

SpaceOps-2025, ID #184

Space Teams: Digital Twins for Real-Time Operations

**Alexandr Sein^{a*}, Waylon Lee^b, Connor Jakubik^c, Neil McHenry^d, Nazareth Bedrossian, Ph.D.^e,
Michel Izygon, Ph.D.^f, Gregory Chamitoff, Ph.D.^g**

^a *Department of Aerospace Engineering, Texas A&M University, United States of America, asein@tamu.edu*

^b *Department of Aerospace Engineering, Texas A&M University, United States of America, waylonlee@tamu.edu*

^c *SimDynamX, United States of America, connor@simdynamx.com*

^d *SimDynamX, United States of America, neil@tamu.edu*

^e *SimDynamX, United States of America, naz@simdynamx.com*

^f *Tietronix Software, Inc., United States of America, mizygon@tietronix.com*

^g *Department of Aerospace Engineering, Texas A&M University, United States of America, chamitoff@tamu.edu*

* *Corresponding author*

Abstract

This paper introduces a novel digital twin paradigm for space mission design and real-time operations called the Mission Operations Digital Twin (MODT). MODT not only visualizes real-time vehicle states, but also simulates entire mission profiles, complex subsystems, and their interactions with the space environment. This platform is built on the Space Teams PRO (STP) application, and allows for mission design, training, and operations to all be conducted within the same environment, streamlining the entire mission execution process and providing rapid response, planning, and situational awareness during complex mission scenarios and spaceflight emergencies. Different specialists can operate and monitor their own parts of a broader mission through virtual reality or interactive computer interfaces. To demonstrate these capabilities, we use MODT to simulate a theoretical Artemis lunar surface operation, incorporating high-fidelity lunar terrain datasets, realistic gravity lighting conditions, and live data displays for both robot and astronaut crew member states. In the scenario, a nominal lunar survey mission with two extravehicular activity (EVA) crew members turns into a rescue operation when the crew's transport vehicle becomes damaged. Mission control operators must determine which mission goals to sacrifice in order to return the crew safely to their habitat. Using this example, we show how MODT can provide in-depth situational awareness for both operators and crew members, all within the same digital twin environment that was used to design the mission.

Acronyms

DUST	DLES Unreal Simulation Tool.
EDGE	Engineering DOUG Graphics for Exploration.
EVAs	Extra-Vehicular Activities.
GMRAD	Georeferenced Multi-Resolution Arbitrary Datasets.
HITL	Human-in-the-Loop.
JEOD	JSC Engineering Orbital Dynamics.
LTV	Lunar Terrain Vehicle.
MCC	Mission Control Center.
MODT	Mission Operations Digital Twin.
PSR	Permanently Shadowed Region.
ROS	Robot Operating System.
SAR	Search and Rescue.
STK	Systems ToolKit.
STP	Space Teams PRO.
SysML	System Modeling Language.
UE5	Unreal Engine 5.
XOSS	XR Operations Support System.

1. Introduction

A digital twin is a virtual representation of an existing or hypothetical real-world object or system. Digital twins are used extensively in manufacturing and construction, but also have applications across many industries. For example, they can be used to visualize real-world systems in an unconstrained virtual environment, to validate the accuracy and efficiency of a manufacturing process, or to detect potential faults in a construction project during the design phase [1]. In the aerospace industry, aircraft digital twins can be used to assist technicians with maintenance procedures or to simulate entire vehicle designs.

With the history of digital twins in mind [2], we aim to extend this evolving paradigm into modern space mission operations. As NASA and other space agencies look to establish a permanent human presence on the Moon and beyond [3], there is a growing need for integrated digital twin environments that allow for the direct, end-to-end simulation of entire Human-in-the-Loop (HITL) missions. The digital twins used in more mature industries are high-fidelity models of real systems and industrial processes [4][5]. As similar infrastructures are developed on the Moon, they will require a new paradigm of digital twin that incorporates the space environment into the simulation of these emerging technologies. It must also accommodate real-time decision-making from both the crew and mission control, and the results of their actions must affect the simulation in realistic and physically-correct ways. In this paper, we show the process of creating Mission Operations Digital Twin (MODT)*, a digital twin platform that satisfies these overarching goals, and then evaluate its success in simulating a hypothetical HITL lunar mission.

2. Background

MODT is a software package that can be used on its own or in conjunction with external software. In order to ensure that it can meet the overarching goal of integrated, end-to-end mission simulation, we present a review of existing software that could potentially support space digital twin architectures. Important capabilities include planetary terrain visualization and rendering, orbital mechanics, space environment modeling, HITL simulation, event and mission management, scripting interfaces, and real-time, multi-user support. There are many existing simulation platforms that satisfy some of these requirements to varying degrees; they are listed below, and their features are compared in Table 2.

- Space Teams PRO (STP): High fidelity physics and Unreal Engine 5 (UE5) visual simulation of space systems and end-to-end missions.
- NVIDIA Omniverse [6]: Visualization and modeling platform for digital twins.
- FreeFlyer [7]: Astrodynamics mission design and analysis platform.
- Systems ToolKit (STK) [8]: Satellite analysis and mission planning software.
- Trick [9]: Generic simulation environment for space system modeling and development.
- Engineering DOUG Graphics for Exploration (EDGE) [10]: 3D visualization tool with direct Trick compatibility.
- DLES Unreal Simulation Tool (DUST) [11]: UE5-based 3D lunar environment visualization.
- XR Operations Support System (XOSS) [12]: UE5-based 3D visualization of lunar and martian environments designed for hyper-realistic XR operations.

MODT was built using Space Teams PRO paired with Omniverse because STP is the only software that supports all the relevant MODT features natively, while Omniverse provides interfaces to numerous external data sources. While other programs and simulation toolkits support some requirements, we have found no other program that covers as broad a range of necessary MODT capabilities. STP is a dedicated space simulation platform, but also has a native capability to interface with a wide array of external programs. In addition to Omniverse, STP also has APIs for Robot Operating System (ROS), Trick, MATLAB, C++, Python, and more. A Trick simulation could be used to drive dynamic states within STP, while a Python script could simultaneously export orbital ephemerides for post-processing in FreeFlyer. However, these external programs are not necessary for MODT to function. Packages such as DUST support high-fidelity terrain visualization, but need an external interface to Trick or another dedicated simulation engine in order to

*MODT was developed with the support of a NASA Phase I Small Business Technology Transfer (STTR).

Table 2. Comparison of existing simulation platforms for MODT support.

Simulation Software	Rendering	Space Environment	HITL Operations	Multi-User/VR	Resource Modeling	Monte Carlo Analysis	Scripting and APIs
STP (Sim-DynamX)	Photo-realistic, full-solar-system rendering, including user-created models.	Integrators, force models, and NAIF SPICE ephemerides	Native HITL support.	Native multi-user and virtual reality (VR) options.	Multi-physics modeling for fluids, power, thermals, etc.	Monte Carlo tools for all simulations.	Cross-platform C++ and/or Python scripting.
Omniverse (NVIDIA)	Any scenario (user-defined).	No native capabilities.	No native capabilities.	Native VR support.	No native capabilities.	No native capabilities.	No native capabilities.
FreeFlyer (a.i. solutions)	Orbits, ground tracks, and simple satellite models.	Integrators, force models, and NAIF SPICE ephemerides	No native capabilities.	No native capabilities.	Spacecraft fuel only.	No native capabilities.	Custom scripting language; Windows and Linux only.
STK (ANSYS)	Orbits, ground tracks, and simple satellite models.	Integrators, force models, and NAIF SPICE ephemerides	No native capabilities.	No native capabilities.	Spacecraft fuel only.	No native capabilities.	No scripting; Windows-only.
Trick (NASA)	Data plotting tools.	High-fidelity integrators.	No native capabilities.	No native capabilities.	Only with external plugins.	Some native capabilities.	C++ and Python scripting required; Linux and Mac only.
EDGE (NASA)	Configurable, semi-realistic 3D rendering.	ISS and Moon only.	Native HITL support.	Native multi-user and VR support.	No native capabilities.	No native capabilities.	No scripting; cross-platform.
DUST (NASA)	Realistic rendering of space environments and vehicles.	Orbital mechanics with external plugins.	Native HITL support.	Native multi-user and VR support.	No native capabilities.	No native capabilities.	Unreal Engine 5 scripting; cross-platform.
XOSS (Buendea/ NASA)	Realistic rendering of select space environments.	Lunar and Martian surface features.	Native HITL support.	Native multi-user and VR support.	No native capabilities.	No native capabilities.	Unreal Engine 5 scripting; Windows-only.

support vehicle dynamics or orbital mechanics. Current MODT development was done within STP precisely because it natively supports these capabilities and more.

3. MODT Architecture

While generic interoperability in space simulation is the core design philosophy behind MODT, there are also specific features within the program that bolster the creation of space mission digital twin architectures.

3.1 Planetary Surface Rendering

MODT can simulate and render the surfaces of all planetary bodies in the solar system simultaneously. It does so using the built-in Georeferenced Multi-Resolution Arbitrary Datasets (GMRAD) system, which reads pre-processed altimetry data at runtime to generate entire planetary surfaces at an arbitrarily-high resolution[†]. Three examples of GMRAD planets are shown in Figures 1–3; these images were all taken in the same solar-system-wide simulation, showcasing the system’s ability to dynamically increase and decrease the rendered resolution of a planet based on the user’s position relative to it. GMRAD terrain has been validated at select locations, and has shown remarkable accuracy when compared to real sites. One example at Big Bend National Park is shown in Figures 4 and 5.

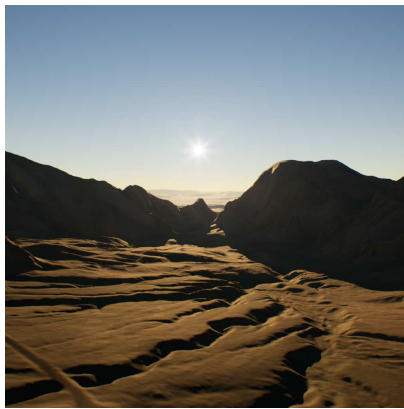


Fig. 1. Big Bend National Park, Earth. Dataset: 2 m per pixel.

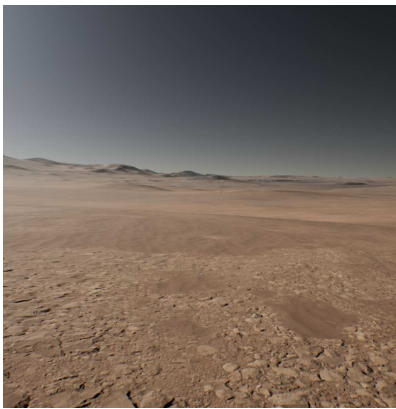


Fig. 2. Jezero Crater, Mars. Dataset: 50 cm per pixel.



Fig. 3. Nobile Rim 1, Moon. Dataset: 20 cm per pixel.



Fig. 4. South Rim at Big Bend National Park, real photo from January 18, 2025.

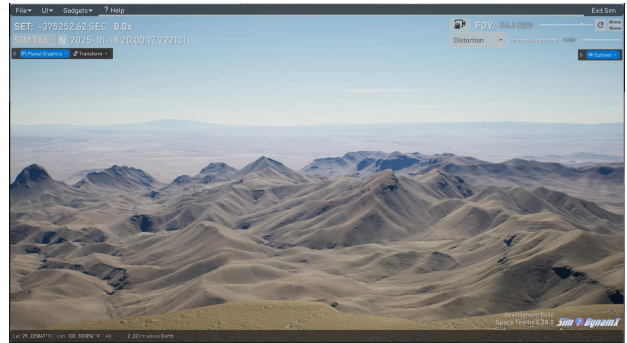


Fig. 5. South Rim at Big Bend National Park, visualized in MODT.

[†]The current resolution limit is 1 cm, which is far lower than any real altimetry dataset.

3.2 Field Effects

Field Effects in MODT are a global way to access simulation-wide models that can be represented as continuous scalar or vector fields. These can include true fields, such as the gravitational and magnetic fields, or pseudo-fields, such as atmospheric properties or the distribution of galactic cosmic radiation in the solar system. The default gravity models, which have all been validated against analytical examples or ephemerides, include:

- Spherical gravity: this is used for point-mass gravitational sources.
- Spherical harmonic gravity: this is used for asymmetric, “lumpy” bodies. Current high-fidelity models include a 360×360 field for Earth, a 150×150 field for the Moon, and a 110×110 field for Mars. All coefficients are obtained from the open-source JSC Engineering Orbital Dynamics (JEOD) software package. [13].
- Parameterized post-Newtonian approximation: an n -body gravity model that uses a first-order approximation of general relativity to model the interactions among all bodies. Our implementation follows the model used to generate the NAIF SPICE DE440 and DE441 planetary ephemerides [14].
- Full general relativity: this includes options for Schwarzschild and Kerr metrics to model an object’s gravitational field, with the geodesic equations expressed in coordinate time [15] and the Christoffel symbols obtained from the Catalogue of Spacetimes [16].

The default magnetic field models (which all follow the spherical harmonic potential model described in [17]) include Mercury [18] [19] [20], Earth [21], Jupiter [22] [23] [24], Ganymede [25], Saturn [26] [27], Uranus [28], and Neptune [29]. All coefficients for these models are taken from peer-reviewed literature or conference papers from subject matter experts.

3.3 Orbital Mechanics

MODT contains a suite of orbital mechanics tools that aid users in visualizing and designing trajectories. Trajectories can be propagated forward in time independently from the simulation time, and any user-defined force models (including field effects) can be used for the trajectory propagation. Additionally, basic targeting algorithms (e.g., Lambert’s algorithm [30]) allow users to design missions around the solar system and then fly those missions, all in the same simulation. Figure 6 shows an example rendezvous mission in Earth orbit. The trajectory can be visualized from the start to the end of the mission, and the orbit lines are drawn with proper perspective in the full 3D world of MODT.

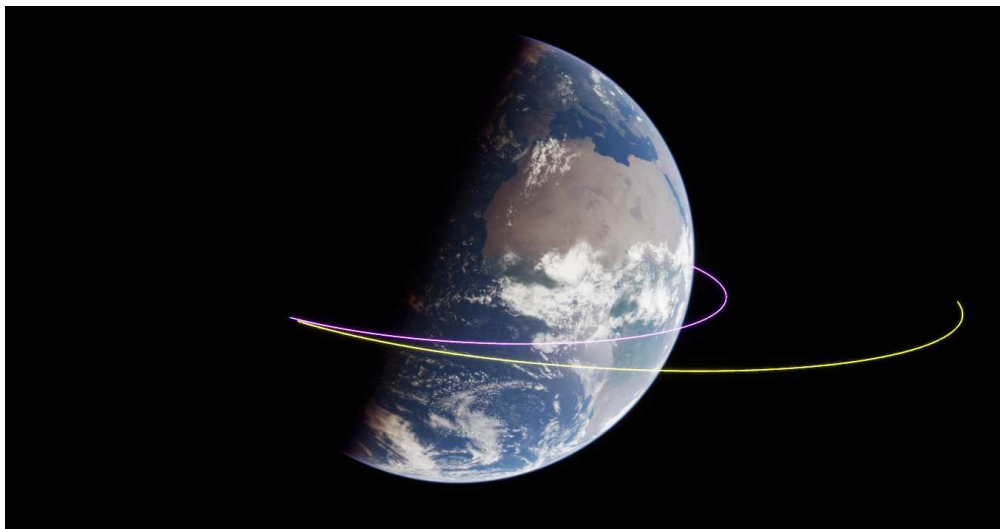


Fig. 6. Orbital trajectory visualization in MODT. In this example, the spacecraft on the yellow orbit executes a rendezvous maneuver with the spacecraft on the magenta orbit.

3.4 Precision in Visualization

MODT leverages the advanced real-time rendering capabilities of UE5 for all space environment visualization. Since both MODT and UE5 use 64-bit-precision coordinate systems, MODT can be used to render objects at both microscopic and solar system scales simultaneously. Another core feature is the built-in space-based reference frame system, which forms the backbone of all object transformations within a simulation. This system supports acceleration-level kinematics in any (inertial or rotating) frame within the solar system, and also supports coordinate basis transformations from one frame to another. It also includes a dynamic, user-controllable “graphics frame” that is always aligned to and centered on UE5’s “global frame.” This allows the user to minimize coordinate precision issues by seamlessly relocating the origin.

3.5 Asynchronous Physics and Multiphysics Simulation

MODT features a fully multi-threaded architecture that enables field effects, physics interactions, and user-written code to all run asynchronously. While this methodology allows for unprecedented computational speed in all simulations, it can also potentially result in non-deterministic physics and, more broadly, non-repeatable simulation results. We address this issue in several ways. First, MODT has a “deterministic mode,” where all threads are guaranteed to execute their actions at perfectly-scheduled times. This deterministic mode allows MODT to behave more like a classical simulator, but sacrifices the parallelism of the default MODT architecture. Second, MODT has low-level, automatic state extrapolation that attempts to sync object states (e.g., position, velocity, rotation, etc.) to a sensible “time context” any time those variables are extracted from the simulation. For example, when a user calls the function to retrieve the velocity of an object, MODT does the following:

1. Retrieves the stored velocity \mathbf{v}_s and stored acceleration \mathbf{a}_s values.
2. Retrieves the “timestamp” t_s associated with those stored values.
3. Computes the time difference between the current, thread-local time t and the stored timestamp: $\Delta t = t - t_s$
4. Extrapolates the thread-local velocity. For example, linear extrapolation would be: $\mathbf{v} = \mathbf{v}_s + \mathbf{a}_s \Delta t$

This methodology helps to minimize errors due to slight differences in thread timing and, combined with frequent syncing of thread-local times, results in a more accurate multi-threaded simulation that still supports the same physics interactions that typically require a single-threaded (or deterministic) simulation.

MODT also has robust, configurable, built-in multi-physics solvers [31] for heat transfer and thermal radiation, fluid flow and gas dynamics, and power and electronics along nodal networks. This resource simulation capability enables end-to-end mission resource management and analysis, including aspects such as in-situ resource utilization, recycling, and resource planning. For example, the availability and distribution of power, water, and oxygen can be tracked during an end-to-end mission, highlighting potential deficiencies in resource distribution. This capability is central to MODT, as it allows for the modeling of systems in a more classical digital-twin framework. Additionally, the ability to connect different networks of solvers together (e.g., a power simulation driving the states and efficiencies of an environmental control and life support system simulation) allows for the direct simulation of emergent effects, such as a partial loss of power affecting the efficiency of a carbon dioxide scrubber.

3.6 Custom Model Import and External Interfaces

Digital twins often include 3D representations of real systems, and MODT is no different. When CAD or 3D models related to a given operation are available, they can be imported into MODT at runtime. An imported model contains mesh data (all vertices and faces representing the object) as well as user-defined materials that control how the model looks in the simulation under various lighting conditions. CAD models, which differ from 3D models in their precision and parametric (as opposed to vertex-based) geometry definitions, can also be imported into MODT as long as they are processed into a valid file format. CAD models often come with supporting documentation, which can be linked and accessed through MODT directly. This feature directly mirrors typical digital twin functionality, and supports the real-time analysis aspect of MODT. In the example in Figure 7, a user examines supporting documentation for a Lunar Terrain Vehicle (LTV) imported into MODT at runtime.

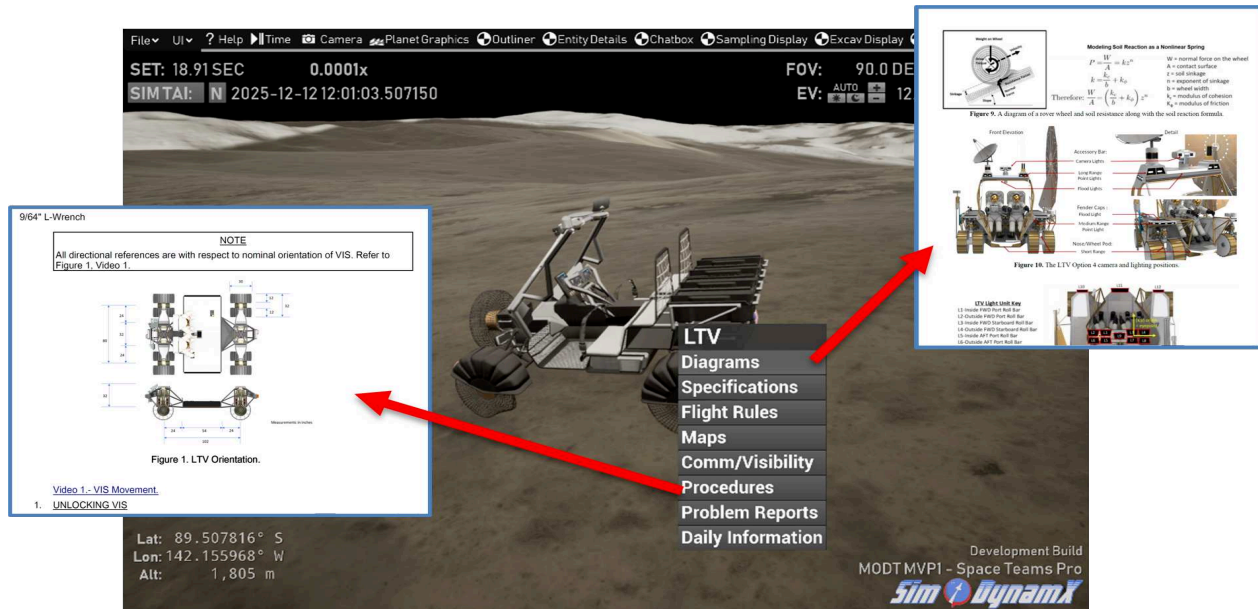


Fig. 7. An example LTV in MODT, alongside supporting documentation that is directly accessible from within the simulation.

3.7 User APIs and Analysis Tools

MODT users can write custom “systems” in Python and/or C++. These systems can be written to affect almost anything in a simulation, and are the basis for many MODT-specific features demonstrated in this work. While C++ systems offer faster execution, Python systems are of particular interest because they require no prior compilation before running a simulation. This allows for rapid iteration on the initial conditions and operational procedures within any simulation, which is a valuable MODT capability.

The ability to analyze and compare mission scenarios is an important application of MODT. This is accomplished with analysis tools that can parameterize almost any aspect of a system design or operational sequence and perform a Monte Carlo or sensitivity analysis. A configurable Monte Carlo interface allows any set of simulation parameters to be scattered; simulations using these parameters can then be run in parallel, and results can be plotted directly within MODT. Additionally, the multi-threaded nature of MODT allows for simulations to be run up to ten times faster than real-time[‡] while maintaining accurate representations of simulation states. This means that Monte Carlo analyses can be conducted in minutes instead of hours, empowering rapid and informed real-time decision-making.

MODT also supports terrain evaluation tools. GMRAD planets can be configured to display terrain elevation and slope, with user-defined minimum and maximum altitudes and slopes. Two examples of these visualization modes are shown in Figures 8 and 9. These visualization modes can be activated at any point during a simulation, and they allow operators to visually validate terrain-related conditions and constraints. More advanced tools include precision lighting, communication line-of-sight, and horizon mapping for surface navigation [32]. Additional tools for space and surface vision-based navigation and other applications are in progress.

3.8 Event and Behavior Management

Simulating operations requires robust management of the behaviors of all simulated objects in response to planned and emergent events. MODT has a built-in event subsystem that allows any C++ or Python system to “dispatch” events and also subscribe “listeners” for those events. Since events are global, any system (or process) can subscribe to an event from any other system. This creates the basis for all communications between objects in the simulation, which is an especially important capability for mission operations. Additionally, users can program objects in the simulation (known as “Entities”) to have specific responses to events: these are known as “entity behaviors.” Working in parallel

[‡]We limited the rate to ten times in this work; in future work, we aim to improve this to hundreds of times the real-time simulation speed.

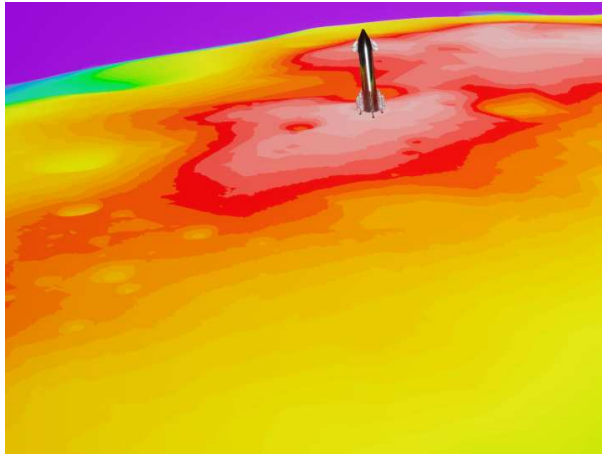


Fig. 8. GMRAD terrain altitude visualization (50 meter tall vehicle for scale).

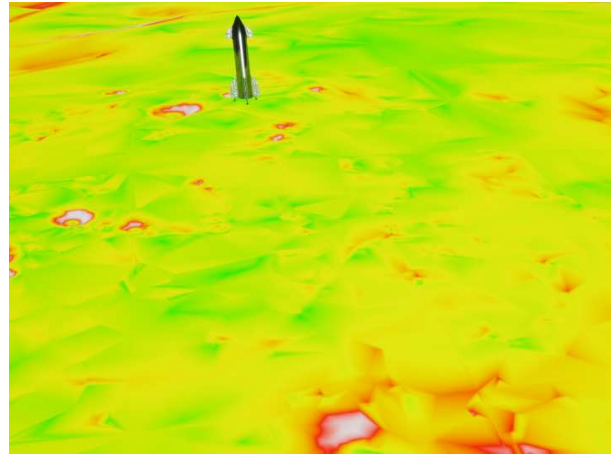


Fig. 9. GMRAD terrain slope visualization (50 meter tall vehicle for scale).

with global events, entity behaviors allow a simulation to evolve on its own from a set of initial conditions, as well as to ingest and respond to user input at any time. Overall, these features of MODT enable real-time, informed decision-making for mission operators.

4. Operational Scenario: Lunar Search and Rescue

This section presents a realistic scenario for part of a lunar mission in order to demonstrate the utility of MODT for mission analysis and decision making. The first lunar Extra-Vehicular Activities (EVAs) in over fifty years are planned to occur during NASA's Artemis III mission, and subsequent crewed missions are expected to increase the number and frequency of surface EVAs. With LTVs entering the scene by Artemis IV or V, EVAs will traverse increasingly longer paths, and go further and further from landers and habitats. Mission operators and crew members alike must be prepared for off-nominal scenarios that may occur far from the safety of the lunar habitat. In these scenarios, the only way to save the stranded crew members may be to deploy a Search and Rescue (SAR) mission in real time.

4.1 Scenario Overview

This concept is the basis for our MODT validation tests. The scenario centers on a distant, LTV-based EVA, with the crew prospecting for water-ice in and around a PSR. They are also accompanied by a tele-operated sampling rover, and are too far from the surface habitat to walk back. A wheel on the LTV suddenly fails—the LTV now cannot be used to continue the nominal operations nor to return the crew to the habitat. The crew reports the wheel failure to Mission Control Center (MCC), and then inspects it thoroughly. The team comes to the conclusion that it cannot be repaired, and there is no spare wheel onboard their LTV. The LTV is now stuck inside a PSR, and the crew's only hope of getting home is for the backup crew at the habitat to mount a rescue mission[§]. A condensed visualization of this scenario is shown in Figure 10; the relevant entities are shown in close proximity to each other, but in reality, they are many kilometers apart on the lunar surface.

[§]The crew also cannot ride the sampling rover back to the habitat as it is not designed to support both of their weights, and also has no designated seats.

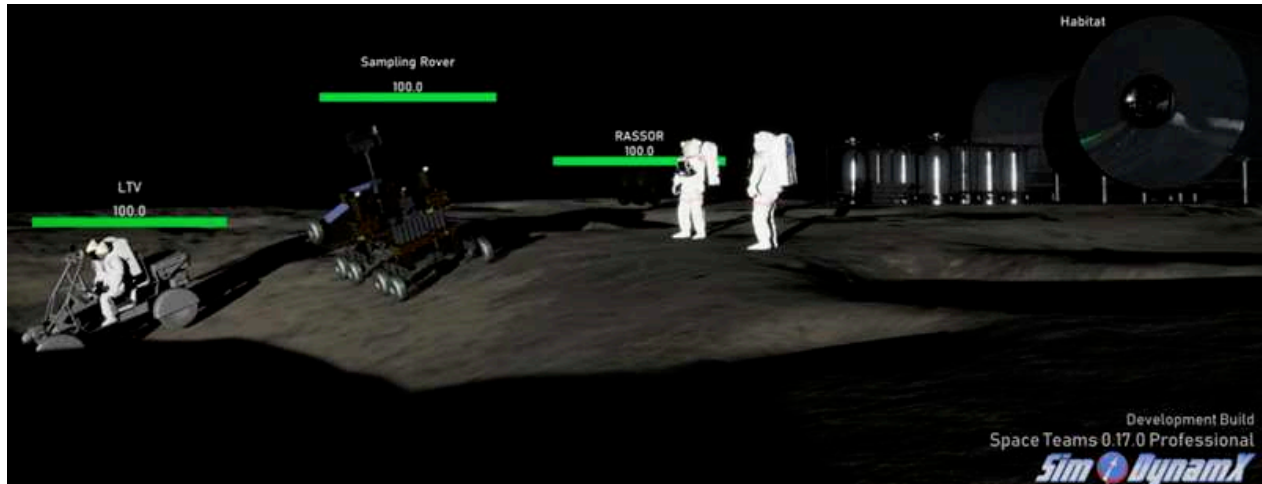


Fig. 10. SAR scenario with two EVA crew members stuck in a Permanently Shadowed Region (PSR) after an LTV wheel failure. The LTV, sampling rover, and habitat models are shown in a realistic lunar surface environment in MODT.

Certain mission- and vehicle-specific constraints help to define the scope of potential SAR options, and also determine the complexity of the operational decisions. First, the sampling rover is typically left on the rim of a crater between EVAs. It can survive here for over a week in “safe” mode, but cannot last more than two days inside a PSR (which is where it is at the start of the off-nominal scenario). Nominal sampling rover operations are power-intensive, so the typical EVA termination procedure requires the crew to replace the sampling rover’s battery before setting it to “safe” mode outside of a PSR. On the other hand, LTVs can survive for much longer inside PSRs (over two weeks), but again only in “safe” mode. Fortunately, the backup LTV at the habitat is fully-charged, and there is a compatible spare wheel there as well.

4.2 Operational Considerations

The moment the crew indicates that there is a failure in their LTV, MCC begins considering the options for returning the EVA crew members safely to their habitat. Before they begin designing (or even considering) a SAR mission, they must answer some critical status questions. First, how much power and oxygen does each EVA crew member have? Can they walk back to the habitat within the flight rules for minimum power and oxygen? Second, can the wheel be fixed or replaced in-situ, and if so, how long would it take? Lastly, if needed, can the backup crew mount a rescue operation?

After considering the flight rules and operational constraints, MCC elects to send a tele-operated backup LTV with a spare wheel. If the original LTV can be fixed, the crew will fix it and ride it back to the habitat. If the LTV cannot be fixed, the crew will abandon it and drive the spare LTV back to the habitat. In a sense, there is security in this mission plan, as the crew should be able to return to the habitat no matter which solution is ultimately implemented. However, the trade-off lies in whether there will be enough time to both fix the LTV and safe the sampling rover, or if the crew will have to choose which one to sacrifice. The key to this scenario is that there are many operational sequences to choose from; these are discussed in Section 4.4. Additionally, there is no flight rule for which vehicle should be sacrificed first; MCC must now evaluate several scenario variations on a concrete set of metrics in order to determine which scenario actually sacrifices the least overall mission objectives.

In order to validate that the MODT can assist in this decision-making, each variation is simulated directly from start to end. The crew are simulated as autonomous agents, with pre-programmed entity behaviors that limit their actions to a predictable, realistic set. The vehicle systems are simulated directly; for example, an LTV’s power draw can be programmed to depend on the grade of the slope it traverses. However, realistic vehicle dynamics are left out of the simulation in favor of straight-line paths that “snap” to the lunar terrain. Not simulating vehicle dynamics allows the simulation to be run many times faster than real-time, which in turn allows for faster evaluation of each mission variation without sacrificing a significant part of each one. Each scenario is given a score after the astronauts are rescued and

the mission is deemed “complete.” The score is based on remaining EVA power and oxygen (compared to flight rule baselines), as well as completed mission objectives (e.g., safing the sampling rover prior to returning to the habitat).

4.3 Scenario Configuration

For this demonstration, scenarios were configured through a scenario configuration graphical user interface (GUI) and through basic Python scripting. More advanced functionality, such as modifying performance metrics or changing Monte Carlo input variables, can be accessed via Python scripts.

4.4 Scenario Variations

We present 5 variations to the scenario described above that MCC might realistically consider and that MODT can be used to evaluate; these are listed in Table 3. The final two variations are of particular interest, as they motivate and then implement Monte Carlo analysis in the MODT.

Table 3. Scenario variations for MODT validation.

Variation	Description	Desired Outcomes
Case 1	The EVA crew members fix the LTV, then replace the sampling rover battery and safe it, and finally return to the habitat.	Based on the initial conditions of the scenario when the wheel failure occurs, it turns out that this variation is not feasible to carry out every task as-written without the EVA crew members’ oxygen and power falling under the flight-rule-mandated minimums.
Case 2	The EVA crew members abandon the failed LTV, use the backup LTV to reach and safe the sampling rover, then return to the habitat.	This is the first of two “sacrifice” variations, and is designed to evaluate the cost of losing the LTV entirely.
Case 3	The EVA crew members fix the LTV and return to the habitat, abandoning the sampling rover entirely.	This is the second of two “sacrifice” variations, and is designed to evaluate the cost of losing the sampling rover entirely.
Case 4	While the EVA crew members are fixing the LTV, they also remotely drive the sampling rover toward themselves. After fixing the LTV, they attempt to safe the sampling rover. The sampling rover’s proximity to them improves the likelihood that they have enough time to safe it on the way back to the habitat.	In this variation, MCC attempts to implement a compromise to avoid sacrificing <i>either</i> vehicle. It is designed as a precursor to the final variation, and shows the capability within the MODT to pause a simulation, save it at that time, and then play it from that time with slightly different inputs.
Case 5	Monte Carlo analysis of the most realistic case, where multiple unknowns within known ranges must be evaluated against each other. In this variation, the crew remotely drives the sampling rover in the direction of the habitat until a minimum power level is achieved (5–12%, or 10–24 hours of survival time).	The sampling rover power level is one of the independent (input) variables in the Monte Carlo analysis. The other two input variables are the remaining EVA oxygen and power levels at the time that the LTV repair is complete. While it is true that remaining oxygen and power depend on the time it takes to repair the failed LTV, we implement them as independent variables to account for slight variations in actual oxygen and power usage during a potentially strenuous repair operation.

The Monte Carlo analysis for Case 5 is shown in separate 3D plots. There are three inputs:

- Sampling rover power
- EVA starting oxygen
- EVA starting power

and there are two outputs:

- EVA final oxygen upon returning to the habitat
- EVA final power upon returning to the habitat

The inputs are plotted in three dimensions in both plots, and their respective outputs (oxygen in Figure 11 and power in Figure 12) are represented as points, with their size and color ranging large and light (best) to small and dark (worst). The motivation for this analysis is to help MCC understand the limits of the current situation. After analyzing the envelope of options, MCC can set constraints on how much remaining time the EVA crew members are allowed to spend repairing the LTV.

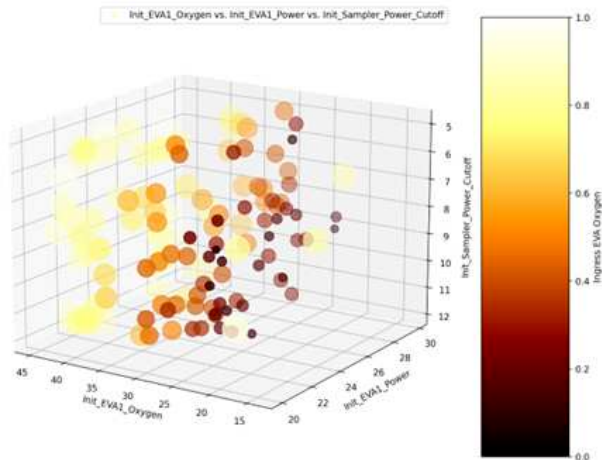


Fig. 11. Monte Carlo analysis for EVA oxygen at habitat ingress v. sampling rover power, initial EVA oxygen, and initial EVA power.

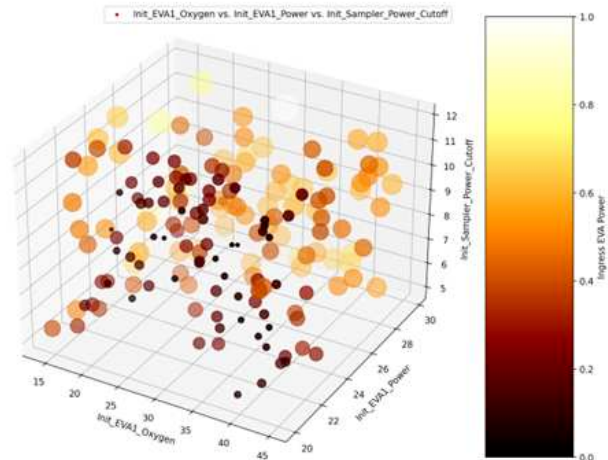


Fig. 12. Monte Carlo analysis for EVA power at habitat ingress v. sampling rover power, initial EVA oxygen, and initial EVA power.

In this case, the plots show that there is a cut-off point for the LTV repair that represents the best condition for completing every task successfully (23% oxygen, 21% suit power, and 12% sampling rover power). Beyond this point, the crew cannot complete every objective and there is a risk of violating flight rules. This result can be used to set a time limit on the LTV repair operations—if the wheel is still not repaired by the time oxygen drops below 23%, the EVA crew members abandon the repair and drive to the habitat using the backup LTV, attempting to safe the now optimally-positioned sampling rover on the way. An interesting secondary result is that the sampling rover power (i.e., its resting position) has little effect on the final EVA oxygen and power. This is likely because the LTV is much faster than the sampling rover, which means that as long as the sampling rover gets close to the LTV path, the crew will be able to make a small detour to safe it. In this scenario, it's clearly beneficial to leave the sampling rover at the highest possible power level to increase its chances of survival, even if that means the crew has to go slightly out of their way to reach it.

5. Results

Evaluation of the SAR scenario described above using MODT demonstrated the capability of end-to-end mission design, simulation, and evaluation. To begin, the entire mission scenario was designed in MODT. First, a site on the lunar surface was selected by exploring the terrain near the south pole in MODT. After the site was selected, 3D models for vehicles, EVA crew members, and surface infrastructure were imported into the simulation and placed on the lunar terrain. The behaviors for all entities were programmed entirely within MODT, while externally-written flight rules were seamlessly integrated into mission constraints. The scenario variations were created in phases, and each step of the mission (including all events and entity-to-entity communication) was validated in real time in the simulation.

We also demonstrated the capability of MODT to act as a real-time situational awareness and decision-making tool for mission operations. For each scenario, we were able to run hundreds of Monte Carlo simulations[¶] and produce informative plots of the results. In a real emergency, there is often very little time to deliberate, and MCC must make a decision in real-time with the available information. The rapid Monte Carlo capability provides an opportunity for flight controllers to make more informed decisions during off-nominal scenarios by asking “what if” questions and quickly evaluating the outcome for operational choices. During a real mission, the overall MODT mission simulation would already be running in MCC in real-time. At the moment an off-nominal event occurred, MCC would log the relevant variables and start Monte Carlo runs to evaluate the effect of those variables on the mission. The simulations would all run in parallel, simulating the mission hours into the future while the flight controllers discussed the situation with the crew. Within minutes, MCC would have an improved understanding of the future of the mission as a function of their current choices, allowing them to inform the crew of the best course of action to maximize crew safety and mission success.

6. Conclusions and Future Work

In this work, we motivated the concept of a space mission operations digital twin (MODT), discussed its main features, and presented test scenarios used to demonstrate its capabilities for operational decision-making. We showed that an integrated design and simulation program such as MODT can be created and used for end-to-end mission design, simulation, evaluation, and operations.

In the future, we intend to expand MODT from a demonstration of the concept to a fully-functional mission evaluation and situational awareness tool. While the current version can already support a wide variety of lunar surface scenarios, it will require several key modifications and innovations. First, we plan to improve the MODT event management system to align more closely with protocols used in other industry standard programs. This would allow for greater compatibility with external tools such as ROS and System Modeling Language (SysML). Second, we aim to expand the set of built-in tools in STP to better support MODT capabilities in three primary focus areas:

- Team situational awareness: for example, visualizing real-time operations on displays and in virtual reality based on live telemetry.
- Custom operator virtual workspaces: for example, allowing operators to customize virtual views of systems and operations they are focused on monitoring, and accessing tools and data within the same environment.
- Advanced decision-making tools: for example, having the ability to examine a wide range of parameters that define operational, design, and task-sequence options to produce the most useful performance measures for mission success.

Lastly, we intend to demonstrate these new capabilities and improved MODT support with more sophisticated simulations, including a dedicated situational awareness simulation occurring during highly dynamic events such as rendezvous or landing.

Acknowledgements

The authors would like to thank everyone at the Texas A&M ASTRO Lab, SimDynamX, and Tietronix for their continued support and contributions to this work.

[¶]In this project, these simulations took several hours to run. Although every run was “headless”—that is, the simulation was not running the graphics engine in the background—each one took approximately ten minutes to run end-to-end. In the future, this performance will be improved with proper parallelization and simulation optimization.

References

- [1] M. Schluse and J. Rossmann, "From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems," in *2016 IEEE International Symposium on Systems Engineering*, Oct. 2016, pp. 1–6. doi: 10.1109/SysEng.2016.7753162.
- [2] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *Journal of Manufacturing Systems*, vol. 58, pp. 346–361, 2021, Digital Twin towards Smart Manufacturing and Industry 4.0, issn: 0278-6125. doi: <https://doi.org/10.1016/j.jmsy.2020.06.017>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612520301072>.
- [3] A. Bowman and D. Baird, *Moon to mars architecture*, Jan. 2025. [Online]. Available: <https://www.nasa.gov/moontomarsarchitecture/>.
- [4] L. Schreiber, "Supercharging the industry transformation with the comprehensive digital twin," Siemens, Tech. Rep., 2025. [Online]. Available: <https://assets.new.siemens.com/siemens/assets/api/uuid:cda2b14c-23bf-4685-8fdf-adb7fd25e3a7/Whitepaper-Digital-Twin-EN.pdf>.
- [5] ANSYS, "Ansys digital twin," ANSYS, Tech. Rep., 2024. [Online]. Available: <https://www.ansys.com/content/dam/product/digital-twin/twin-builder/ansys-digital-twins-technical-datasheet.pdf>.
- [6] N. Corporation, *Nvidia omniverse*, 2025. [Online]. Available: <https://www.nvidia.com/en-us/omniverse/>.
- [7] a.i. solutions, *Leverage the power and flexibility of freeflyer astrodynamics software in your next mission*, 2022. [Online]. Available: ai-solutions.com/wp-content/uploads/2022/07/FreeFlyer_Capabilities.pdf.
- [8] ANSYS, *Ansys stk: Software for digital mission engineering and systems analysis*, 2025. [Online]. Available: <https://www.ansys.com/products/missions/ansys-stk#tab1-2>.
- [9] J. Penn and A. Lin, "The trick simulation toolkit: A nasa/opensource framework for running time based physics models," in *AIAA Modeling and Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics, 2016. doi: 10.2514/6.2016-1187. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2016-1187>. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2016-1187>.
- [10] N. J. VR-Lab, *Dynamic onboard ubiquitous graphics (doug)*, Sep. 2018. [Online]. Available: www.nasa.gov/sites/default/files/atoms/files/doug_user_guide_doc_final_0.pdf.
- [11] L. Bingham, J. Kincaid, B. Weno, N. Davis, E. Paddock, and C. Foreman, "Digital lunar exploration sites unreal simulation tool (dust)," in *2023 IEEE Aerospace Conference*, 2023, pp. 1–12. doi: 10.1109/AER055745.2023.10115607.
- [12] Buendea, *Xoss*, 2025. [Online]. Available: <https://www.buendea.com/xoss>.
- [13] N. JSC, *Jsc engineering orbital dynamics (jeod)*, 2025. [Online]. Available: <https://github.com/nasa/jeod/>.
- [14] R. S. Park, W. M. Folkner, J. G. Williams, and D. H. Boggs, "The jpl planetary and lunar ephemerides de440 and de441," *The Astronomical Journal*, vol. 161, no. 3, p. 105, Feb. 2021. doi: 10.3847/1538-3881/abd414. [Online]. Available: <https://dx.doi.org/10.3847/1538-3881/abd414>.
- [15] C. M. Will, *Theory and Experiment in Gravitational Physics*, 2nd ed. Cambridge University Press, 2018.
- [16] T. Mueller and F. Grave, *Catalogue of spacetimes*, 2010. arXiv: 0904.4184 [gr-qc]. [Online]. Available: <https://arxiv.org/abs/0904.4184>.
- [17] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12.
- [18] B. J. Anderson *et al.*, "Low-degree structure in mercury's planetary magnetic field," *Journal of Geophysical Research: Planets*, vol. 117, no. E12, 2012. doi: <https://doi.org/10.1029/2012JE004159>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2012JE004159>. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012JE004159>.

- [19] I. Wardinski, B. Langlais, and E. Thebault, “Time-varying magnetic fields of mercury,” in *European Planetary Science Congress*, Sep. 2018, EPSC2018-1209, EPSC2018–1209.
- [20] I. Wardinski, B. Langlais, and E. Thébault, “Correlated time-varying magnetic fields and the core size of mercury,” *Journal of Geophysical Research: Planets*, vol. 124, no. 8, pp. 2178–2197, 2019. doi: <https://doi.org/10.1029/2018JE005835>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018JE005835>. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JE005835>.
- [21] P. Alken *et al.*, “International geomagnetic reference field: The thirteenth generation,” *Earth, Planets and Space*, vol. 73, no. 1, p. 49, Feb. 2021, issn: 1880-5981. doi: 10.1186/s40623-020-01288-x. [Online]. Available: <https://doi.org/10.1186/s40623-020-01288-x>.
- [22] J. E. P. Connerney *et al.*, “A new model of jupiter’s magnetic field from juno’s first nine orbits,” *Geophysical Research Letters*, vol. 45, no. 6, pp. 2590–2596, 2018. doi: <https://doi.org/10.1002/2018GL077312>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2018GL077312>. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2018GL077312>.
- [23] J. E. P. Connerney *et al.*, “A new model of jupiter’s magnetic field at the completion of juno’s prime mission,” *Journal of Geophysical Research: Planets*, vol. 127, no. 2, e2021JE007055, 2022, e2021JE007055 2021JE007055. doi: <https://doi.org/10.1029/2021JE007055>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2021JE007055>. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021JE007055>.
- [24] J. E. P. Connerney, M. H. Acuña, N. F. Ness, and T. Satoh, “New models of jupiter’s magnetic field constrained by the io flux tube footprint,” *Journal of Geophysical Research: Space Physics*, vol. 103, no. A6, pp. 11 929–11 939, 1998. doi: <https://doi.org/10.1029/97JA03726>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/97JA03726>. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/97JA03726>.
- [25] M. Kivelson, K. Khurana, and M. Volwerk, “The permanent and inductive magnetic moments of ganymede,” *Icarus*, vol. 157, no. 2, pp. 507–522, 2002, issn: 0019-1035. doi: <https://doi.org/10.1006/icar.2002.6834>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S001910350296834X>.
- [26] M. K. Dougherty *et al.*, “Saturn’s magnetic field revealed by the cassini grand finale,” *Science*, vol. 362, no. 6410, eaat5434, 2018. doi: 10.1126/science.aat5434. eprint: <https://www.science.org/doi/pdf/10.1126/science.aat5434>. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.aat5434>.
- [27] M. Burton, M. Dougherty, and C. Russell, “Model of saturn’s internal planetary magnetic field based on cassini observations,” *Planetary and Space Science*, vol. 57, no. 14, pp. 1706–1713, 2009, issn: 0032-0633. doi: <https://doi.org/10.1016/j.pss.2009.04.008>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S003206330900107X>.
- [28] N. F. Ness *et al.*, “Magnetic fields at uranus,” *Science*, vol. 233, no. 4759, pp. 85–89, 1986. doi: 10.1126/science.233.4759.85. eprint: <https://www.science.org/doi/pdf/10.1126/science.233.4759.85>. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.233.4759.85>.
- [29] J. Connerney, M. H. Acuña, and N. F. Ness, “The magnetic field of neptune,” *Advances in Space Research*, vol. 12, no. 8, pp. 239–248, 1992, issn: 0273-1177. doi: [https://doi.org/10.1016/0273-1177\(92\)90394-D](https://doi.org/10.1016/0273-1177(92)90394-D). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/027311779290394D>.
- [30] R. H. Battin and R. M. Vaughan, “An elegant lambert algorithm,” *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 6, pp. 662–670, 1984. doi: 10.2514/3.19910. eprint: <https://doi.org/10.2514/3.19910>. [Online]. Available: <https://doi.org/10.2514/3.19910>.
- [31] C. Jakubik, L. Floyd, and G. E. Chamitoff, “Asynchronous multiphysics for spacecraft system modeling and simulation,” in *2024 IEEE Aerospace Conference*, 2024, pp. 1–9. doi: 10.1109/AERO58975.2024.10520948.
- [32] W. Lee, A. Sein, C. Jakubik, and G. Chamitoff, “Celestial and horizon matching navigation techniques for lunar surface operations,” in *AAS/AIAA Space Flight Mechanics Meeting*, Kauai: AAS/AIAA, Jan. 2025.