improving their autonomous capabilities becomes more and more relevant. Additionally, with various new rovers and potentially legged robots from different commercial and scientific providers potentially being used together in these medium to long-term missions, efficient control of these heterogeneous multi-robot teams is needed. Existing control approaches used in many terrestrial multi-robot applications, such as in logistics, cannot easily be used in these situations, due to the difficult communication restrictions and the high degree of uncertainty.

We developed a scalable autonomy approach to control teams of arbitrary robots on various levels of autonomy, from fully autonomous missions to assisted teleoperation. In the autonomous mode, robots independently schedule and execute missions using a shared collection of tasks. By communicating the tasks they plan to execute and the associated expected utilities, efficient cooperation between the robots can be performed without the need of a centralized planner. This increases redundancy, allows for autonomous operation when no communication with a centralized mission control is possible and allows for more dynamic behaviors, as robots can easily adjust their plans when their internal or external situation undergoes unexpected changes. By considering their functional health and additional resources such as battery charge, the task allocation in the team can also dynamically adapt to possible damages.

The developed system was, with some changes, tested in three realistic scenarios. As part of the intelliRISK2 project we performed a planetary exploration analogue in the Tabernas desert, using a team of three different walking robots. Additionally, we utilized it in the ESA ESRIC Space Resources Challenge as part of a realistic lunar exploration and resource prospecting scenario, which we were able to win primarily due to our high degree of autonomy. Finally, we tested it in a nuclear power plant emergency scenario.

We will, thus, also present how this system performed in these realistic scenarios, what possible challenges exist in further increasing the technological readiness level of the

*[1] Department of Interactive Diagnosis and Service Systems (IDS), FZI Research Center for Information Technology, Haid-und-Neu-Straße 10–14, 76131 Karlsruhe, Germany.

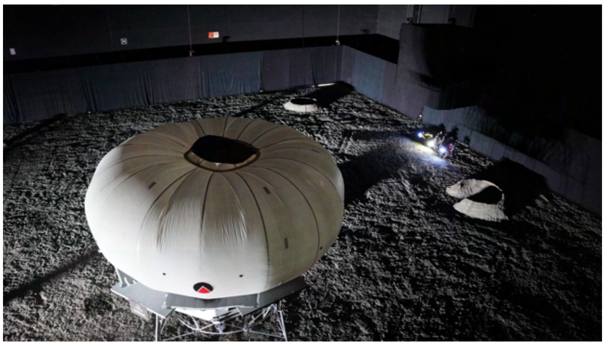

Figure 1. Robots ANYmal, Spot and Husky in a lunar analogue area. At the beginning of the mission, taken as part of the ESA ESRIC Space Resources Challenge, the robots get ready for initial exploration, mapping and resource prospecting.

architecture and what future steps we will take to further improve this control approach.

Key words: robot autonomy; heterogeneous multi-robot teams; analogue missions.

1. INTRODUCTION

The pursuit of a deeper comprehension of the natural world has extended the boundaries of human exploration, including domains like space, the deep sea and contaminated regions. Through the utilization of robots, we can explore areas that are inaccessible, hazardous, or arduous for humans to reach.

Using robots in these situations is difficult, especially as the costs of these missions are very high and failure thus cannot be risked. One way to increase efficiency and redundancy of robotic missions is the usage of robot teams. Operating robots in teams, though, comes with additional challenges: Actions have to be coordinated, data fused and the insights gained by intelligent autonomous agents shared.

On the other hand, once robots are able to efficiently share tasks and cooperate, even more complex interactions with unfamiliar environments are possible. Additionally, a distributed approach can enhance autonomy of individual robots, allowing a higher level of decision making of the systems while guaranteeing redundancy and robustness.

Significant work in multi robot systems has been performed over the last years. Exploration of unknown environments can be defined as a utility maximization problem, defining distance to a goal as cost and unknown area around them as gain, considering active tasks of other robots [1] [2].

Other approaches [3] [4] use a response threshold model, in which every robot individually has a specific thresholds for each available task. These create a swarm-like behavior without any communication between the robots.

For dealing with the problem of communication under suboptimal network conditions, [6] presents a model for automatic data synchronization between robots when communication is possible.

Currently, heterogeneous teams usually have very specific tasks defined and assigned to robots offline with little to no online task distribution. Especially if the team changes throughout the mission, there is little or no possible adaptivity. Full multi robot planning often requires full knowledge of every bit of relevant information at a centralized planner, which is problematic with more and more complex robotic platforms and is worsened in low connectivity scenarios.

We present our experiences using efficient cooperation of heterogeneous robots with a high degree of individual autonomy. It allows complex systems without detailed knowledge of each other to work together. Additionally, it still ensures efficient distribution of tasks by considering both future tasks and best tasks for other robots in planning. Finally, it is adaptive towards changes in the team, such as loss of or damage to a robot.

We accomplish this, using a dynamic, decentralized storage and management of tasks. Each robot determines an optimal task execution sequence individually, using a utility comparison and future robot state estimation. This utility considers the costs calculated by path planning, the robot-specific capabilities and future tasks. Additional relevant information is also considered, such as the battery status and the robot's health.

This paper is structured as follows: In section II, we discuss the architecture, specifically the planning systems and algorithm used. Section III presents our experiences in 3 realistic scenarios. Finally, section IV summarizes the results and planned future work.

2. APPROACH

By employing a multi-stage planning and an utility based task evaluation and approach, we establish a robust, autonomous, and decentralized task allocation system across multiple robots. We store all task decentralized on each robot. Furthermore, we design a communication node which runs on each robot and shares relevant task information. The communication nodes continuously synchronize the databases and resolving conflicts that may arise due to communication failures, such as when two robots have claimed the same task.

We examine a collaborative team consisting of various walking robots with a limited inter-robot communication and no comprehensive understanding of the capabilities and real-time states of other robots within the team. We

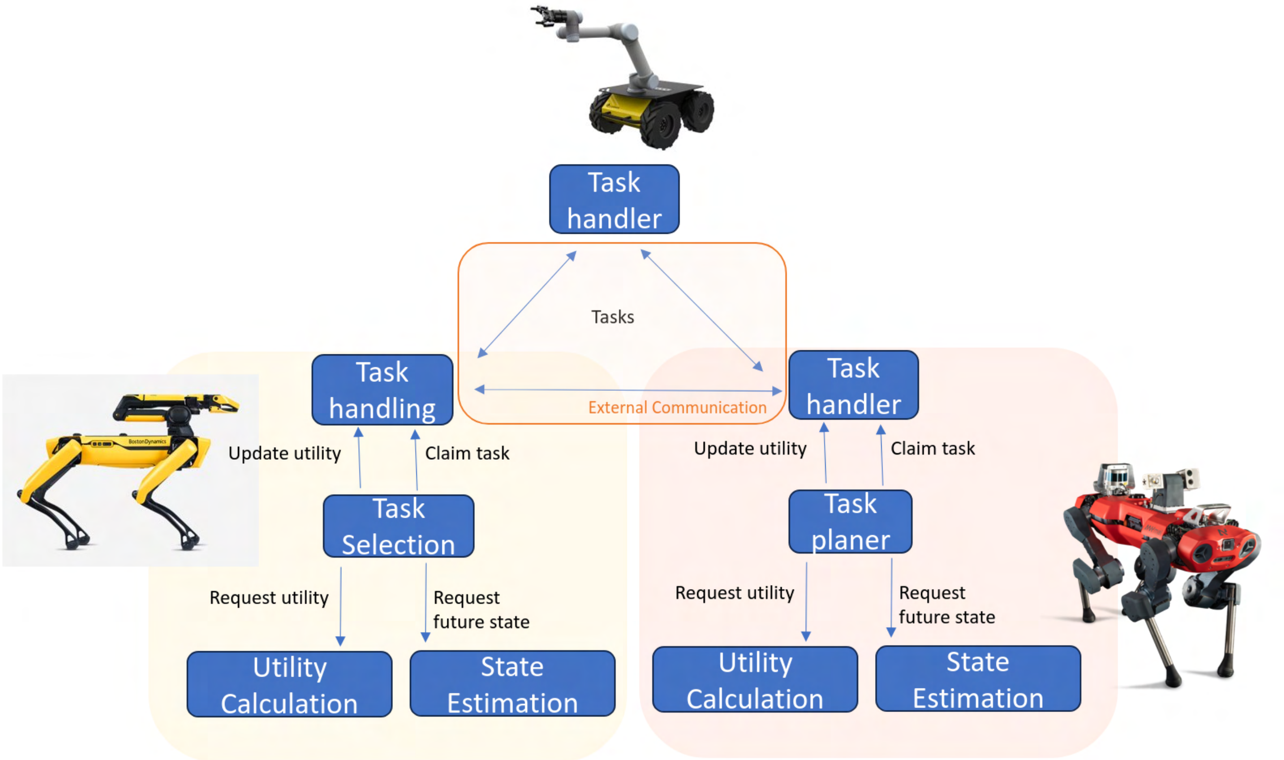

Figure 2. Overview of the system architecture. Each robot has its own task handler instance, which synchronize when possible. Based on this, each robot performs individual utility calculation and state estimation, which in turn is used for task selection and planning.

further consider a path planner that generates paths for given poses and a continuous generation of new tasks.

2.1. Task selection

We are interested in a long-term efficient task allocation without a direct communication. Given the latest received task list A, each robot task planer continually computes the desired order of the most suitable tasks for its instance.

Let U_i be a list of the utilities of the other robots, and $u(i,s)$ be one's own utility regarding task i and robot state s. We define the selection optimality as $o_i = u_i - max(U_i)$. Let O be the list of selection optimality values for each task; we choose the task with the highest selection optimality as the next task.

Calculating the utility can be computationally intensive, as it e.g. includes a detailed path planning. To reduce the calculation, we utilize a rapid estimation of the maximal utility to filter irrelevant tasks. This significantly reduces the selection time. Our selection approach improves the overall exploration quality by enhancing the distribution of exploration areas among the robots.

While the selection of one optimal task may provide a short-term exploration gain it is not necessarily the best long-term task-allocation strategy. To address this subsequent tasks are determined based on an estimation of the state of the robot s after the execution of the selected task. Next, the utilities are recalculated based on this estimated state while accounting for uncertainties. The next task is then chosen using the estimated robot state and the newly computed utilities, as former described. As a result, tasks that are located close to each other are generally more valuable due to their proximity, but they might be less interesting for other robots.

Algorithm 1 Task Planning Algorithm

$Plan \leftarrow$ empty list
$s \leftarrow$ CurrentState
while $PlanningDepth$ not reached **do**
  for i in $Tasks$ **do**
    $u(i,s) \leftarrow \sum w_j u_j$ // *Utility calculation*
  end for
  $Task_{\text{selected}} \leftarrow$ Task with $\max(u_o - \max(U))$
  Insert $Task_{\text{selected}}$ into $Plan$
  $s \leftarrow$ Estimated state after execution of $Task_{\text{selected}}$
end while
for n in $Plan$ **do**
  Communicate(u_n)
end for

2.2. Utility calculation

Given a state of the robot s and a task i, we want to calculate the utility $u_{i,s}$ of executing the task. We define the utility as a sum of weighted utility features

$$ui, j = \sum w_j * u_j$$

for task j. We use the weights to individualize the behavior of the various robots. We define the following utility features:

Type We categorize tasks into different types. For instance, in an exploration task, the robot moves to the task's designated position, whereas in a manipulation task, it has to interact with the environment at the specified position. The prioritization is chosen individually for each robot. Thus, we define the type feature

$$u_{\text{type}} = c(\text{type})$$

in which c is a type-specific constant.

Path length In order for robots to prioritize nearby tasks, the path planner calculates a path from the robot's position to the task. We define the path length feature as the negation of the path length:

$$u_{\text{path}} = -\sum_{n=0}^{m-1} |\overrightarrow{p_{n+1}} - \overrightarrow{p_n}|$$

in which m is the number of positions in the path.

Battery level In continuous exploration missions, robots must regularly return to their base station to recharge their batteries. By defining the battery level feature

$$-u_{\text{battery}} = (b \cos \frac{\pi}{2})^5 * |\overrightarrow{p_{\text{base}}} - \overrightarrow{p_{\text{robot}}}|$$

, in which $b \in [0,1]$ is the battery level and p the position, we ensure that tasks near the base station are prioritized as the battery level decreases. This effectively utilizes the impending return trip. This feature is negligibly small at a high charge level.

Health During the exploration of unknown areas, robots are exposed to high risks, and operational robots should prioritize tasks over those that are damaged. Therefore, the robot's health condition is estimated using data-based techniques. The health value is directly employed as a utility feature

$$u_{\text{health}} = h$$

. Consequently, the robot evaluates its utility lower when its own health is low, for instance due to damage or physical impairment.

Additionally, we estimate the maximum achievable utility for a given robot state and task. By utilizing straightforward metrics, the maximum utility estimation is considerably faster compared to the full utility calculation. For instance, this estimation incorporates Euclidean distance instead of planning a path to the task.

2.3. State estimation

The robot's state comprises its pose, a health value, and battery status. In the most basic implementation utilized in most of our field work, we only roughly estimate the robots future state. The position of the robot after task execution is estimated to be at the intended pose to complete the task successfully. The battery status is estimated by multiplying the path length with the average battery consumption per distance. The average battery consumption is continually adjusted to ensure correct estimation, as this is the most important estimate state variable. For health estimation, in the basic case we can assume static health or utilize available risk estimations for terrain presented in other work.

Additionally, we have implemented a more complex, context sensitive state estimation which closely integrates the terrain the robot will walk over. This utilizes a novel transformer based prediction model, but at the time of our field missions it could not be tested robustly enough to integrate into the full control stack and we will present it in more detail in upcoming work.

2.4. Conflict resolution

Each robot has a communication node that transmits changes to the task database, such as when the robot claims a task. The ROS 2 middleware ensures that the data is distributed to all interested subscribers. In a decentralized task distribution, each robot autonomously makes decisions without synchronous communication. If multiple robots make changes to the tasks in parallel without considering the modifications made by other robots, conflict situations can arise, especially during network failures. Therefore, the communication node of each robot synchronizes incoming messages with its own state of the database, detects conflicts, and resolves them. For instance, if two robots select the same task during a network outage, the communication nodes will identify this conflict once communication is reestablished between the two robots. The robot with the lower task utility will halt the execution of the task, while the robot with the higher utility will disregard the other robot's execution and continue its own.

2.5. Adaptivity to robot damage/loss

The system is designed such that damages or failures of robots do not threaten the exploration mission. Internal and (external) health estimations evaluate a robot's health status and the risk of exploring into specific areas. In the event of damage, the robots health and, subsequently, its utility for tasks are reduced, which automatically causes other robots within the team to assume these tasks.

The task allocation plan increases the likelihood of cohesive tasks being executed by the same robot. However,

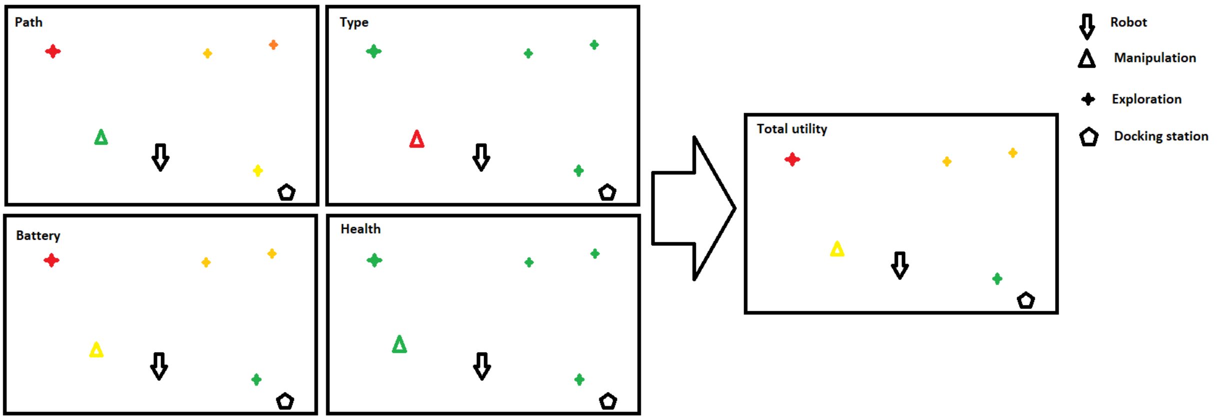

Figure 3. Example of a merged utility calculation for different tasks and utility types. Total utility is calculated as a weighted sum of the path-based cost (top left), the type-specific gain of performing the task (top right), the estimated battery loss (bottom left) and the estimated robot health (bottom right)

tasks are not preemptively reserved. Should a robot experience a complete failure, its tasks will be redistributed among other functioning robots.

3. RESULTS AND EXPERIMENTS

The architecture was developed for use in realistic scenarios for teams of autonomous robots. As it is intended to be used in arbitrary mission types, with different robots and to accomplish varying goals, we evaluated in 3 use cases.

Initially, we developed and utilized it in the ESA ESRIC Space Resources Challenge, a lunar exploration and resource prospecting scenario. In a mars exploration analogue mission as part of the intelliRISK2 project, we generally tested the approach and evaluated several additions such as risk-aware planning and behavior adaptation systems, which are beyond the scope of this paper. Finally, as part of the European Robotics Hackathon we adapted it to a nuclear power plant emergency scenario, including manipulation of valves and buttons and zones of no network connectivity.

3.1. ESA ESRIC Space Resources Challenge

The ESA ESRIC Space Resources Challenge was a realistic lunar exploration and prospecting analogue mission scenario. Given 4 hours of time, an area of about 2400 square meters had to be explored and mapped. Potentially interesting rocks in this area were to be located and analyzed concerning their geological composition. Additionally, areas of the ground contained valuable resources, specifically titanium oxides, which could not be detected visually. All of this had to be done in realistic lunar network conditions, including a limited bandwidth,

Figure 4. Spot and Husky performing tasks in the ESA ESRIC Space Resources Challenge. Spot explores the challenge area and finds potentially interesting rocks to be analyzed (left), while Husky later on independently analyzes these rocks (right).

artificial delay of all communication and sudden full network blackouts.

For this challenge, we used three commercially available robots: the Boston Dynamics Spot, the ANYbotics ANYmal and the Clearpath Husky using a Universal Robots UR10. While none of these platforms are space certified, our approach is inherently robot independent and would feasibly work on any robot hardware providing the same base low-level capabilities and some amount of processing power.

Overall, as used in the architecture description, we defined four task types:

- *Exploration*
- *Rock Candidate Checking*
- *Rock Analysis*
- *Ground Analysis*

Exploration covered the basic task of mapping the challenge area completely. Exploration tasks where automatically created by the robots using a frontier detection algorithm on a merged challenge map and could be, theoretically, completed by all robots. *Rock Candidate Checking* consisted of autonomously taking a context image of a potentially interesting object to be sent to the operators. These tasks were automatically generated whenever a point cloud based rock detection algorithm continuously running on the robots found an object that might a rock to be analyzed. The pictures taken of these objects were then checked by an operator to make a final decision if the object should be geologically analyzed. If marked thusly, a *Rock Analysis* task was created. Its execution included approaching the rock, removing the top layer with an angle grinder, then taking a close-up image of this spot and lastly performing a geochemical measurement using an x-ray spectrometer. Finally, the *Ground Analysis* task consisted of taking the same type of geochemical measurement at a certain ground spot. These tasks were generated my initial rough sampling of the full challenge area, with automatic closer sampling tasks being defined once titanium was found to fully map the resource deposit.

The two analysis tasks could, due to the necessary equipment, only be performed by the Husky platform and were defined as the highest value. The other two tasks, accordingly were shared between Spot and ANYmal. While Husky would have been able to perform these as well, it had always more valuable tasks available.

Using our distributed control structure, the robot team was able to very efficiently explore and map the environment. After only about 35 minutes of challenge time, the area was fully mapped and all rocks found. Throughout this, Spot and ANYmal explored separate areas, with Spot covering about two thirds of the full area. This difference resulted from two factors: Firstly, Spot had a higher top speed and could thus move significantly faster over the relatively simple challenge terrain. Secondly, the area covered by ANYmal included a lunar habitat prototype, which caused many potential "Rock candidates" to be found which ANYmal had to take context images of.

Throughout the same time, Husky measured about a quarter of the full ground in the challenge and analyzed one found and confirmed rock.

Overall, we were able to show highly efficient, robust autonomy of a heterogeneous robot team in a challenging scenario, which allowed us to win the challenge.

3.2. Mars exploration analogue mission

As part of the intelliRISK 2 project, we performed a mars analogue exploration mission using a robot team in the Tabernas Desert. We used a very similar basic team and architecture approach to the lunar challenge described above, substituting Husky with our own six-legged walking robot prototype.

The tasks performed and the mechanics to create and execute them were similar exploration and image tasks, which is why they are not presented in detail here, and we were able to confirm the general results of the Space Resource Challenges in a different scenario and environment.

Figure 5. Mars exploration analogue mission in the Tabernas Desert in Spain.

The main focus of this mission, though, was to evaluate the extendibility of the architecture with more advanced components. Specifically, instead of our basic global navigation planner built to find the shortest traversable path from a robot to a given target point, we developed a risk aware path planner that actively considers the dangers and advantages of different ground types. Additionally, we extended the low-level robot control system to adapt to unexpected sensor stimuli with an automatic cautious slowdown to increase robustness. Finally, our basic frontier detection approach to generate exploration tasks was replaced with a novelty based learning approach, that specifically generates exploration targets in areas that seem more "interesting", for example due to them including more objects of interest or different terrain than previously seen.

While all of these individual systems cannot be discussed here and will be presented in other works, of special note here is their implementation into the core multi-robot control architecture. Due to the distributed system used, we were able to drop-in replace the relevant components, without any additional modifications needed. Theoretically, it would even be possible to dynamically swap online during robot operation when needed.

This highlights that our approach is not only feasibly to create efficient robot collaboration in realistic scenarios, but is also easily extendable towards different robots and more complex individual system components.

3.3. European Robotics Hackathon

The final evaluation scenario to be mentioned was the European Robotics Hackathon ENRICH, which presented a nuclear power plant emergency scenario. The tasks themselves, which included regular and radiation mapping and manipulation of valves, were again similar to the previous missions, though performed in the much more constrained environment of a nuclear power plant.

We again used Husky, ANYmal and Spot, though the latter was equipped with a manipulator, and a similar task

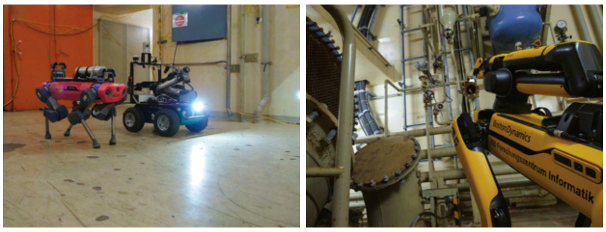

Figure 6. Anymal, Husky and Spot in the ENRICH scenario. ANYmal and Husky (left) start initial 3D and radiation mapping. Spot (right) gets ready to turn a valve using its manipulator.

structure. The primary mission difference was the presence of network dead zones. Instead of timed loss of connectivity, certain rooms were built to make communication with the robot impossible while inside.

To combat this, we developed an additional extension that mapped the connectivity in visited areas. When communication was necessary, this allowed the robot to return autonomously to an area where a good connection was possible. Once again, this extension easily integrated into the full architecture without modifications needed.

4. CONCLUSIONS AND FUTURE WORKS

We presented a decentralized approach to allow for cooperation in teams of autonomous robots. Utilizing a high degree of robot-level intelligence, this approach can foster efficient collaboration, without requiring a single planning system to closely understand all robots and their capabilities. This creates a high degree of modularity and allows scaling towards both different and bigger robot teams. We showcased our system in multiple planetary and terrestrial field test scenarios with different robotic team members and task structures. Overall, the ease of adapting our approach to different robots and scenarios proves its scalability. The implementations usage in multiple realistic mission tests allows shows its continuing robustness and thus push towards higher TRLs. Especially our results in the ESA ESRIC Space Resources Challenge showcase the strengths of the presented control structure and its ability to utilize a high degree of robot autonomy in difficult planetary scenarios.

This provides a significant step towards bringing safe and robust, but also efficient autonomy into high-risk scenarios. With an increasing effort towards more lunar and planetary missions in the near future, these more efficient control architectures are required for long-term operation of projects such an active lunar base and larger scale mining of resources. Towards this goal, we will continue increasing the system's robustness and safety and evaluate it in more and increasingly difficult scenarios. This will bring us closer to our goal of high-TRL autonomous robot teams.

ACKNOWLEDGMENT

The research leading to these results has received funding from the DLR Space Administration under grant agreement No. 50RA2026 by the German Bundestag.

REFERENCES

[1] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1. IEEE, pp. 476–481. [Online]. Available: http://ieeexplore.ieee.org/document/844100/

[2] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," vol. 21, no. 3, pp. 376–386, conference Name: IEEE Transactions on Robotics.

[3] J. de Lope, D. Maravall, and Y. Quiñonez, "Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems," vol. 163, pp. 47–55. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215004531

[4] A. Kanakia, B. Touri, and N. Correll, "Modeling multi-robot task allocation with limited information as global game," vol. 10, no. 2, pp. 147–160. [Online]. Available: https://doi.org/10.1007/s11721-016-0123-4

[5] T. Schnell, T. Büttner, L. Puck, C. Plasberg, G. Heppner, A. Rönnau, and R. Dillmann, "Risikobewusste roboter für erhöhte autonomie in planetaren explorationsmissionen," publisher: Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. [Online]. Available: https://publikationen.dglr.de/?tx_dglrpublications_pi1%5bdocument_id%5d=480140

[6] M. F. Ginting, K. Otsu, J. A. Edlund, J. Gao, and A.-A. Agha-Mohammadi, "CHORD: Distributed data-sharing via hybrid ROS 1 and 2 for multi-robot exploration of large-scale complex environments," vol. 6, no. 3, pp. 5064–5071, conference Name: IEEE Robotics and Automation Letters.