DIGITAL TWIN FOR IN-ORBIT ROBOTIC OPERATIONS: A POWERFUL TOOL FOR MONITORING AND RECOVERY PURPOSES

Isabelle Maroger, Clément Bazerque, and Thierry Germa

Magellium, Ramonville-Saint-Agne, France, isabelle.maroger@magellium.fr

ABSTRACT

In-space robotic operations can revolutionize the way in-orbit missions are approached. Being able to refuel, fix, upgrade or even assemble a satellite paves the way to a sustainable use of Earth's orbits. Achieving such operations is challenging due to the inherent constraints of space operations. One of the main challenges is the simulation to both prepare and monitor in-orbit mission. Indeed, it is essential for the operator to visualize the operations, beforehand, to anticipate potential problems and, during the mission, so that he rapidly spots when an unexpected behaviour occurs and easily understands how to fix the potential issue. A relevant solution to address this problem is to develop a simulated digital twin of the system in orbit. Magellium is currently working on such a tool's development.

Key words: In-obit robotic operations; Simulation; Digital twin.

1. INTRODUCTION

1.1. Context

With the growing number of satellites (whether operational or defunct) orbiting Earth, new challenges have emerged in recent decades. How can we safely de-orbit obsolete satellites? How can we ensure the proper maintenance of those still in service? Robotic in-orbit operations provide promising answers to both issues. Robots could, for example, dismantle satellites that are no longer functional, or integrate new components into active ones to extend their operational lifetime. They could also play a key role in assembling large structures directly in space. A fully automated station for in-orbit experiments can even be considered to replace the International Space Station (ISS) after the end of its operational life in 2030. Despite this potential, such robotic interventions remain rare today because of the significant constraints of space environments. This underlines the importance of developing reliable methods to simulate viable robotic missions from Earth and to supervise their execution once deployed.

To solve this issue, Magellium has developed, for the last 5 years, a framework named Ground station Architecture for In-space Assembly (GAIA). This framework, composed of modular components, aimed to provide a set of simulation-based tools to prepare and monitor in-orbit robotic operations for assembly or servicing by multiple robots.

1.2. GAIA overview

The GAIA framework pursues two main goals. On the one hand, it plays a key role in mission design by validating the Concept of Operations (ConOps) through simulation. On the other hand, it supports the mission execution by supervising operations through a digital twin. Consequently, this framework is relevant at two distinct stages: (i) the ground preparation phase before launch and (ii) the in-orbit activities phase during the mission. In addition, the framework is conceived to be modular and adaptable, making it suitable for a wide variety of space missions. It is based on the ROS2 middleware [1].

As the focus of this manuscript is more on the simulation and digital twin, only a brief description of each component of GAIA is provided below but more details are available in Bazerque et al. [2]. The different Building Blocks (BBs) and their interfaces are presented in Fig.1.

1.2.1. Mission preparation

The main purpose of the Mission Preparation BB is to support the design of the mission scenario. It is structured around three core elements:

- Simulation Framework: This component provides a 3D virtual environment that accurately reproduces mission conditions. Its objective is to act as a test bench, allowing the validation of operational safety before deployment in space. More details about this tool are provided in section 2.
- Robotic Framework: This module contains all the software capabilities needed to interpret and execute sequences of operations. Since the Ground Segment must accurately reflect the Flight Segment, the skill libraries (sequencing, motion planning, perception) must be identical to those onboard. However, they may rely on different middleware. To address this, the Robotic Framework integrates a communication

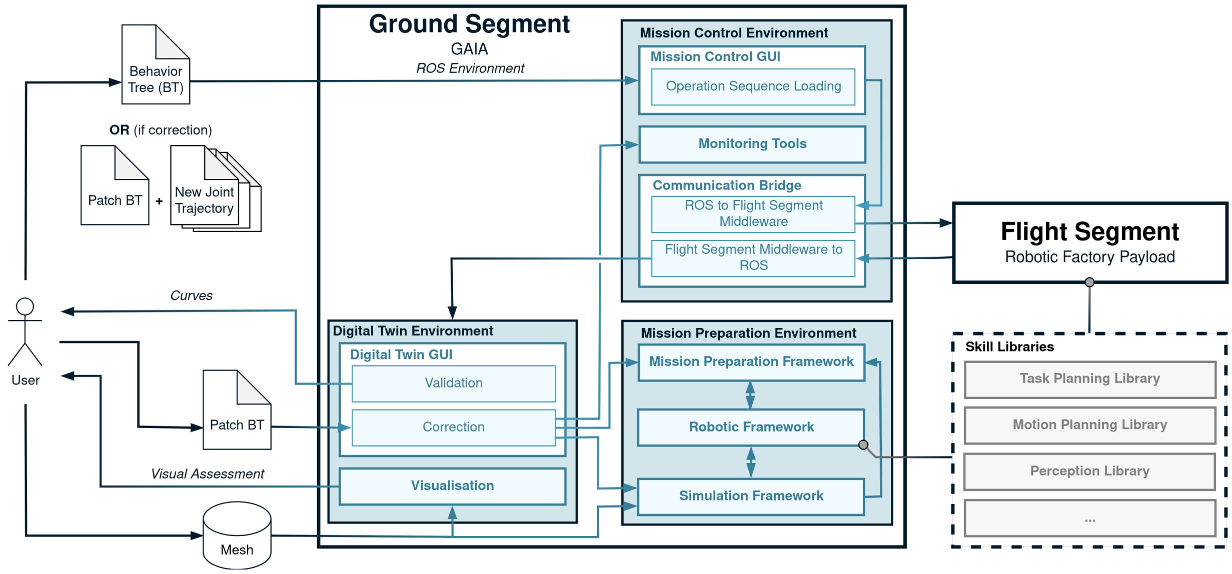

Figure 1: GAIA architecture.

bridge that translates messages between the Ground Segment and Flight Segment middleware formats.

- Mission Preparation Graphical User Interface (GUI): This graphical interface allows the operator to launch and monitor the mission preparation. Through it, the operator can fully control the execution of test sequences—pausing them, running them step by step, or stopping them to load an alternative. The main goal of the Mission Preparation Framework is to equip operators with monitoring and control tools that help them refine and validate an appropriate sequence of operations.

1.2.2. Mission control

The Mission Control BB is the component that enables communication with the Flight Segment (in-orbit or ground-based, real or virtual). The operator provides a sequence of actions (a Behaviour Tree (BT) for example), which the Mission Control BB interprets and translates into unit commands compatible with the Flight Segment middleware, using the same communication bridge as in the Robotic Framework. Once executed, the resulting telemetry is processed and displayed through monitoring tools, such as the status panel and the digital twin, to give the operator a clear overview of system state.

1.2.3. Digital twin

The GAIA framework includes a software component designed to help the operator visualize the execution of the robotic operations at the Flight Segment level and to respond in case of anomalies. This capability is essential for in-orbit demonstrations, where direct visual inspection of the spacecraft is impossible. The Digital Twin BB aims to serve as a highly realistic replica of the onboard environment, supporting both the validation of upcoming operation sequences and the monitoring and analysis

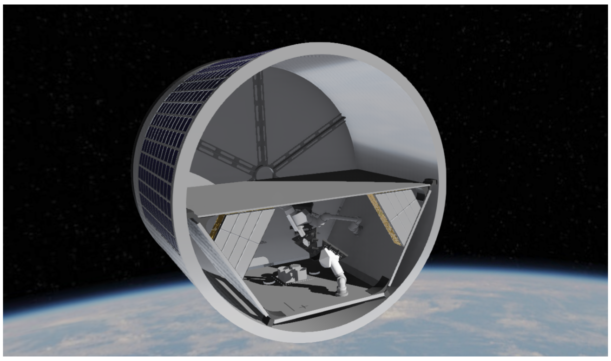

Figure 2: Simulation in Webots of the In-Orbit Demonstration (IOD) in the contexte of the BPI-DEMARLUS project.

of executed ones. To achieve this, the status of each system components is updated based on the received telemetry during each visibility window. Additionally, it offers the possibility to replay operation sequences for post-analysis. In such cases, the digital twin functions primarily as a visualizer rather than a simulator, since no physics engine is needed for replay. More details about the digital twin functioning are available in section 3.

1.3. Past and current projects

GAIA was developed and used over different projects:

- ESA-ISAAC: This project, which ended in February 2025, aimed a ground demonstration of a large structure assembly by a multi-arms robot on a floating bed. Magellium worked on the simulation of the IOD, the mission control GUI and a tool to validate the robot's arms trajectories for the ground demonstration.

- BPI-DEMARLUS [3]: This project, which ended in

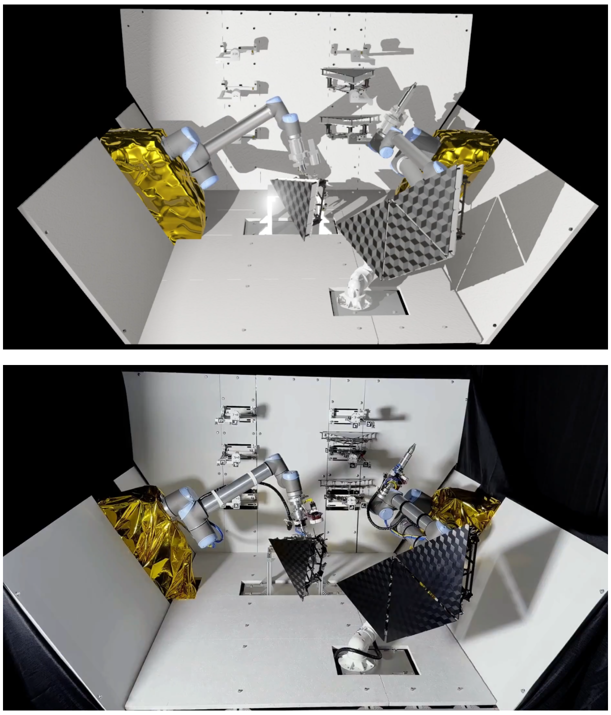

Figure 3: Comparison between the simulation (on the top) and the real demonstrator (on the bottom) in the context of the BPI-DEMARLUS project.

January 2025, targetted the assembly of an antenna composed on 6 tiles using 3 robotic arms on ground. Magellium provided the simulation for both the IOD (see Fig.2) and the final ground demonstration (see Fig.3) along with the motion planning skill, the mission control GUI and a first version of a digital twin.

- HE-EURISE [4]: This project, which is expected to end in December 2025, aimed a ground demonstration of a set a functions useful for in-orbit operations (refuelling, assembly and testing). Magellium designed the simulation, the digital twin and the perception skill as part of this project.

2. SIMULATION FOR MISSION PREPARATION

2.1. Concept

As quickly explained in section 1.2.1, in the simulation, the whole demonstration setup is virtually reconstructed to allow the operator to validate the ConOps and to test the Flight Segment software.
In more detail, the simulation encompasses a set of tools for modelling the key mission elements, both in terms of dynamics and graphics, based on the operator's inputs. For each simulated object, the operator needs to specify the:

- Graphical description: objects geometry and texture, required in particular for vision algorithms.
- Physical description: parameters such as mass, inertia matrix and bounding box, used for contact computations.

The simulation also offers 3D visualization of the scene with interactive tools to navigate, move objects, change the lighting conditions and access or modify the state information about any simulated component. Sensors and actuators can be integrated into mission elements, with support for a wide range of devices: RGB and depth cameras, accelerometers, gyroscopes, force/torque and contact sensors. Actuators include all sort of motorized joints (hinge, slider, ball joints) for robotic arms, lighting systems and locking mechanisms to simulate standard interface behaviours. The simulator can be extended to represent mission-specific sensors and actuators. In addition, analysis tools are provided to log and display telemetry data generated during the simulation.

2.2. Technical choice

From a technical point of view, in the GAIA framework, the simulation is implemented using Webots [5]. Webots is a robot simulation software developed since 1998 by Cyberbotics Ltd.. It is widely used in industry, education, research, and numerous EU-funded projects. In December 2018, it was released as free and open-source software under the Apache 2 license. The simulator provides an extensive library of robots, sensors, and actuators commonly employed in robotics, with the possibility to extend and model additional devices. Webots relies on a custom physics engine built on the Open Dynamics Engine (ODE) and a rendering engine (WREN) based on OpenGL for realistic visualization. Each simulated element is associated with an executable called a "controller" which manages its sensors and actuators. This design emulates the behaviour of hardware driver components in real robotic systems.

2.3. Building process

A simulation building process requires accurate Computer-Aided Design (CAD) models of the scene to emulate.

CAD models are detailed 3D representations created using parametric surfaces, commonly used for precise manufacturing and engineering purposes. The mechanical team in charge of providing the hardware generally always works with this type of format and can provide them to initiate the simulation. However, these models need to be converted into meshes - simplified representations of geometry using polygons or triangles - before they can be used in simulation environments. This conversion is necessary because simulation tools rely on physics engines, which operate with polygonal meshes to compute collisions and dynamics and do not support CAD formats directly. Additionally, meshes are optimized for real-time rendering, enabling efficient memory usage and computational performance, especially in large simulations. It is necessary to simplify the topology of the converted mesh

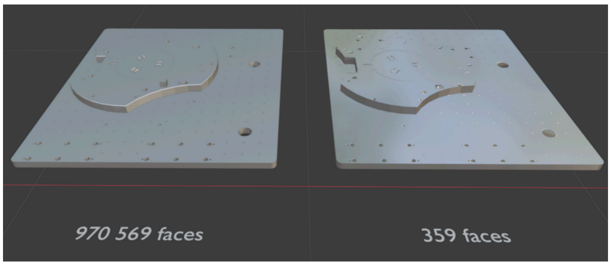

Figure 4: Comparison between the raw mesh (on the left) and the simplified textured mesh (on the right). They look similar but the second mesh has much less faces. To achieve that all the holes in the plate were removed from the geometry and added to the texture using a normal map.

to improve the performances of the simulation. Indeed, it is even recommended to create two simplified versions: one slightly simplified (< 10000 faces) for the visual and one very simplified (< 1000 faces) for the bounding geometry to be used by the physics engine.
Moreover, unlike CAD models, meshes also support textures and materials, allowing for enhanced visual realism. The conversion process often includes simplifying unnecessary details from CAD models, such as internal components or fine features, to ensure the simulation remains computationally feasible while retaining accuracy for physics interactions. This step is essential for creating simulation-ready models that are compatible, efficient, and visually realistic. All this process can be performed using Blender [6]. An example of the transformation from a raw mesh to a simplified texture mesh is shown in Fig.4.

Once the visual and the collision meshes are built, the simulated model can be created. The format in which this model is written depends on the chosen simulator. In Webots, models are described using a hierarchical structure defined by the PROTO file format, which is a reusable and modular way to define objects and their properties. A PROTO file encapsulates the description of a 3D model, including its appearance, physical properties, and behaviour, using a tree-like structure of nodes and fields. The model can also include scripts, sensors, and actuators for interactive or dynamic behavior. Once created, the PROTO model can be inserted into a Webots scene like any other built-in object and it can be further manipulated or nested within other models to build complex simulations.

2.4. Use cases

2.4.1. Sensor and actuator dimensioning

The simulation acts as a versatile sand box platform to test various sensors and actuators characteristics. It can be used for example to model the intrinsics parameters (focal length, Field of View (FoV), resolution, depth of field) of an existing camera and check that the optic suits the requirements for the different steps of the assembly. A

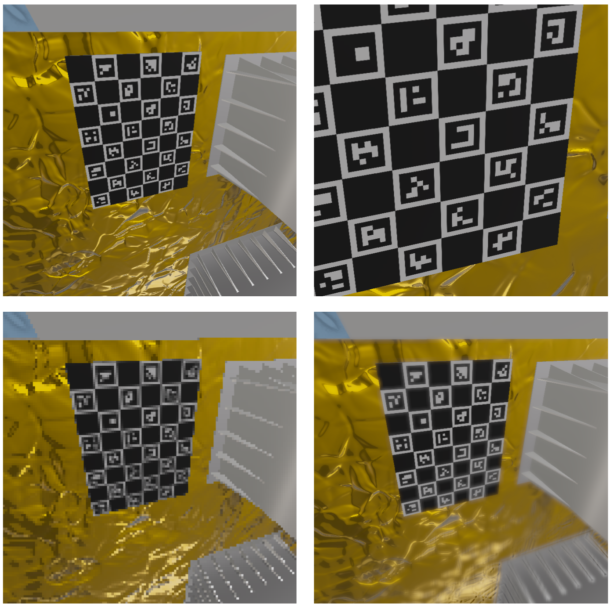

Figure 5: Test of different camera specification (FoV, resolution, focus) in simulation.

comparison of 4 images taken from the same camera pose but with different specifications is presented in Fig.5.

The simulation can also be used to test different camera angles and ensure the intrinsic parameters allow a correct visibility of the target. This is typically used to check if a camera calibration can be done in a constrained workspace such as the one inside a potential spacecraft. An example of such a situation is shown in Fig.6. Another use case for perception testing is the check of the markers visibility. Indeed, the lightning conditions or the markers size and pose can easily be changed in a simulated environment.

Apart from the sensor dimensioning for perception applications, the simulation can also help the operator dimen-

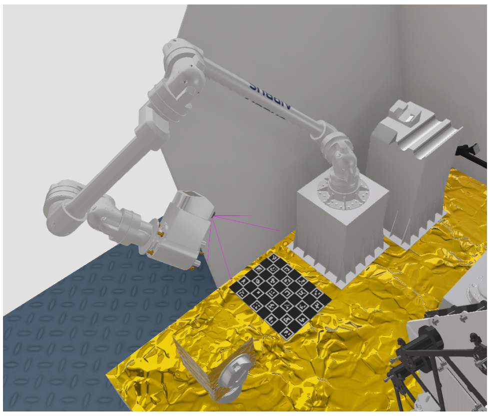

Figure 6: Example of a camera calibration in a constrained environment with a lot of obstacles which may hinder the robot workspace in the context of the HE-EURISE demonstration.

sion the Attitude and Orbit Control System (AOCS) by computing the torques and forces applied by the robotic operations on a spacecraft or actuators like arm motors by assessing the torques required for specific motions during the operations. The principle is mostly the same as for the camera dimensioning: define a set of parameters, run a simulation, log the sensors data and check if they meet the expectations. This iterative process helps refine design and reduce errors when transitioning from a simulation to a real-world implementation.

2.4.2. Reachability analysis

The simulation allows the operator to test the sequence of operations and the planned robot trajectories. First, simulating all ConOps steps confirms that the planned operations are kinematically feasible. The simulation ensures that a trajectory can be computed for each motion and that the sequencing of operations is coherent.

Moreover, the simulation may reveal several design flaws leading to collisions or unreachable configurations. For instance, the initial placement of the tool dispenser prevented the robotic arm from accessing a specific tool, while some operations resulted in collisions. Once these issues were identified, minor modifications were applied to the hardware design to resolve them.

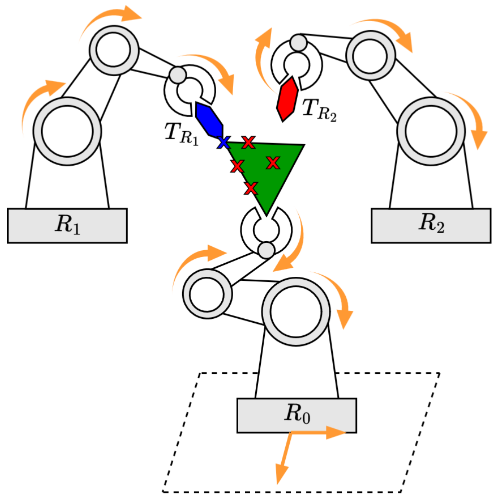

Figure 7: Interaction between 3 manipulator robotic arms. R_0 holds a green object, R_1 and R_2 respectively hold a blue and red tool. R_1 has one target (blue cross) to access. While R_1 is in position, R_2 has to move toward multiple goals (red cross). The possible motions are represented in orange and the doted square represents the area where the base of R_0 can be fixed.

Concerning reachability, the simulation can highlight some accessibility issues. As some operations are quite complex and demand for multiple robots to reach multiple poses, the simulation can help find a solution by testing multiple configuration until finding one that makes all the operations feasible. Such a situation, faced in the context of the HE-EURISE project, is represented in Fig.7. However, such a process is quite time-consuming and laborious. This is why Magellium developed a tool, as an addition to the simulation, to automatically find the robots'configuration which allow them to reach all their targets. This tool is specific to the situation described in Fig.7 but it can easily be generalized to other use cases. Indeed, this tool takes as inputs the robots'URDF description and the list of targets for each robot which contain the end effector frame, the goal frame and a potential approach phase. It, then, solves a basin-hopping optimization problem [7] to find the configuration (pose + joint state) for the first robot R_0 which maximizes the number of targets reached by R_1 and R_2. To compute if a target is reachable, this tool solves an Inverse Kinematics (IK) problem using the Pinocchio library [8] for forward kinematics and a Quadratic Problem (QP) solver [9] for resolution under constraints (joint limits and limited workspace of R_0). This optimization process is presented in Fig.8.

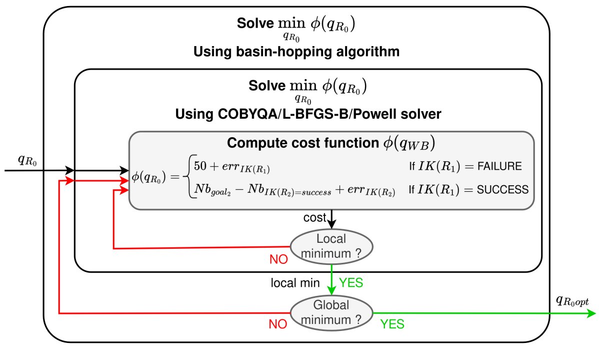

Figure 8: Optimization problem diagram. q_{R_0} is the configuration of R_0, ϕ is the cost function, $IK(R_i)$ is the solution of the IK problem for the robot R_i with $err_{IK(R_i)}$ the distance between the desired and the reached position if no solution was found to the IK problem.

2.4.3. Realistic representation

One major asset of the simulation is to provide a realistic representation of the system not only for visualization, as shown in Fig.3, but also for software testing. As part of the HE-EURISE project, the simulation was developed to be interfaced with the flight segment software provided by one of the partner. In that case, the goal of such process is to completely replace the real hardware by simulated ones to test the skill libraries. Thus, the definition of the interfaces is very important to swiftly switch from the simulated to the real environment.

3. DIGITAL TWIN FOR MISSION MONITORING

3.1. Concept

As briefly presented in section 1.2.3, the digital twin component serves three main purposes. First, its visualization mode enables the operator to quickly assess the current state of the Flight Segment and identify any deviations from the expected behaviour. This function is essential

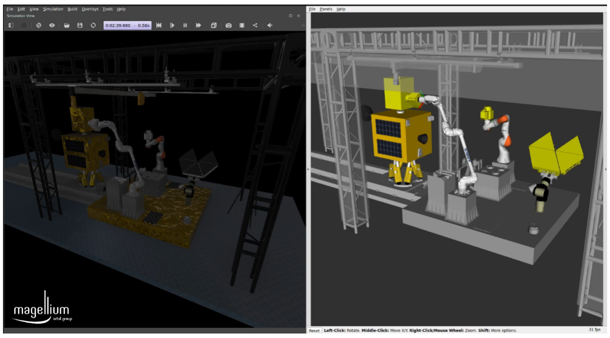

Figure 9: Online digital shadow on the right with the running simulation on the left where the light was switch off to simulate closed curtains.

for making timely decisions on whether to continue or interrupt ongoing operations. Second, the correction mode is key for recovery in case of anomalies: it not only updates the simulation with the actual state but also generates new commands consistent with the current in-orbit conditions. Finally, the visualization mode also supports replaying executed operations, compensating for the inability to directly observe the Flight Segment.

3.2. Building process

The digital twin is based on the simulation presented in section 2. Using the supervisor device in Webots, Magellium developed a tool to both import and export the scene state as a URDF file.
First, the URDF export from the simulation is used to initialized the 3D representation of the scene in RViz. This visualization, named digital shadow, is detailed in section 3.3.1. At any time during a demonstration, the state of this visualizer can also be exported as a URDF file.
Then, the URDF import is used to update the state of the scene in the simulation. Thanks to this capability, at any time all elements in the simulator can be teleported to another location in order to make the simualtion state match with the real system state.

3.3. Use cases

3.3.1. Digital shadow

As part of the in-orbit assembly projects (BPI-DEMARLUS, HE-EURISE), Magellium was in charge of the design of the digital twin. This tool was expected to be a key feature to help the operator to visualize what happened in orbit and to plan a recovery scenario in case of unexpected behaviour. Moreover, this tool was also expected to help the operator to monitor the ground demonstration which were planned as part of these projects. Thus, one important part of the digital twin is a 3D visualizer to both display the current state of the demonstrator for a ground demonstration and the past states of a potential in-orbit demonstrator. This visualizer will be appointed as digital shadow with online and offline modes in this manuscript.

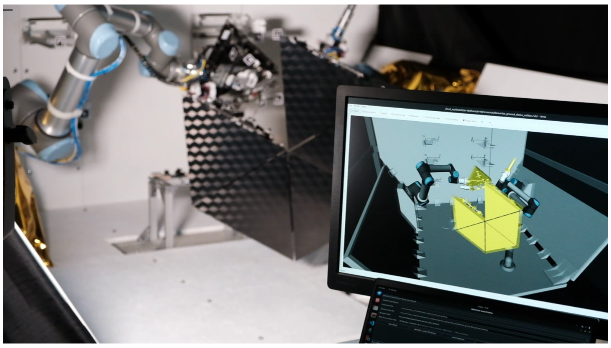

Figure 10: Picture of the digital shadow running during the BPI-DEMARLUS ground demonstration.

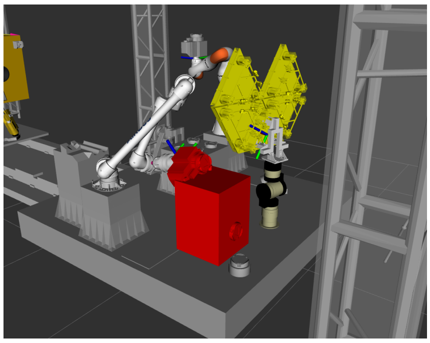

Figure 11: Collision visualized in RViz using the offline digital shadow.

Online mode The goal of the online mode is to show the operator the current state of the demonstrator in real time. It should allow the operator to quickly assess the current situation even without a direct view on the demonstrator. This will be not only the case in orbit but also when the demonstrator curtains are closed to mimic the same illumination condition as in orbit. This mode displays in RViz the current state of a demonstrator either real or simulated. The architecture stays the same in both cases as it is represented in Fig.12.
Given some inputs detailed in Fig.12, the digital shadow will compute in real time the poses of all the objects in the scene and keep up-to-date the potential attachments. Thus, the main output of the digital shadow online mode is the 3D visualization of all the objects in the scene in real-time. In RViz, the markers corresponding to non-attached object are grey while those of attached object are yellow. The visual meshes are plain while the collision ones are transparent. An example of visualization of a scene in RViz is shown in Fig.9.
This mode was used during the final demonstration of the BPI-DEMARLUS project as it is shown in Fig.10.

Offline mode The goal of the offline mode is to show the operator the past states of the demonstrator. It should allow the operator to quickly assess what happened during the operations execution. Moreover, it should also help him to easily spot and analyse unexpected events due to non-nominal behaviour. This tool is particularly useful in case of an in-orbit demonstration where the telemetry data are only received by batch during the visibility

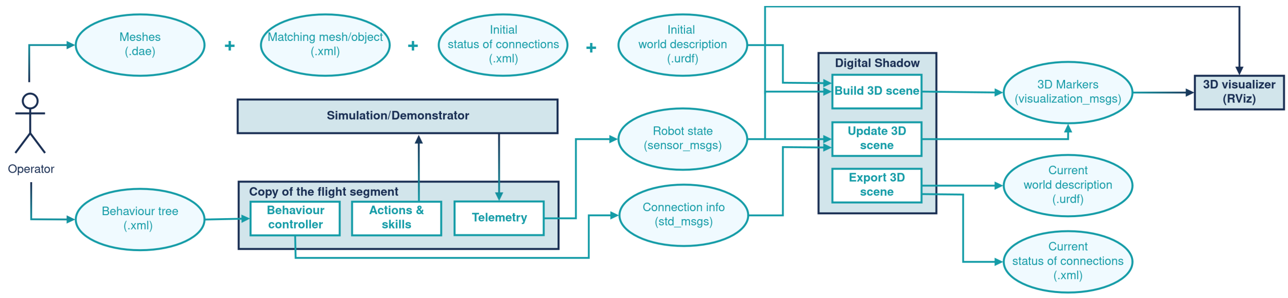

Figure 12: Online digital shadow (connected to either simulated or real hardware) architecture.

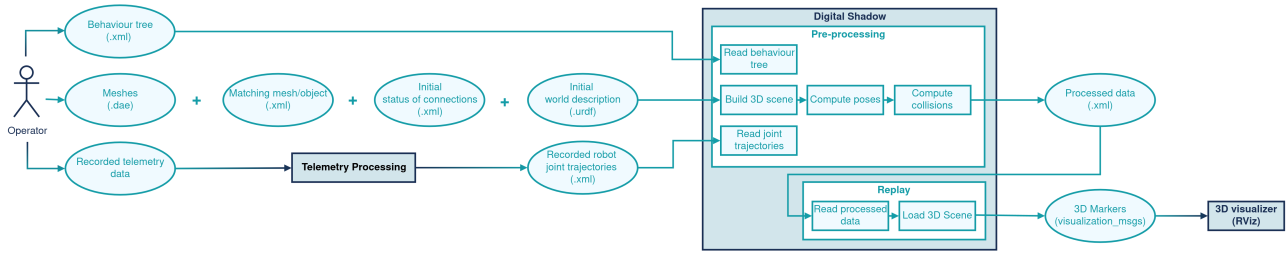

Figure 13: Offline digital shadow architecture.

window. Indeed the potential low frequency of visibility windows could prevent real-time visualization.
Given some inputs detailed in Fig.13, the digital shadow can compute for each timestep the poses of all the objects in the scene and keep up-to-date the potential attachments. It can also detect the potential collisions which happened during the demonstration. All this data can be stored for each motion in a XML file. This process is named preprocessing. Once generated, these motion data files can be loaded to be replayed. This process is named replay. It loads all the demonstration in RViz. The user can use a slider plug-in to move through time, jump from motion to motion or to the next collision. Collided marker are red. An example of a collision is shown in Fig.11.

3.3.2. Recovery

As previously explained, the digital twin can be used for recovery, this functionality is expected be tested as part of the HE-EURISE project. The expected workflow of the Digital Twin BB during a potential in-orbit mission will be the following. During visibility windows, sensor data are collected by the Mission Control BB, which triggers the validation mode. This mode provides the operator with the position, velocity and acceleration profiles of both planned and executed motions for comparison purposes. Based on this analysis along with the replay of operations using the difital shadow offline mode, the operator can quickly assess whether the planned sequence should continue.
If the current state of the Flight Segment is deemed unsatisfactory, the operator may activate the correction mode. In this case, a corrective sequence is tested using the Mission Preparation BB with an updated simulation that mirrors the actual onboard state. If the correction proves successful in simulation, a new sequence of operations is transmitted as commands to resolve the anomaly in orbit (illustrated in Fig.14). Conversely, if the validation

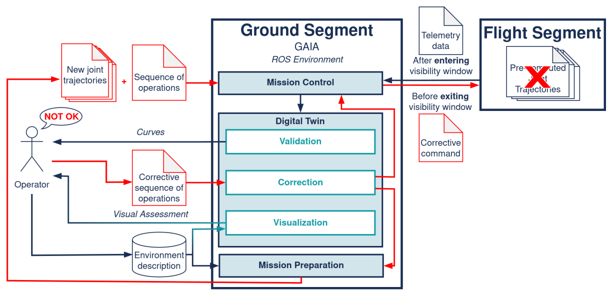

Figure 14: Digital twin workflow in case of a recovery process is needed to continue the in-orbit mission.

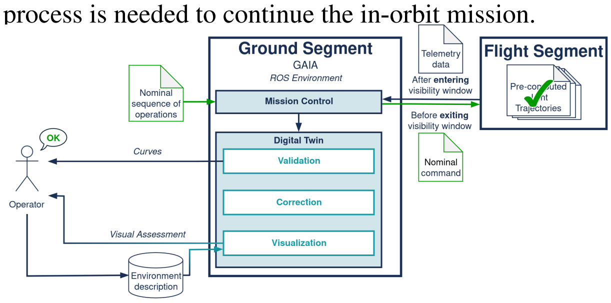

Figure 15: Digital twin workflow in nominal case, i.e. the mission unfolded as planned.

mode confirms nominal performance, operations proceed as initially planned with the trajectories defined during mission preparation (illustrated in Fig.15).

3.4. Sim-to-real gap

One major problem which was faced during the BPI-DEMARLUS project with the digital twin was the gap between the simulation, based on the CAD models, and the real demonstration set-up. Indeed, as the set-up is assembled by people, the poses of all elements in the scene may differ within a few centimeters from the theoretical poses. This is a known issue called the "sim-to-real" gap.

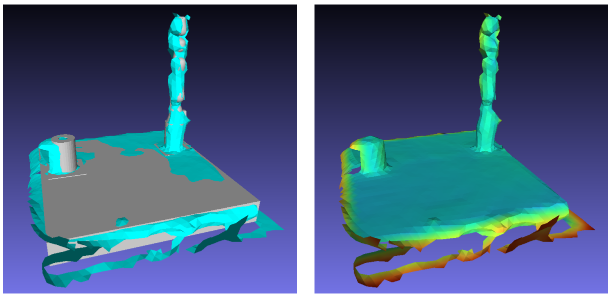

Figure 16: Comparison with MeshLab of the reconstructed mesh of the real setup and the initial model (on the left) and the reconstructed mesh colorized with the vertex quality corresponding to the computed distance with respect to the initial mesh (on the right).

To solve this problem, Magellium is currently working on the development of a solution to scan the demonstration set-up, compute the difference of poses between the 3D reconstruction of the scene and the original 3D model and readjust different element poses in the simulated scene. To scan the scene, multiple solutions to build 3D point cloud were considered: LiDAR, stereo-camera, photogrammetry. Dense LiDAR might be the most accurate solution but LiDARs are not commonly used in-orbit. This is why a vision-based method is favoured as the sim-to-real gap will have to be solved in-orbit for potential space mission. Thus, as part of the HE-EURISE project, only stereo-processing and photogrammetry were tested. As this kind of processes are computationally intensive, they will not run onboard but will be completed once at the start of the mission. During operations, updates to the digital twin will rely solely on telemetry data.

The first tests were performed on a simplified version of the scene (one robot and stands only) using a ZED-X camera from Stereolabs [10]. The 3D reconstruction using the depth information was completed with the ZED-SDK, the alignment of this mesh with the original model was done on Blender and the final comparison between the two meshes was performed with MeshLab [11]. Meshlab allows to compute the distance vertex-to-vertex between two meshes the result is shown on Fig.16. This first study demonstrated the feasibility of this method. However, it is still to show that the reconstruction's accuracy is sufficient to detect gaps of 2-5 cm. Better and more exhaustive datasets need to be acquired to precisely assess this method's accuracy. Moreover, photogrammetry algorithms were run without success on the same datasets. One hypothesis to explain these failures is that the acquisition were extracted from a video stream which may introduce some blur in the images.

4. CONCLUSION

Magellium is currently working on GAIA, an extensive framework to prepare and monitor in-orbit robotic missions using simulation means. GAIA encompasses a digital twin for monitoring, fault detection and recovery purposes.

Prior to the mission, GAIA helps operators to design robotic operations. This mission preparation BB includes a 3D simulator linked to a copy of the flight segment software, which can be used to test algorithms for perception or path planning in a representative simulated environment.

Another key feature of GAIA during mission operations is its digital twin. It allows operators to visualize current and past states of the flight segment. Telemetry data updates the digital twin during each visibility window, providing a powerful analysis tool to replay past operations and detect unexpected behaviours such as collisions. The simulator, linked to the mission preparation BB, can then be used to test new operations before sending commands to the flight segment, which is crucial in case of deviations from the planned operations. However, the digital twin might be ineffective without verifying that its initial state matches reality. Therefore, a sim-to-real alignment process is planned before operations begin.

REFERENCES

[1] Macenski S., Foote T., Gerkey B., Lalancette C.and Woodall W., *Robot Operating System 2: Design, architecture, and uses in the wild*, 2022, Science Robotics, vol. 7

[2] Bazerque C., Maroger I., Germa T., Fieschi P., Delort V. et al., *Designing Future In-orbit Missions: a Simulation and Monitoring Framework for Robotics Operations*, 2024, IAC-24, B6, 3, 9, x87343

[3] Caujolle R., Aridon G., Marchive T., Girault S., Figus C. et al., *In-Orbit Demonstration of Robotic Operations on a reusable service module - DEMARLUS*, 2024, IAC-24, D1, IP, 15, x85175

[4] Caujolle R., Aridon G., Figus C., Jankovic M., Manz M. et al., *European Robotics for Space Ecosystem - EU-RISE*, 2024, IAC-24, D1, 1, 2, x85038

[5] Webots, an open-source mobile robot simulation software, `http://www.cyberbotics.com`

[6] Blender, a 3D modelling and rendering package, `http://www.blender.org`

[7] Wales D. J. and Doye J. P. K., *Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms*, 1997, Journal of Physical Chemistry A, 101, 5111.

[8] Carpentier J., Saurel G., Buondonno G., Mirabel J., Lamiraux F. et al., *The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives*, 2019, IEEE International Symposium on System Integrations (SII)

[9] Goldfarb D. and Idnani A., *A numerically stable dual method for solving strictly convex quadratic programs*, 1983, Mathematical Programming, vol. 27, p. 1-33.

[10] Stereolabs Docs: Installation, API Reference, Tutorials, and Integration, `https://www.stereolabs.com/docs`

[11] Cignoni P., Callieri M.,Corsini M., Dellepiane M., Ganovelli F. et al., *MeshLab: an Open-Source Mesh Processing Tool*, 2008, Sixth Eurographics Italian Chapter Conference, p. 129-136