2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

The iMETRO Dynamic Simulation: An Open-Source Simulator for Intravehicular Space Robotics Research

Nikki Hart[1,2], Nathan Dunkelberger[2,3], Erik Holum[2,4], Lydia E. Kavraki[1,5], Emma Zemler[2], Shaun Azimi[2]
[1]Rice University [2]NASA Johnson Space Center [3]CACI International [4]METECS [5]Ken Kennedy Institute

Abstract*— We present the iMETRO Dynamic Simulation, the first open-source dynamic simulation environment for research in the use of robot manipulators inside space vehicles for maintenance and logistics tasks, or intravehicular robotics (IVR). IVR has great potential to facilitate science and exploration on the Moon by saving crew time, but there are limited open-source resources that would enable researchers to identify the next set of challenges in manipulation for IVR. We provide a full-featured, high-fidelity dynamic simulation of the real-world iMETRO IVR test facility, which includes mockups representative of the interior of a future space vehicle as well as an 8-DoF manipulator that serves as an example robot platform for this research. Our modular simulator enables new software, hardware, and operational paradigms to be tested in a reconfigurable mockup environment. To improve the accessibility and extensibility of this simulation environment, we also provide ROS 2 hardware control interfaces to MuJoCo as well as a model conversion tool such that the same models may be used with ROS 2 and MuJoCo. To evaluate the sim-to-real transfer capabilities of this simulation, we present an open-source example application demonstration developed in the simulation and transfer it to the real-world iMETRO facility in less than a day. Finally, we identify the challenges and opportunities in modeling a real-world facility to aid future simulation efforts. The open-source simulation and application can be found at `https://github.com/NASA-JSC-Robotics`. The MuJoCo and ROS 2 integration tools have migrated to the ros-controls organization and can be found at `https://github.com/ros-controls/mujoco_ros2_control`.

I. Introduction

As NASA plans to return humans to the Lunar surface by 2028, the use of robots for surface and orbital maintenance and logistics tasks has emerged as a promising way to save crew time and facilitate exploration objectives [1]. In particular, intravehicular robotics (IVR), or the use of robots inside space vehicles and habitats, has the potential to improve mission outcomes leveraging existing technologies. Unlike the algorithms required for extravehicular robotics (EVR), those required for IVR can assume stable lighting conditions, reliable access to sensing and power, and a relatively well-known environment. Similarly, hardware for IVR need not consider certain environmental conditions of space, such as lunar dust or large temperature fluctuations. IVR research is key to identifying the next set of challenges for robots to advance mission objectives in space.

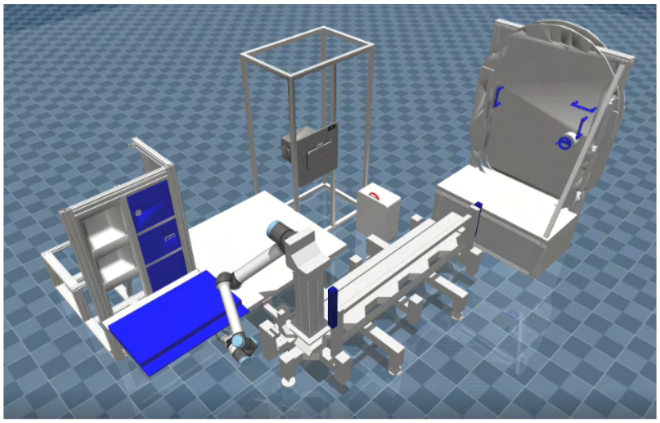

Fig. 1: The iMETRO Dynamic Simulation environment visualized in MuJoCo, including the 8-DoF manipulator robot and IVR mockups.

NASA has recently made efforts to open-source aspects of real or realistic intravehicular environments [2, 3] to facilitate research in this area. One such open-source resource is NASA's Integrated Mobile Evaluation Testbed for Robotics Operations (iMETRO) facility [3]. The ground test facility, located at NASA's Johnson Space Center, features full-scale, high-fidelity mockups of real and potential space vehicle components. The facility also provides an 8-DoF manipulator as a test platform for IVR software and an example robot archetype for IVR activities. In addition to the ground test facility, NASA has developed an open-source kinematic simulation of the facility and robot. Using the ros2_control mock hardware framework [4], joint position commands sent to the simulated robot are replicated exactly without simulating dynamics or control, allowing users to visualize the position of the robot among the mockups in response to commands. However, kinematic simulations pose limitations for testing contact-rich or perception-based behaviors, as neither sensor data nor contact dynamics are simulated. This is a step towards making IVR research accessible to the broader robotics community, but further effort is required to create a realistic, publicly-accessible testbed for a breadth of IVR research.

Dynamic simulations, which model sensor observations and contact dynamics in addition to a robot's proprioception, enable the rapid but realistic testing of robot behaviors while minimizing the effort required to transfer these behaviors to

---

This work was supported by the NASA Johnson Space Center Engineering Innovation Fund. NH is supported by the Ken Kennedy Institute Computational Science and Engineering Recruiting Fellowship, funded by the Energy HPC Conference and the Rice University Department of Computer Science, and by the Rice University School of Engineering and Computing Graduate Fellowship. Work on this project by LEK is supported in part by NSF CCF-2336612 and Rice University Funds.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

the real world. Further, these simulations enable testing end-to-end policies and system-based approaches to manipulation, rather than focusing on a single area of research such as perception or planning. While dynamic simulations have emerged as widely-used platforms for benchmarking robots' ability to tackle a wide variety of tasks [5–12], intravehicular environments differ greatly from the home and industrial environments explored in most manipulation research, both in the tasks performed and in the objects and mechanisms with which robots interact.

To fill the gap of open-source dynamic simulations for intravehicular robotics research, we introduce the iMETRO Dynamic Simulation (Figure 1). Based on NASA's open-source iMETRO facility resources, our dynamic simulation provides a modular simulation environment in which new robotics software, platforms, and operational paradigms can be evaluated for IVR use cases. We provide high-fidelity, reconfigurable simulated space vehicle mockups and an example 8-DoF robot manipulator with which a breadth of IVR research may be conducted. We design the simulator to be modular and extensible, enabling alternative robot platform archetypes to be tested in the mockup environment. To facilitate the integration of new software and hardware components into the simulation, we provide two software tools for integration of ROS 2 and MuJoCo [12]: first, ROS 2 hardware control interfaces to MuJoCo for robots and sensors, and second, a conversion tool for the widespread unified robot description format (URDF) to the MuJoCo description file format (MJCF). Finally, to exemplify the use of the simulation environment and a typical IVR application that can be developed in the iMETRO Dynamic Simulation, we develop an open-source application demonstration of a logistics task in simulation and transfer the task to the real-world facility.

In this paper, we discuss the technology gaps and driving principles that motivate this work, then describe the features of the iMETRO Dynamic Simulation in detail. Next, we discuss our approach to the example application demonstration we provide, and we describe the transfer of an IVR application built completely in simulation to the real-world facility. Finally, we detail lessons learned from the sim-to-real transfer experiment and provide future directions of research in this area. In summary, our contributions are as follows:

- We present the first open-source dynamic simulation environment for robot manipulation research in IVR applications;
- we provide tools for MuJoCo and ROS 2 integration that facilitate the integration of new software and hardware components into the simulated environment;
- we demonstrate the sim-to-real transfer capabilities of the dynamic simulation environment through an example application demonstration;
- and we isolate and discuss the challenges we faced to aid future simulation efforts.

II. RELATED WORK

A. Benchmarking and Simulation for Robot Manipulation

Driven by the rapid progression of algorithms and learned policies for robot manipulation, the robotics research community as a whole has identified the need for widely-accessible, standardized benchmarks for robot manipulation. While some have proposed open-source hardware (OSH) such as the NIST Taskboard [13] for this purpose, high-fidelity dynamic simulations have emerged as the most common platforms for these benchmarks [5–12, 14]. In particular, the reinforcement learning and imitation learning communities have adopted dynamic simulations as not only a benchmark of policy performance but a training arena for new learned policies. As a result, many works seek to diversify the objects [15], manipulation skills [5–7, 16, 17], and long-horizon tasks composed of multiple skills [8–11, 14] available for training and testing. Certain benchmarking environments aim to simulate a specific application, such as furniture assembly [9, 14], that involves objects or skills not covered by more general-purpose benchmarking environments [8]. Similarly, these existing benchmarks and simulation environments are not suitable for IVR research, as the mechanisms and objects present in an intravehicular environment vary greatly from the commercial off-the-shelf (COTS) mechanisms and objects usually found in home or industrial environments. Namely, hardware found on a space vehicle is typically custom designed to meet spaceflight-specific requirements such as space and weight saving or serve spaceflight-specific purposes such as cabin pressurization. Further, benchmarking environments for manipulation primarily focus on tabletop manipulation, while IVR operates on a larger scale.

B. Testbeds for Intravehicular Robotics

In order to develop and evaluate concepts for IVR, it is necessary to have access to a testbed. However, few suitable and accessible testbeds exist (Table I). Due to the unique conditions of space vehicle environments, tests of IVR systems often occur on the International Space Station (ISS) [22–25]. For earlier-stage research, NASA robotics researchers and their close collaborators often use realistic, large-scale ground mockups, such as the Granite Lab at Ames Research Center [26] or the iMETRO test facility at Johnson Space Center [3]. For most researchers, however, the cost or requirements for access to these resources can be prohibitive. As a result, some research in this area uses only small-scale ground mockups, such as those of cargo transfer bags (CTBs) [18, 19]. To facilitate IVR research in academia and industry, both NASA and the Japan Aerospace Exploration Agency (JAXA) have made efforts to open-source aspects of real or realistic IVR environments. Namely, NASA has open-sourced a dataset of images of the inside of the ISS and associated data from the Astrobee free-flying robot that took the images [2]. However, this dataset is best suited for research in perception or planning for free-flyers, not IVR manipulation. Three additional efforts to open-source resources for IVR research are described in Section II-C.

Category	Name	Dynamic Sim	Manipulation	Articulated Objects	IVR Assets	OSS/OSH*
Manipulation benchmarks	FurnitureBench [14]	✓	✓	✓		✓
	RL Bench [8]	✓	✓	✓		✓
	NIST Taskboard [13]		✓	✓		✓
IVR Environments	Ground mockups [18, 19]		Varies	Varies	✓	
	Astrobee Dataset [2]				✓	✓
	Astrobee Simulator [20]	✓			✓	✓
	Int-Ball2 Simulator [21]	✓			✓	✓
	iMETRO Kinematic Simulation [3]		✓	✓	✓	✓
Ours	iMETRO Dynamic Simulation	✓	✓	✓	✓	✓

TABLE I: Comparison of our work to closely related work. Our open-source dynamic simulation serves as a platform for IVR manipulation research and includes articulated IVR mockups. *Open-source software (OSS) or open-source hardware (OSH).

C. *Simulations for Space Robotics*

There are many simulation environments designed for extravehicular robotics applications [27–31], in part due to the public availability of satellite images and terrain information for the Lunar and Martian surfaces. Among those that currently simulate robot manipulation, both closed-source [29] and open-source [30, 31] simulators are used. However, due to the lack of open-source real or realistic models of the inside of a space vehicle, there are few simulations of IVR environments. NASA and JAXA have both open-sourced IVR-focused ISS simulators: the Astrobee simulator [20] and the Int-Ball2 Simulator [21], both based in Gazebo [32]. While both simulators provide photo- and depth-realistic simulation of the ISS, the models are not articulated, and they therefore are not suitable testbeds for IVR manipulation. Open Robotics provides Gazebo models of NASA's Robonaut 2 [22] robot and the ISS [33, 34], including graspable handles, but there are no large-scale articulated assets for testing IVR logistics and maintenance tasks. Finally, NASA has also open-sourced models of the aforementioned iMETRO facility, which contains mockups of potential architectures for a future space vehicle [3]. However, the simulator deals only with robot kinematics; therefore, contact dynamics and uncertainty are not considered for manipulation applications, and sensor data cannot be simulated to represent the state of the simulated world. Still, open access to high-fidelity models of an IVR ground test facility facilitates further work in simulation for IVR.

III. PRINCIPLES FOR THE IMETRO DYNAMIC SIMULATION

The iMETRO Dynamic Simulation is designed to facilitate the development of internal applications in simulation as well as to foster collaboration with external partners from a wide variety of robotics disciplines. The following four objectives, illustrated in Figure 2, have informed our architecture and implementation decisions for the simulation software.

Full-featured simulation. In order to support the integration of new robotics technologies, it is important for the iMETRO Dynamic Simulation to provide realistic simulation of all aspects of the iMETRO facility, including the 8-DoF manipulator robot, the surface habitat mockups, RGBD cameras, and the robot's force torque sensor. It therefore must provide high-quality simulation of contact dynamics and sensors to support the development of the contact-rich and perception-based behaviors frequently used in the facility. It also must integrate with the existing software stack for the mobile manipulator robot, which is written using ROS 2.

Accessibility. In order to simplify the development of new applications and facilitate the integration of new robot software and hardware into the simulation, ROS 2 integration is a priority in the iMETRO Dynamic Simulation. ROS 2 is widely used and allows us to provide an easy entrypoint for creating complex robotics applications in the simulation. Further, to encourage collaborations from a variety of robotics disciplines, the iMETRO Dynamic Simulation must not have strict requirements for the hardware specifications of users' machines; for example, access to GPUs should not be required.

Extensibility. The iMETRO test facility was designed as a general-purpose testbed for a variety of IVR systems; as such, it was important that the iMETRO Dynamic Simulation also support a wide variety of future applications. We therefore prioritize writing open-source software and using an open-source physics simulation. Our use of open-source software enables collaborators to extend the iMETRO Dynamic Simulation to support new robots, sensors, software, and application-specific simulation capabilities as needed. Our lack of strict hardware requirements also facilitates the development of a wide variety of systems for test; we encourage not only new software but new hardware for IVR to be tested in the simulation.

Sim-to-real transfer. The final principle involved in the design of the iMETRO Dynamic Simulation is minimizing the sim-to-real gap. As a result, we prioritize high-fidelity contact dynamics and support for the existing sensors in the iMETRO

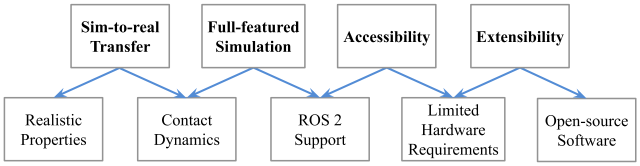

Fig. 2: Principles behind the design of the iMETRO Dynamic Simulation and the resulting requirements.

test facility in our choice of underlying physics simulation. In comparison to previous work [3], we also prioritize assigning realistic mass properties to the simulated mockups.

IV. DESIGN OF THE iMETRO DYNAMIC SIMULATION

A. Choice of Physics Simulator

We build the iMETRO Dynamic Simulation off of MuJoCo [12], an open-source physics simulator with a strong focus on contact dynamics and real-time performance on CPUs. Compared to alternatives like IsaacSim [35] and Gazebo [32], MuJoCo facilitates our goal of providing realistic simulation for contact-rich manipulation behaviors, grasping, deformable objects, and sensors while also aligning with our philosophy of open-source software for extensibility. We extend MuJoCo's accessibility and features with a ROS 2 control hardware interface for MuJoCo, which provides expanded functionality compared to previous open-source implementations [36]. With support for multiple sensors, including RGBD cameras, LiDAR, and force-torque sensors, we enable the development of internal applications and facilitate the integration of new hardware and software into the simulation stack. Further, we provide a tool to convert unified robot description format (URDF) files, which are commonly used in ROS 2, to the MJCF file format used by MuJoCo. As a result, robots currently being used in ROS 2 can easily be integrated into the simulation.

B. Realistic Models of Real-world Mockups

The iMETRO Dynamic Simulation simulates the real-world space vehicle mockup environment found in the iMETRO facility. The default environment found in the simulation environment reflects the current state of the facility; it therefore includes an EXPRESS Rack [37] for housing modular equipment, a MERLIN freezer [38] for cold stowage of science samples, a pressurized hatch door, and a stowage unit with a hinged bench seat. Additionally, the environment can be reconfigured to include a second, distinct pressurized hatch door and additional instances of the existing mockups. These mockups provide some of the unique mechanisms with which a robot may be expected to interact in an IVR scenario, including hatch handles, drawer pull mechanisms, a locking latch on the MERLIN freezer door, and hatch pressurization wheels. A mockup of a cargo transfer bag is also included as a free body in the environment; while the CTB can deform slightly in the real world, it is modeled as a rigid object for simplicity.

The mockups are represented both in the unified robot description format (URDF) and MuJoCo's MJCF file format to preserve their articulation and enable them to be used with ROS 2 applications. To minimize uncertainty when planning with a virtual model of the mockups in the real-world facility, we use a touch calibration procedure on the real robot to update the relative locations of the mockups in the URDF model of the facility. The mockups are assigned mass properties based on estimated material densities, and frictional properties are approximated based on visual inspection of the mockups' behavior in simulation relative to observations in the real world.

C. Facility Robot Platform

While the iMETRO facility is designed to be robot agnostic, the facility provides an example robot consisting of a UR10e manipulator on a Vention linear rail [39] and an Ewellix vertical lift kit [40], and these assets are simulated in the iMETRO Dynamic Simulation. The robot also features a Robotiq Hand-E parallel jaw gripper with custom fingers designed for IVR [41], a force-torque sensor located at the tool flange, and two RealSense cameras mounted at the wrist and vertical lift. Further details on this robot can be found in prior work describing the facility [3]. This robot replicates the reachable workspace of a mobile manipulator while requiring only proprioception to ascertain the position of the robot in the workspace. Most importantly, simulating this example robot in the iMETRO Dynamic Simulation provides a robotic platform on which new behaviors can be developed and transferred easily from simulation to the real-world facility.

D. Modularity

Like the iMETRO facility, the iMETRO Dynamic Simulation is designed to be highly modular. The iMETRO facility is represented in the simulation using the xacro format [42], which enables individual components of the facility to be included or excluded from the simulation using configurable flags. As a result, the mockups in the simulated facility may be reconfigured, and the provided robot may be removed and replaced with another embodiment easily. To facilitate frequent changes to the simulated environment, we provide a conversion tool that generates MJCF files from URDFs, allowing these modular configurations to be seamlessly integrated with both ROS 2 and MuJoCo. Similarly, our ROS 2 control hardware interface for MuJoCo enables the use of ROS 2 with the simulation, and new ROS 2 enabled hardware and software components can integrate with the simulation via standard ROS 2 interfaces.

E. Open-source Application Demonstration

Finally, we provide an example application, including procedures to run the application in simulation and in the physical iMETRO facility. This example application highlights key features of the simulation, demonstrates a typical approach to robot behaviors in the iMETRO facility, and validates the sim-to-real capabilities of the dynamic simulation. The example application uses MoveIt [43] and other open-source software packages, making it extensible to new applications. The methods used to create the example application demonstration are described below.

V. APPLICATION DEMONSTRATION TECHNIQUES

A. Motion Planning

The example application demonstration is executed via a series of waypoints, including robot joint configurations, end-effector poses, and relative moves in the end-effector frame. We use the OMPL [44] planners in MoveIt for motion planning due to their integration with ROS 2. Both in simulation and on the real robot, we use a virtual model for motion planning; that is, we represent the mockups in the URDF format such that MoveIt can plan safe trajectories among them, and we update the states of the mockups to reflect the real or simulated world. In simulation, the MuJoCo-to-ROS2-control hardware interface publishes joint states for these mockups onto ROS 2 topics, keeping the state of the digital twin up-to-date. When running the application demonstration on the real robot, we call a ROS 2 service within the application demonstration to update the world model's state once the bench seat open is complete, enabling safe motion planning.

Two actions in the application demonstration do not use hard-coded waypoints: the bench seat open and the CTB grasp. We generate an arc of waypoints around the current location of the bench seat's revolute hinge joint, allowing this step to be integrated with perception-based mockup localization methods in the future. Similarly, we identify a grasp point for the CTB using perception, and the demonstration can accommodate small variations in the CTB's location in the facility. This step is described further in Section V-B.

B. Color Mask Segmentation

In the iMETRO Dynamic Simulation, the CTB is modeled in MuJoCo but not in ROS 2 and MoveIt, as it can be manipulated by the robot but is never rigidly attached to the robot. Its position in the world therefore must be identified using perception. In the iMETRO facility, small augmentations can be made to the mockups to facilitate robot behaviors; in this case, the CTB handle is marked with red tape to facilitate perception and manipulation. To identify the ideal grasp location on the handle, we segment the image from the wrist-mounted RealSense camera into named colors [45] and identify the centroid, size, aspect ratio, and principal component of the segmented region of the color of interest, red. Finally, we use the centroid as well as depth information from the camera image to identify a grasp pose.

C. Demonstration Procedure

The example application demonstrates a logistics task within the iMETRO facility using its 8-DoF manipulator. First, the robot approaches and opens the bench seat. The robot then traverses across the facility to approach the CTB. Using the color mask segmentation approach, the robot identifies a grasp pose on the CTB handle, grasps the CTB, and lifts the CTB. Finally, the robot carries the CTB to the bench seat and places the CTB upright in the seat's stowage area.

VI. RESULTS

The iMETRO Dynamic Simulation is a high-fidelity simulation environment that minimizes the sim-to-real gap. To evaluate the sim-to-real transfer capabilities of the simulation, we transferred the example application demonstration described above to the real-world iMETRO facility, as shown in Figure 3. In this section, we describe the adjustments made to the simulation or behaviors based on the performance of the demonstration. Here, success for each step of the demonstration is defined as completing the action without unintended contact between the robot and its environment or in-hand objects and the environment, and overall task success is defined as the CTB being placed in an upright, stable position in the bench seat once the robot has retreated from the bench seat.

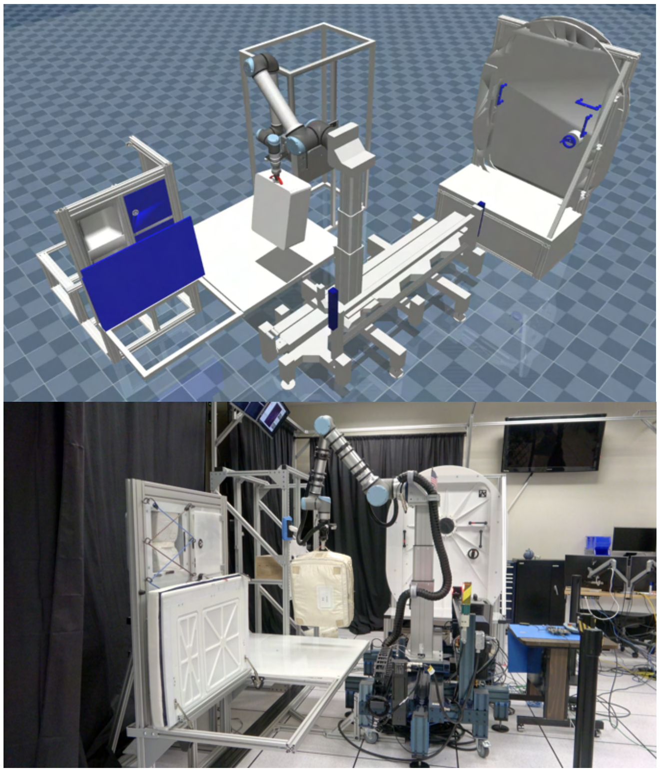

Fig. 3: Side-by-side images of the 8-DoF manipulator placing the CTB into the bench seat in simulation (top) and in the physical iMETRO facility (bottom).

A. Trials in Simulation and on Hardware

Prior to transferring the application demonstration to the real world, four trials were run in simulation with a task success rate of 100%. After the changes made to the simulation and behaviors described in Section VI-B, two trials were run in the real-world iMETRO facility, and both were fully successful. To ensure changes made to the behaviors on the real robot did not impact task success in simulation, three additional trials of the adjusted demonstration were run in the updated simulation, and all were fully successful.

B. Transfer to the Real Robot

Transferring the example application demonstration built only in the iMETRO Dynamic Simulation was completed in less than a day. The sim-to-real transfer experiment of the application demonstration was used to identify modeling shortfalls, as described and categorized in Table II; however, we found that some of these shortfalls result from multiple contributing factors and are best mitigated through changes to the robot behaviors.

Demonstration Step	Failure Mode	Modeling Shortfall
Bench Seat Open	N/A	N/A
Traverse to CTB	E-stop due to robot's proximity to obstacles	Unmodeled camera cable housing
CTB Grasp	CTB handle out of camera frame	Unmodeled structural beams on floor
CTB Lift	Failure to plan	
CTB Place	CTB collision with mockups	CTB modeled as rigid

TABLE II: Failure modes that occurred during each demonstration step in the iMETRO facility and their associated modeling shortfalls.

The first failure mode occurred when the robot traversed the iMETRO test facility to reach the CTB. While this revealed the pitfalls of failing to model the wrist camera cable housing on the robot, other factors contribute to the sim-to-real gap: for example, human operators tend to prefer e-stopping the robot to allowing a collision with the real robot, making operator preferences for the robot's proximity to obstacles more conservative than in simulation, and uncertainty in the mockups' positions may result in the robot approaching obstacles more closely than the trajectory visualized in the world model. Due to the difficulty of accurately modeling the flexible camera cable housing, this modeling shortfall is likely best mitigated through changes to the robot's behaviors, such as optimization-based motion planning to increase clearance from obstacles; however, cable management and modeling remain open problems.

A second set of failure modes occurred when grasping and lifting the CTB, resulting from failing to model structural metal beams on the floor of the test facility. This caused the floor height to be different in the test facility and the simulation, affecting the field of view of the camera. Further, the approach configuration selected in simulation had limited manipulability, resulting in the robot not being able to lift the CTB the same distance off the higher ground height.

Finally, the failure mode that occurred when placing the CTB resulted both from the stretching of the CTB handle when lifted in real life, an aspect of the CTB that was not modeled in simulation, and from the placing behavior developed in simulation not being designed with uncertainty in the CTB's position in mind. This failure was mitigated by adjusting the robot's behavior to provide more clearance between the CTB and bench seat, but modeling the deformability of the CTB handle is an open area of future work.

Notably, no failures were observed when opening the bench seat, and there were no perception-related failures on runs when the CTB handle was in the camera frame. These are promising results for the transfer of perception-based behaviors, contact-rich behaviors, and the manipulation of heavy objects from the iMETRO Dynamic Simulation environment to the physical world.

VII. LESSONS LEARNED

Identify changes to models. The iMETRO facility is a dynamic, complex, and modular environment; the robot, the mockups, and other assets in the facility may change at any time. This introduces challenges when maintaining an accurate model of the facility for a simulated environment or digital twin. While sim-to-real transfer is a burden often placed on robot policies and behaviors, unexpected results during sim-to-real transfer can be used to identify modeling shortfalls and detect changes in a dynamic environment. Detecting changes to semi-structured environments is an open problem in IVR research, and perception-based methods for detecting discrepancies between a model and a real-world environment can also be used in the future [26, 46].

Simulate uncertainty. In benchmarking literature, benchmarking problems are often procedurally generated with some degree of randomness or variation to evaluate generalization [8, 14, 47]. While we do provide a mechanism by which the relative positions of the mockups in the environment can be changed at runtime to simulate uncertainty, we do not randomize the position of the CTB in the environment. This contributed to challenges with sim-to-real transfer, as the CTB could appear in slightly different locations in the real world depending on the human operator's placement. Extensions of this work should consider strategies of procedurally and reproducibly embedding uncertainty into the simulation environment, both to better simulate the real world and to improve the robustness of behaviors developed in simulation.

Maintain state of the world. Due to the approach of using digital twins for motion planning in the example application demonstration, it was important to maintain an accurate and up-to-date state of the mockups in simulation and in the real world. While the mockup joint states are available in simulation, the states of relevant mockups in the digital twin must be manually or programmatically updated when running real-world experiments. There are trade-offs associated with any world representation—for example, a digital twin may facilitate a semantic or symbolic understanding of the state of the world, while point clouds ensure the most up-to-date collision information for motion planning—and merging these disparate representations is not trivial. Future work should consider methods of maintaining an up-to-date state of the world that lends itself to both geometric and symbolic representations, such as using AprilTags to identify both the position of a hatch door and whether it is fully open, partially open, or closed.

VIII. FUTURE WORK

High-fidelity simulation of complex mechanisms. While we have demonstrated the sim-to-real transfer capabilities of contact-rich and perception-based tasks developed in simulation, we also plan to evaluate the iMETRO Dynamic Simulation's performance as a testbed for dexterous manipulation tasks. We provide a variety of complex mechanisms with which a robotic manipulator may interact, such as hatch handles, drawer pull mechanisms, the MERLIN freezer locking latch, and hatch pressurization wheels. In future work, we plan to evaluate the sim-to-real transfer of interactions with these mechanisms, as well as to better model the behavior of complex features in the environment, such as the locking mechanisms on the MERLIN freezer and hatch door.

Quantitative evaluation of simulation fidelity. While sim-to-real transfer of behaviors and policies is a positive indicator towards the fidelity of a simulation, quantitative methods must be used to fully characterize the properties of the robot and mockups in simulation and the real world. To evaluate the simulated robot, we plan to leverage proprioceptive data or proposed system identification tools [48]. Similarly, we plan to use the high-fidelity robot model and force-torque sensor data, joint torques, and other proprioceptive observations to evaluate the physical properties of the simulated mockups.

Model additional robot embodiments. The robot currently modeled in the iMETRO Dynamic Simulation and used in the iMETRO test facility provides many of the benefits of mobile manipulators while simplifying applications through the assumption of accurate proprioception. Meanwhile, NASA has proposed a mobile manipulator archetype for use in the iMETRO test facility [3]; work to model this robot in the simulation environment is open-source and ongoing. In future work, we plan to model additional robot embodiments in the iMETRO Dynamic Simulation to explore related simulation challenges such as localization, contact dynamics between mobile robots and the ground, and additional sensing modalities.

Demonstrate use of new sensors. As part of the ROS 2 control hardware interface for MuJoCo, we provide implementations of simulated RGBD cameras, LiDAR sensors, force-torque sensors, and IMUs. However, we only simulate RGBD cameras and force-torque sensors as part of the simulation of the 8-DoF manipulator, and we only use an RGBD camera as part of the example application demonstration. As part of our future efforts to simulate additional robot embodiments and develop a wider variety of applications in the iMETRO Dynamic Simulation, we also plan to evaluate additional simulated sensors' performance through applications.

REFERENCES

[1] National Aeronautics and Space Administration (NASA) Exploration Systems Development Mission Directorate (ESDMD). *Moon to Mars Architecture Definition Document (ESDMD-001)*. https://www.nasa.gov/wp-content/uploads/2024/12/esdmd-001-add-rev-b.pdf?emrc=5ffbf4. Version B.1. 2024.

[2] S. Kang, R. Soussan, D. Lee, B. Coltin, A. M. Vargas, M. Moreira, K. Browne, R. Garcia, M. Bualat, T. Smith, J. Barlow, J. Benavides, E. Jeong, and P. Kim. "Astrobee ISS Free-Flyer Datasets for Space Intra-Vehicular Robot Navigation Research". In: *IEEE Robotics and Automation Letters* 9.4 (2024), pp. 3307–3314.

[3] N. Dunkelberger, E. Sheetz, C. Rainen, J. Graf, N. Hart, E. Zemler, and S. Azimi. "Design of the iMETRO Facility: A Platform for Intravehicular Space Robotics Research". In: *2025 22nd International Conference on Ubiquitous Robots (UR)*. 2025, pp. 390–397.

[4] ros2_control Development Team. *Mock Components*. https://control.ros.org/humble/doc/ros2_control/hardware_interface/doc/mock_components_userdoc.html.

[5] P. Atreya, K. Pertsch, T. Lee, M. J. Kim, A. Jain, A. Kuramshin, C. Eppner, C. Neary, E. Hu, F. Ramos, et al. "RoboArena: Distributed Real-World Evaluation of Generalist Robot Policies". In: *Proceedings of the Conference on Robot Learning (CoRL 2025)*. 2025.

[6] Y. Urakami, A. Hodgkinson, C. Carlin, R. Leu, L. Rigazio, and P. Abbeel. "Doorgym: A scalable door opening environment and baseline agent". In: *arXiv preprint arXiv:1908.01887*. 2019.

[7] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, S. Nasiriany, and Y. Zhu. "robosuite: A Modular Simulation Framework and Benchmark for Robot Learning". In: *arXiv preprint arXiv:2009.12293*. 2020.

[8] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. "RLBench: The Robot Learning Benchmark & Learning Environment". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3019–3026.

[9] Y. Lee, E. S. Hu, and J. J. Lim. "IKEA Furniture Assembly Environment for Long-Horizon Complex Manipulation Tasks". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 6343–6349.

[10] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. "LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 44776–44791.

[11] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su. "ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills". In: *International Conference on Learning Representations*. 2023.

[12] E. Todorov, T. Erez, and Y. Tassa. "MuJoCo: A physics engine for model-based control". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.

[13] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji. "Benchmarking Protocols for Evaluating Small Parts Robotic Assembly Systems". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 883–889.

[14] M. Heo, Y. Lee, D. Lee, and J. J. Lim. "FurnitureBench: Reproducible Real-World Benchmark for Long-Horizon Complex Manipulation". In: *Robotics: Science and Systems*. 2023.

[15] Y. Mu, T. Chen, Z. Chen, S. Peng, Z. Lan, Z. Gao, Z. Liang, Q. Yu, Y. Zou, M. Xu, et al. "Robotwin: Dual-arm robot benchmark with generative digital twins". In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 27649–27660.

[16] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning". In: *Conference on Robot Learning*. PMLR. 2020, pp. 1094–1100.

[17] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine. "FMB: A Functional Manipulation Benchmark for Generalizable Robotic Learning". In: *The International Journal of Robotics Research* 44.4 (2025), pp. 592–606.

[18] I. Naramura, Y. Dong, H. Kaneko, T. Kurebayashi, M. Sudo, R. Itakura, S. P. Yamaguchi, M. Muromachi, and T. Inagaki. "Identifying Challenges and Validating Solutions for Dexterous Operations in Space Environments with Communication Delays". In: *2024 International Conference on Space Robotics (iSpaRo)*. 2024, pp. 349–356.

[19] C. Kaeser, N. Melenbrink, A. Karp, and J. Werfel. "Physical Simulation with Force Feedback Aids Robot Factors Design". In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 2025, pp. 9018–9024.

[20] NASA. *nasa/astrobee*. https://github.com/nasa/astrobee.

[21] JAXA. *jaxa/int-ball2_simulator*. https://github.com/jaxa/int-ball2_simulator.

[22] M. Diftler, J. Mehling, M. Abdallah, N. Radford, L. Bridgwater, A. Sanders, R. Askew, D. Linn, J. Yamokoski, F. Permenter, B. Hargrave, R. Platt, R. Savely, and R. Ambrose. "Robonaut 2 - The first humanoid robot in space". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 2178–2183.

[23] GITAI USA Inc. https://gitai.tech/2021/10/28/iss-tech-demo-ja/.

[24] M. Ekal and R. Ventura. "Disturbance effects observed during micro-gravity experiments on the Astrobee free-flyer and mitigation measures". In: *2024 International Conference on Space Robotics (iSpaRo)*. 2024, pp. 8–14.

[25] D. Hirano, S. Mitani, K. Watanabe, T. Nishishita, T. Yamamoto, and S. P. Yamaguchi. "Int-Ball2: On-Orbit Demonstration of Autonomous Intravehicular Flight and Docking for Image Capturing and Recharging". In: *IEEE Robotics & Automation Magazine* (2024), pp. 2–14.

[26] J. Santos, H. Dinkel, J. Di, P. Borges, M. Moreira, O. Alexandrov, B. Coltin, and T. Smith. "Unsupervised Change Detection for Space Habitats Using 3D Point Clouds". In: *AIAA SCITECH 2024 Forum*. 2024.

[27] S. Santra, K. Uno, G. Kudo, and K. Yoshida. “Risk-Aware Coverage Path Planning for Lunar Micro-Rovers Leveraging Global and Local Environmental Data”. In: *2024 International Conference on Space Robotics (iSpaRo)*. 2024, pp. 42–47.

[28] A. Richard, J. Kamohara, K. Uno, S. Santra, D. van der Meer, M. Olivares-Mendez, and K. Yoshida. “OmniLRS: A Photorealistic Simulator for Lunar Robotics”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 2024, pp. 16901–16907.

[29] V. Verma and C. Leger. “SSim: NASA Mars Rover Robotics Flight Software Simulation”. In: *2019 IEEE Aerospace Conference*. 2019, pp. 1–11.

[30] A. Orsula, A. Richard, M. Geist, M. A. Olivares Mendez, and C. Martinez Luna. “Towards Benchmarking Robotic Manipulation in Space”. In: *Conference on Robot Learning (CoRL) Workshop on Mastering Robot Manipulation in a World of Abundant Data (MRM-D)*. 2024.

[31] A. B. Mortensen and S. Bøgh. “RLRoverLAB: An Advanced Reinforcement Learning Suite for Planetary Rover Simulation and Training”. In: *2024 International Conference on Space Robotics (iSpaRo)*. IEEE. 2024, pp. 273–277.

[32] N. Koenig and A. Howard. “Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sendai, Japan, Sept. 2004, pp. 2149–2154.

[33] Open Robotics. *Robonaut 2*. https://app.gazebosim.org/OpenRobotics/fuel/models/Robonaut%202.

[34] Open Robotics. *ISS*. https://app.gazebosim.org/OpenRobotics/fuel/models/ISS.

[35] NVIDIA. *Isaac Sim*. https://github.com/isaac-sim/IsaacSim. Version 5.0.0.

[36] *moveit/mujoco_ros2_control*. https://github.com/moveit/mujoco_ros2_control.

[37] A. Sledd, M. Danford, and B. Key. “EXPRESS Rack: the extension of International Space Station resources for multi-discipline subrack payloads”. In: *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*. Vol. 1. 2003, pp. 1–35.

[38] The University of Alabama at Birmingham (UAB) Engineering and Innovative Technology Development (EITD). *MERLIN*. https://www.uab.edu/engineering/eitd/projects/cold-stowage/merlin.

[39] Vention. *Enclosed Ball Screw Actuator, 2295mm Length*. https://vention.io/parts/enclosed-ball-screw-actuator-2295mm-length-1287.

[40] Ewellix. *TLT*. https://www.ewellix.com/en/products/lifting-columns/tlt.

[41] A. Sharp, A. Lovan, E. Herrera, and E. Laske. “Practical End-Effector Development Through Task Interface Taxonomy Analysis”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 7: 46th Mechanisms and Robotics Conference (MR). Aug. 2022, V007T07A043.

[42] *ros/xacro*. https://github.com/ros/xacro.

[43] *MoveIt 2*. https://github.com/moveit/moveit2.

[44] I. A. Şucan, M. Moll, and L. E. Kavraki. “The Open Motion Planning Library”. In: *IEEE Robotics & Automation Magazine* 19.4 (Dec. 2012), pp. 72–82.

[45] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. “Learning Color Names for Real-World Applications”. In: *IEEE Transactions on Image Processing* 18.7 (2009), pp. 1512–1523.

[46] J. Ham, R. Soussan, B. Coltin, H. Chun, and P. Kim. “Drift-Free Visual Compass Leveraging Digital Twins for Cluttered Environments”. In: *IEEE Space Mission Challenges for Information Technology / Space Computing Conference Space Robotics Workshop*. 2025.

[47] C. Chamzas, C. Quintero-Pena, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki. “Motionbenchmaker: A tool to generate and benchmark motion planning datasets”. In: *IEEE Robotics and Automation Letters* 7.2 (2021), pp. 882–889.

[48] K. Zakka, Y. Tassa, and MuJoCo Menagerie Contributors. *MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo*. http://github.com/google-deepmind/mujoco_menagerie. 2022.