Preprint, accepted for publication in the *AIAA Journal*'s upcoming virtual collection on Smart and Resilient Lunar Habitats.

HabSim: A Modular Coupled Virtual Testbed for Simulating ExtraTerrestrial Habitat Systems

Mohsen Azimi*
Mississippi State University

Alana Lund†
University of Waterloo

Yuguang Fu‡
Nanyang Technological University

Herta Montoya§ and Luca Vaccino ¶, R. Murali Krishnan ‖ and Seungho Rhee**
Purdue University

Leila Chebbo††
University of Connecticut

Adnan Shahriar‡‡
The University of Texas-San Antonio

Zixin Wang§§
Purdue University

Amin Maghareh¶¶
Verus Research

Shirley J. Dyke***
Purdue University

Extraterrestrial habitats involve a tightly coupled combination of hardware, software, and humans while operating in an unforgiving environment that poses many risks, both anticipated and unanticipated. Traditional approaches with such systems-of-systems focus on reliability, robustness, and redundancy. These approaches seek to avoid failure rather than reduce overall risk. However, faults are inevitable, and understanding and managing the complex and emergent

*Assistant Professor, Department of Mechanical Engineering, Starkville, MS USA 39762, AIAA Member, azimi@me.msstate.edu (Corresponding Author)
†Assistant Professor, Department of Civil and Environmental Engineering, Engineering 2-2314, 200 University Ave. W., Waterloo, ON, Canada, N2V 2T1, alana.lund@uwaterloo.ca
‡Assistant Professor, School of Civil and Environmental Engineering, 50 Nanyang Ave, N1 01c-96m, Singapore 639798, yuguang.fu@ntu.edu.sg
§Graduate Research Assistant, Lyles School of Civil Engineering, West Lafayette, IN, 49707, AIAA Student Member, montoyah@purdue.edu
¶Graduate Research Assistant, School of Mechanical Engineering, AIAA Student Member, lvaccino@purdue.edu
‖Graduate Research Assistant, School of Mechanical Engineering, AIAA Student Member, mrajase@purdue.edu
**Graduate Research Assistant, School of Mechanical Engineering, West Lafayette, IN, 49707, AIAA Student Member, rhee18@purdue.edu
††Graduate Research Assistant, Department of Electrical and Computer Engineering, Storrs, CN, USA 06269, leila.chebbo@uconn.edu
‡‡Postdoctoral Researcher, Department of Mechanical Engineering, San Antonio, TX, USA 78249, adnan.shahriar@utsa.edu
§§Graduate Research Assistant, Lyles School of Civil Engineering, West Lafayette, IN, 49707, wang4591@purdue.edu
¶¶Senior Research Engineer-Control Systems, Albuquerque, NM, United States, amin.maghareh@verusresearch.net
***Donald A. and Patricia A. Coates Professor of Innovation in Mechanical Engineering, School of Mechanical Engineering and Lyles School of Civil and Construction Engineering, West Lafayette, IN, 49707, AIAA Member, sdyke@purdue.edu (Corresponding Author)

behavior and cascading events of such a complex system is critical. This study describes the development of HabSim, a computational simulation environment intended to support research to establish the know-how to design and operate resilient and autonomous SmartHabs. HabSim is a modular virtual testbed comprised of many of the coupled dynamical systems expected in a typical SmartHab. A heterogeneous set of interconnected physics-based and phenomenological models are used to represent the essential functions of a SmartHab. HabSim further considers disruptions, and models damage and repair of certain components. This paper discusses: (a) system and subsystem requirements of the deep space habitat included in the HabSim platform; (b) architectural choices made in response to the requirements; (c) technical considerations for developing, verifying, configuring, and executing HabSim; and (d) illustrative sample results from a simulation of a representative disruption scenario.

I. Introduction

The quest to send humans to the Moon and Mars – this time to stay – has engaged the world's space community [1] [2] [3] [4]. Developing habitat systems that are suited for supporting life in extreme conditions is essential to enable a robust, long-term human presence on the Moon and then into deep space. These ambitious plans require the development of resilient and smart habitats that can operate safely and as intended in extra-terrestrial environments under extreme conditions and with limited resources. Such habitats, known as *SmartHabs, would have the ability to autonomously sense, anticipate, respond to, and learn from disruptions*. This quest is likely to be the most challenging mission ever pursued to date by humanity.

This ambitious objective requires overcoming several major technical challenges that will require the highest application of engineering knowledge. The Lunar Surface Innovation Consortium (LSIC) was established in 2019 as a virtual community focused on these new challenges for the aerospace community [4]. The Lunar environment is more extreme than any found on Earth, with high levels of radiation and abrasive dust posing risks to both humans and equipment, micrometeorites that may penetrate the structural system, large temperature swings, vacuum, and moonquakes that are expected to be quite different from their counterparts on Earth [5] [6]. Even familiar hazards, such as fires, leaks, and faults in habitat system components, take on a new dimension in the harsh and isolated Lunar environment. Thus, habitat systems will require a significant level of sensing, for both fault detection and overall habitat systems health management [7]. Disruptions are inevitable, and because space habitats are highly interconnected systems-of-systems, they have the potential to propagate through the habitat system and produce cascading consequences. Actions that may be taken to support habitat system recovery need to also be considered to expedite decision-making in a resource-constrained, time-critical situation. A resilient approach to the design and operation of such a habitat is prudent [5] [8] [9].

Due to the high costs associated with the transportation of materials and operations in space, resource constraints, including both physical materials and consumables, and computational capacity, drive several of the technical requirements as well as many SmartHab operational procedures. Replacement parts will either need to be manufactured in place, or shipped from Earth. Maintainability of the subsystems will be essential, and robotic maintenance should be exploited whenever feasible to reduce the workload on the crew. In addition, deep space habitats will experience long dormant periods. Dormancy introduces additional risks, and although many of the specifics of dealing with a dormant state can be anticipated, there are likely to be many more that have not yet been examined [10]. Without human crew-members who are able to rapidly recognize pitfalls and systematically diagnose unforeseen faults and disruptions, autonomy will need to be exploited as much as is possible. As we travel and explore further from the Earth, long communication delays will further hinder the ability for personnel on Earth to support the habitat crew.

Simulations are essential to study the complex interdependencies and operational vulnerabilities in SmartHabs. For instance, to perform flight control simulations in the Lunar Gateway project NASA developed Gateway in a Box as a low-fidelity model that focused on software emulation [11]. Subsequently, as a later step in the development, a medium-fidelity partial hardware emulator called Gateway in a Rack was adopted. Simulation offers a low-risk mechanism to test and improve operational techniques and technologies [12], including resilient design strategies, decision-making, systems health management, agent actions, hazard mitigation, and contingency planning. Furthermore, simulation enables one to analyze system behaviors under a wide variety of operational conditions (e.g., dormancy or recovery), and over a SmartHab life-cycle (considering, for example, aging, degradation and reconfiguration).

While several physical space habitat analogs exist, they are intended mainly for studying human factors and human behavior (for instance, see [13]). Efforts to understand and extrapolate the performance of various habitat subsystems have instead been primarily studied in simulation. For example, there has been substantial effort to computationally simulate the complex dynamics of environmental control and life support systems (ECLSS), including faults and repairs. For instance, ELiSSA (Environment for Life-Support Systems Simulation and Analysis) [14] and MELiSSA (Micro-Ecological Life-Support System Alternative) [15] have been developed to simulate specific ECLSS capabilities for space flight. Also, Virtual Habitat (V-HAB [16]), a computational simulation tool, has been built in the Matlab environment to facilitate the design process of life support systems in space habitats [17]. Through fully dynamic simulation, V-HAB considers the impact of component failures or a changing crew schedule, in which all of the faults considered are internal to ECLSS. Behjat et al., prototyped a Python-based computational framework called the control-oriented dynamic computational modeling tool meant to perform trade studies on systems-of-systems, and demonstrated its use for trade studies including one with a low-fidelity model of a space habitat [18][19]. Of course, more general tools also exist for modeling systems-of-systems, such as Modelica [20] which is a modeling language, and TRICK [21] which is a C/C++ library developed by NASA. However, to the best of our knowledge, no open-source options exist for the integrated simulation of the several subsystems that comprise a space habitat system on the Lunar surface. Matlab is an appropriate platform for such a research tool as it is widely available at research institutions. Simulink's multi-domain simulation environment is ideal to build and manage hierarchical systems. It supports model-based, system-level design and system-of-systems integration, and it is widely used for real-time experimental verification.

The Resilient Extraterrestrial Habitats institute (RETHi) has developed a system-of-systems platform for modeling and simulation of a lunar surface-based space habitat system to enable fundamental research toward the objective of establishing resilient SmartHabs on the Moon. Denoted HabSim, the model is built in Matlab/Simulink [17], and serves as a modular and coupled virtual testbed (MCVT) [5]. It aims to capture the dynamics and emergent behavior of a lunar surface-based SmartHab, simulate situational awareness, assess system resilience, and enable the study various degrees of autonomy under either internal or external disruptions. Taking a systems engineering viewpoint, HabSim consists of a heterogeneous set of models representing each subsystem and modules essential for building a functional SmartHab. A reference habitat concept (RHC) is developed using a formal architectural approach, and is used as the basis for defining the integrated habitat system on the lunar surface with the various environmental disturbances. The RHC is scaled to match the dimensions of laboratory-scale hardware. Each subsystem model is built following a standard notation, with damageable and repairable properties. The interdependencies of these subsystem models are defined, documented, described and managed through a design structure matrix (DSM) to support model development and subsystem integration [22].

The purpose of this paper is to outline the motivation for and capabilities of HabSim. We examine the technical challenges in developing this novel space habitat system-of-systems model that is meant to support a variety of research in resilience and autonomy of such complex systems. HabSim simulates SmartHab operation, monitoring, and management by incorporating a variety of disruptions and disturbances, and thus provides a preliminary proving ground for such techniques needed for the development of long-term space settlements. The main subsystems are represented as interconnected physics-based and phenomenological models having a variety of damageable and repairable properties [23]. Using a systems engineering viewpoint, each subsystem model is built with a standard notation, and the various types of interdependencies between these subsystem models are handled with the DSM. The relative physical location of various subsystems is encoded into the platform to directly support research on resilience, recovery, and decision-making.

HabSim serves as a versatile tool to facilitate the development of SmartHabs in the space community, and the process developed for building this simulation platform can also serve as an informative reference for similar tool development. Several of the papers in this virtual collection utilize HabSim and discuss the details of individual subsystem models. This paper discusses the capabilities of the HabSim platform and how we addressed the technical challenges in developing such a platform to support research into autonomy and resilience. Section II is focused on the HabSim platform and the platform-level requirements used to develop the architecture. The subsystems are described along with the modeled disruptions and degradation, and the associated repair and recovery. Section III considers several specific aspects of the technical approach taken to overcome challenges. Section IV provides an illustrative sample of the types of simulations that are possible, Section V discusses several specific research efforts that are using HabSim, and Section VI provides closing comments including lessons we learned.

II. Description of HabSim

The realization of a modular, multi-physics model for an extraterrestrial habitat has great potential to enable research in resilience and autonomy. However, this task comes with several challenges, suggesting that a systematic approach is needed that also leverages existing systems engineering tools that have been developed for such purposes. For instance, careful thought must be given to the choice of the modeling platform used, standards must be established in advance for proper data exchange, and model requirements must be identified deliberately and verified throughout the development. Each of these steps is essential to ensure that the product of this research is a simulation environment that can support the needs of a variety of researchers.

Enabling research on resilient design methods first and foremost means that the model must be able to inject a variety of disruptions that may cause a cascading sequence of disruptive events. The consequences of those events must then influence the functionality of both the individual subsystems as well as the entire system, and repair to those systems must then reverse the damage and return functionality to nominal as appropriate. Similarly, supporting research on autonomous operations means that the habitat must have the tools and sensors to detect and localize faults, and schedule and execute relevant agent repair actions to be defined by the researcher. These action settings include parameters related to the repair priority, repair rate, and the time needed to reach the repair task (which is related to the distance that must be traversed). Thus, in this section we consider the following key aspects in the development of the HabSim platform: (a) developing the platform architecture and data flow; (b) establishing SmartHab operational requirements; (c) scenarios that drive the necessary disruptions and disturbances.

A. Establishing System-level and Platform Requirements

In building a system-of-systems testbed to enable research, the first step is to generate requirements for both the models and the platform. Models able to capture the subsystem dynamics and the highly interconnected nature of this system-of-systems are necessary to generate a meaningful and sufficiently complex representation of a SmartHab system. The modeling will require an understanding and formulation of the relevant physics. The strong coupling among the subsystems must be represented in those model formulations. For certain subsystems, phenomenological models are used to capture the input-output behavior. This approach is chosen, for instance, when the physics of a system or phenomenon is more complex than needed for the purpose of the HabSim platform and the associated computation would be excessively demanding.

The platform implemented must be accessible to the modelers, support modularity and interoperability, and be extensible. The Matlab/Simulink environment is selected here because it has a user-friendly environment, models can be readily integrated in a graphical manner, it is widely used and accessible by researchers at universities, and a wide range of existing functions and other building blocks are already provided to the users in the Simulink library. Additionally, the MATLAB/Simulink environment has the capability to perform real-time simulations. HabSim subsystem models can therefore be used directly to establish cyber-physical testing capabilities, as discussed further in Section V. Because integrating the subsystem models with physical testing is an important element of our vision for the validation of new technologies, an important requirement for the overall SmartHab model is the capability for real-time execution [24].

Realistically, each of the models cannot be more detailed than is needed to perform the research. High-fidelity models require large computational effort, and, therefore, have prohibitively long run-times. A balance between having the model fidelity needed to do the research and a reasonable run-time needs to be achieved to ensure that the intended research can be performed without unnecessary computational overhead. Damage, repair, and recovery must also be modeled appropriately to support research in resilience and autonomy. Damage has consequences that affect the functionality of the habitat subsystems, and those consequences may also cascade into other subsystems due to the complex system interdependencies. Agents are necessary to simulate the process of performing repairs to support system recovery. Sensors must also be included in each subsystem model to enable situational awareness and, consequently, emulate fault detection and diagnosis (FDD). Note that the sensors themselves can also be faulty, damaged and repaired in HabSim.

In the requirements-gathering phase, the high-level requirements and specifications at the system-level are determined by analyzing the needs of the researchers expecting to use the model. To support research in resilience and autonomy, we define several disruption scenarios with various intensity levels along with their potential consequences, damage, safety controls, and actions taken, or repairs made. The scenarios are then analyzed to extract system-level and subsystem requirements. The modeled disruptions in these scenarios include micrometeorites, fire, moonquakes, airlock leakage, and nuclear system coolant leakage. Persistent disturbances such as dust, vacuum, and thermal changes are also captured in the models. This process is meant to determine the required functionalities of the models included in the system-of-systems. The HabSim User Manual and associated sample codes describe and demonstrate how each of these capabilities may be defined and implemented in simulations [23].

B. Architecture of the HabSim Platform

The overall architecture of HabSim is shown in Fig. 1. The platform is divided into the electro-mechanical subsystems (EMS) and the health management system (HMS). The habitat EMS subsystems in this dynamic Lunar habitat system model include: (i) a structural protective regolith layer (SPL) and a structural system (ST), each having both mechanical and thermal properties and dynamics, with appropriate coupling between them; (ii) a lumped-parameter model capturing the coupled temperature and pressure of the two-zone physical air volume inside the habitat's interior environment (IE); (iii) a thermal and pressure management system capable of regulating each of the zones independently (ECLSS); and, (iv) a power generation, energy storage, and distribution system (PW). Each of these electro-mechanical subsystems has built-in sensors to gather physical response data, and synthetic FDD components to provide diagnostic information. An agent is included to perform the actions needed for the repair of the damaged components in the habitat subsystems. Here we refer to the health management system (HMS) as the system composed of the communication network, data repository and data service, and command and control (C2). The communication network, data repository, and data service are collectively called the communication and data handling service (CDHS). In concert, they transfer the data between EMS and HMS and store data. C2 takes in synthetic diagnostic information and makes decisions to schedule specific agent actions. To enable the study of how communication delays impact resilience and autonomy, a ground control node representing mission control on Earth is included with similar CDHS capabilities. The latency may be set by the user.

A key element in HabSim is the ability to capture system interdependencies. The coordination block serves to manage interdependencies in the simulation platform and pass data among the subsystem models. Similarly, the disturbance block (DB) initiates and coordinates disruption events and persistent disturbances. FDD adopts a distributed (i.e., decentralized) approach where each subsystem detects faults and passes the health state values corresponding to damageable components within that subsystem to C2 through the CDHS.

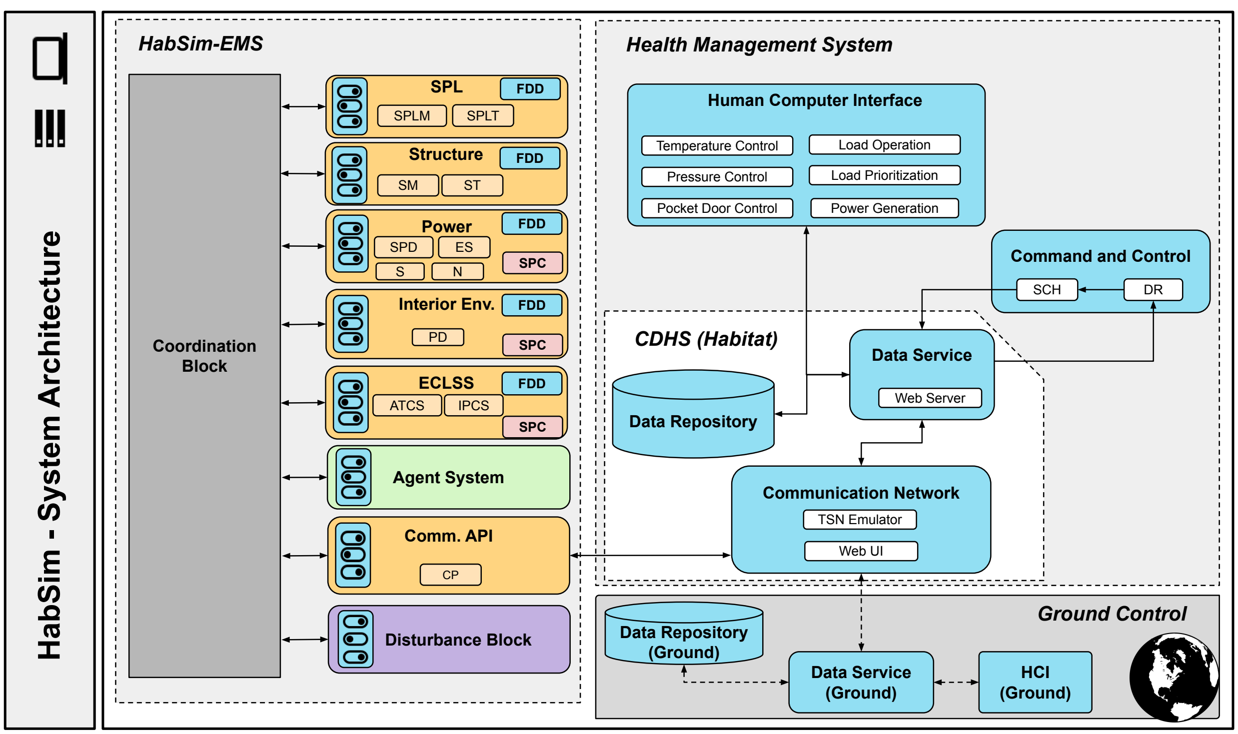


Fig. 1 Architecture of the HabSim Simulation Environment

Several input files are used to set up the simulation environment prior to execution, including the habitat environment and layout, the scenarios and relevant parameters for running a given realization, and to characterize the agent actions. These files include: `Input` (system-level scenario description), `Sim_set` (simulation settings and subsystem details), `Configuration` (habitat features and exterior environment), and `Run_scenario` (loads all variables and executes simulation). `Configuration` sets the habitat scale, design parameters associated with the system and subsystems, and constants that define the ambient environment (e.g., the gravitational constant, surface temperature). The habitat layout is captured in an Excel file that contains the locations and distances of relevant physical components. A comprehensive list of possible agent actions is captured in another Excel file where the user can specify the actions, priorities, repair rate, repair time, etc. A reflexive, model-based decision-making approach is adopted for this initial implementation of the HabSim, which simply performs the necessary repair actions in the priority defined by the user. Simulation outputs are recorded in Matlab files based on the user's pre-selected preferences.

The HMS components run in a docker container and an application programming interface (API) has been developed to connect EMS and HMS. The EMS and HMS can be executed on one computer, or can be partitioned to accelerate simulations by using two machines, one for EMS and one for HMS. This option is especially useful when real-time execution is needed.

C. Habitat Models and Subsystems

The functionality, damage, and repair of the EMS subsystems that comprise the habitat system are selected deliberately to achieve a sufficiently representative and complex system with opportunities to simulate disruption scenarios of various intensity levels that could follow a variety of paths with different consequences.

The HabSim-EMS includes the following subsystems:

- **Structural protective layer (SPL)** is a layer of Lunar regolith to protect against solar radiation and insulate the habitat from large thermal swings. In this work, the mechanical dynamics of SPL is considered to model the interaction force with the ST, and thermal dynamics are included to model the heat exchanged with the exterior and the structure throughout a Lunar day. Damage to and repair of the SPL is considered, which affects the thermal behavior when that occurs. The interactions between the thermal and the mechanical dynamics are considered in this model.
- **Structural system (ST)** accounts for the mechanical dynamics of the habitat structure in response to pressure changes inside the habitat and the interaction forces with the SPL . This model also reflects the thermal dynamics of the habitat structure, influenced by thermal loads of the SPL and interior habitat environment. Damage to and repair of the structure is considered, affecting both thermal conditions and pressure retention. The interacting thermal and mechanical dynamics are also considered in this model.
- **Interior environment (IE)** quantifies the temperature and pressure changes of the air within the interior of a habitat. In this model, temperature and pressure are coupled [25]. The model is divided into two zones, and a pocket door can be

triggered to divide the interior volume into two separate zones each with its own temperature and pressure.

- **Environmental control and life-support system (ECLSS)** is composed of an active thermal control system (ATCS) and an interior pressure control system (IPCS) [25]. ECLSS consumes energy to operate the physical components, such as compressor, cooling fans, and heaters. Dust inside the habitat will affect the ventilation fan, and maintenance is required to prevent loss of thermal management system performance. Damage to and repair of the cooling fan, evaporator coil, condenser coil, air storage tank, air supply valve, and compressor for each zone are considered. Even though ECLSS does not directly interact with the ST, SPL, and IE, these systems indirectly influence the behavior of ECLSS through temperature and pressure changes in the habitat.
- **Power (PW)** consists of power generation (solar cells and nuclear), energy storage (ES), and a smart power distribution (SPD) system. SPD has a scheduling model responsible for allocating the power that is generated and a load prioritization model to manage the power distribution during critical situations as described in [26]. The main components of the SPD are: three boost converters, converter 1, converter 2, and converter 3, to convert the different generated voltage levels from nuclear, solar, and ES, respectively, to a common voltage level; a DC generation bus that connects the total power generated with different loads; and, three buck converters, converter 4, converter 5, and converter 6, to convert the distribution voltage level to the rated voltages of life support loads, monitoring loads, and all other habitat loads, respectively. Damage to and repair of the different power system components, including nuclear and solar power generators, ES, power converters, and generation bus, are modeled. Damage and repair of the converters and generation bus is modeled as binary and can further be defined as random. For the nuclear, solar, and ES systems, the damage and repair features modeled are gradual (not binary), as they can be damaged/repaired in discrete increments.

The HabSim-HMS includes those components that pass, store, and process data from the sensors and systems in the habitat, in part supporting decision-making.

- **Communication and data handling service (CDHS)** emulates a communication network and manages the data exchange among individual EMS subsystems, the database, and the ground node, representing mission control on Earth. A fixed network topology is used in the communication network emulator, which adapts when a communication node is faulty by removing the affected node. When enabled, communication with ground control can be configured with a programmable delay to emulate latency and restrict throughput.
- **Command and control (C2)** is intended for autonomous decision-making and scheduling agent (AG) interventions based on a suite of built-in tests in HabSim that use the information provided by sensors (SN) and FDDs in an automated fault-reasoning mechanism to provide test-to-effect information. Consequently, C2 uses the information provided by the built-in tests in a dependency matrix (D-matrix) for the decision-making. C2 monitors the operation of the HabSim subsystems and outputs commands to trigger interventions, repairs and maintenance, to restore the nominal operating condition of the detected damaged components. For instance, C2 would schedule the robotic AG to repair the SPL

damage after repairing the ST damage.

- **Human computer interface (HCI)** enables human operators to monitor and manually control the state of pressure and temperature in each IE zone independently, and close or open the pocket door to isolate zone 1 and zone 2 from each other. The HCI can optionally be connected to HabSim to enable manual decision-making, in which case it would replace the functionality of C2. In future versions of the platform, we would enable both to function in tandem.

In addition to EMS and HMS subsystems the following subsystem, models, and blocks have a significant contribution to building HabSim:

- **Disturbance block (DB)** facilitates the realization of a variety of disruptive hazard events and their consequences that are included in the models that have the potential to impact the habitat. The user defines the disruption characteristics in the `Input` file. This model also imparts the ambient environmental conditions to the habitat. For instance, the DB provides the temperature on the surface of the Moon, the angle of the sun, vacuum conditions, and the amount of dust raised by a micrometeorite strike, launch event, or simply from environmental conditions.
- **Sensors (SN)** take measurements of physical quantities of interest to be used for monitoring components in various subsystems or for local closed-loop control of certain subsystems. To add uncertainty, the data acquired by a given sensor can be contaminated with random noise, or a drift or bias can be added. SN system outputs are also used for habitat system health management.
- **Fault detection and diagnostics (FDD)** is embedded in each EMS subsystem (see the architecture diagram in Fig. 1) to monitor the health of the damageable components and send reliable health state information to C2. FDDs take in SN measurements such as accelerations, temperature, pressure, etc. and output health state values. All but one of the FDDs herein are currently synthetic, or idealized, and know the exact health state of the subsystem. However, the FDD for the ST is a true built-in test that uses acceleration data to detect when and where structural damage has occurred.
- **Agent (AG)** performs interventions including the repair of damaged components or the maintenance of certain components. This simple model captures repair rate and can represent a human crew member or a robot depending on the parameters chosen. In the current implementation of HabSim, only one agent is available to perform such tasks. It is assumed that the AG has sufficient power charge to complete all scheduled repairs and is not dependent on the habitat PW. It is also assumed that AG is deterministic and ideal in performing repairs.

D. Disruptions and Disturbances

HabSim is developed mainly to support research in resilience and autonomy of deep space habitats. Thus, in simulation the habitat model is exposed to several disruption scenarios and disturbances. Six types of disruptions are modeled in HabSim, including micrometeorite impact, fire, moonquakes, airlock leakage, nuclear system coolant leakage, and sensor failure. Individual or combinations of disruptions are realized by embedding 28 damageable and repairable components in the subsystem models (see Table 1). In addition, sensors in the subsystems may also become faulty. Certain disruptions are associated with a location, based on the layout of the RHC. Many of the disruptions are also characterized by an intensity level defined by an integer that may range from 1 to 5. The intensity level is used to pass phenomenological damage indicators to subsystems whose interactions with the disturbance would otherwise be too complex to effectively model. The user may define the disruption, time of occurrence, location and intensity, and affected components, in the `Input` file.

One or more components can be damaged in any given disruption scenario. The options for damageable and repairable components are listed in Table 1. For instance, a micrometeorite impact can hit the habitat and damage the SPL, ST, and certain components in PW or ECLSS in sequence. Outside the habitat a micrometeorite impact scenario can damage solar PV arrays, nuclear power generator panels, or radiator panels. Inside the habitat, a fire could damage the PW or ECLSS components when it starts or spreads within their proximity. A moonquake could damage components in the PW or ECLSS subsystems. In a scenario with airlock leakage, the pressure inside one zone of the habitat may drop slightly due to air leakage. The coolant system cascade scenario may result in a drop in efficiency of nuclear power generation combined with increase power consumption by the pump. Each SN in HabSim also has the potential to experience faults. Such faults result in data losses, erroneous data, but in some cases may lead to more substantial consequences such as disruptions in the regulation of temperature and/or pressure. Each of these events may be repaired through an agent action with a predefined priority. Clearly, HabSim offers many opportunities to study contingency planning and response actions and priorities.

In addition to the disruptions, three environmental disturbances may be considered in a simulation. The first environmental disturbance is nominal external dust accumulation that affects the efficiency of external components including solar PV arrays, nuclear radiators, and ECLSS radiators. The loss in performance of the affected systems propagates through other habitat components. The second environmental disturbance is the solar angle, which modifies the heat transferred between the interior environment and the exterior environment. Through a Lunar day, the temperature outside the habitat fluctuates between -130 °C and +130 °C. The ECLSS compensates to maintain a safe temperature inside the habitat, unless damage yields improper functionality. The third environmental disturbance is a small amount of air leakage that is almost impossible to avoid in a real pressurized system.

Table 1 Damageable and Repairable Features of Habitat Subsystems

Subsystem	Damageable & Repairable Components	
SPL	SPL penetrated	
ST	ST penetrated	
IE	Fire in zone 1	Fire in zone 2
	Airlock leakage	
ECLSS	Radiator panels dusty	Radiator panels paint degradation
	Ventilation fan filter need to be replaced	Air storage tank leakage
	Supply valve malfunction	Condenser coil deformation
	Compressor degradation	Evaporator coil deformation
	Heater degradation	Cooling fan degradation
PW	PV arrays dusty	PV arrays damaged
	Nuclear panels dusty	Nuclear Panels damaged
	ES damaged	Converter 1 damaged
	Converter 2 damaged	Converter 3 damaged
	Converter 4 damaged	Converter 5 damaged
	Converter 6 damaged	DC generation bus damaged
	Nuclear system coolant leakage	
SN	SN failures	

E. Verification and Validation

Model verification and validation was implemented by following structured engineering software approaches that enabled systematic and efficient system-of-systems integration for HabSim [12] [27]. First, the functional requirements at the subsystem level were defined, as described in Section II.B, including damageable and repairable elements, followed by system-level requirements, such as disruption propagation. These requirements included assessing of the relevant hazardous states for each subsystem and identifying the specific safety controls to include in the model. Each subsystem modeler was tasked with establishing a reference model, typically consisting of a high-fidelity physics-based model. At this step, the modeling assumptions and limitations were delineated to ensure a clear understanding of the capabilities of each subsystem model. Modelers had to provide verification that each model behaved according to the specified dynamics and requirements that represented the hazardous state changes of interest and expected behavioral properties.

Following the high-fidelity verification, reduced-order models were developed when necessary to reduce the computational demands for the system-of-systems integration. The resulting medium-fidelity models were then verified by comparing the dynamic responses of the reduced-order model to those of the higher fidelity reference model. Then, unit testing was conducted to verify that the simulation results accurately represented the model description and requirements. System integration was then performed to integrate the models and gradually verify the functional requirements. At this stage, we confirmed and determined the degree to which the integrated results accurately represented a SmartHab behavior for the intended use. Then, user testing was also performed to ensure that the system-level requirements and specifications were met. Several of the papers included in this virtual collection provide details on individual models, and a full description of these steps is discussed in the documentation that will be provided with the code [23].

III. Technical Considerations

HabSim requires the simultaneous execution of a number of complex and interdependent subsystem models. The interdependencies between these subsystems take several forms, and also grow exponentially with the number of subsystems, thus expanding the challenge of tracking the required inputs and generated outputs for each subsystem. Several specific technical considerations need to be taken into account to simulate a system-of-systems with the level of complexity used in HabSim. These considerations are explained in this section, and offer lessons learned for those developing simulation platforms for systems-of-systems.

A. Design Structure Matrix

The DSM is a network representation technique used for handling data exchange and enhancing efficiency during the development and management of a system-of-systems [22]. This tool has been adopted in a wide range of research and industrial practices as a compact and scalable representation of subsystem dependencies in a system-of-systems. The primary purpose of developing the DSM is to identify and manage the data flow, interface conditions, and operational dependencies and modes. The DSM also aims to assist model developers in making choices regarding signals, and communicating those needs, for the inputs to each subsystem as well as the outputs to be provided to other subsystems. In addition to the graphical representation of the DSM, we examined the interdependencies (i.e., connections among subsystems) in a holistic manner to facilitate the system-of-systems integration. The interdependencies in a system-of-systems are characterized by multiple complex connections and relationships, including feedback and feedforward loops that vary in dimensionality and purpose [28].

For this reason, we defined four types of interdependencies. In HabSim these are referred to as:

- **Physical Interdependencies** occur when the physical output(s) of either subsystem progress or depends on the dynamical behavior of the other. Examples include the heating and cooling loads provided by the thermal management to the interior environment or the power consumed by various components.
- **Cyber Interdependencies** occur when the signal is carrying data or information that typically pass through the CDHS or information infrastructure. Examples include measured temperature or pressure acquired by sensors in the interior environment, structural sensors that capture accelerations for health management, and FDD health states from various subsystems.

- **Intervention Interdependencies** are commands sent from C2 to the agent to trigger maintenance, improve system performance, or repair habitat components. Examples include a command to the agent to clean the dust that accumulated on the solar arrays or to patch a hole caused by a micrometeorite.
- **Disruption Interdependencies** are external inputs (disruptions and disturbance) that originate outside of the habitat or fault initiation mechanisms within habitat subsystems. Examples include a micrometeorite impact, fire inside the habitat, and sensor failure.

Habitat subsystems, and the other models included for simulations are identically labeled and ordered by their numbering in both the rows and columns, as shown in Fig. 2. The input sources and output destinations corresponding to each subsystem are identified in the off-diagonal cells of the matrix. An off-diagonal 'X' indicates the presence of an interaction, while an unmarked cell signifies its absence. The convention followed here is that subsystem outputs are specified as the rows "system giving the signal." Subsystem inputs are in the columns denoted "system receiving the signal". More explicitly, reading across a matrix row will reveal all the outputs from the subsystem denoted by that row. For example, reading across row three shows that the ST outputs to the following subsystems: (11) SPL, (8) IE, (3) PW, (6) AG and (10) CN. Examining down a column will specify all the inputs to the subsystem denoted by that column. For example, reading down column three shows that the ST has inputs from the following subsystems: (11) SPL, (8) IE, (3) PW, (6) AG and (9) DB.

The interactions and corresponding categories described in Section 2 were properly identified using the RHC to develop the DSM for HabSim. The DSM for the crewed mode of operation is shown here in Fig. 2. As human exploration pushes the current boundaries and technology advances to enable more autonomous operation of deep space habitats, dormant conditions will be more common [10]. Thus, HabSim is being expanded to consider a dormant mode as well as simulating the transitions between dormant and crewed modes.

The HabSim DSM is further developed by creating a tool to manage the interaction specifications, the realization of which is in Fig. 2. Moreover, this tool graphically specifies an interaction and directly links a user to a page that provides complete interaction specifications. In each interaction page, the specifications shown for the interdependencies include the name, size, sampling rate, units of either the signal measurement or interchanged physical conditions, and any helpful additional notes. A general sample of such an interaction page is shown at the bottom of Fig. 2. In the initial phase of HabSim development, all model developers were involved in developing the DSM to capture all requirements and essential behaviors. Discussions held during the development of the DSM yielded a greater understanding among model developers and improved system modeling requirements. The DSM was updated in parallel with the development and integration of the subsystems. Continuous communication was maintained among model developers to verify and validate the integration of systems through the DSM. It provided a clear purpose to the model developers, as they no longer modeled for the sake of modeling but to meet the expectations of other subsystems that were not easily captured in written documentation. The level of detail captured within the DSM is appropriate for the intended purposes, providing sufficient knowledge to model developers about the data flow without understanding all details of subsystem model dynamics.

The system-level diagram in Fig. 3 is used to visualize the actual interdependencies captured within this system-of-systems model, and the specific class for each interaction.

System receiving the signal

System giving the signal

Sub-Systems (Level 1)		HMS	SPL	ST	IE	ECLSS	PW	AG	CN	DB
Health Management System	HMS								X	
Structural Protective Layer	SPL			X			X	X	X	
Structural System	ST		X		X		X	X	X	
Interior Environment	IE			X		X	X	X	X	X
ECLSS	ECLSS				X		X	X	X	X
Power System	PW		X	X	X	X	X	X	X	X
Agent System	AG		X	X	X	X	X		X	X
Communication API	CN	X			X	X	X	X		
Disturbance Block	DB		X	X	X	X	X			

I-G-R: Giving-System to Receiving-System

Number of Cyber Interdependencies: #
Number of Physical Interdependencies: #
Number of External Disturbances: #
Number of Interventions: #

Cyber-Interdependencies:

Cyber Signal 1: Name Signal
Size: n_a x dim
Sampling rate: -
Unit: Signal Units
Notes: n_a is the degrees of freedom/sensors for that signal; dim is the number of dimensions

Physical-Interdependencies:

Physical Signal 1: Name Signal
Size: n_b x dim
Sampling rate: -
Unit: Signal Units
Notes: n_b is the degrees of freedom at the system's interface with another system; dim is the number of dimensions

⋮

⋮

External Disturbances:

Disturbance 1: Name Signal
Size: n_c x dim
Structure name: MATLAB structure name
Sampling rate: -
Unit: Signal Units
Notes: n_c is the degrees of freedom at the system's interface; dim is the number of dimensions

Interventions:

Intervention 1: Name Signal
Type of Intervention: Describe action
Size: n_d x dim
Failure Mode: [0000000] ID
Sampling rate: -
Unit: Signal Units
Notes: n_d is the degrees of freedom at the system's interface; dim is the number of dimensions

Fig. 2 HabSim design structure matrix for crewed operation

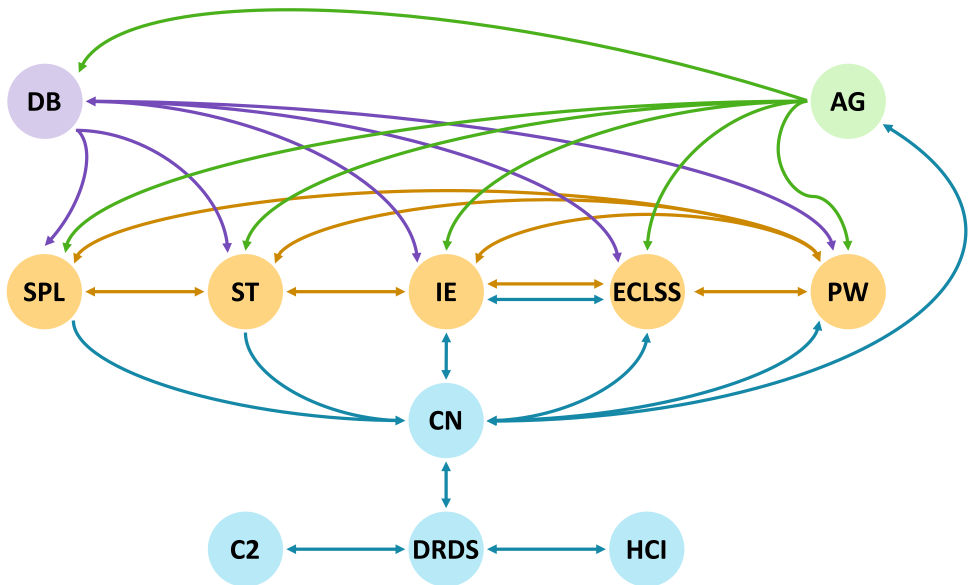


Fig. 3 Diagram representing the EMS and HMS subsystems and their interdependencies

B. Standard Notation

Developing a plug-and-play model drives the need to define a standard notation to which each subsystem model must adhere to exchange data between the various subsystems. Defining this common language helps the domain experts and subsystem model developers to identify and communicate requirements for the exchange of information between the subsystems or their constituent components. Modelers and future developers interested in using HabSim for their own research are expected to put their models in this standard notation to integrate it with the adjacent subsystem models and the disturbance block in HabSim.

The behavior of these systems may be described by functional relations, which depend on their states and interdependencies subject to the disturbances and disruptions they face in simulation. In Figure 4, we show a generic HabSim subsystem and describe its standard notation below.

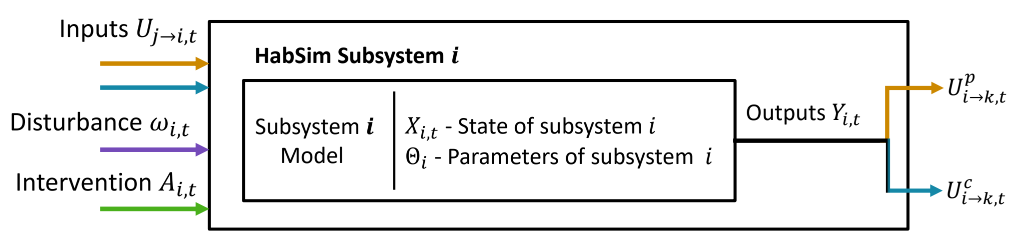


Fig. 4 Decomposition of complex interconnected subsystem i

Assume the RHC consists of $i = 1, \cdots, N$ constituent subsystems. The behavior of subsystem i is described by $X_{i,t}$, which are state variables that comprise the physical and health states of the subsystem. Subsystem i's input physical or cyber interdependencies with another subsystem j is denoted by $U_{j\rightarrow i,t}$. The input of a subsystem i from subsystem j is denoted by $U_{j\rightarrow i,t}$. $U_{j\rightarrow i,t}$ is a structured collection of physical or cyber interdependencies, which is denoted as

$$U_{j\rightarrow i,t} = \left\{U^{p}_{j\rightarrow i,t}, U^{c}_{j\rightarrow i,t}\right\}, \tag{1}$$

where, $U^{p}_{j\rightarrow i,t}$ and $U^{c}_{j\rightarrow i,t}$ are a collection of physical and cyber interdependencies from subsystem j to i, respectively. Further, the time-independent parameters of subsystem i are denoted by Θ_i. The disturbance interdependence on subsystem i is denoted by $\omega_{i,t}$, capturing all exogenous processes that affect that subsystem's performance. Finally, the HabSim-HMS may provide an intervention for subsystem i if its performance needs to be recovered. We denote this intervention interdependency on subsystem i as

$$A_{i,t} = \left\{A^{p}_{i,t}, A^{c}_{i,t}\right\}, \tag{2}$$

where $A^{p}_{i,t}$ is any interventions performed by the AG subsystem, and $A^{c}_{i,t}$ which refers to a set-point control command issued by the HabSim-HMS on subsystem i. Intervention from the AG subsystem is considered to be a physical interdependence, while the set-point control command from HabSim-HMS is a cyber interdependence. The temporal evolution of the states of a subsystem is described by its state dynamics. The state dynamics of subsystem i is defined as

$$X_{i,t+1} = f_i\left(X_{i,t}, A_{i,t}, \Theta_i, \left\{U_{j\rightarrow i,t}\right\}_{j\neq i}, \omega_{i,t},\right), \tag{3}$$

where $f_i(\cdot)$ defines the time evolution of the states of interest in subsystem i. Finally, all sensor, FDD, and built-in test telemetry of subsystem i, $Y_{i,t}$, is an outcome of a measurement process

$$Y_{i,t} = g_i\left(X_{i,t}, \phi_{i,t}\right), \tag{4}$$

where $g_i(\cdot)$ is the measurement process and $\phi_{i,t}$ is the measurement noise. The telemetry may be transmitted to another subsystem k from subsystem i, denoted by $U_{i\rightarrow k,t}$, such that $U_{i\rightarrow k,t} \subset Y_{i,t}$. Since $U_{i\rightarrow k,t}$ may be physical or cyber, we define the structured collection of output dependencies of subsystem i with another subsystem k is

$$U_{i\rightarrow k,t} = \left\{U^{p}_{i\rightarrow k,t}, U^{c}_{i\rightarrow k,t}\right\} \tag{5}$$

where $U^{p}_{i\rightarrow k,t}$ and $U^{c}_{i\rightarrow k,t}$ are a collection of physical and cyber interdependencies, respectively, from subsystem i to subsystem k.

C. Physical Layout of the Reference Habitat Concept

Encoding the physical locations of the RHC components and resources into simulations is essential for research on resilience and autonomy. This information is used in the simulation for considering the time required for travel, as well as for repair and recovery actions. These details affect the resilience of a habitat system and, thus, are needed to quantify metrics for resilience and to make decisions about the best course of action to take [29],[30].

The layout of the interior and exterior of the RHC is shown in Fig. 5. In a simulation, the location of a disruption, such as a micrometeorite impact, is used to determine which subsystems or components are affected by that hazard event. The location can also be combined with the intensity level to define how much debris is produced and, thus, which other subsystems may be affected by the specific disruption being simulated. The distance between certain components inside the habitat and the initiation point of the fire will determine the elapsed time associated with heat or fire spreading to those components. Furthermore, outside of the habitat, the distance between components will influence the time needed for repair actions. Note that HabSim considers the AG traveling speed, which is especially important to consider when repairs are needed to more than one damaged component. Also, the model and dimensions in HabSim are scaled to one-fifth of the RHC, which is compatible with the dimensional scaling of the cyber-physical testbed (CPT) [5].

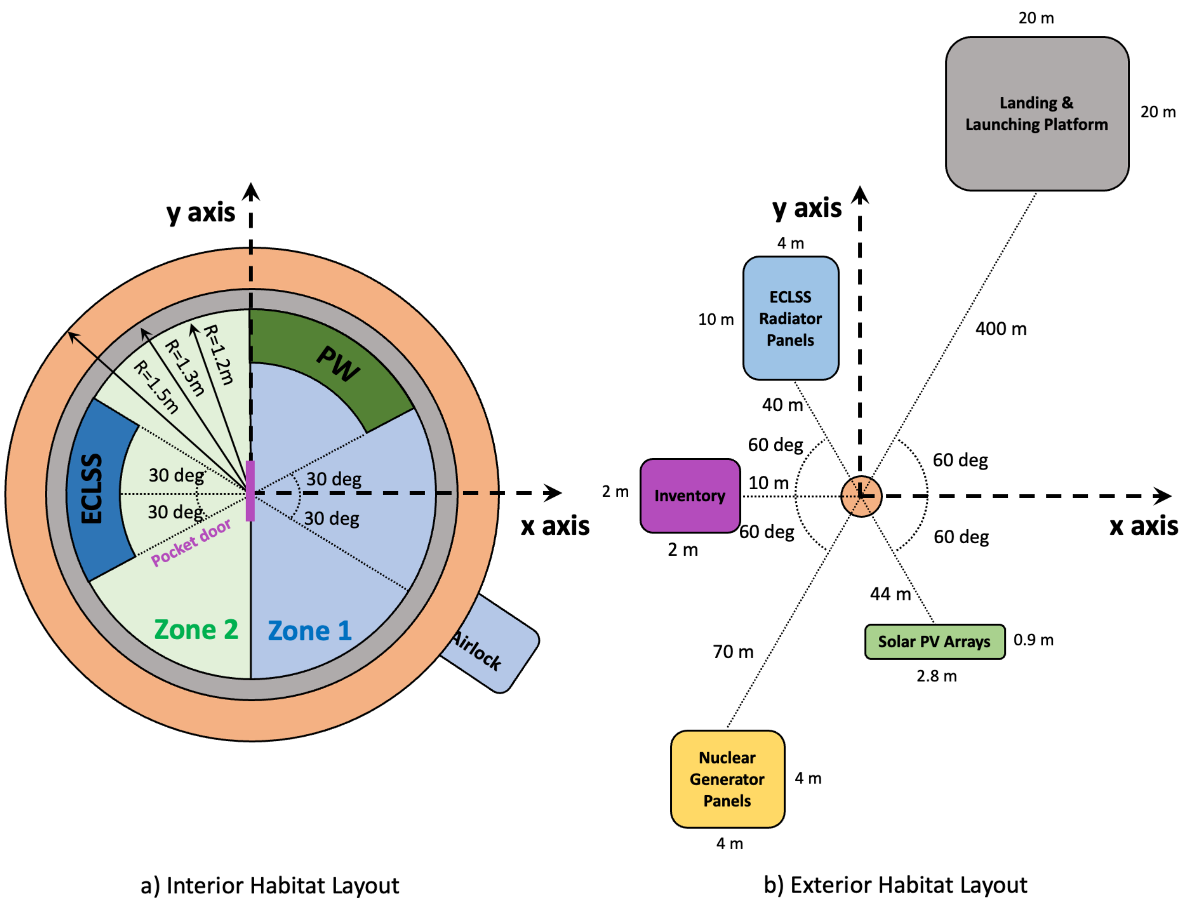


Fig. 5 Layout of the one-fifth scale reference habitat concept

D. Setting the Simulation Initial Conditions

Setting the initial conditions for all states and variables is always necessary, but is especially important for simulations involving multi-physics models with physical interdependencies. Large transient responses may occur at the beginning of the simulation if the initial values for the model states are not properly set prior to the simulation. States that typically have non-zero nominal values, such as temperature and pressure, are especially important. For instance, if the angle of the Sun does not correspond to the initial conditions set for the ST and the SPL models, the dynamics of the habitat system will result in the system responding as if step inputs are applied to the system. Although these transient responses fade away as the system reaches its steady state, the behavior impacts the interpretation of the results and unnecessarily adds computational effort and time.

Setting the initial conditions requires that one run the simulation for a relatively long time to reach steady state, and then those values are stored to be used as initial states for specific cases. This task was completed for HabSim by determining the appropriate boundary conditions for ST and SPL according to the initial location of the Sun and determining the solar irradiation on the outer layer of SPL and the desired initial pressure and temperature inside the habitat thus imposing it to the inner layer of ST. This step determines the initial temperature, and deformation of SPL and ST. Having this information, the initial condition related to the thermal and pressure management system can be determined. The final step is to determine the initial conditions in the PW subsystem based on the operational condition of all the components in the habitat.

Several choices of initial conditions are made available based on the position of the Sun and its associated habitat system conditions, such as temperature, pressure, and displacements. The user must select one of the five distinct solar angles in the input file, which are 0°, 45°, 90°, 135°, and 180°, and correspond to the sunrise, late morning, noon, late afternoon, and sunset of a Lunar day, respectively. Based on the selected angle, the corresponding file is loaded containing all initial state values needed to run the simulation. Simulation configuration parameters, such as the simulation time step, are also specified therein.

IV. Illustrative Sample Scenario

In this section we demonstrate some of the capabilities of HabSim. A disruption scenario is presented involving a micrometeorite strike followed by the propagation of the effects through the habitat system. Faults are then detected and repairs are made in an order designated by the user before starting the simulation. This scenario serves as a sample to illustrate how the HabSim platform can be used to simulate the consequences of immediate decisions and emergent events, and thus support research into resilience and autonomy.

A. Disruption Scenario: Micrometeorite Strike

The storyboard and timeline for this simulation are shown in Fig. 6. The simulation begins in a crewed state while the habitat is operating under nominal operating conditions. A micrometeorite strike occurs 500 sec into the simulation which begins early in the Lunar morning before solar power generation is available. The micrometeorite intensity level in this sample scenario is set to 5 (the highest intensity), corresponding to a case in which it penetrates both SPL and ST, resulting in a hole with a radius of 0.8 cm. In this scenario, the strike is close to PW, and the resulting debris causes damage in certain PW components inside the habitat, including power converter 1, power converter 2, and a portion of the battery cells. In this scenario, five components are damaged out of the possible 28 damageable components, each with direct and indirect consequences on the overall habitat functionality. The activities scheduled for the crew during that day will also be affected, as noted herein. More details about the nominal schedule for a typical day are discussed in [31].

Damage that occurs to the components as a consequence of this impact is detected using synthetic FFDs. When damage is detected, the health state of the affected components shifts from a healthy state (set at 0 in the FDD) to an unhealthy state (set to 1 in the FDD), as shown in Fig. 7. Then the HMS takes in the FDD information and schedules a robotic AG to perform the necessary repairs. The repair priority designated in this simulation is: battery > ST > converter 1 > converter 2 > SPL, as shown in Fig. 6.

The AG starts replacing the damaged battery cells first at 510 sec and completes this task at 910 sec. Having access to an energy source is essential as ECLSS requires a significant amount of power to compensate for the air leak. Thus the damage results in a drop in the IE pressure and temperature, and the habitat potentially enters a hazardous state.

The batteries can provide power for only a limited amount of time. Therefore, as soon as the batteries are replaced, the AG begins repairing the hole in the ST to rectify the main source of the pressure and temperature drops, after which the power consumption returns to a more typical level. Once the battery cells and the ST are repaired, converter 1 repair has priority. Note that power converter 1 handles nuclear power generation, which is typically the most reliable source of power generation for the SmartHab. Once the batteries, ST, and power converter 1 are repaired, the AG moves on to less critical components during the remainder of the simulation, and at that point, power converter 2 and SPL are repaired.

In this specific scenario the solar angle is zero, so there is a negligible amount of power being generated from the solar panels. The output from converter 2 is thus zero. Additionally, repair of the SPL in the early stages does not have a significant effect on the IE pressure and temperature and can be scheduled at the very end of the recovery. It is helpful to point out that the simulation also considers the AG travel time between the two damaged components, based on the pre-defined AG speed and the layout of the RHC. This travel time shows up in the simulation results as gaps between the repair actions in Fig. 6. For instance, the ST repair finishes at 1260 sec and the repair of converter 1 begins 50 sec later at 1310 sec.

The time required to perform these recovery actions and travel between components does have an influence on the

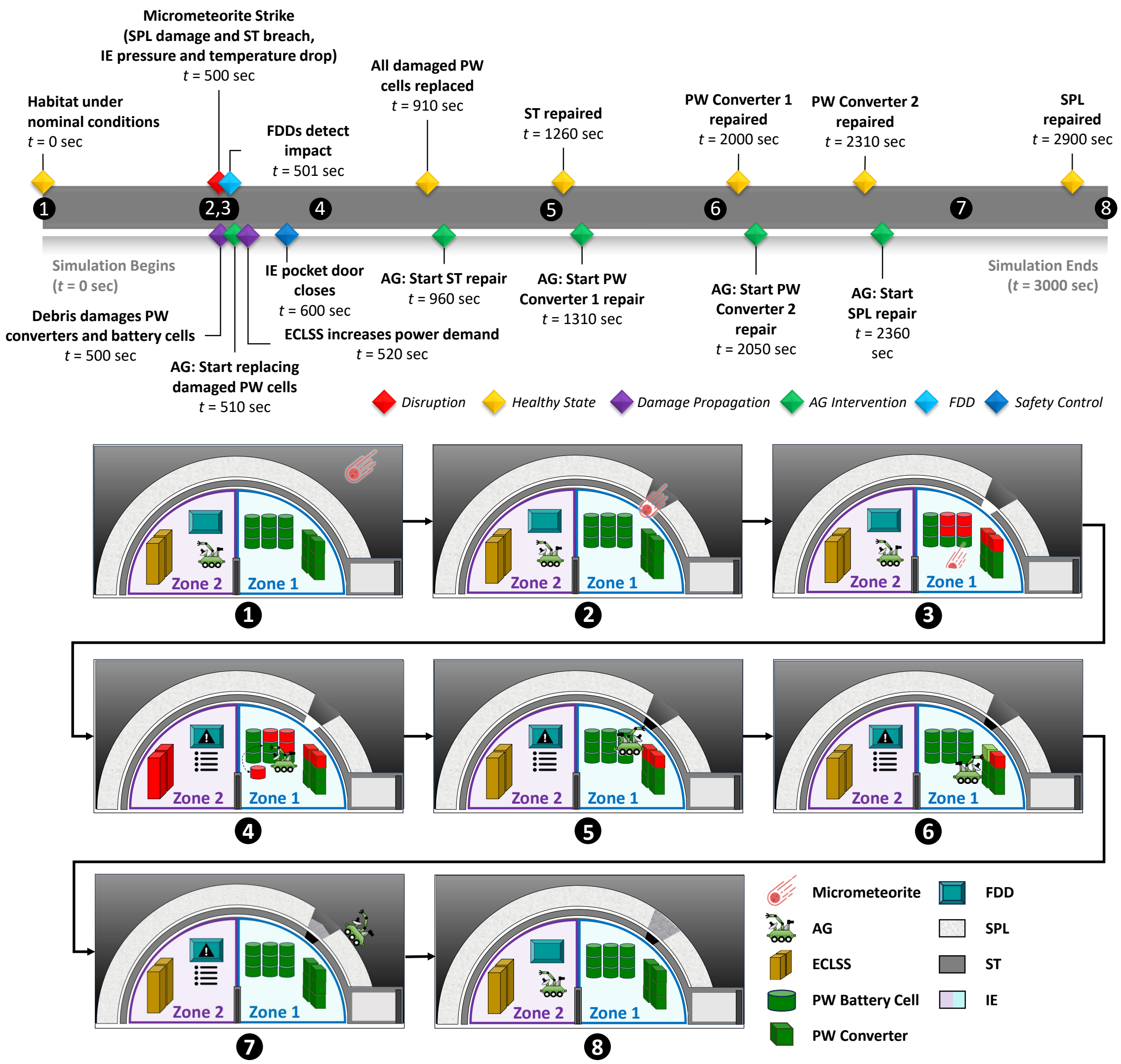
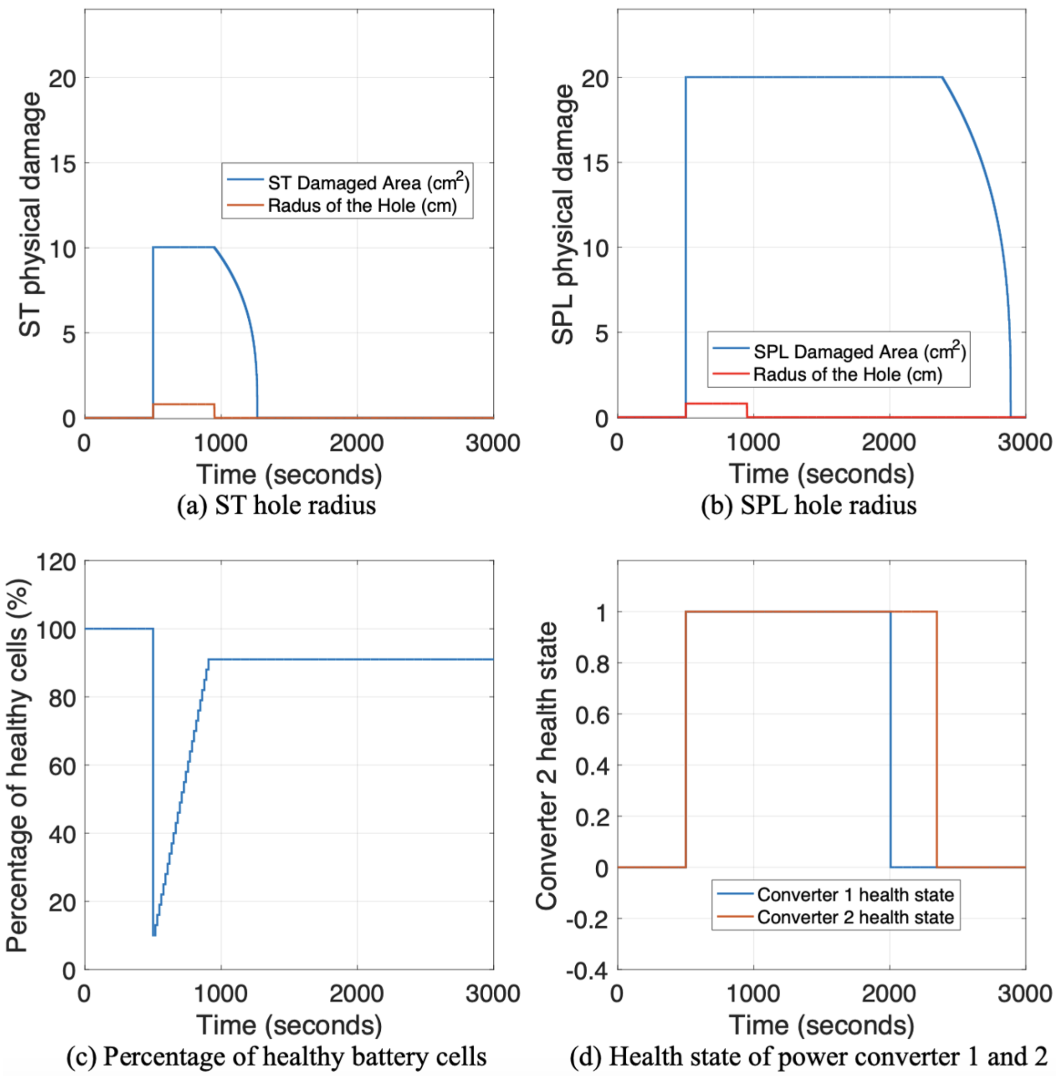


Fig. 6 Storyboard and timeline illustrating the chain of events in the sample disruption scenario

recovery process. For example, due to the hole in the ST and SPL, the pressure in zone 1 drops to $0.3 \cdot 10^4$ Pa, which is the pressure that ECLSS pressure management system is sized to sustain while there is a hole in the habitat with a radius of 0.8 mm. Figure 8 shows this change in both pressure and temperature in zones 1 and 2 over time. Although the micrometeorite strikes in zone 1, and the pocket door is closed at 600 sec to protect zone 2, there is drop in the pressure and temperature of zone 2 to $0.72 \cdot 10^4$ Pa and 278 K, respectively. This behavior persists for only a short period of time corresponding to the delay between the time of detection of the hole and the time the command is sent to close the door. Note that when the pocket door is closed, the crew is isolated in the unaffected interior zone, zone 2. Here the crew does not have access to the airlock and thus is unable to perform extravehicular activities that may be scheduled for that day. Even if the repair is fully autonomous, they will likely be unable to have meals or conduct scientific experiments until the repairs are completed.



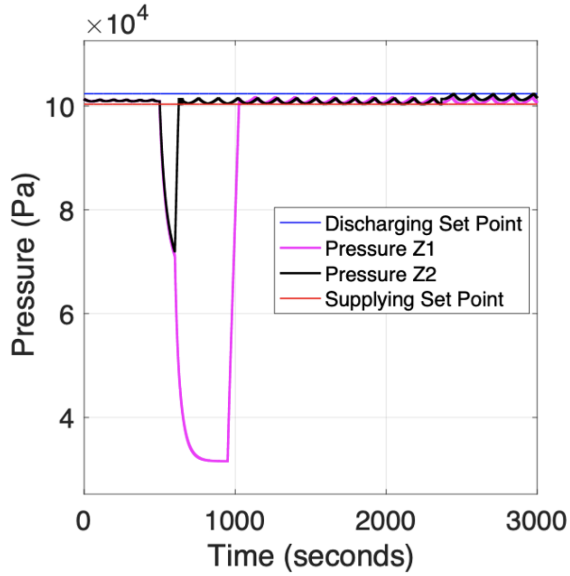
(a) ST hole radius

(b) SPL hole radius

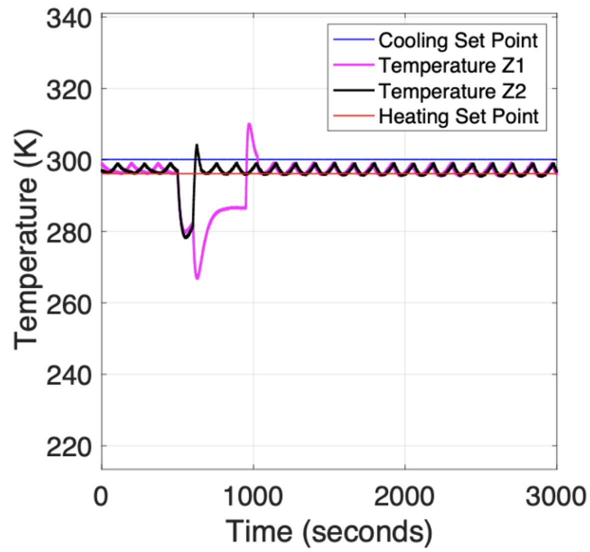
(c) Percentage of healthy battery cells

(d) Health state of power converter 1 and 2

Fig. 7   Health of select components



(a) Habitat pressure



(b) Habitat temperature

Fig. 8 Air pressure and air temperature inside the habitat

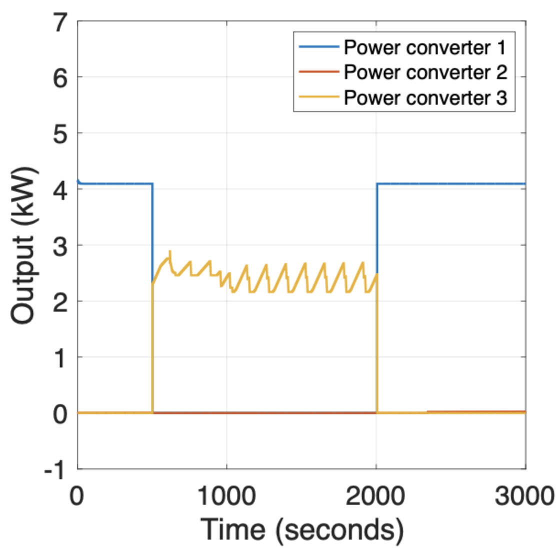
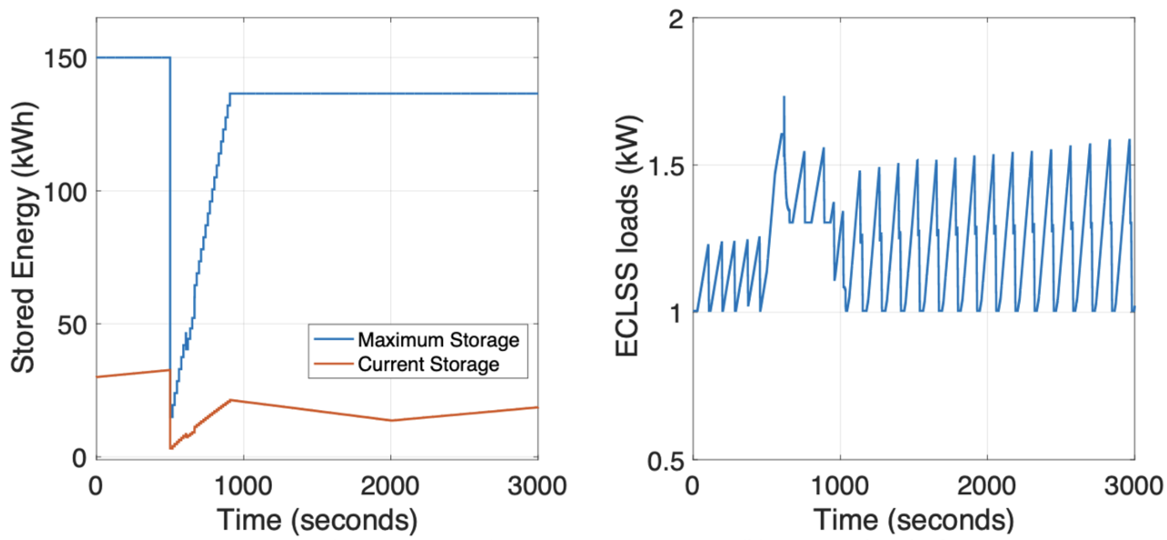


Fig. 9 Output of power converters 1, 2 and 3



(a) Energy storage capacity and stored energy (b) Power loads for ECLSS

Fig. 10 Power system stored energy and loads

Figure 9 demonstrates how the power supplied to the habitat (power converters 1-3) changes throughout this simulation based on the SPD. Note there is no solar power generation during this scenario (converter 2). As a result, the power distribution system reverts to the battery cells (converter 3) at 500 sec, right after the micrometeorite damages converter 1, until 2000 sec at which point the agent completes the repair of converter 1 (nuclear power generation).

Figure 10 shows how the number of available battery cells and the percentage of batteries charged changes over time. At the beginning of the simulation not all battery cells are charged, so a part of the nuclear power being generated is consumed right away due to the need to charge those battery cells. The micrometeorite strike at 500 sec results in damage to a large percentage of the battery cells, and the stored energy thus decreases immediately. After 510 sec when the AG starts replacing damaged cells with new fully charged cells, the available stored energy gradually increases. Figure 10a shows that the rate at which the stored energy increases is less than the rate at which new cells are added because the new battery cells are being used by ECLSS for regulating the IE pressure and temperature. However, the total stored energy increases because new battery cells are added at a faster rate than energy is consumed by ECLSS, shown in Fig. 10b. This process continues until 910 sec when all damaged batteries have been replaced. At this point the total stored energy starts decreasing until 2000 sec when the AG repairs converter 1 and the SPD system starts using the nuclear power being generated rather than the battery cells. The total stored energy in the batteries begins to increase again. The pocket door can be openedand the crew can resume their scheduled activities for the day.

This sample provides just one illustration of how HabSim can be used to investigate fundamental research questions related to the design and operation of space habitat systems exposed to disruptions. Here we demonstrated how the choice of repair priorities influences the ability to maintain a nominal state in the habitat. Many more disruption scenarios and variations on this scenario may be considered. Further samples are provided in the HabSim user manual [23].

V. Discussion

The HabSim platform is already enabling numerous research efforts. For instance, researchers have already used HabSim (formerly referred to as MCVT) for research into power systems and microgrids [26] , life-support systems, fault-tolerance, resilience [8] [29] [30], operational vulnerabilities [31] [32], and decision-making. Ongoing research also considers fundamental questions involved in commissioning and decommissioning the habitat when a mission involves dormancy. HabSim also provides sufficiently complex scenarios to address key questions in human-computer interface design. Researchers have also worked with individual subsystem models, outside of the broader HabSim model, to address specific research questions.

We anticipate that there are many additional opportunities to use the HabSim platform to explore habitat operation and autonomy. The code is being posted and made available for the space research community. The systems engineering tools and processes described herein, such as the DSM and our versatile standard notation, provide the mechanisms and building blocks that can enable a researcher to further expand the habitat system using this plug-and-play architecture. For instance, humidity, water recovery, inventory, consumable resources, etc., could all be added to the habitat model. Modifications to the layout of the overall habitat could readily be made in the input files. We do, however, recommend systematically following the procedures described herein.

In addition to its use as a stand-alone research platform, HabSim is directly facilitating the CPT developed by the RETH institute [9]. HabSim is modeled at one-fifth full scale specifically to match the size of the physical parameters of the surface habitat structure in the laboratory. The CPT combines physical and cyber (or virtual) models of subsystems to perform experiments using a broad range of conditions and disruptions [5] [9]. By design, the virtual models used in the CPT leverage HabSim subsystem models and their real-time capabilities. Environmental conditions and disruptions can be enforced in a controllable manner in the CPT, and reconfigurability is achieved mainly by modifying the virtual models. Having the ability to use these models directly for cyber-physical testing has proven to be quite useful for investigations into autonomous operations. Transfer systems are employed to properly account for the interactions at the interfaces between the physical and virtual components [33].

To directly ensure that the models in HabSim would be useful for the CPT, two specific considerations have been guiding our efforts since the beginning of this development effort: (a) the need for real-time execution of the models in HabSim; and (b) the extension of the DSM to support the development of the CPT. Real-time execution is a critical requirement for the CPT. Thus, an important requirement enforced during the development of each of the HabSim subsystem models is that it must run in real time. To ensure that this is the case, the balance between model fidelity and run-time needs is quite important to ensure that the intended testing can be done. Also, after completing the DSM for

HabSim, the subsystems were rearranged into physical and cyber to facilitate the partitioning process for using HabSim subsystem models to facilitate the RETHi CPT. The identified interdependencies, architecture, data flow, interface conditions, and operational dependencies are instrumental in making the CPT a reality.

VI. Conclusion

HabSim, a modular and coupled model and testbed that is able to capture the essential dynamics of an extraterrestrial habitat system, is available for fundamental research into the operation of deep space habitats. HabSim has been developed for researchers to carry out a wide array of quantitative research related to the resilience and autonomous operation of extraterrestrial habitats. The systematic integration of several physics-based and phenomenological models for habitat subsystems is discussed, built upon Matlab/Simulink for the modeling and simulating the integrated habitat as a system-of-systems. Access to a simulation framework like HabSim is essential for examining a variety of questions related to habitat design and operations, including resilience, contingencies, systems health management, and autonomy.

Several noteworthy tools adopted for HabSim development have facilitated the proper handling of data exchange and heterogeneous simulation for this complex, multi-physics system, such as the standard notation, design structure matrix, and real-time computational architecture. Defining a standard notation is critical to enable model developers to communicate requirements and streamline the exchange of information between the subsystems. Additionally, adopting scenarios as a means to drive model development provides a way to achieve a balance between model complexity and computational demand. Dynamics that are not needed for the simulations may not need to be included in the models. For research into contingency planning and resilient design in SmartHabs, the scenarios that are most relevant to NASA and partner agencies should be considered. The modeled disruptions in these scenarios include micrometeorites, fire, moonquakes, airlock leakage, and nuclear system coolant leakage. Persistent disturbances such as dust, vacuum, and thermal changes are also captured in the models. This process is meant to influence the functionality of the subsystem models. To demonstrate the functionalities and performance of HabSim, a sample disruption scenario is discussed and the repair priorities are defined. HabSim is modular and can be expanded or modified to consider alternative technologies, fault detection strategies, task scheduling, decision-making approaches, etc. Thus, this testbed can serve as a platform to support a variety of future research efforts.

Code Availability

HabSim, the user manual, and the files to run this sample simulation will be posted on GitHub RETHi repository soon. This paper corresponds to a description of version 6.3 of the HabSim MCVT simulation code.

Acknowledgments

This work was supported in part by a Space Technology Research Institute Grant (80NSSC19K1076) from NASA's Space Technology Research Grants Program. The authors would like to acknowledge the valuable contributions of RETHi team members including: Karen Marais, Ilias Bilionis, Ali Bazzi, Song Han, Arturo Montoya, Dawn Whitaker, Davide Ziviani, Mohammad Jahanshahi, Antonio Bobet, Justin Werfel, Jiachen Wang, Chuanyu Xue, Meghan Cilento, Jacqueline Ulmer, Takaharu Igarashi, Kenneth Pritchard, Matthew Winkler, Emily Testin, Ben Krugman, Dylan Bohanan, Ethan Vallerga, Sofia Velarde, Audrey DeKoninck, Alexa Barron, and Robert Kitching.

References

[1] NASA, "Moon to Mars Objectives," https://www.nasa.gov/press-release/nasa-s-stakeholder-collaborations-help-inform-moon-to-mars-planning, 2022. Accessed September 25, 2023.

[2] NASA, "NASA'S Moon to Mars Strategy and Objectives Development: A Blueprint for Sustained Human Presence and Exploration throughout the Solar System," https://www.nasa.gov/feature/nasa-details-strategy-behind-blueprint-for-moon-to-mars-exploration, 2023. Accessed September 25, 2023.

[3] NASA, "NASA's Lunar Exploration Program Overview," https://www.nasa.gov/wp-content/uploads/2020/12/artemis_plan-20200921.pdf, 2020. Accessed September 25, 2023.

[4] Lunar Surface Innovation Consortium, "LSIC Web Portal," https://lsic.jhuapl.edu/, 2019. Accessed June 30, 2024.

[5] Dyke, S., Marais, K., Bilionis, I., Werfel, J., and Malla, R., "Strategies for the Design and Operation of Resilient Extraterrestrial Habitats," *Proc. SPIE Smart Structures + Nondestructive Evaluation Conference*, 2021.

[6] Nunn, C., Garcia, R., Nakamura, Y., Marusiak, A., Kawamura, T., Sun, D., Margerin, L., Weber, R., Drilleau, M., Wieczorek, M., Khan, A., Rivoldini, A., Lognonné, P., and Zhu, P., "Lunar Seismology: A Data and Instrumentation Review," *Space Sci Rev*, Vol. 216, 2020, p. 89. https://doi.org/10.1007/s11214-020-00709-3.

[7] Holloway, E., D. Taylor, and Badger, J., "When the Eyes Don't Have It: Autonomous Control of Deep Space Vehicles for Human Spaceflight," *Proc of the AIAA ASCEND*, 2020. https://doi.org/https://doi.org/10.2514/6.2020-4163.

[8] Kitching, R., Mattingly, H., Williams, D., and Marais, K., "Resilient Space Habitat Design Using Safety Controls," *Proc. Earth and Space Conference, ASCE*, Seattle, Washington, 2021. https://doi.org/https://doi.org/10.1061/9780784483381.

[9] Maghareh, A., Lenjani, A., Dyke, S., Marais, K., Whitaker, D., Bobet, A., Ramirez, J., Melosh, H., Modiriasari, A., and Theinat, A., "Resilience-oriented Design of Extraterrestrial Habitat Systems," *Proceedings of the AIAA Propulsion and Energy Forum*, 2019. https://doi.org/10.2514/6.2019-3972.

[10] Badger, J., and Frank, J. D., "Spacecraft Dormancy Operational Design for a Crewed Martian Reference Mission," *2018 AIAA SPACE and Astronautics Forum and Exposition*, American Institute of Aeronautics and Astronautics, Orlando, FL, 2018. https://doi.org/10.2514/6.2018-5117.

[11] Muri, P., Hanson, S., and Sonnier, M., "Gateway Avionics Concept of Operations for Command and Data Handling Architecture," *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–7. https://doi.org/10.1109/AERO50100.2021.9438539.

[12] Oberkampf, W. L., and Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press, 2010.

[13] Heinicke, C., and Arnhof, M., "A Review of Existing Analog Habitats and Lessons for Future Lunar and Martian Habitats," *Reviews in Human Space Exploration*, 2021, p. 100038. https://doi.org/10.1016/j.reach.2021.100038.

[14] Detrell, G., Belz, S., and Keppler, J., "ELISSA: A Life Support System (LSS) technology selection, modelling and simulation tool for human spaceflight missions," *Proc. 42nd COSPAR Scientific Assembly*, Vol. 42, 2018, p. F4.3–5–18.

[15] Hendrickx, L., De Wever, H., Hermans, V., Mastroleo, F., Morin, N., Wilmotte, A., Janssen, P., and Mergeay, M., "Microbial Ecology of the Closed Artificial Ecosystem MELiSSA (Micro-Ecological Life Support System Alternative): Reinventing and Compartmentalizing the Earth's Food and Oxygen Regeneration System for Long-haul Space Exploration Missions," *Research in Microbiology*, Vol. 157, No. 1, 2006, pp. 77–86. https://doi.org/10.1016/j.resmic.2005.06.014.

[16] Czupalla, M., Zhukov, A., Schnaitmann, J., Olthoff, C., Deiml, M., Plotner, P., and Walter, U., "The Virtual Habitat – A Tool for Dynamic Life Support System Simulations," *Advances in Space Research*, 2015. https://doi.org/10.1016/j.asr.2014.12.029.

[17] Mathworks, "Simulation and Model-Based Design," https://www.mathworks.com/products/simulink.html, 2020. Accessed last on November 13, 2023.

[18] Behjat, A., Ibrahimov, R., Lenjani, A., Barket, A., Martinus, K., Maghareh, A., Whitaker, D., Bilionis, I., and Dyke, S., "A Computational Framework for the Evaluation of Resilience in Deep Space Habitat Systems," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 3A: 48th Design Automation Conference (DAC), 2022, p. V03AT03A036. https://doi.org/10.1115/DETC2022-89132.

[19] Behjat, A., Liu, X., Forero, O., Ibrahimov, R., Dyke, S., Bilionis, I., Ramirez, J., and Whitaker, D., "A Computational Framework for Making Early Design Decisions in Deep Space Habitats," *Advances in Engineering Software*, Vol. 195, 2024. https://doi.org/https://doi.org/10.1016/j.advengsoft.2024.103690.

[20] Fritzson, P., *Principles of Object-oriented Modeling and Simulation with Modelica 3.3: A Cyber-physical Approach*, John Wiley & Sons, 2014.

[21] Penn, J., and Lin, A., "The Trick Simulation Toolkit: A NASA/Opensource Framework for Running Time Based Physics Models," *Proc. Modeling and Simulation Technologies Conference, AIAA*, 2016, p. 1187. https://doi.org/https://doi.org/10.2514/6.2016-1187.

[22] Eppinger, S., and Browning, T., *Design Structure Matrix Methods and Applications*, MIT press, 2012.

[23] RETH institute, "HabSim v6.3 User Manual," https://www.purdue.edu/rethi/files/HabSim%20User%20Manual.pdf, 2024. Accessed June 30, 2024.

[24] Maghareh, A., Lenjani, A., Krishnan, M., Dyke, S., and Bilionis, I., "Role of Cyber-Physical Testing in Developing Resilient Extraterrestrial Habitats," *Earth and Space 2021*, ASCE, 2021, pp. 1059–1068. https://doi.org/10.1061/9780784483374.098.

[25] Rhee, S., Noble, Z., Park, J., Lial, A., Collazo, L., and Ziviani, D., "Development of a Damageable ECLSS and IE Virtual Testbed Model to Simulate Future Resilient Deep Space Habitats," *Proceedings of the 52nd International Conference on Environmental Systems*, Calgary, Canada, 2023.

[26] Chebbo, L., Gultekin, M., Bazzi, A., Tomastik, R., Pattipati, K., Vaccino, L., Azimi, M., and Lund, A., "Modeling and Operation of Microgrids for Deep Space Habitats Under Environmental Disturbances," *IEEE Power and Energy Conference at Illinois*, 2022. https://doi.org/10.1109/PECI57361.2023.10197727.

[27] Engel, A., *Verification, Validation, and Testing of Engineered Systems*, 2010.

[28] Rinaldi, S. M., Peerenboom, J. P., and Kelly, T. K., "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies," *IEEE Control Systems*, Vol. 21, No. 6, 2001, pp. 11–25. https://doi.org/10.1109/37.969131.

[29] Cilento, M. V., "Resilient Extra-Terrestrial Habitat Design Using a Control Effectiveness Metric," Master's thesis, Purdue University Graduate School, West Lafayette, IN, 2022.

[30] Ulmer, J., "A Resilience Oriented Extra-terrestrial Habitat Design Process," Master's thesis, Purdue University Graduate School, West Lafayette, IN, 2023.

[31] Pritchard, K., "A Simulation-based Study of Operational Vulnerabilities and Contingency Planning for Smart Extraterrestrial Habitats," Master's thesis, Purdue University Graduate School, West Lafayette, IN, 2023.

[32] Vaccino, L., Pritchard, K., Azimi, M., and Dyke, S., "Simulation-Based Assessment of Hazardous States in a Deep Space Habitat," *Proceedings of the 52nd International Conference on Environmental Systems*, Calgary, Canada, 2023.

[33] Montoya, H., Dyke, S. J., Silva, C. E., Maghareh, A., Park, J., and Ziviani, D., "Thermomechanical Real-Time Hybrid Simulation: Conceptual Framework and Control Requirements," *AIAA Journal*, Vol. 61, No. 6, 2023, pp. 2627–2639. https://doi.org/https://doi.org/10.2514/1.J062857.