ON-GROUND KINEMATIC SIMULATOR OF THE MOTION SUSPENSION SYSTEM FOR SPACE MANIPULATOR TESTING

Ferdinand Elhardt[1], **Andreas Stemmer**[1], **Peter Lehner**[1], **Marco De Stefano**[1], **Anton Shu**[1], **and Maximo A. Roa**[1]

[1]*Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany, name.surname@dlr.de*

ABSTRACT

This work presents kinematic planning for a cable-driven parallel robot (CDPR) that support a robotic manipulator by usage of a single collision body that represents the CDPR. Our approach has the benefit of computationally efficient planning and compatibility with established path planners. CDPR are used in ground-based verification and validation of space robots to partially offload the added gravitation loads of the ground environment. The German Aerospace Center (DLR) has developed the space robotic manipulator CAESAR which is tested with the Motion Suspension System (MSS). The kinematic complexity of this coupled robotic system demands careful analysis of possible collisions between the manipulator, the MSS cables, and laboratory elements before executing any motions. This work presents the design, implementation, and demonstration of a kinematic simulator for analyzing the workspace envelope of the MSS by considering cable forces and cable-environment collisions. A method for using the Unified Robot Description Format (URDF) for parallel kinematics is proposed and a heuristic approach is employed for cable force computation. The implementation allows a seamless integration in existing collision framework. The simulator demonstrated practical value by identifying improved laboratory configurations, supporting reconfiguration of on-ground space manipulator tests.

Key words: space robotics, gravity compensation, cable-driven parallel robot, gravity offloading, motion suspension system, collision detection, workspace analysis, kinematic simulation.

1. INTRODUCTION

As space missions become more sophisticated, robotic manipulators play increasingly critical roles in In-Space Operations and Services (ISOS) tasks [Pap+21; EC25; MMS82]. Already today, robotic manipulators such as the Canadarm2 support docking maneuvers, assembly tasks, and maintenance operations [GS02]. For this reason, the DLR has developed the three-meters space robot arm *CAESAR* [Bey+18; Elh+24] with force/torque control. As this robot is designed for zero gravity, it cannot support its own weight on ground. Thus, a gravity compensation system is required for on-ground tests.

Verification and validation by on-ground system test presents a fundamental challenge [De +21]. For satellite servicing missions [Roa+24], ground-based testing additionally encompasses physical interactions such as grasping and transferring objects, conducted in laboratory testing facilities. To simulate a relative motion between servicer and client [Roa+24], a movable mockup satellite can implemented. Hereby, orbital and most planetary robotic arms are designed to operate in zero or low gravity, but must be tested under the influence of Earth's gravity. Furthermore, robotic arms are usually limited in the torque necessary to move on ground, i.e., they cannot withstand their own weight in Earth's gravity [Pap+21].

For that reason, many space manipulators need to be supported during on-ground tests to compensate for the gravity influence. The *Institute of Robotics and Mechatronics* at DLR and the *Chair of Mechatronics* at the *University of Duisburg-Essen* have developed the MSS [Elh+23a], shown in Figure 1.

The MSS is a 3-degree of freedom (dof) CDPR that is designed for compensating the weight of a space robotic arm system. The guiding principle of the design is to reduce robot joint load caused by the gravity by suspending it. It is designed for enabling the space manipulator to move in a 6 degrees of freedom work envelope with high observation capability and usability [Elh+23b]. A CDPR is based on cables connected to a mobile platform (point D) that can be spatially moved. The cables are guided by pulleys and actuated by winches. This system can be commanded by a suitable algorithm that enables the computation of suspension forces by formulating an optimization problem designed to minimize the joint torques of the space robotic arm [De +23].

The coupling interface at point E connects the MSS cable connection point with the space manipulator. At this point, the MSS applies the suspension force which results in supporting the space manipulator's joints. The coupling interface is equipped with two passive joints, marked as red circles in Figure 1.

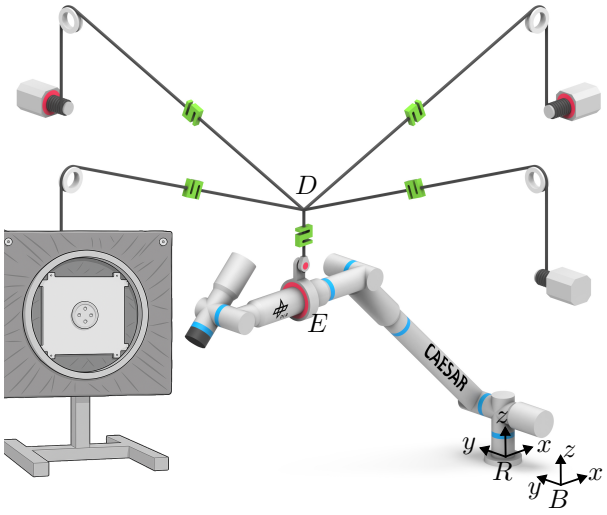

Figure 1. The MSS is a CDPR to support a space manipulators under gravity, here the DLR CAESAR. The laboratory frame is represented with B and the robot base with R.

The problem statement is formulated in the analysis of the available workspace envelope of the MSS. This is important because the workspace of the MSS defines the maximum workspace of the space manipulator during on-ground tests. Firstly, the maximum and minimum allowed cable forces limit the workspace of the MSS [PBM09] which has been analyzed in our previous work [Elh+23a]. Secondly, collisions of the MSS cables with the static laboratory environment must be avoided, which has not been analyzed yet for the MSS. While the existing collision detection tool of the space manipulator accounts for environment-manipulator and cable-manipulator collisions, it does not support cable-environment collisions of CDPR due to its parallel kinematic structure. However, the MSS collision model shall be considered in the existing space manipulator collision detection tool for its trajectory planning.

Hereby, the physical environment of the laboratory, including mockup satellites and test fixtures, forms obstacles that pose a risk of unplanned collisions in the cluttered environment. Since the MSS cables form direct lines between ceiling-mounted pulleys and the manipulator, they move dynamically with the arm, creating a large, static collision volume that restrict the usable workspace. Without accurate prediction of potential collisions before trajectory execution, physical testing becomes inefficient, frequent manual interventions are required and the risk of equipment damage increases.

Therefore, an integrated kinematic simulation tool that models the non-feasible volume of MSS is needed. To interface this simulator with the existing manipulator collision detection tool, three-dimensional non-feasibility maps are created which can be imported into the manipulator collision checker as static collision meshes.

1.1. State of the Art

Collision detection algorithms are a fundamental component of robotic systems, enabling safe navigation, manipulation, and interaction in complex environments. The field is distributed into three critical domains: self-collision detection, environment collision detection, and trajectory verification. Fundamental geometric algorithms have undergone significant improvements in the last few years, particularly through the integration of GPU acceleration and machine learning techniques. Latest, the transition from purely geometric algorithms to hybrid approaches combines classical methods with machine learning [Wel13].

Collision algorithms for CDPR [NG15] are separated into four distinct categories: cable-cable [Mer04; Oti+09], cable-platform [Bur+19], platform-environment, and cable-environment [MCC18]. While algorithms for discrete checking methods are simpler and computationally efficient, modern collision checking methods move towards continuous approaches, which provide exact time-of-impact computation. In this context, Bury et al. developed a specialized collision detection approach specifically designed for a robotic arm mounted on a CDPR platform [Bur+19]. However, this method requires a complete integration of the parallel kinematics into the collision detection algorithm, which presents significant implementation challenges when existing collision detection infrastructure is already established.

1.2. Contribution

We adopt an approach that leverages the existing manipulator collision checker without requiring modifications to its core architecture. It was developed to analyze the workspace envelope of the MSS by considering cable forces constraints and cable-environment collisions. The CDPR is modeled in the URDF format and a feasibility algorithm generates a static STL collision map that represents the non-feasible regions of the MSS. These maps can be imported as static collision objects into the existing manipulator collision checker software. This approach maintains the integrity of the established collision detection system while extending its capabilities to account for MSS constraints. Additionally, the simulator enables identification of collision-prone elements, supports reorganization of the laboratory environment, and provides a fast, visual validation method for robotic test scenarios – prior to hardware execution.

The remainder of this paper is organized as follows: Section 2 presents the URDF-based modeling of the MSS and laboratory environment. Section 3 describes the kinematic simulation method including inverse kinematics, collision detection, and cable force analysis. Section 4 details the Python implementation and STL map generation. Section 5 demonstrates workspace analysis results and laboratory reconfiguration outcomes. Section 6 dis-

cusses limitations and comparisons with existing methods, and Section 7 concludes with future work directions.

2. MODELING

This section introduced on how the MSS and robotic manipulator are modeled for this work.

In order to compute the MSS workspace, a suitable modeling of the system needs to be established. Modeling of the kinematic chain of robotic systems are usually performed in the URDF format [QGS15]. The URDF is an XML-based specification used to describe the kinematic and dynamic properties of robotic systems, including joint configurations, link geometries, inertial properties, and visual representations. URDF files define the structural relationships between robot components through a tree-like hierarchy of links connected by joints. This allows modeling of robot morphology and physical characteristics for simulation and control applications.

The primary advantages of URDF include its widespread adoption across the robotics community, for example a native integration with the Robot Operating System (ROS) ecosystem. While it is machine-readable, it can easily be interpreted by the user due to its XML structure. This allows manual editing and version control. However, URDF has limitations including its restriction to tree topologies that cannot represent closed kinematic chains, lack of support for advanced features such as flexible bodies or complex contact models, and verbose syntax that can become unwieldy for complex robotic systems with many degrees of freedom.

2.1. Modeling of the MSS

The MSS is modeled using the URDF format to maintain compatibility with standard robotic simulation environments. However, URDF is inherently limited to tree topologies and cannot represent the closed kinematic chains characteristic of parallel robots such as the MSS. To overcome this limitation, we propose a modified approach that reverses the conventional kinematic chain representation. While traditional serial robot URDF structures originate from the robot's base and extend outward, the MSS URDF is structured in reverse, with the kinematic tree extending from point D (the MSS end effector) outward to the ceiling-mounted pulleys. This reverse approach transforms the closed parallel kinematic chain into multiple open kinematic branches that can be represented within the URDF tree structure. The modeling approach incorporates the following assumptions: cables are considered massless and maintain straight-line geometry between attachment points, while cable sagging and dynamic swinging effects are neglected for computational simplicity.

The proposed URDF structure of the MSS is illustrated as a two-dimensional simplified form in Figure 2. Three

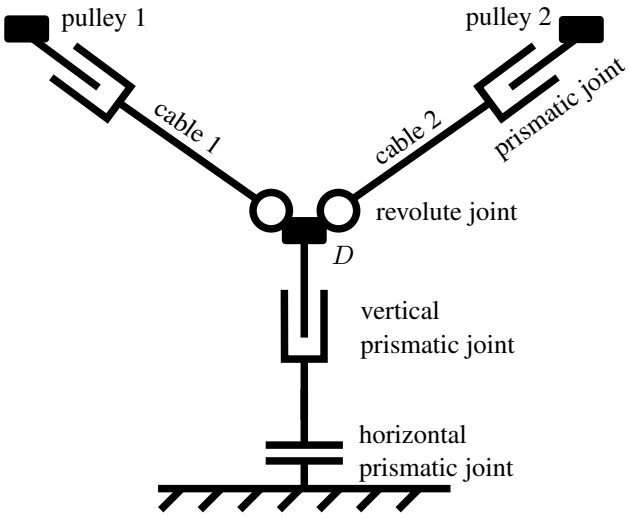

Figure 2. Two-dimensional illustration of the URDF modeling of the MSS.

virtual linear joints connect the MSS cable connection link D to the fixed environment, enabling three dof positioning. The cable connection link is modeled as a sphere-shaped object. Each of the four cables are connected to point D with two revolute joints, respectively. One for the cable's rotation in the horizontal xy plane, the other for its inclination. The cables themselves are modeled as cylinder-shaped links between point D and the pulleys. To change the length of the cables, they are connected to the pulleys by prismatic joints. This allows to move point D closer and further from the individual pulleys.

However, the length of the collision mesh of the cables are fixed, so it is assumed that parts of the rope mesh protrude behind the pulley. Although this does not correspond to reality, it serves the purpose since we do not expect any collisions in this area.

This results in a four-branch URDF tree with freely movable tails. Finally, the pulley's locations must be fixed using positional constraints. However, URDF does not allow this constraint. This is why the inverse kinematic needs to be computed externally (see Section 3.1).

2.2. Modeling of the Robotic Manipulator and Environment

The robotic manipulator is already modeled in the existing collision checker which includes a representation of the environment. These objects are usually exported from the CAD program, represented as mesh files, and manually placed at the corresponding location. The laboratory walls are neglected.

3. METHOD

This section describes the approach used in this work for the kinematic simulator. First, the inverse kinematics is described, then the collision checker is introduced. The last subsection introduces the cable force check. Hereby, the simulator consists of an inverse kinematic method that takes the URDF as input described in detail in the first subsection. Collision and cable force checking decides if a single position of point D is feasible or not described in the preceding two subsections. The last subsection describes an extended workspace analysis creating non-feasibility maps of the MSS.

3.1. Inverse Kinematics

The inverse kinematics computation is based on the MSS modeling described in the previous section. The simulator accepts the space manipulator position at point E as input and determines the corresponding MSS end effector position at point D using a predefined vertical offset of fixed length between the coupling interface and the cable attachment point.

Given the parallel kinematic structure of the MSS, the inverse kinematics solution is computed according to [Lan84]. For each cable i, the required length l_i is calculated by

$$l_i = ||\boldsymbol{p}_{D,i} - \boldsymbol{p}_{pulley,i}|| \tag{1}$$

which represents the Euclidean distance between point D position $\boldsymbol{p}_{D,i} \in \mathbb{R}^3$ and the corresponding pulley position $\boldsymbol{p}_{pulley,i} \in \mathbb{R}^3$. The cable direction vectors are simultaneously computed for subsequent force analysis. This direct geometric approach provides the joint angles and cable lengths according to the URDF specification for any given end effector position within the workspace.

3.2. Collision Check

Now, we have the MSS description in the URDF, the collision meshes, and the joint angles/lengths. This serves as input for the collision check. The algorithm checks each potential collision by using the Flexible Collision Library (FCL). The library implements hierarchical, convex polytope collision checks based on the Gilbert-Johnson-Keerthi and Expanding Polytope Algorithms [PCM12]. It results in a binary information per space manipulator position E if there is a collision or not.

3.3. Cable Force Check

The force in each cables of CDPR need to be in the range $f_{min} \leq f_{\text{ropes},i} \leq f_{max}$ [Bou24]. The lower limit f_{min} keeps the cable tensed and prevent it from sagging. The upper limit f_{max} comes from the safety parameters based on available drive torques and the strength of the mechanical parts. The cable forces introduce a limitation of the workspace because certain points in the workspace are not feasible within the allowed cable force range. For example, the cable forces increase when point E is gets higher until it reaches f_{max}.

The exact determination of cable forces of CDPR is an over-constrained problem in most cases [Pot18]. This is why we use a simplified heuristic approach. A matrix $\boldsymbol{v}_{\text{ropes}} \in \mathbb{R}^{ix3}$ contain the normalized i cable vectors. The cable force vector $\boldsymbol{f}_{\text{ropes}} \in \mathbb{R}^i$ is computed by the equation

$$\begin{aligned} \mathcal{P}(p_i): \quad & \text{find } \boldsymbol{f}_{\text{ropes}} \\ & \text{subject to:} \\ & \boldsymbol{v}_{\text{ropes}}(p_i) \cdot \boldsymbol{f}_{\text{ropes}} = {}_B\boldsymbol{f}_C \\ & f_{\min} \leq f_j \leq f_{\max}, \quad \forall j = 1, 2, \ldots, m \end{aligned} \tag{2}$$

with the suspension force

$${}_B\boldsymbol{f}_C = \begin{bmatrix} 0 \\ 0 \\ f_z \end{bmatrix}. \tag{3}$$

Hereby, f_z is given by the space manipulator. If a solution can be found then the cable forces are feasible for this point.

3.4. Workspace Analysis

With the previously introduced methods, the feasibility of individual endeffector positions can be checked. The goal of the workspace analysis is to apply the method on a workspace volume to create virtual collision objects that represents the non-feasible workspace volumes. For this, the continuous workspace of the MSS in the Cartesian task space is discretized into an equally spaced grid of candidate points representing positions of point D. The initial workspace is defined as a rectangular cuboid with dimensions $l \times w \times h$ (length, width, height)

$$\mathcal{W}_0 = \{\boldsymbol{p}_{D,i} \in \mathbb{R}^3 \mid i = 1, 2, \ldots, N\} \tag{4}$$

where each point $\boldsymbol{p}_{D,i}$ represents a potential D position, and N is the total number of discretization points. The points are distributed equidistantly with uniform spacing δ throughout the workspace volume.

The workspace analysis creates two set of non-feasible locations: $\mathcal{W}_c$ for the collision constraint and $\mathcal{W}_f$ for the

cable force constraint. The non-feasible workspace due to collision is

$$\mathcal{W}_c = \{p_{D,i} \in \mathcal{W}_0 \mid \mathcal{C}(p_i) = \text{false}\} \tag{5}$$

where $\mathcal{C}(p_i)$ is a Boolean function provided by external collision detection software that returns true if point p_i is collision-free.

The non-feasible workspace due to the cable force constraints is created by

$$\mathcal{W}_f = \{p_i \in \mathcal{W}_0 \mid \nexists \boldsymbol{f}_{\text{ropes}} \text{ such that } \mathcal{P}(p_i) \text{ is feasible}\} \tag{6}$$

considering function $\mathcal{P}(p_i)$ defined in Equation 2. This method can be improved by applying the Bisection Method along the z-axis (vertical direction), which more efficiently identifies the boundary between collision and non-collision regions. While this approach reduces computational effort, it assumes that collision volumes are convex—an assumption justified by our knowledge of the collision objects' geometry

Based on the resulting point cloud $\mathcal{W}_f$, a volume can be created by connecting the non-feasible points. This volume is exported as STL object.

4. IMPLEMENTATION

The kinematic simulation tool is implemented as a modular Python framework consisting of three main components: the *MSS helper* module for inverse kinematics, the FCL-based *collision detection module*, and the *cable force analysis module*. The implementation follows the workflow shown in Figure 3.

The MSS modeling module creates the URDF of the MSS. The MSS helper module serves as the core inverse kinematics solver for the parallel kinematic system. Key implementation details include:

Input Processing The module accepts the manipulator position at point E and computes the corresponding MSS end effector position at point D using the predefined vertical offset.

Pulley Constraint Integration Ceiling-mounted pulley positions are incorporated as fixed constraints in the kinematic chain, with cable routing calculated as direct line segments between pulleys and attachment points.

URDF Compatibility Joint angles and cable lengths are computed in accordance with the provided URDF specification of the MSS.

Parallel Kinematics Solver The implementation uses the straightforward nature of parallel kinematic inverse solutions, computing cable lengths directly from geometric relationships by using the standard *math* Python library without iterative numerical methods.

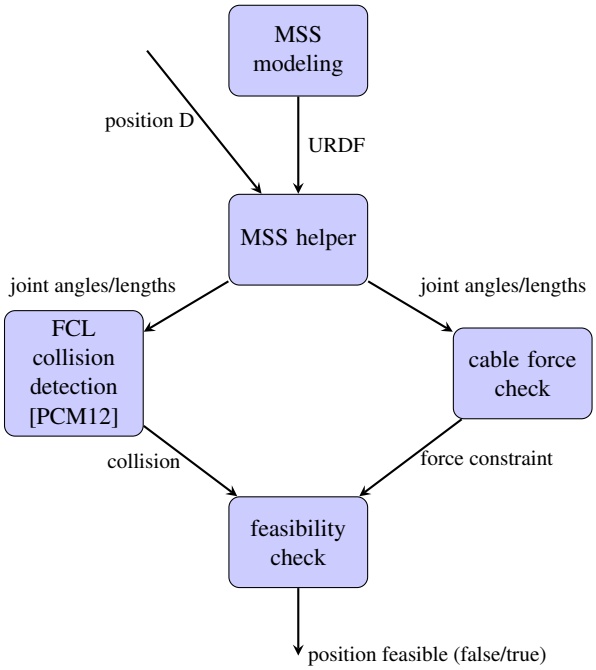

Figure 3. Workflow diagram of the MSS kinematic simulation tool showing the integration of modeling, inverse kinematics, collision detection, and force analysis components

For collision detection, FCL is used. It is integrated through Python bindings. The cable geometries are imported as cylindrical collision meshes. Laboratory environment elements, including mockup satellites and test fixtures, are loaded as static collision meshes from CAD-derived formats.

To estimate the cable forces, the cable direction matrix $\boldsymbol{v}_{\text{cables}}$ is assembled dynamically based on current cable configurations, with each row containing the normalized vector from pulley to end effector.

The linear programming approach *lsq_linear* of the Python package *scipy.optimize*[1] is used to solve the force distribution problem within the specified force bounds $[\boldsymbol{f}_{min}, \boldsymbol{f}_{max}]$. The solver returns a binary feasibility flag indicating whether a valid force distribution exists for the given configuration.

4.1. Workspace Discretization and Map Generation

The workspace is discretized using a *StructuredGrid* in the Python package *pyvista*[2]. Grid boundaries are defined based on physical laboratory constraints and MSS mechanical limits.

For each discrete point in the workspace, the feasibility assessment combines:

[1] `docs.scipy.org`
[2] `docs.pyvista.org`

1. Kinematic Feasibility: Verification that the *MSS helper* can compute valid joint angles/lengths
2. Collision Feasibility: Confirmation that no cable-environment collisions occur
3. Force Feasibility: Validation that cable forces remain within acceptable bounds

The single-point method returns a binary feasibility result for each tested position. A Binary search algorithm for the z axis is implemented for searching the boundary between non-feasible and feasible volume. Hereby, a convex non-feasibly volume is assumed.

Two distinct 3D non-feasibility maps are generated: the collision map indicating regions where cable-environment collisions occur and the cable force constraint map indicating regions where cable force limits are violated. The generated feasibility maps are exported in STL format, compatible with existing manipulator collision checkers for seamless integration with trajectory planning systems. Hereby, the collision object should be only applied on the position of point D.

4.2. Visualization

Apart as using the individual collision checking for trajectory planning and using the resulting non-feasibility maps in the manipulator collision checker, the non-feasibility can be visualized for manual investigation. Hereby, any viewer for STL files can be used. This supports reorganization of the laboratory environment by moving large collision elements where less impact of the MSS is expected.

5. RESULTS

For the demonstration of the method, we perform an analysis of the MSS workspace within the on-ground testing facility. The goal is to identify non-reachable regions and possible improvement ideas for the laboratory arrangement.

The cables are considered with a diameter of $20\,\mathrm{mm}$. The cable connection link D is modeled as $50\,\mathrm{mm}$ sphere. The initial workspace $\mathcal{W}_0$ defined as a rectangular cuboid are defined with the size $5.5\,\mathrm{m}\times5\,\mathrm{m}$ and the height $2.5\,\mathrm{m}$. The discretization in xy was performed with $\delta_{xy} = 0.25\,\mathrm{m}$ and in z with $\delta_z = 0.05\,\mathrm{m}$ leading to $|\mathcal{W}_0|= 24\,633$ number of elements in the initial workspace.

Figure 4 shows the non-feasibility map $\mathcal{W}_c$ based on MSS cable collisions. The map reveals that a substantial volume in the lower region of the theoretical workspace is blocked, severely restricting the space manipulator's range of motion. Figure 5 contextualizes this non-feasibility map within the laboratory environment,

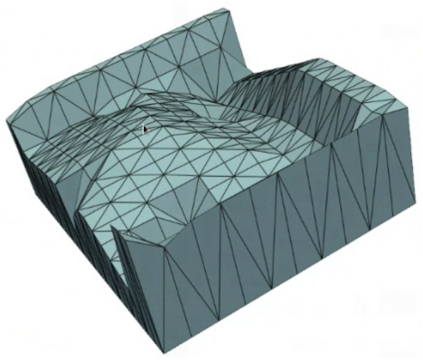

Figure 4. Non-feasible volume map $\mathcal{W}_c$ based on MSS cable-environment collisions.

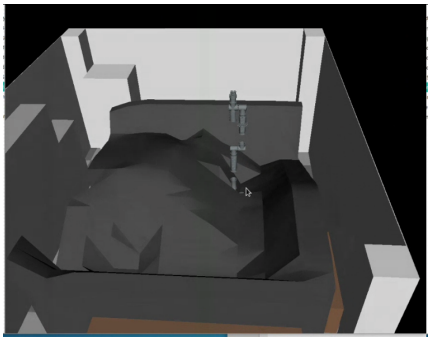

Figure 5. The MSS collision non-feasible map imported in the laboratory setup with illustrated CAESAR space manipulator.

providing a clear sense of scale and dimensional constraints. This illustration helps to understand the possible workspace of the MSS by virtually rearranging collision objects and comparing the resulting non-feasible volume.

Figure 6 shows the cable force constraint map defining set $\mathcal{W}_f$. This map reveals that a substantial region in the upper area of the laboratory environment is blocked, extending to the lateral boundaries of the workspace. These force-based constraints are fundamental limitations of cable-driven systems due to force and tension limits that cannot be modified, but must be considered during motion planning. For instance, trajectories that require the MSS to reach high positions are not feasible due to these constraints.

The results were used to rearrange the laboratory environment. In this example, the mockup satellite was placed to a different location which lead to a strongly increased workspace of the MSS.

6. DISCUSSION

The cable force computation used in this work uses a heuristic approach that provides estimates rather than exact cable forces. When checking the upper and lower

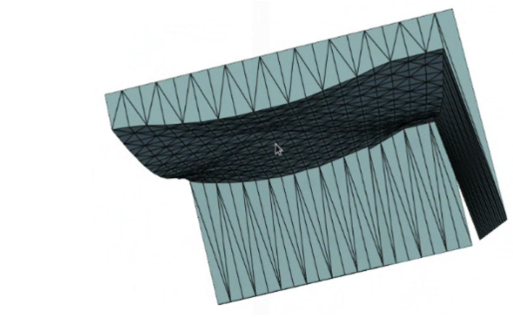

Figure 6. Non-feasible volume map $\mathcal{W}_f$ based on the MSS cable force constraints.

bound condition, this can result in both false-positive and false-negative workspace points. However, this limitation has minimal impact on the intended application, as the simulator is designed primarily for non-accurate visualization and preliminary workspace assessment rather than precise trajectory planning. This aligns with the purpose of a rapid workspace visualization for laboratory reconfiguration and initial test scenario evaluation.

The current implementation focuses specifically on cable-environment collision detection, which integrates into the manipulator-environment collisions checking by creating static collision meshes. Manipulator-cable collisions are considered in the manipulator collision detector by using a static approximation of the cable volume. This separation approach allows using a standard collision checker which is naturally not compatible for parallel kinematics.

The discrete point-based workspace analysis approach, while the computational effort is easily adaptable, presents limitations compared to continuous collision detection methods. The discrete approach may miss potential collision scenarios that occur between sampled points, whereas continuous methods provide exact time-of-impact computation and more comprehensive collision coverage.

The simulator has demonstrated a strong practical value in improving the laboratory configurations to maximize the available workspace. A notable success was the identification of an improved location for the mockup satellite through trial-and-error exploration within the simulation environment, resulting in a substantial increase in the MSS workspace envelope. Apart from this specific case, the tool helped by the placement of manipulation objects in areas with good workspace coverage. This capability enables efficient laboratory setup optimization without requiring physical rearrangement trials, reducing setup time and improving test scenario feasibility.

The simulator's architecture emphasizes modularity through the adoption of the URDF. This allows integration from CAD software to the simulation environment. However, due to the limitations of the URDF format, the custom solver for the inverse kinematics is necessary to close the kinematic chain of the CDPR.

Several technical limitations present opportunities for future development. The integration of all collision types (cable-environment, manipulator-cable, and manipulator-environment) into a unified collision detection framework would provide a better workspace analysis. Improving the cable force computation from the current heuristic approach to a more exact method would enhance the accuracy of workspace predictions. Additionally, transitioning from the discrete point-based approach to continuous collision detection methods would provide more accurate collision predictions and eliminate potential gaps in coverage. Despite these limitations, the current simulator successfully fulfills its primary objective of providing rapid workspace visualization and supporting laboratory reconfiguration decisions for MSS-supported space manipulator testing.

7. CONCLUSION

This work presents a kinematic simulation tool for analyzing the workspace envelope of the MSS used in on-ground space manipulator testing. The developed simulator addresses the challenge of integrating CDPR collision detection with existing space manipulator collision checkers without requiring modifications to established software infrastructure. The key achievements of this work include the development of a URDF-based modeling approach that overcomes the limitations of tree-structured representations for parallel kinematic systems through reverse kinematic chain formulation. The implementation of an integrated Python framework combining inverse kinematics computation, FCL-based collision detection, and heuristic cable force analysis enables a workspace feasibility assessment. The generation of STL-format non-feasibility maps provides seamless integration with existing manipulator collision detection tools.

The practical validation demonstrates the simulator's effectiveness in laboratory reconfiguration scenarios, where workspace analysis led to substantial improvements in available MSS workspace through strategic repositioning of mockup satellites and test fixtures.

REFERENCES

[Pap+21] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello. "Robotic Manipulation and Capture in Space: A Survey". In: *Frontiers in Robotics and AI*. 2021. DOI: 10.3389/frobt.2021.686723.

[EC25] EC. *European Commission: The Space Act: Regulation of the European Parliament and of the Council on the Safety, Resilience and Sustainability of Space Activities in the Union*. 2025.

[MMS82] R. H. Miller, M. L. Minsky, and D. B. S. Smith. *Space Applications of Automation, Robotics and Machine Intelligence Systems (ARAMIS). Volume 4: Supplement, Appendix 4.3: Candidate ARAMIS Capabilities*. Tech. rep. NAS 1.26:162083. NASA, 1982.

[GS02] G. Gibbs and S. Sachdev. "Canada and the International Space Station Program: Overview and Status". In: *Acta Astronautica* 51.1 (2002), pp. 591–600. DOI: 10.1016/S0094-5765(02)00077-2.

[Bey+18] A. Beyer, G. Grunwald, M. Heumos, M. Schedl, R. Bayer, W. Bertleff, B. Brunner, R. Burger, J. Butterfaß, R. Gruber, T. Gumpert, F. Hacker, E. Krämer, M. Maier, S. Moser, J. Reill, M. A. Roa, H.-J. Sedlmayr, N. Seitz, and A. Albu-Schäffer. "CAESAR: Space Robotics Technology for Assembly, Maintenance, and Repair". In: *Proc. Int. Astronautical Congress*. Bremen, 2018.

[Elh+24] F. Elhardt, M. Ekal, M. A. Roa, R. Bayer, A. Beyer, B. Brunner, M. De Stefano, S. Moser, Schedl Manfred, H.-J. Sedlmayr, Stelzer Martin, A. Stemmer, B. Thomas, R. Burger, J. Butterfass, T. Gumpert, F. Hacker, E. Krämer, J. Reill, N. Seitz, T. Wimmer, R. Boumann, T. Bruckmann, R. Heidel, P. Lemmen, W. Bertleff, J. Heindl, D. Reintsema, F. Steinmetz, G. Grunwald, G. Hirzinger, K. Landzettel, and A. Albu-Schäffer. "Video: DLR's Advancements in Space Robotic Manipulation". In: *40th Anniversary of the IEEE Conference on Robotics and Automation*. Rotterdam, 2024.

[De +21] M. De Stefano, H. Mishra, A. M. Giordano, R. Lampariello, and C. Ott. "A Relative Dynamics Formulation for Hardware-in-the-Loop Simulation of On-Orbit Robotic Missions". In: *IEEE Robotics and Automation Letters* (2021). DOI: 10.1109/LRA.2021.3064510.

[Roa+24] M. A. Roa, A. Beyer, I. Rodríguez, M. Stelzer, M. De Stefano, J.-P. Lutze, H. Mishra, F. Elhardt, G. Grunwald, V. Dubanchet, H. Renault, F. Niemeijer, C. Jacopini, P. Atinsounon, S. Behar-Lafenetre, J. A. Béjar-Romero, S. Torralbo-Dezainde, M. Alonso-Alonso, A. Jakubiec, Ł. Kozłowski, A. Lukasiak, A. Merlo, M. Lapolla, and K. N. Gregertsen. "EROSS: In-Orbit Demonstration of European Robotic Orbital Support Services". In: *2024 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, 2024, pp. 1–9. DOI: 10.1109/AERO58975.2024.10521010.

[Elh+23a] F. Elhardt, R. Boumann, M. De Stefano, R. Heidel, P. Lemmen, M. Heumos, C. Jeziorek, M. A. Roa, M. Schedl, and T. Bruckmann. "The Motion Suspension System – MSS: A Cable-Driven System for On-Ground Tests of Space Robots". In: *Advances in Mechanism and Machine Science*. Ed. by M. Okada. Vol. 148. Cham: Springer Nature, 2023, pp. 379–388. DOI: 10.1007/978-3-031-45770-8_38.

[Elh+23b] F. Elhardt, R. Boumann, M. De Stefano, R. Heidel, P. Lemmen, M. Heumos, C. Jeziorek, M. Roa, M. Schedl, and T. Bruckmann. "System Requirements Elicitation and Conceptualization for a Novel Space Robot Suspension System". In: *Proc. Symp. on Advanced Space Technologies in Robotics and Automation (ASTRA)*. Leiden, Niederlande, 2023.

[De +23] M. De Stefano, R. Vijayan, A. Stemmer, F. Elhardt, and C. Ott. "A Gravity Compensation Strategy for On-ground Validation of Orbital Manipulators". In: *Proc. IEEE Int. Conf. on Robotics and Automation*. London, UK: IEEE, 2023, pp. 11859–11865. DOI: 10.1109/ICRA48891.2023.10161480.

[PBM09] A. Pott, T. Bruckmann, and L. Mikelsons. "Closed-Form Force Distribution for Parallel Wire Robots". In: *Computational Kinematics*. Ed. by A. Kecskeméthy and A. Müller. Springer, 2009, pp. 25–34. DOI: 10.1007/978-3-642-01947-0_4.

[Wel13] R. Weller. "A Brief Overview of Collision Detection". In: *New Geometric Data Structures for Collision Detection and Haptics*. Ed. by R. Weller. Heidelberg: Springer International Publishing, 2013, pp. 9–46. DOI: 10.1007/978-3-319-01020-5_2.

[NG15] D. Q. Nguyen and M. Gouttefarde. "On the Improvement of Cable Collision Detection Algorithms". In: *Cable-Driven Parallel Robots*. Ed. by A. Pott and T. Bruckmann. Cham: Springer International Publishing, 2015, pp. 29–40. DOI: 10.1007/978-3-319-09489-2_3.

[Mer04] J.-P. Merlet. "Analysis of the Influence of Wires Interference on the Workspace of Wire Robots". In: *On Advances in Robot Kinematics*. Ed. by J. Lenarčič and C. Galletti. Dordrecht: Springer Netherlands, 2004, pp. 211–218. DOI: 10.1007/978-1-4020-2249-4_23.

[Oti+09] M. J.-D. Otis, S. Perreault, T.-L. Nguyen-Dang, P. Lambert, M. Gouttefarde, D. Laurendeau, and C. Gosselin. "Determination and Management of Cable Interferences Between Two 6-DOF Foot Platforms in a Cable-Driven Locomotion Interface". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 39.3 (2009), pp. 528–544. DOI: 10.1109/TSMCA.2009.2013188.

[Bur+19] D. Bury, J.-B. Izard, M. Gouttefarde, and F. Lamiraux. “Continuous Collision Detection for a Robotic Arm Mounted on a Cable-Driven Parallel Robot”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 8097–8102. DOI: 10.1109/IROS40897.2019.8967836.

[MCC18] A. Martin, S. Caro, and P. Cardou. “Geometric Determination of the Cable-Cylinder Interference Regions in the Workspace of a Cable-Driven Parallel Robot”. In: *Cable-Driven Parallel Robots*. Ed. by C. Gosselin, P. Cardou, T. Bruckmann, and A. Pott. Cham: Springer International Publishing, 2018, pp. 117–127. DOI: 10.1007/978-3-319-61431-1_11.

[QGS15] M. Quigley, B. Gerkey, and W. D. Smart. *Programming Robots with ROS*. First edition. Beijing Boston Farnham Sebastopol Tokyo: O’Reilly, 2015.

[Lan84] S. E. Landsberger. “Design and Construction of a Cable-Controlled, Parallel Link Manipulator”. Thesis. Massachusetts Institute of Technology, 1984.

[PCM12] J. Pan, S. Chitta, and D. Manocha. “FCL: A General Purpose Library for Collision and Proximity Queries”. In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 3859–3866. DOI: 10.1109/ICRA.2012.6225337.

[Bou24] R. Boumann. “Damage Prevention After Cable Failure in Redundant Parallel Cable Robots”. PhD thesis. DuEPublico: Duisburg-Essen Publications online, University of Duisburg-Essen, Germany, 2024. DOI: 10.17185/DUEPUBLICO/81869.

[Pot18] A. Pott. *Cable-Driven Parallel Robots: Theory and Application*. Springer Tracts in Advanced Robotics. Springer, 2018. DOI: 10.1007/978-3-319-76138-1.