IAC-22-D1-6-10x74101

Motion planning for relocatable robots performing on-orbit locomotion and manipulation tasks

Anatoliy Huzynets[a], Ismael Rodriguez[b], Mathieu Deremetz[c], Pierre Letier[c], Máximo Roa[b*]

[a] Politecnico di Milano, *20133 Milano, Italy*, anatoliy.huzynets@mail.polimi.it

[b] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), *82234 Wessling, Germany*, firstname.lastname@dlr.de

[c] Space Applications Services, *1932 Sint-Stevens-Woluwe, Belgium*, firstname.lastname@spaceapplications.com

[*] Corresponding author

Abstract

In-space assembly is a key technology for the future development of large infrastructures in space, from space stations and telescopes, to solar power plants or planetary bases. Such structures are much larger than cargo areas in current launchers, therefore they must be sent in separate pieces that are assembled in situ, typically using relocatable robotic manipulators. The efficient exploitation of the locomotion and manipulation (loco-manipulation) abilities for such robotic systems requires suitable planning tools. In this paper, we present a motion planning approach for exploiting loco-manipulation abilities of self-relocatable space robots, assuming that they move over specific interconnects that provide the required mechanical, power and data connectivity. The proposed approach consists of three planning layers: a high-level planning for obtaining the contact sequence, a low-level planning for the joint trajectories, and a validation layer. The motion planner provides plans for single locomotion and manipulation tasks, as well as combined loco-manipulation tasks. The approach is illustrated with examples for two robotic systems: MOSAR-WM, a relocatable walking manipulator, and a multi-arm robot (MAR) equipped with two arms attached to a central torso.

Keywords: space robotics, relocatable robot, multi-contact planning, motion planning, loco-manipulation.

Acronyms/Abbreviations

International Space Station (ISS), Standard Interconnect (SI), Degrees of Freedom (DoF).

1. Introduction

Significant developments of the space sector in recent years are pushing space technology toward the development and construction of large space structures. Those structures must be launched in pieces and deployed in situ due to their dimensions, much larger than current cargo areas available in present launchers. A key technology to allow large-scale cost-effective on-orbit manufacturing and assembly is autonomous robotics, typically using relocatable robotic manipulators. Such approach allows to overcome dimensional constraints imposed by launchers, and to significantly reduce costs associated to launch and on-orbit operations. In fact, if structures are planned to be assembled on-orbit, they do not undergo launches in assembled state, which relaxes their structural requirements. Robots offer great flexibility for multiple manufacturing tasks, and can be exploited even for tasks that are not expected at the moment of the launch. They can be programmed or tele-operated, leading to a significant increase in safety for the astronauts, whose time and energy can be devoted for more scientific purposes.

Several robotic manipulators are currently operational in space. Perhaps the most known is Canadarm2[1], together with Dextre (a smaller two-arm robot)[2], mounted outside the ISS and used for operations such as relocation of loads and berthing. It is the successor of Canadarm[3], which was in service from 1981 to 2011 onboard the Space Shuttle and, among other tasks, was used to support the ISS construction. Similarly, the European Robotic Arm (ERA) was recently mounted on the Russian segment of the ISS for handling external payloads, supporting extra-vehicular activities, and for inspection tasks[4]. The Japanese Experiment Module Remote Manipulator System (JEMRMS)[5] consists of a large arm for heavy tasks and a small arm, mountable as end effector on the large arm, for operations that require additional accuracy. Both robots are currently in service on the ISS. The Tiangong Robotic Arm is currently mounted on the Tiangong Space Station[1], it can relocate itself on top of the station, and is used for handling supplies and to support reparation tasks.

To efficiently exploit the versatility of such robotic systems for autonomous assembly tasks and similar operations, suitable motion planning tools are required. The aim of such a motion planning tool

[1] https://www.asc-csa.gc.ca/eng/iss/canadarm2/default.asp

[2] https://www.asc-csa.gc.ca/eng/iss/dextre/

[3] https://www.asc-csa.gc.ca/eng/canadarm/default.asp

[4] https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/International_Space_Station/European_Robotic_Arm

[5] https://iss.jaxa.jp/en/kibo/about/kibo/rms/

is to consider a generic high-level task and find a suitable sequence of actions and motions required to fulfill the task. In assembly operations, the most relevant tasks are walking and manipulation. Planning locomotion and manipulation actions has been separately addressed in the past [2]. In general, motion planning involving contacts is generally divided into two stages: a search for motions in the obstacle-free space, and a search for configurations that require contact with the environment. Probabilistic methods can be applied to the former, but fail with the latter [3]. Locomotion and manipulation are conceptually similar problems, as they both deal with underactuation and solve it through suitable contact forces [4]. In recent years, both problems have been considered together in whole-body motion planning approaches [5]. The presence of contact and whole-body motions in the planning problem are two major common features of humanoids and space robots.

This paper presents an approach to plan activities of locomotion, manipulation, and loco-manipulation of objects with relocatable space manipulators. The approach is based on a hybrid planner that combines a high-level layer, representing the discrete transitions from one contact state to another, and a low-level layer considering the joint trajectories to move from a contact state to the next one. They are complemented with a validation layer that verifies the fulfillment of all the considered constraints. The developed planning tools are applied to two different relocatable robots, the MOSAR-WM and the MAR.

2. Problem Statement

This paper develops a loco-manipulation planner, i.e. a framework of tools and algorithms that provides solutions to the motion planning problem for locomotion and manipulation tasks, either isolated or in combination in the so-called loco-manipulation tasks (or whole-body tasks).

The input, or action to be performed, is specified as actions, such as "take an object and move it to a desired pose," or "relocate to a new position in the environment". The output contains the actuation instructions, i.e. trajectories and torques at the joint level, as well as latching/unlatching commands for end effectors. A valid output motion must fulfill the following constraints:

1. geometric: collision avoidance,
2. kinematic: joint range, velocity and acceleration limits,
3. dynamic: joint torque limits and contact force/torque limits at coupling interfaces.

The robot interacts with the world by latching its end effectors to discrete supports. End effectors and supports are equipped with dedicated coupling devices called standard interconnects (SI). When two SIs latch, the two bodies are rigidly connected, as long as mechanical limits of the interface are not violated. Cartesian motions are used for establishing/removing a contact. In all other cases, motions are planned directly in joint space.

Within the paper, two different robots are considered: (i) the MOSAR Walking Manipulator (WM), which introduces the planning problem in a relatively simple case, and (ii) the MIRROR Multi-Arm Robot (MAR), which adds multiple end effectors. For our cases, we will use HOTDOCKs [6] as SIs. Their mating geometry exhibits a 90deg symmetry, thus 4 latching configurations are admissible, identified with a rotation index r, as illustrated in Fig. 1. The rotation index of the tip of the robot is denoted with r_t, and the rotation index at the base with r_b.

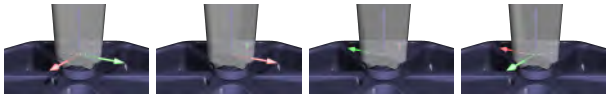


Fig. 1: Rotation index for the HOTDOCK SI. The bottom HOTDOCK is fixed. The upper one rotates by 90 deg around its z axis in each figure from left to right (the reference frame represents its pose), showing the rotation indices $r = 1, 2, 3, 4$.

3. Planning method overview

Generally, two paradigms are generally used for motion planning involving contact: (i) *all-at-once* methods, or (ii) *contact-before-motion* methods [3, 7]. The former usually formalizes the overall problem as a large optimization task; the latter approach breaks it down into sub-problems and typically reduces the planning complexity.

The contact-before-motion approach is implemented within this work, using two planning layers:

Contact planning (high-level layer): provides a sequence of contacts, representing the robot interactions with the environment (i.e. physical steps or operations such as grasping/releasing objects). At this stage, we define the sequence of (i) grounding states, (ii) end effectors, (iii) associated poses for the end effector, and (iv) latching state of end effectors.

Path planning (low-level layer): provides a set of geometrical paths in the joint space allowing the transition from one contact state to the next one. From the geometric path, a time trajectory is generated and used as the feed-forward reference for the joint control.

Note that the contact planning relies on path planning to evaluate validity and assign costs to transitions, i.e, the two layers are coupled. A final third layer of validation simulates the overall motion to verify that all constraints are satisfied. We have employed a similar two-layer planning structure in previous work for assembly planning problems [7, 8]. This paper extends such planning structure to the loco-manipulation domain.

4. MOSAR Walking Manipulator

The Walking Manipulator (WM) is a robotic arm for space developed within the European project MOSAR [9]. It features a $1.6m$ long, 7-DoF serial symmetric kinematic chain and two end effectors, one at each extremity. They are referred to as EE_A and EE_B, and are both equipped with SIs. The MOSAR demonstrator consists of a mock-up modular reconfigurable satellite scenario, as shown in Fig. 2, comprising a servicer (left of the black stripe) and a client satellite (right of the black stripe).

Fig. 2: Experimental setup during the ground demonstration of MOSAR-WM [9].

4.1. Operational strategy

Walking is performed by alternating the end effectors: while EE_A is grounded, EE_B is used to reach a target SI, and vice versa. When manipulating, an end effector is grounded, and the other one performs the pick and place operations using the required SIs. The rigid robot-ground connection, even if alternated, allows to use the so-called *switching fixed-base* approach [10] to simplify the planning problem. Essentially, the robot base frame is always located at the grounded SI. Suitable transition equations are introduced when the robot configuration is reversed with respect to the initial one [10]. For simplicity of implementation, after a contact transition, a duplicate figurative kinematic chain is introduced and overlapped with the actual one. The planning is then performed on the twin chain and mapped back to the physical robot. Thanks to this technique, the same planning tools can be used, independently of whether EE_A or EE_B is grounded.

4.2. Contact planning

4.2.1 Locomotion planner

The robot displays a hybrid dynamics, and will be represented using a directed graph $G = \{V, E\}$, where the vertices (or nodes) $v \in V$ represent robot states, and the edges $e \in E$ represent transitions between states. A state is formalized as $s = \{b, r_b, \mathbf{q}\}$, where b is the base support (grounding), r_b the rotation index of the base, and $\mathbf{q}$ the joint vector. The objective of the planner is to connect the start and target node. Fig. 3 provides an example of a simplified representation of a locomotion graph, where each node represents b, and the graph is overlapped with the physical environment.

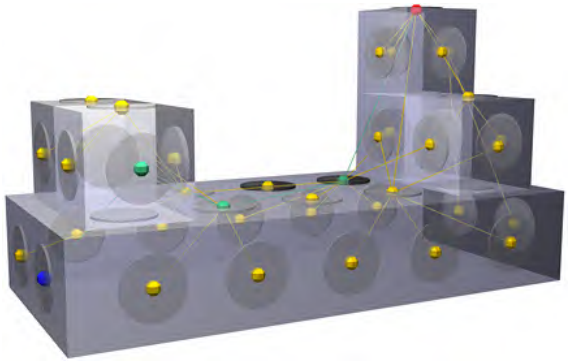


Fig. 3: Simplified locomotion graph, where the start node is represented in red, the target node in blue, the nodes/edges of the planning graph in yellow, and the final trajectory through the graph (physical transitions to be executed) in green.

The graph is constructed with a breadth-first search with priority queue reordering. Essentially, the algorithm evaluates all possibilities for the first step, then all possibilities for *all* possible second steps, and so on. The priority queue reordering allows to break this scheme while still preserving breadth. In particular, new edges are always stemmed towards the node with the lowest current cost. If a branch becomes too expensive, it will not be explored any further. The graph construction comprises (a) assessing the existence of transitions and (b) assigning costs to nodes and edges. Point (a) is performed with 3 incremental layers: (i) inverse kinematics, (ii) path planning, (iii) dynamic simulation; when all three checks are satisfied, an edge is added. For point (b), the *node cost* of v_i is the Euclidean distance between b_i, i.e. the position of the support associated to v_i, and the target support b_{target}. This cost is then normalized with the Euclidean distance between b_{parent}, i.e. the support of the parent node of v_i, and the target support b_{target}. The *edge cost* is the maximum torque saturation for the robot joints while performing the transition motion, where the (normalized) torque saturation is defined as $\hat{\tau}_i = \tau_i / \tau_{lim_i}$. The cost for the start node is null. The cost of vertex v is $cost_v = cost_{parent} + w_v c_v + w_e c_e$, i.e. a weighted sum of vertex and edge costs, cumulative along the graph traversal path.

4.2.2 Manipulation planner

Task inputs are (i) the starting state of the robot s, (ii) the SI to grasp, and (iii) the pose of the manipulated object that must be attained by manipu-

lation. The planning outputs are (i) a set of poses of the manipulating end effector, and (ii) a set of cost-ordered plans out of the four possible grasping options to grasp the initial SI (i.e., considering the rotation index r). The assumption for a pure manipulation problem is that the operation is feasible without relocating the robot.
Generation of poses: The algorithm defines reference poses for the manipulating end effector. Specifically, Cartesian approaches must account for the presence of neighboring objects to allow proper coupling/decoupling of all involved mating devices. The direction of motion is solely determined by the inputs and the environment geometry.
Manipulation optimization: The rotation index of the tip r_t is the design variable to optimize in this contact planning layer. Its domain is R_{SI}, that formalizes the 4 symmetries of the SI. The path planning layer solution for moving from one pose to the other is fixed, with a joint trajectory is $\mathbf{q} = \mathbf{q}(t, r_t)$, and corresponding joint torques $\tau = \tau(\mathbf{q}(t, r_t))$. Torques are then normalized with torque limits and the time dependency is removed by introducing the cost of motion c_m:

$$\min_{r_t \in R_{SI}} c_m(r_t) = \max_{t \in [t_1, t_2], i \in I} \hat{\tau}_i \left(\mathbf{q}(t, r_t) \right) \qquad (1)$$

which defines the manipulation cost as the maximum torque saturation among all joints $I = \{1, \ldots, 7\}$ along the motion. The objective is to find the SI rotation index r_t that minimizes the cost c_m.

4.2.3 Loco-manipulation planner

This planner allows to plan *relocate-pick-place* operations, assuming that there exists a support for the robot base such that the pick and place targets are within the robot's workspace.
Optimization: First, reference poses are defined as in the manipulation case. An optimization problem is defined, now with two cost terms: c_m, associated to manipulation and analogous to eq. 1, except that it now depends on both grounding mode (b, r_b) and grasping mode r_t; and $c_l(b)$ associated to locomotion and computed as $c_l(b) = \|b - b_{start}\| / \|\sigma_m - b_{start}\|$, with σ_m the centroid between the pick and the place tool center points. The objective is to find (b, r_b, r_t) that minimizes the weighted sum of both costs:

$$\min_{b \in A', r_b, r_t \in R_{SI}} w_l \, c_l(b) + w_m \, c_m(b, r_b, r_t) \qquad (2)$$

Graph search: The outcome of the minimization is a set of N candidate manipulation plans $cmp_i = \{b, r_b, r_t\}_i$, ordered by increasing total cost. The locomotion planner is iteratively asked to search for a locomotion plan from $(b, r_b)_{start}$ to $(b, r_b)_i$, for $i = 1, \ldots, N$. When a valid locomotion plan is found, the loop is interrupted and the outputs are returned.

4.3. Path planning

The trajectory generation is implemented with a lines-and-parabolas method. First, time instants associated to each waypoint are computed for minimum actuation time considering the joint velocity/acceleration limits. Then, joint trajectories are defined through segments connecting those waypoints. Finally, parabolic blends are added between segments. The outcome trajectory alternates then constant acceleration and constant velocity sections.

4.4. Results

Both the contact and the path planning layers are implemented in CoppeliaSim, from Coppelia Robotics. For inverse kinematics and collision avoidance we use Coppelia Kinematic Routines[6], and for planning algorithms (in particular, RRTConnect) we use the Open Motion Planning Library[7].

4.4.1 Simulation results

Several WM capabilities are demonstrated through five exemplary tasks reported hereafter, numbered D1 to D5. D1 is a manipulation task with the aim of transferring a module from the servicer to the client spacecraft. The planner finds three candidate plans and selects the one corresponding to the rotation index $r = 2$. The planning is successful and takes 229.1 s, while its validation takes 107.9 s. Fig. 4 shows the execution of D1. Note that the support highlighted in red is where the arm is grounded; the one in purple is the grasp target. The module release is performed with an elbow-down configuration.

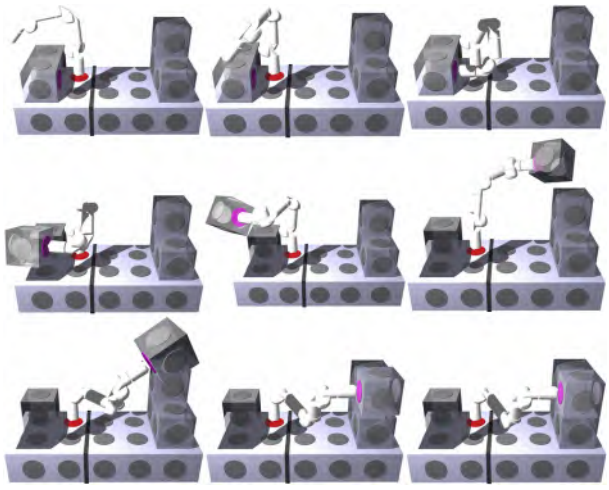


Fig. 4: WM executing demonstration task D1.

D2 is a manipulation task with the aim of transferring a module from the client to the servicer spacecraft. The planner finds three candidate plans, and

[6] https://github.com/CoppeliaRobotics/coppeliaKinematicsRoutines
[7] https://ompl.kavrakilab.org/

selects the one corresponding to the rotation index $r = 3$. The planning is successful and takes 187.8 s, while its validation takes 91.4 s. Fig. 5 shows the execution of D2. The grasping is performed with an elbow-down configuration.

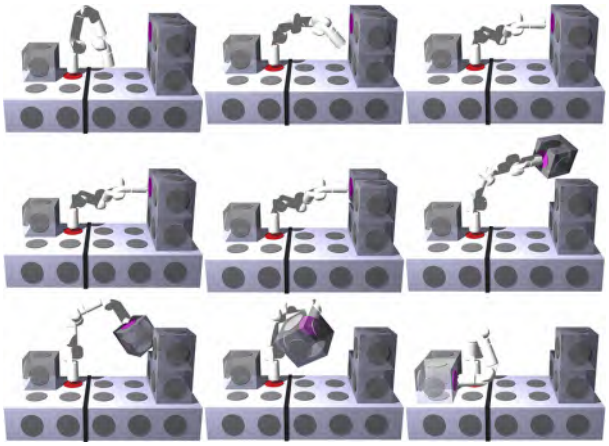


Fig. 5: WM executing demonstration task D2.

D3 is a loco-manipulation task, illustrated in Fig. 6, and involving locomotion to an optimal support and then manipulation of a module from the servicer to the client spacecraft. After 63.4 s, the planner concludes that the manipulation task is not feasible from the initial grounding state. The planner finds a new optimal base location, and plans 53 candidate manipulation plans in 6605.0 s. The locomotion planning to reach the optimal support takes 38.1 s. The overall plan is validated in 85.6 s. Note in Fig. 6 that indeed the initial grounding of the robot does not allow to pick the target module, as it is too close to the WM base. Thus, the arm first relocates from the initial (red) support to the optimal (blue) one, using a single step; the two supports are 0.4 m apart and lie on the same horizontal plane. After grasping, the WM performs a redundant clockwise rotation around its base. To avoid this, a path optimization should be introduced in the path planning layer.

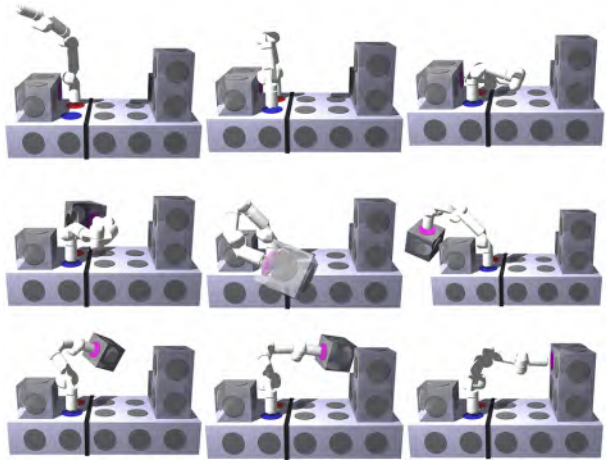


Fig. 6: WM executing demonstration task D3.

D4 is a locomotion task, illustrated in Fig. 7, where the starting support is shown in red, and the final one in blue. The planner creates a graph of 17 possible nodes in 2019.4 s before returning an output. The planning is successful and the resulting plan comprises 3 steps on two different horizontal planes, as shown in Fig. 7.

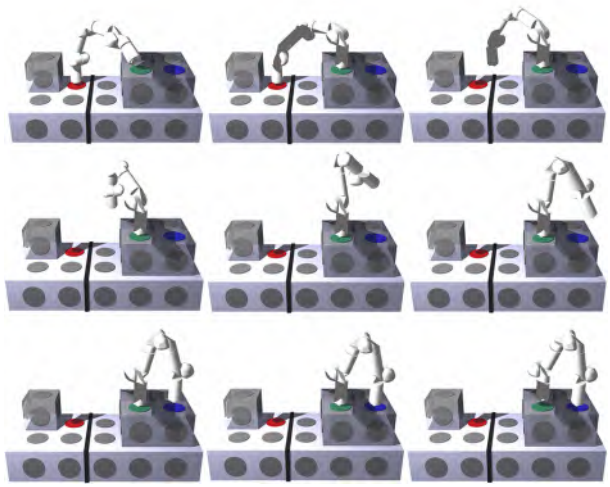


Fig. 7: WM executing demonstration task D4.

D5 is a locomotion task, illustrated in Fig. 8. The planner creates a graph of 11 nodes in 1495.6 s before returning an output. The planning is successful and the resulting plan comprises 3 steps. Note that D5 is the inverse plan of D4. However, the return path found here is slightly different.

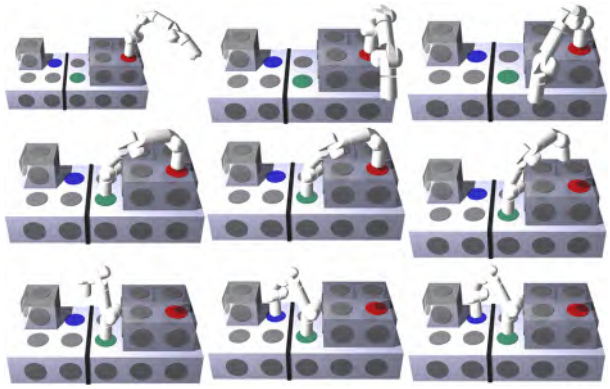


Fig. 8: WM executing demonstration task D5.

Planning and validation are successful in all considered tasks. Contact transitions cause discontinuities in the dynamic loads (torques and interaction forces), corresponding to changes in grounding support or grasping/releasing the manipulated objects. The simulations validated the capability of the planner to provide feasible plans for locomotion, manipulation and loco-manipulation tasks for the WM.

4.4.2 Experimental results

The previous subsection shows the results obtained with the planning tools tested in simulation. Analogous tasks were performed with the hardware setup. A video of execution of the planned tasks is available online[8]. The test campaign validated the capability of the WM to perform walking on horizontal and vertical surfaces, and manipulation operations with 10 kg payload in 1g environment.

5. MIRROR Multi-Arm Robot

The Multi-Arm Robot (MAR) is a robotic manipulator for space, in development under the ESA project MIRROR [11]. The MAR is composed by a torso and two arms. The torso is a truncated tetrahedron with one actuated end effector (1 DoF). Each arm is 7-DoF and 1.8 meters long (based on the WM). All end effectors are equipped with coupling SIs. MAR operates on a large-scale modular telescope made of hexagonal mirror tiles. To simulate the microgravity during operation, we assume a $g/20$ gravity.

5.1. Operational strategies

MAR features three end effectors. This multiplicity introduces additional freedom to decide which end effector to use for each action. The following operating modes are selected for implementation:

locomotion: the robot relocates by alternating one arm with the other for grounding;

manipulation: the robot stands on one arm and manipulates with the other one; also, the robot can stand on any of the arms and manipulate with the torso;

loco-manipulation: arises as a combination of the previous strategies.

5.2. Extension of planning tools

The planner for the MAR is composed by the same layers and features as the one for WM, and the capabilities are extended to cope with the enhanced operational modes. A switching fixed-base strategy for modelling is once again employed. The major novelty is the possibility of specifying which end effector to use (in all calls to the inverse kinematics, path planning, manipulation and loco-manipulation contact planners). Note that in some cases an arm is not involved in the execution of a task, e.g. during manipulation with the torso. In those cases, the configuration of the unused arm is left free, but still respects joint limits and avoids collisions.

5.3. Results

The planning tools are tested with CoppeliaSim. Five locomotion, five manipulation and five loco-manipulation tasks, depicted in Fig. 9, Fig. 10 and Fig. 11, are used to test the approach. The results are summarized in Table 1, Table 2, Table 3. The simulations validate the capability of the planner to provide feasible plans for locomotion, manipulation and loco-manipulation tasks for the MAR.

The loco-manipulation plan LM10 is presented with additional details: Fig. 12 reports the MAR executing the plan; Fig. 13 and Fig. 14 report the associated joint torques; Fig. 15 reports the associated contact forces at the grounding of the robot, and Fig. 16 reports the graph associated to the locomotion task. From the start to $t = 175s$, the right arm is significantly more loaded (Fig. 13) than the left arm (Fig. 14). After the contact transition, the trend is reversed, and so on after each change in grounding end effector ($t = 480s$, etc.). As soon as the manipulated module is picked up, also a discontinuity in total vertical force F_z is visible in Fig. 15 at $t = 1460s$. At the end of the task, the tile is released and the total vertical force resumes its value, corresponding to the total weight of the robot. In general, due to the important masses of the manipulator, the joint drives and mating interfaces are heavily loaded. Thus, it is preferable to stay within a reduced volume of the workspace during planning, forcing the robot to perform multiple small steps that are less demanding on the drive units.

6. Conclusions

This work proposed a two-layer planner that provides motion plans for different robots, different environments and various input tasks, including manipulation, locomotion and loco-manipulation actions. A final validation layer provides a final check of the feasibility of the plan considering all the robot and environmental constraints. The developed planner produces locomotion plans ranging from single-steps to plans that traverse the entire operating environment. Manipulation tasks were planned with diverse grounding, grasping, and placement conditions, including horizontal and vertical surfaces. Combined walking and manipulation operations were also accomplished.

Planning parameters and weights can be tuned to influence the desired output, e.g. to minimize number of steps, minimize torque saturation, limit distance between steps, etc. The planning time is suitable for offline use. Overall, the success rate of the planner is high, both in the planning and validation layers. This is mainly a result of the proposed formalization, and of the implemented strategy for graph search and numerical optimization. The two-layer strategy allows a quick elimination of non-feasible plans, e.g., the high-level planner already discards plans that violate the constraints considered at that stage. When a plan results invalid, it is possible to (i) reduce velocities and accelerations in the trajectory planning, (ii) modify or re-plan the path layer, (iii) modify or re-plan the contact layer. In this regard, the plan-

[8] MOSAR Final Demo: `https://youtu.be/kqwKEPRXEYU`

Table 1: Locomotion planning results for MAR.

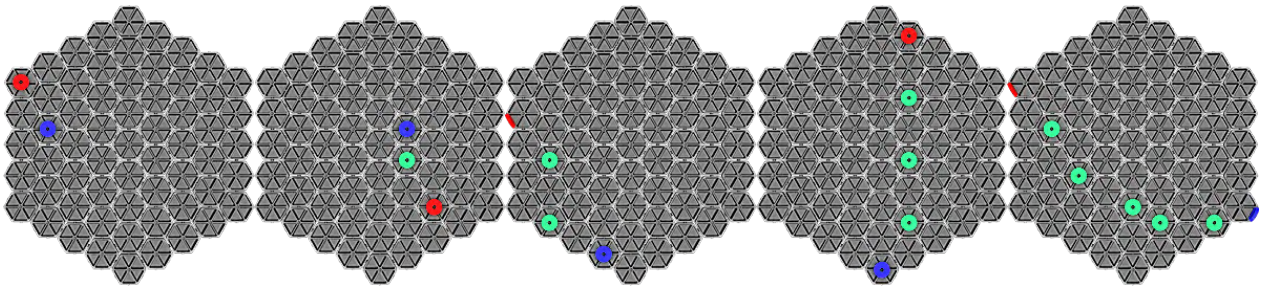
Plan ID	L6	L7	L8	L9	L10
Fig.	9a	9b	9c	9d	9e
$\#v$	2	26	47	71	112
t_{plann} $[s]$	30.2	412.1	581.1	837.8	1968.0
t_{valdt} $[s]$	188.3	236.2	248.5	836.9	483.5

Table 2: Manipulation planning results for MAR.

Plan ID	M6	M7	M8	M9	M10
Fig.	10a	10b	10c	10d	10e
$\#plans$	3	3	4	4	4
t_{plann} $[s]$	146.5	161.9	186.0	204.3	197.6
t_{valdt} $[s]$	179.0	238.3	192.4	252.6	226.6

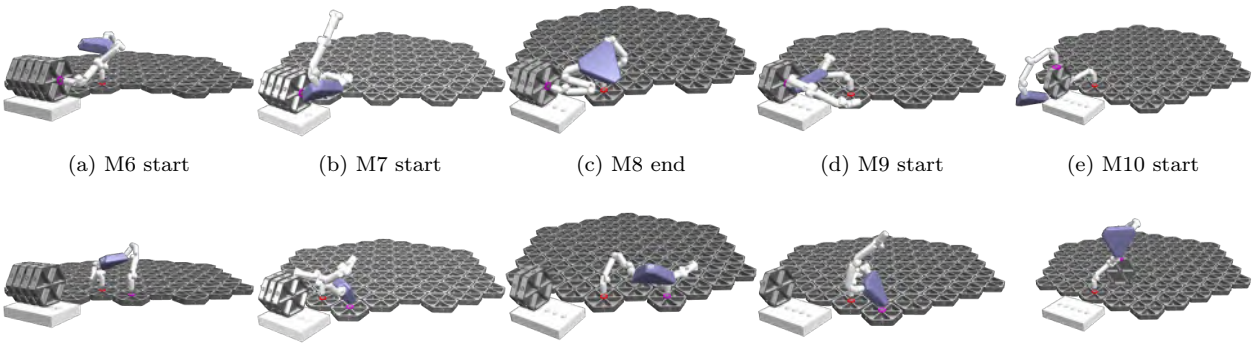
Table 3: Loco-manipulation planning results for MAR.

Plan ID	LM6	LM7	LM8	LM9	LM10
Fig.	11a	11e	11i	11m	12
# candidate manip. plans	57	52	44	37	53
# nodes in locomotion graph	2	22	19	23	98
Time to find candidate manip. plans [s]	3355.6	1959.8	2184.1	2245.9	3070.6
Time to find locomotion plan [s]	55.5	266.9	206.9	480.6	5101.4
Time to validate [s]	205.1	325.8	297.3	366.1	683.5



(a) L6 (b) L7 (c) L8 (d) L9 (e) L10

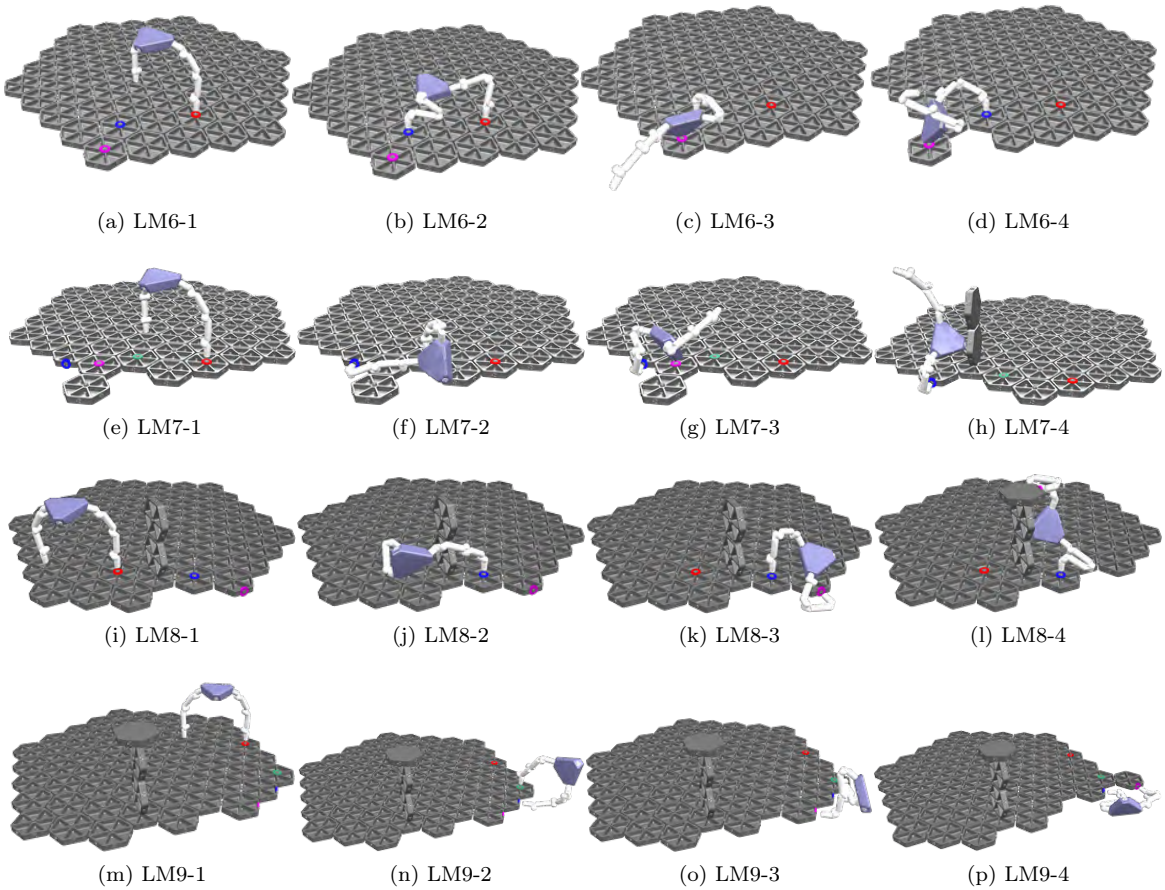
Fig. 9: Locomotion plans for the MIRROR MAR. The colors highlight the starting support (red), intermediate supports (green) and target support of the plan (blue).



(a) M6 start (b) M7 start (c) M8 end (d) M9 start (e) M10 start

(f) M6 end (g) M7 end (h) M8 end (i) M9 end (j) M10 end

Fig. 10: Manipulation tasks for the MIRROR MAR. The robot/environment is shown at the beginning and end of each task.



(a) LM6-1 (b) LM6-2 (c) LM6-3 (d) LM6-4

(e) LM7-1 (f) LM7-2 (g) LM7-3 (h) LM7-4

(i) LM8-1 (j) LM8-2 (k) LM8-3 (l) LM8-4

(m) LM9-1 (n) LM9-2 (o) LM9-3 (p) LM9-4

Fig. 11: Loco-manipulation plans for the MIRROR MAR. For each plan, the start (-1) and the end (-2) of the locomotion, the pick (-3) and the place (-4) operation are shown.

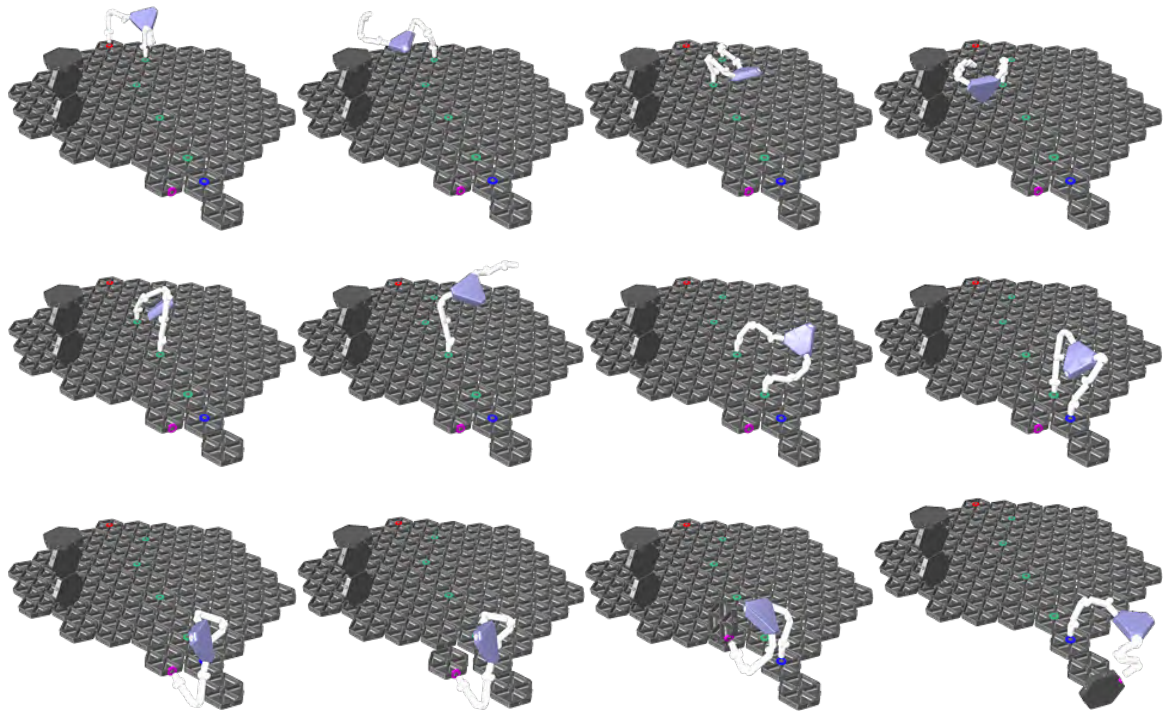


Fig. 12: Snapshots of the MIRROR MAR executing the loco-manipulation plan LM10.

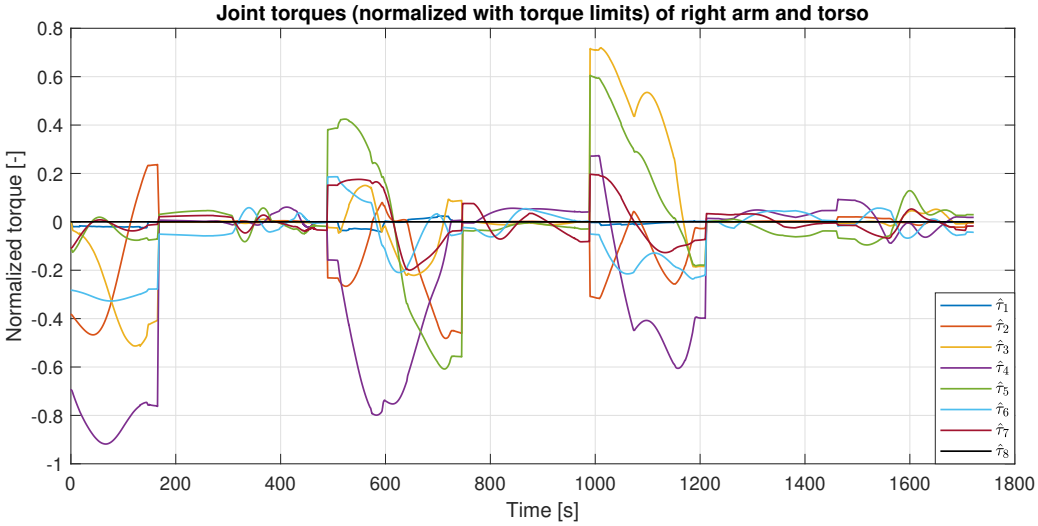


Fig. 13: Joint torques of right arm/torso for the LM10 plan.

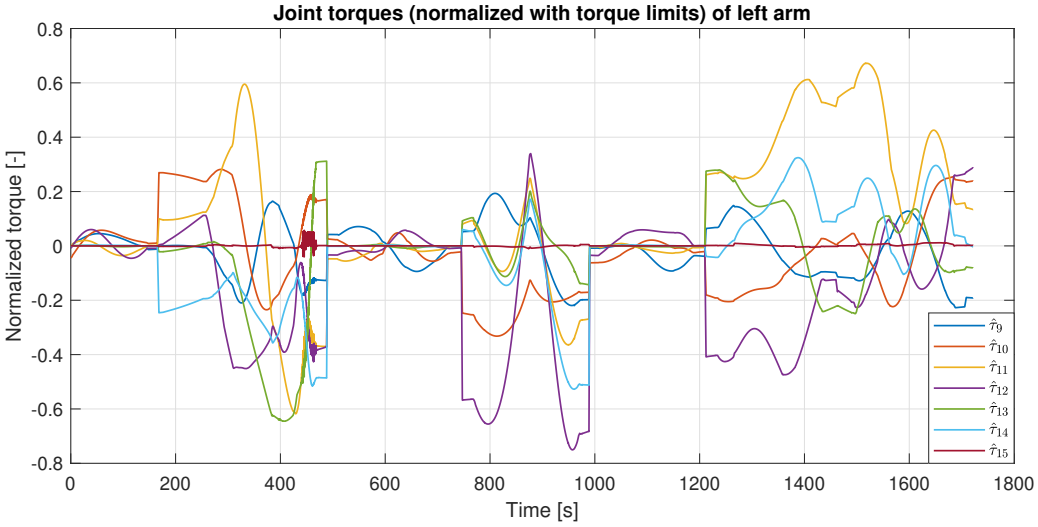


Fig. 14: Joint torques of left arm for the LM10 plan.

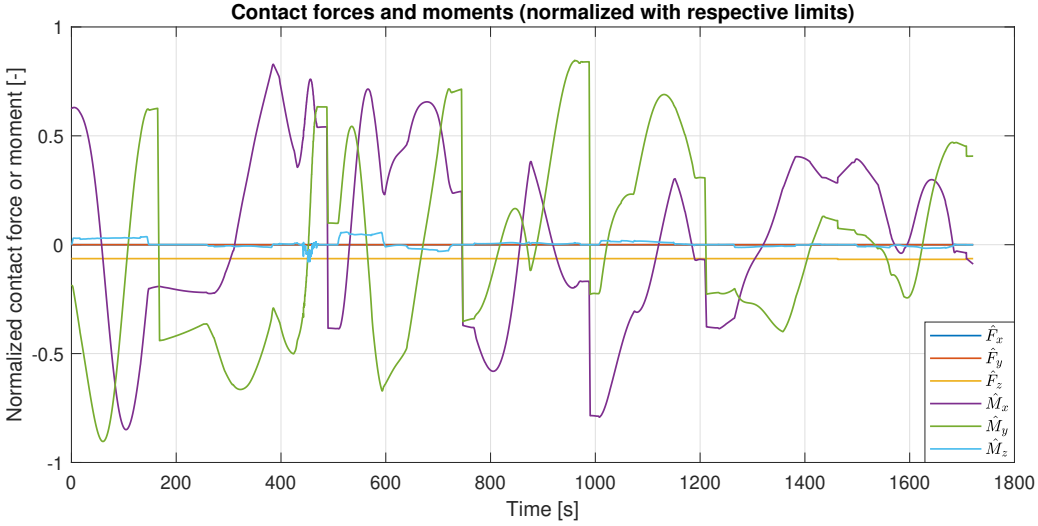


Fig. 15: Contact forces at the current grounding SI for the LM10 plan.

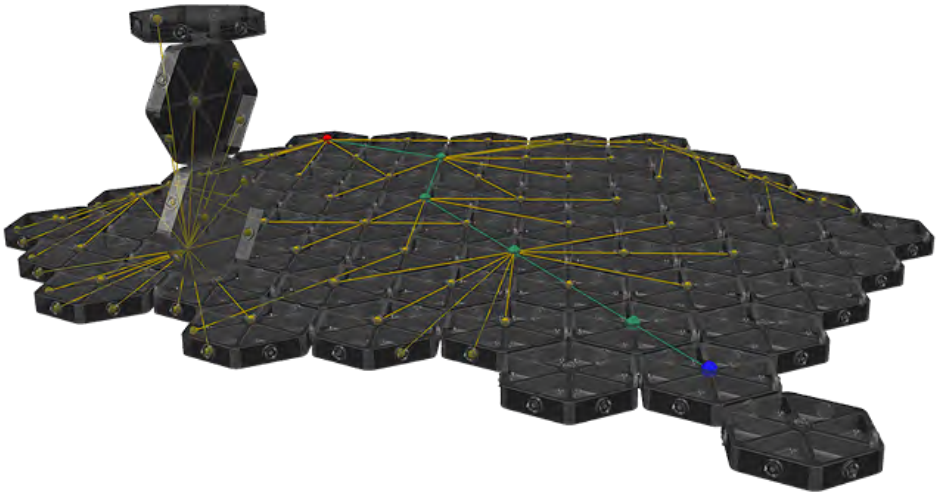


Fig. 16: Locomotion graph for LM10. The start node is shown in red, the target node in blue, the nodes/edges of the planning graph in yellow, and the final trajectory through the graph (physical transitions to be executed) in green.

ner provides not a single plan but a set of candidate plans based on the graph representation. The exploitation of this available (but not optimal) paths is left as future work.
The planner only provides solutions if at least one exists. Otherwise, it only returns a failure. This happens for instance in cases when the inputs are not valid, when a target support is already latched to something else, when the specified target support cannot be reached without violating some constraints, when the specified target pose of the manipulated object causes collisions with the environment, or when the object to manipulate is too heavy.
A limitation of the path planning layer is that it relies on random sampling of the RRT algorithm. This introduces a non-deterministic component in the generation of joint paths. As a result, also the contact planning layer is not deterministic, as the costs in contact planning are based on torque (which is path-dependent). However, the effect of the random sampling on the contact planning layer is limited. It might be not negligible when the solution space is even more constrained.
In future, other operating modes of the MAR may be exploited, e.g. grounding the torso and using both arms to manipulate different objects simultaneously. Also, *pick-relocate-place* operations can be considered, as well as autonomous reconfigurability, e.g. removing one arm or attaching the two arms in series. Additional constraints can be included in the high-level planning. Processing time may be reduced by avoiding simulations in that stage, or introducing a depth-first search. It is also helpful to optimize the path planning with e.g. an optimal kinetostatic configuration generator. The large null-space of the MAR can be exploited for this purpose.

7. Acknowledgements

This study was partially funded by the European Union H2020 Program under Grant No. 821966 (MOSAR: Modular Spacecraft Assembly and Reconfiguration) and by the European Space Agency (ESA) under contract No. 4000132220/20/NL/RA (MIRROR: Multi-arm Installation Robot for Readying ORUS and Reflectors).

References

[1] Y. Liu, S. Cui, H. Liu, M. Jin, F. Ni, Z. Li, and C. Li. Robotic hand-arm system for on-orbit servicing missions in Tiangong-2 space laboratory. *Assembly Automation*, 2019.

[2] K. Harada and M. A. Roa. Manipulation and task execution by humanoids. In *Humanoid Robotics: A Reference*, pages 1633–1655. Springer, 2019.

[3] K. Bouyarmane, S. Caron, A. Escande, and A. Kheddar. Multi-contact motion planning and control. In *Humanoid Robotics: A Reference*, pages 1763–1804. Springer, 2019.

[4] B. Henze, M. Roa, and C. Ott. Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *Int. J. of Robotics Research*, 35(12):1522–1543, 2016.

[5] E. Yoshida, F. Kanehiro, and J. P. Laumond. Whole-body motion planning. In *Humanoid Robotics: A Reference*, pages 1575–1599. Springer, 2019.

[6] P. Letier, T. Siedel, M. Deremetz, E. Pavlovskis, B. Lietaer, K. Nottensteiner, M. A. Roa, J. Sánchez, J. L. Corella, and J. Gancet. HOTDOCK: Design and validation of a new generation of standard robotic interface for on-orbit servicing. In *Int. Astronautical Congress (IAC)*, 2020.

[7] J. Martínez-Moritz, I. Rodríguez, K. Nottensteiner, J. Lutze, P. Lehner, and M. A. Roa. Hybrid planning system for in-space robotic assembly of telescopes using segmented mirror tiles. In *IEEE Aerospace Conf.*, 2021.

[8] I. Rodríguez, A. Bauer, K. Nottensteiner, D. Leidner, G. Grunwald, and M. A Roa. Autonomous robot planning system for in-space assembly of reconfigurable structures. In *IEEE Aerospace Conf.*, 2021.

[9] M. Deremetz, P. Letier, G. Grunwald, M. A. Roa, B. Brunner, A. Ferrán, M. Szydelko, J. Gancet, and M. Ilzkovitz. MOSAR-WM: Integration and tests results of a relocatable robotic arm demonstrator for future on-orbit servicing missions. In *Int. Astronautical Congress (IAC)*, 2021.

[10] S. Traversaro, D. Pucci, and F. Nori. A unified view of the equations of motion used for control design of humanoid robots. *Online*, 2017.

[11] M. Deremetz., G. Grunwald, F. Cavenago, M. A. Roa, M. De Stefano, H. Mishra, M. Reiner, S. Govindaraj, A. But, I. Sanz Nieto, J. Gancet, P. Letier, M. Ilzkovitz, L. Gerdes, and M. Zwick. Concept of operations and preliminary design of a modular multi-arm robot using standard interconnects for on-orbit large assembly. In *Int. Astronautical Congress (IAC)*, 2021.