53rd International Conference on Environmental Systems
21-25 July 2024, Louisville, Kentucky ICES-2024-53

In-Situ Resource Utilization Modeling of a Lunar Water Processing System

Avery Carlson[1] and Noah Andersen[2]
Jacobs Technology, Houston, TX, 77058, USA

Jacob Collins[3]
NASA Johnson Space Center, Houston, TX, 77058, USA

A key element of achieving a sustained surface presence, such as defined in NASA's Artemis plan, is In-Situ Resource Utilization (ISRU). ISRU is the practice of using local resources to provide mission consumables that reduce system launch mass requirements, and regenerate resources (chiefly, water and oxygen) for propulsion and life support supporting both Lunar and Martian missions. ISRU systems require multiple complex processes, such as excavation, chemical reactors, and electrolysis subsystems that must operate in harmony to optimize the overall system process from beginning to end. The Mission Analysis and Integration Tool (MAIT) was previously developed with MATLAB in FY22 to connect individual subsystem models into a customized, flexible framework for the purpose of technology downselect, optimization, and end-to-end process planning. Beginning in FY24, MAIT was updated and became the capital program in the Systems Engineering and Integration (SE&I) ISRU Modeling and Analysis (SIMA) project. Prior work was leveraged and evolved using MATLAB/Simulink due to its ability to communicate with a vast number of other programming languages and makes up the backbone of data flow between inputs and outputs to the subsystem models. MAIT initially evaluated a suite of ISRU-related technologies, including the water processing Lunar Auger Dryer for ISRU (LADI) system with integrated upstream excavation and downstream electrolysis subsystems. With individual models consolidated, the MAIT tool generated over 5,000 cases during its first round of parametric sweeps on the water processing architecture at multiple production targets; the system analysis produced valuable insight into the optimal LADI geometry that minimized energy demands, estimated effects to cold trap size and radiator requirements, and calculated the power dynamics of the electrolysis unit and liquid oxygen storage volume. Additional efforts are being made to demonstrate the ability to scale ISRU technologies supporting the Space Technology Mission Directorate's (STMD) commercialization strategy and increase the MAIT software capability. Work is ongoing to handle a wide array of ISRU system models beyond the Lunar environment, e.g. production of propellant for a Martian lander.

Nomenclature

AIAA	=	American Institute of Aeronautics and Astronautics
ANSI	=	American National Standard Institute
API	=	Application Program Interface
ESM	=	Equivalent System Mass
FY	=	Fiscal Year
ISRU	=	In-Situ Resource Utilization
JSC	=	Johnson Space Center
KPP	=	Key Performance Parameter
LADI	=	Lunar Auger Dryer for ISRU
MAIT	=	Mission Analysis and Integration Tool

[1] Chemical Engineer, Life Support Systems, 2224 Bay Area Blvd, Houston, TX 77058.
[2] Thermal Analysis Engineer, Thermal and Life Support Systems, 2224 Bay Area Blvd, Houston, TX 77058.
[3] Aerospace Engineer, Energy Conversion Systems Branch (EP3), 2101 NASA Parkway, Houston, TX 77058.

MBSE	=	Model-Based System Engineering
mdot	=	mass array
MGA	=	Mass Growth Allowance
MRE	=	Molten Regolith Extraction
mt	=	metric ton (1,000 kg)
PSR	=	Permanently Shadowed Region
PEM	=	Proton Exchange Membrane
SE&I	=	Systems Engineering and Integration
SIMA	=	SE&I ISRU Modeling and Analysis
STMD	=	Science Technology Mission Directorate
TRL	=	Technology Readiness Level

I. Introduction

A critical element of NASA's return to the Lunar surface with the Artemis missions and future planned missions to the Martian surface is the ability to generate usable material without compelling resupply launches from Earth into low earth orbit and to the final destination. Previous studies of the Gear Ratio Effect estimate approximately 8 and 13 kg of propellant and rocket stages to deliver 1 kg of payload to the Lunar or Martian surface, respectively, and that factor increases considerably if accounting for return journeys.[1] Incorporation of in situ resource utilization (ISRU) practices—i.e., the process of producing those mission consumables from local lunar/planetary resources—minimizes those system launch mass requirements, mainly through the production of propellant fuel components and life support resources. Modeling full ISRU technology architectures significantly impacts the design and optimization of individual unit processes, and facilitates their introduction into service with NASA's logistical planning and operation space. The Space Technology Mission Directorate (STMD) calls for ISRU-enabling technologies throughout their recently published "Go, Land, Live, and Explore" initiatives to advance propulsion and promote logistical mobility within missions on the first two thrusts, while outcomes adjacent to ISRU, such as power systems, thermal management, excavation, and mobile platforms are the focus of the latter two thrusts.[2] Furthermore, the Moon to Mars Architecture documents specify ISRU system objectives and priorities that will be required to transition to more permanent human environments.[3] The complexity, number of unique technologies, and scale of this undertaking necessitates the use of modeling techniques to develop accurate predictions of potential ISRU plant designs, and focus efforts on the most critical design bottlenecks.

The Mission Analysis and Integration Tool (MAIT) is a framework that connects individual ISRU subsystem models for the purposes of technology downselect, scaling & optimization, and end-to-end process planning. The aim in building MAIT is to address specific STMD shortfalls by simulating ISRU systems, and producing models that future NASA project managers and engineers can use to link Key Performance Parameters (KPP) and environmental requirements by employing Model-Based System Engineering (MBSE) techniques. MAIT will remain for the time being as an internal tool to run by NASA personnel, but was designed with interoperability and the convenience of developer-driven models in mind. The framework is adaptive, flexible, and suitable for collaboration among government, private sector, and international entities should they decide to partner with the SIMA team. The team's current approach seeks to provide conventional outputs, such as mass, power, and volume, to analyze whole system trade spaces. Moreover, the framework was designed to validate current breadboard subsystems, simulate future process design capacity, and allow enough flexibility to any developer-driven subsystem models outside of JSC that are in a state of evolution. MAIT consists of modular elements that read and pass information between individual subsystems, which can then be arranged into a customized full architecture. These modular elements are robust enough to access files written in multiple software languages, and the "drag-and-drop" data manipulation can modify or substitute process flow diagrams according to the desired end user need. This tool is currently being used to assess various interconnected ISRU technologies in water processing and oxygen extraction from lunar regolith; however, the standardized modular design is generic in order to implement a wide array of ISRU subsystem models.

The objective of this paper is to provide an overview of the MAIT framework and logic connecting ISRU-related technologies in early application. In doing so, the hope is to expose the larger ISRU community to MAIT's standards and interoperability requirements. MAIT can rapidly (on the order of a couple of days) perform thousands of parametric sweeps, and as a benefit for partnering with SIMA, NASA partners and stakeholders will gain insight as to how their technology would perform within a complete ISRU system on a faster timescale. The aim is to use this example study to foster unique design configurations with multiple subsystems, and/or trade studies with different architectures.

II. Overview of MAIT Framework

The framework for MAIT consists of three parts: a user-manipulated spreadsheet containing the master list of subsystem variables, the process flow environment encoded and built in Simulink (part of the Mathworks graphical programming suite),[4] and the subsystem model files prepared by technology developers. The master list is a spreadsheet that contains a library of variables organized under specific categories to be stored in a MATLAB data file and called later by wrapper functions within the process flow environment. Variables are designated as *Global* (pertinent environmental or mission-specific conditions), *Local* (applicable to an individual subsystem), or *Passed*. Examples of a global variable include the Lunar or Martian gravity constant, the density and composition of excavated material (such as an assumed Lunar regolith in the example below), mission duration, or system operating time. Local variables are designated as process-specific, so all input and output variables required to run a subsystem model are recorded in the master list and then called in order of the system configuration. Passed variables describe dynamic information that cascades through the subsystem model blocks and is constantly being updated with the results of model runs or parametric sweeps. One type of *Passed* variable is a standardized dataset containing critical chemical, fluid, and thermodynamic parameters, which are sequentially used as an input and updated by each block module before being passed to subsequent modules. The other type would only be applicable for a specific case where a subsystem is split into two separate file types, and therefore modeling the complete subsystem requires connecting an additional array of non-standardized data from one module to another.

The MATLAB framework stores Global, Local, and Passed variables separately so they can be directed more easily to the appropriate downstream module (see Figure 1). The overall data handling strategy pursued was one that avoided having to pass every individual input and output to each consecutive module. The MATLAB base code configuring the block modules was written to be compatible with a number of software languages, allowing the subsystem model to run in its original form based on its required inputs and transfer the output data back into their respective location in the MAIT matrix. The outputs are delineated in an output directory between common parameters of interest to the user (standard results) and the local results relevant to each individual unit process.

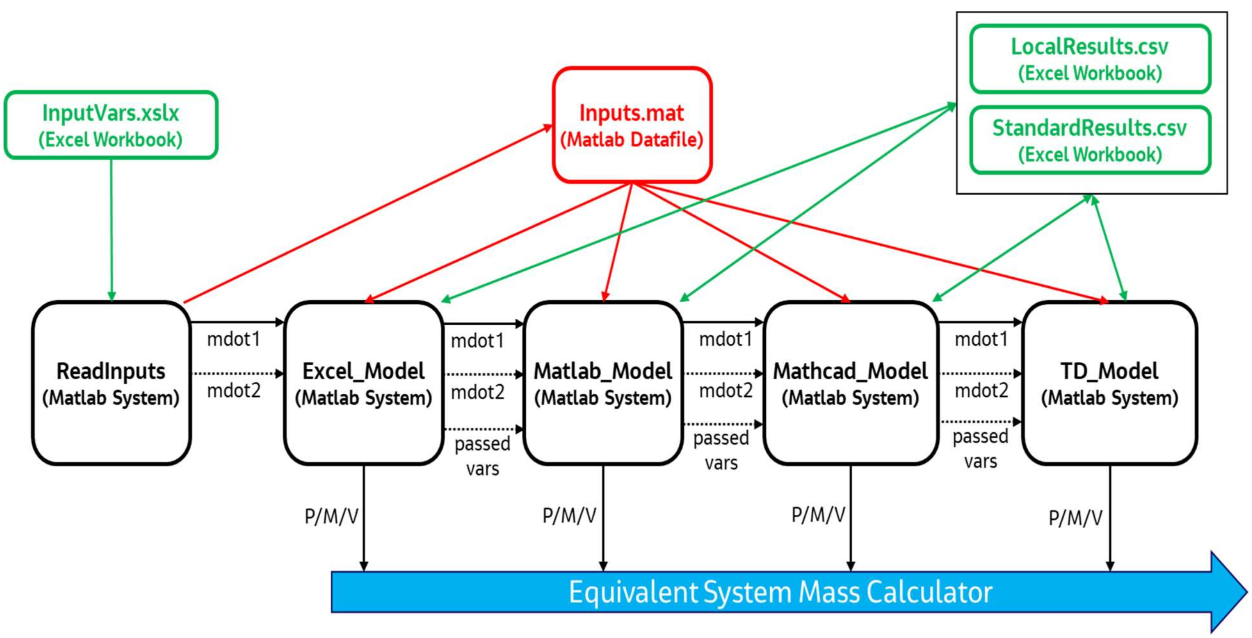

Figure 1: MAIT process flow diagram. *Example subsystem modules are shown in black, and external or saved files are shown in red and green. Black arrows indicate the flow of passed variables and are directly visible in the Simulink environment. Red and green arrows indicate flow of all variables, and are not visible in the Simulink environment, but instead automatically executed for every run of the model.*

A. Variable Master List and Input Structure

The library of variables used within the MAIT framework are arranged into a master list. The purpose for centralizing all variables in an external location is that it allows MAIT to index and reference all input and output variable names in the correct order and enables communication to external subsystem files. As the overall architecture complexity increases, the variable list can drastically expand, mostly due to a growing number of local variables to describe the connected subsystems. A simple example of some items in the master list input array is shown in Table 1. A script (*ReadInputs* depicted in Figure 1) accesses the master list (*InputVars*) to fill in values for the Global and Passed variables array. Local variables are organized by model name and stored in separate sheets to simplify the addition or removal of subsystems without affecting the overall architecture. *ReadInputs* stores all variable names and values in an internal MATLAB data file. Additionally, the script reads into the data file all variables desired for a parametric sweep run, such as variable name, range of values to include in the sweep, and the number of iterations required (i.e., step size). Parametric sweeps can be performed on any variable type, i.e., global, passed, or local, depending on the analysis goals. This comprehensive set of data (*Inputs.mat*) is called sequentially by the module functions as the modeling run progresses to completion.

B. Process Flow Environment & Module Description

The majority of architecture modeling work is completed in a Simulink process flow environment, a MBSE application[4] that connects the library of variables (or master list) mentioned previously to module blocks that, in turn, access and run individual subsystem model files. The graphical user interface of MAIT consists of blocks and arrows representing the modular functions and standardized flow of data that connects input and output ports. Inputs consist of the Passed variables described previously. Commonly reported output components used to calculate equivalent system mass (ESM), such as heat rejection requirements, power demand, and subsystem mass, are recorded for downstream accounting (see Figure 1). Other localized results taken from subsystem modeling runs are written to an accessible file directory.

Table 1: Simple Example of MAIT Input Tale *displaying variables & headers used to index within internal array*

Category	Input/Output Flag	Variable Name	Units
Global	Input	Mission_Duration	d
Global	Input	Temperature	°C
Global	Input	Density	kg/m^3
Passed	Input	Inlet_Mass	kg/hr
Passed	Input	Inlet_Pressure	Pa
Passed	Input	Inlet_Temperature	K
[Model Name] Local	Input	Dimension1	m
[Model Name] Local	Input	Dimension2	m
…	…	…	-
Passed	Output	Outlet_Mass	kg/hr
Passed	Output	Outlet_Pressure	Pa
Passed	Output	Outlet_Temperature	K
…	…	…	-
[Model Name] Local	Output	…	
[Model Name] Local	Output	…	
…	…	…	

The modules are purposefully generic in order to copy and paste directly in the flow environment. Instead of designing around a specific unit process or piece of equipment, functional scripts (wrapper functions) were written to communicate with and run a model written in an external software language. One of the biggest difficulties in establishing a modular ISRU system modeling tool is that subsystem models are often created independently by different NASA centers or commercial partners. Each model has different naming conventions, software, assumptions, and levels of fidelity. In addition to creating the software capable of interfacing different subsystem models, MAIT has established a standard wrapper and architecture for how a subsystem model should be interfaced with the MAIT tool. MAIT can currently communicate with programming languages in Excel, Mathcad, MATLAB, and Thermal Desktop. Future work is planned that will allow additional modeling software packages such as COMSOL or Aspen.

The inputs to a generic module are:

1. a standardized mass flow array with components needed for mass and energy balances (*mdot*), which contains chemical species mass flow rates (kg/hr), temperature (°C), pressure (Pa),
2. any supplied heat or power (W),
3. global variables that describe environmental conditions at the plant site,
4. local variables necessary to run a particular subsystem model.

Because the module is dependent on the software language, the wrapper will index local variables in the larger information matrix by the unit process name dictated in the master list. Boolean indexing is used to pull all relevant variables needed for a particular subsystem model from the master list. The module code, after interpreting the model order, directs the input mass array to the first port, retrieves the global variables, looks in the stored matrix for its corresponding local variables based on the model's name, and submits this total information package to an individual subsystem file (Figure 2). The subsystem model runs, and the outputs are organized according to a standard structure. The mdot output array consists of the same variables as the input. However, any chemical transformations or updates to the constituent flow rates will be stored there.

Using this generic modular framework, the whole system configuration can be written as processes in series (as Figure 1 implies), or in parallel, enabling the user ability to customize the configuration for many project types. In other words, one unit process with two outlet flow streams can communicate with two (or more) subsystem models operating independently from each other, facilitating the creation of fit-for-purpose ISRU architectures characterizing any number of unique unit processes (a more detailed discussion of use cases is provided below).

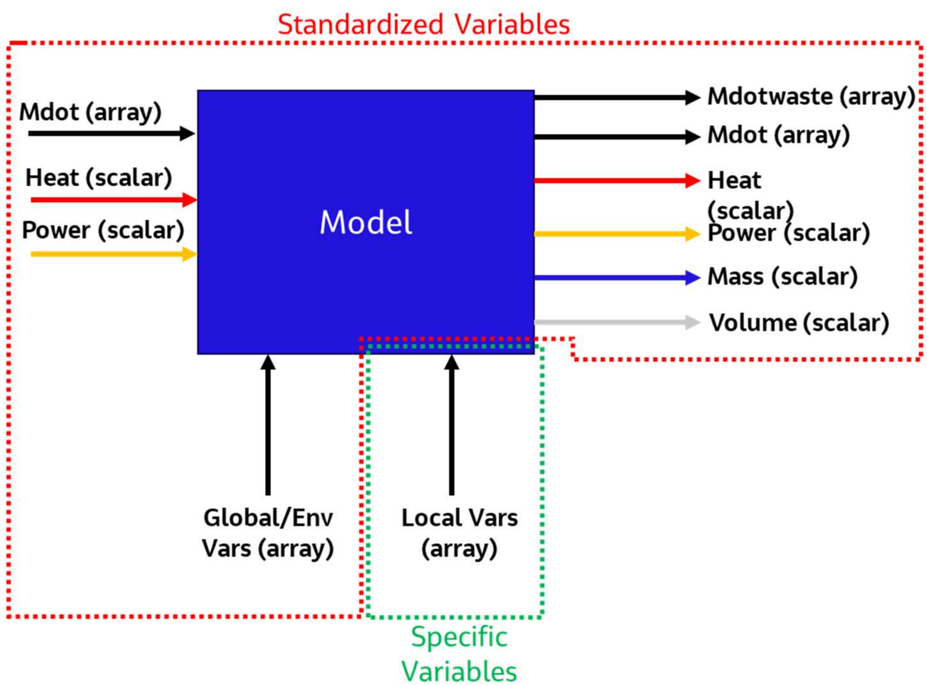

Figure 2: Visual of the Generic Module approach to subsystem modeling in MAIT. *Input and output flow streams shown in color.*

C. Subsystem Models

Underlying the module code are external models that describe individual subsystem unit processes, each characterized in as much detail as possible to appropriately describe the transformation of material. MAIT is intentionally built to be "software agnostic", i.e., it would work conveniently with the technology developer's software of choice. Currently, MAIT can communicate with a span of five to ten subsystem model files in four different software languages, but in theory is limited only by computing resources necessary to iterate through the desired runs. Most subsystem models written in a Mathworks or Microsoft Office software language can be accessed via the *actxserver* command, which establishes a server connection to transfer designated MATLAB commands sending input data to VBA-encoded programs, and transferring back output data. The original intention was to utilize MAIT to model high-fidelity systems, however, the modular design provides the means for breadboard data validation and can provide useful feedback for subsystems at lower technology readiness levels (TRLs). Internal NASA and external private industry subsystem models exist at varying levels of fidelity, therefore significant effort was made to ensure that model assumptions, first principle fundamentals, and any critical design conditions or flaws were documented. As part of the quantitative analysis, mass growth allowances (MGAs) were taken into account for any mass estimates using the American National Standards Institute (ANSI) & American Institute of Aeronautics and Astronautics (AIAA) mass properties standards.[5] These standards helped translate the level of model uncertainty into an appropriate level of mass margin. Basic mass (also known as calculated mass) is recorded in MAIT as the mass of the subsystem model, whereas predicted mass includes the appropriate MGA based on the components being modeled and the level of model fidelity (e.g., higher fidelity models have lower MGAs). Similar methods were used to find the calculated power demand and added power growth allowance. This logic was built into the ESM calculator.

III. MAIT Setup and Operation

Prior to running a simulation in MAIT, the user must ensure global and passed variable names were written into all subsystem model files, and update all local variable names in the master list. This allows the MAIT base code to keep the integrity of its internally-stored arrays—i.e., the names of passed variables and global variables will be the same between runs unless the user is recharacterizing the environmental conditions at an ISRU plant location or mission-specific parameters. Engineering judgement should be used when assembling a run, determining which subsystems will be used, how modules connect in the flow environment (i.e., the ISRU architecture configuration), any important assumptions underlying the subsystem models, and the important variables to include in a parametric sweep.

A. Parametric Sweeps

Parametric sweeps are a method of testing the boundaries of uncertainty in a given complex system by repeatedly testing independent parameters in a model to accumulate a set of calculated outputs.[6] In the context of subsystem modeling, not only does this method provide the user information on key relationships between independent variables and key performance criteria in trade study analysis, it can also be used as a crude first step in optimizing the overall architecture and identifying performance bottlenecks. MAIT performs parametric sweeps by first calculating a permutation of the number of variables desired in the analysis. If X number of variables are flagged in the input master list with Y number of step changes, the MAIT *ReadInputs* script fills a [Y^X by X] linear space with the different combinations of inputs. The total number of rows in the linear space then accounts for the total number of model iterations to be performed, and each value is read into the process flow environment in succession. Simulink's built-in time-dependent functions were used to generate plots to show a chronological sequence of the data as the parametric sweep is performed. The user can then use these plots to infer relationships between the variables in the sweep and whole system optimization.

One of the biggest difficulties in running parametric sweeps on system models is the computational resource usage. Small changes in an upstream process may have vast impacts downstream, therefore parametric sweeps must be run on variables across all system models. On the other hand, as the number of variables being parameterized increases, the total runs needed to record those results grow nearly exponentially. To improve computer resource requirements and minimize runtime, additional logic was included in MAIT to leverage data recorded in prior runs of a parametric sweep as a way of reducing the total number of system iterations. For instance, during a parametric sweep where upstream subsystems are held static, code at the subsystem level checks the input flow against a database created from all previous iterations. If the inputs are the same, it is determined that the subsystem model has the same inputs, and thus same outputs, as the previous run. This built-in capability greatly reduces the computational resources required by MAIT to run system models, particularly for system models where recycle loops (discussed in the next section) are needed.

B. Recycling Streams

On the system process side, there were numerous situations requiring logic for a recycle loop, i.e., unreacted, or unused material from a unit process recycled back to the input. The change had a rather outsized effect on the MAIT process flow environment, as it required modification to the time-dependent methodology used in the parametric sweeps mentioned above. The problem was two-fold: first, the recycle loop must maintain the conservation of mass and energy of component streams until steady state is achieved; second, there must exist an iterative format to maintain a linear timeline used in the parametric sweep logic. To answer the former issue, two functions were created that operate as a "splitter" and a "mixer" at the outlet and inlet sides of the subsystem, respectively. As the names suggest, these functions take individual component mass streams and either separate them at the outlet (in the case of the splitter) or combine them as one stream to the inlet (in the case of the mixer). For the current application of MAIT, these "splitter" and "mixer" functions are strictly arithmetic, and do not account for the physical processes needed to split or merge flows. They also do not allow for accurate calculations on the impact of merging or splitting the streams on the thermodynamic properties of the flow (e.g. temperature). In the future, these functions will be upgraded by subsystem models that directly model the physics required to split streams (e.g. removing unreacted water downstream of an electrolysis process from the flow of H_2 and O_2).

Equilibrium between inputs and outputs to a subsystem is determined by the recycle tolerance, a conditional flag specified by the user that is satisfied when the inlet and outlet mass are the same as the previous iteration within the value of the tolerance. This logic determines the number of recycle loops to perform. While recycle loops can increase computational resources significantly, the database reuse code discussed in the prior section prevents overrunning of unneeded subsystem models that are not included in the recycle loop. Along the recycle flow path is a signal delay function, an internal Simulink function that allows a time step delay between each iteration to avoid the recycle getting caught in an infinite loop. The linearity of the process is maintained by implicitly setting each iteration of the parametric sweep equal to one unit of time and each iteration of the recycle loop (partitioned by the signal delay) prior to equilibrium as a partial unit of time. For instance, if 5 iterations of a parametric sweep are determined from the range and step size of a variable, and 5 iterations are required per pass of the recycle loop, variables will be plotted against time units beginning at 1.0, 1.2, 1.4, 1.6, 1.8 for the first parametric case, 2.0, 2.2, …2.8 for the second parametric case, and so on. Current iterations of MAIT require the user to specify the number of recycle loops. The code will identify when equilibrium has been reached, but the user will need to manually increase the specified iteration loops if it is found that more iteration loops are needed.

Overall, this methodology affords the modeler the ability to perform parametric sweeps and recycle loops simultaneously. An example of a recycle loop is discussed in the example case below: recycling water in the electrolysis subsystem.

C. System Requirements Management

Significant effort was expended investigating each subsystem to document their first principles, key assumptions, and critical issues. Then, grades were assigned to each model based on NASA's TRL scale, and an informal weighting system. The use of requirements management tools is the next aspect of system engineering that house environmental specifications, project standards, and KPPs as a way of maintaining uniform standards. Future ISRU designs modeled in MAIT will be connected to a requirements management software (Jama Connect) via an application program interface (API). The next version of MAIT will utilized Jama Connect to verify success or failure for multiple hard production targets, ranging from pilot scale (1 mt/year) to full scale (10 mt/year and 50 mt/year), and validate whether the MAIT simulations hit larger environmental specifications and program goals and objectives.

IV. ISRU Use Case: Lunar Water Mining and Processing Architecture

The power of ISRU system modeling with MAIT can be exemplified through a use case. The water mining architecture—one in a number of ISRU pilot plant designs in the trade space—is being considered for the production of propellant from icy Lunar regolith.[7,8] The Lunar water mining architecture has numerous assumptions built into the modeling framework, starting with a designated location of the ISRU plant and a general operation to retrieve the material and provide the power necessary for extraction. The outputs provide insights into integrated system performance and the relationship of key subsystem variables.

The ISRU water processing system is expected to operate in both a Permanently Shadowed Region (PSR) and a lunar ridge environment as shown in **Error! Reference source not found.**~~Figure 3~~ (yellow & purple boxes, respectively). The PSR consists of Highland mare regolith, which is believed to contain a low concentration of frozen water.[9] The excavation and water processing are planned to take place in the confines of the PSR, while downstream refinement and storage in a sunnier region above the PSR. The robotic excavator can be scaled up in the subsystem model to meet production values using data from ground testing and assuming a planned operation within a certain radius of a Lunar ISRU installation. The excavator sends its aggregated regolith to an actuating hopper that both size sorts the regolith feedstock to prevent large rocks from entering and stores enough capacity to ensure continuous flow into the downstream auger. The model accounts for the excavator drum geometry in estimating the regolith flow rate and equivalent number of hoppers needed to process deliveries from the excavator.

The Lunar Auger Dryer for ISRU (LADI) is a unit process consisting of a thermal casing that transfers heat to a plug of regolith as it passes down the length of rotating screw (i.e., screw conveyor).[10] The environment inside LADI is characterized by certain physical properties of the rotating regolith in developing a plug formation at the end of the unit and thermal properties such as the heat transfer unit that affects how energy is directed towards the desired phase changes (e.g., heating the ice within the regolith and sublimating the ice to water vapor at a certain temperature). Due to the complexity of this step, the model then validates the thermal property assumptions through different methods, given material properties of the heater and casing.

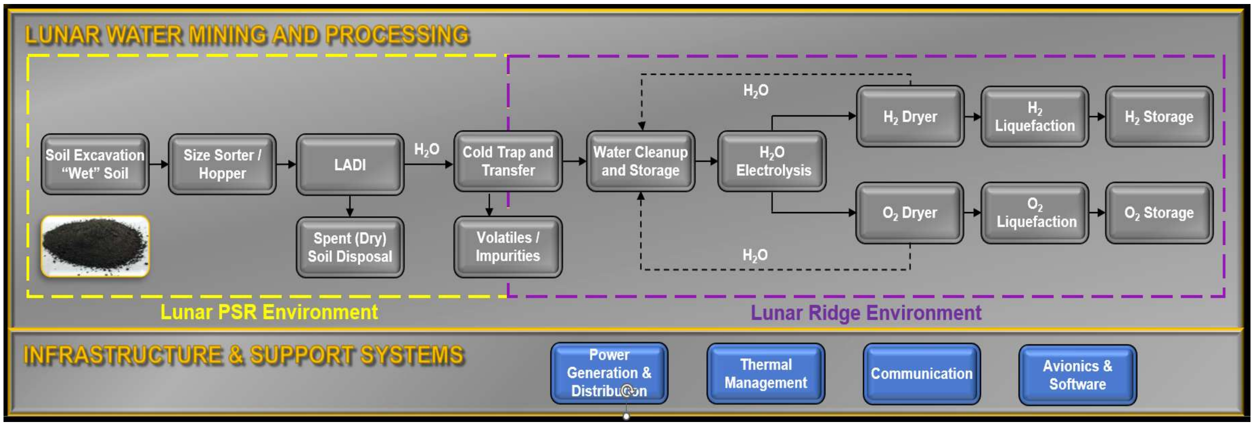

Figure 3: Flow diagram of the proposed Lunar water mining and processing architecture. *The assumed locations of the critical unit processes are low-sunlight regions of a PSR (yellow), and sunnier regions above a PSR (purple).*

Water vapor formed in the LADI is subsequently collected in a cold trap based on the dynamic diffusion-limited growth of a dense ice layer along the surface of a cold wall. The cold trap surface area is assumed to exist within a cylindrical reactor space with internal fins to maximize the surface area to volume ratio, and the calculated heat flux from this specified configuration lends itself to prerequisite radiator heat rejection. All of the cold trap's individual components are calculated within its subsystem model. Finally, after transport of the collected frozen ice up to the edge of the PSR via a mobility platform, a heat exchange process occurs to melt it to a liquid stream. Downstream electrolysis is performed to split water into hydrogen and oxygen gas streams, which is modeled via the proton exchange membrane (PEM) electrolyzer.[11] These gases are presumed to be saturated and thus require drying before ultimately sending these desired products to liquefaction storage. This is essentially a cryogenic process used as a placeholder for the storage phase prior to transfer to a mobile shuttle. The storage of the separated O_2 and H_2 marks the end of this project scope; however, the auxiliary infrastructure necessary for power generation & distribution were also included in this study using information from (sources). MAIT used the output system power demand working backwards to input the flow of energy into Vertical Solar Array Technology (VSAT) mass sizing model. Furthermore, it took into account 5 km of high voltage transmission wire and the mass of essential step-up/step-down converters. The outputs from the system architecture, and power generation & distribution were accounted for in the architecture ESM, and are now available to trade against other ISRU system designs with alternative subsystem technologies or configurations. Note that while the technologies described in this section were used for this investigation, the modularity of MAIT would allow for many different configurations, provided subsystem models were made available.

An initial parametric sweep of the water processing architecture was conducted using the variables below. These were chosen based on model assumptions and their level of anticipated criticality to the subsystem design. The sweep consisted of between 2 and 4 values for each variable, which totaled roughly 5,000 individual iterations of the entire system architecture across three production targets (1, 10, and 50 mt/year). The values were refined from the parametric results from an earlier version of MAIT (data not shown), as well as any limiting conditions that could place minimums or maximums on the subsystem dimensions. Although exclusive, these constituents represent some of the major design variables underlying the most important ISRU unit processes.

- Number of auger dryer screw conveyors
- Auger dryer screw diameter
- Length of each auger dryer
- Heater temperature
- Observed cold trap ice density (kg/m^3)
- Number of electrolysis modules
- Electrolysis Radiator Temperature
- Layers of Multi-Layer Insulation (MLI) for gas storage

The nearly 300 inputs not included in this parametric sweep were contained with their designated values and units in the master list. These included all Global variables describing the Lunar environment and excavated lunar regolith, Passed variables needed for the connections in the process flow environment, and the Local variables characterizing other elements of the design required for the subsystem model to run. More parametric sweeps could be conducted on other variables in the future.

A. Parametric Study Analysis & Results

An example subsystem mass, volume, and power analysis are summarized in Figure 4 below. After conducting the wider parametric sweep, one run was pulled from the larger data set that comprised the lowest ESM at each of the three production targets. The optimal run for the 1 mt/year production target can be seen below. MAIT aggregates mass, volume, and power for easy comparison of critical design parameters and organizes them by subsystem. The plots clearly demonstrate which subsystems are the major contributors to the integrated system design. For instance, the largest source of volume in the 1 mt/year water processing system shown in Figure 4 are the liquefaction & product storage subsystems. The combined storage volume makes up nearly 90% of the overall total volume for this design. On the other hand, total system power demand originates mainly from the electrolysis and H_2 liquefaction subsystems. The screw conveyor (auger dryer) similarly operates in challenging thermal conditions and requires a phase change (sublimation), which translates to higher power requirements than the other subsystems located in the PSR. However, optimization efforts would only start to save approximately 10% of the total system power. The type of results realized using MAIT will better inform NASA personnel and its partners during critical decision points. Detailed qualitative information such as system mass, volume, & power, plainly identifies existing bottlenecks, but also aids in the prediction of optimal design configurations and variables.

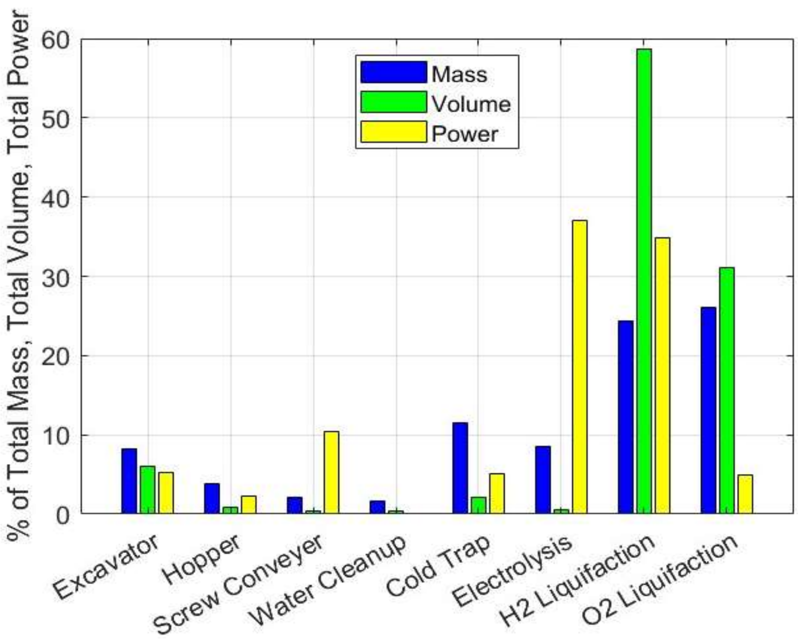

Figure 4: Subsystem Contributions to Total Mass, Volume, Power Estimates. The following characterizes the optimal iteration in the 1 mt/year production parametric sweep.

The mass, power, and volume data were further dissected in MAIT to (1) estimate how much power demand would influence the overall system mass by sizing the required power generation & distribution steps, and (2) calculate the necessary mass growth allowance (MGA) needed for each subsystem component. The individual contributions of each ISRU subsystem in Figure 5 (right of dotted line) and the power & distribution processes (left of dotted line) are calculated as a percentage of the total system mass and tabulated in each column. The basic mass (blue) represents a summation of the components within a subsystem model that was completed using the model inputs, followed through by the cascading flow of data leading from one module to the next. Additionally, the figure includes the appropriate MGA of each component as represented by predicted mass (orange), per ANSI & AIAA standards. The standards' recommended allowances typically range 10-25%, but are up to 40% for some components, which in the water processing model translated to 5-15% of upmass, based on the component type and level of model design maturity.

Examining the results for a 1 mt/year design target (A), it is clear that the excavation and water sublimation processes in the PSR constitute only a small portion of the overall system mass (<8%), and become essentially negligible at larger production targets (B: 10 mt/year and C: 50 mt/year). On the other hand, generating electricity via the VSAT, distributing that electricity to the PSR along a length of power cables, and storing the gaseous product after electrolysis add up to roughly 60%. Efforts to reduce or redesign the latter subsystem infrastructure could result in significant mass savings, and therefore it can be recommended to focus optimization efforts on those subsystems. Considering that the properties of hydrogen preclude it from easily being stored as a liquid, and moreover, the stored product is the least dense of all liquid propellants, liquid oxygen accounts for the majority of propellant mass required on an ascent vehicle (liquid O_2/H_2 mixture ratio between 5:1 and 6:1). One mission architectural strategy would be to eliminate the hydrogen liquefaction and storage subsystems, or alternatively recycle this gas without storage. According to the results, eliminating hydrogen liquefaction could save 20-25% of the overall system mass, as well as its corresponding energy demand.

Because these snapshots indicate the lowest ESM configurations, the design parameters and dimensions can be evaluated. MAIT logic can also account for other limiting conditions, such as dimension constraints during launch or the deployment phase, that may disqualify the numerically lowest ESM as an unfeasible design. As part of the model verification steps, these limitations were documented, and implemented either within the subsystem model internal logic, or as part of a post-parametric data quality filtering step.

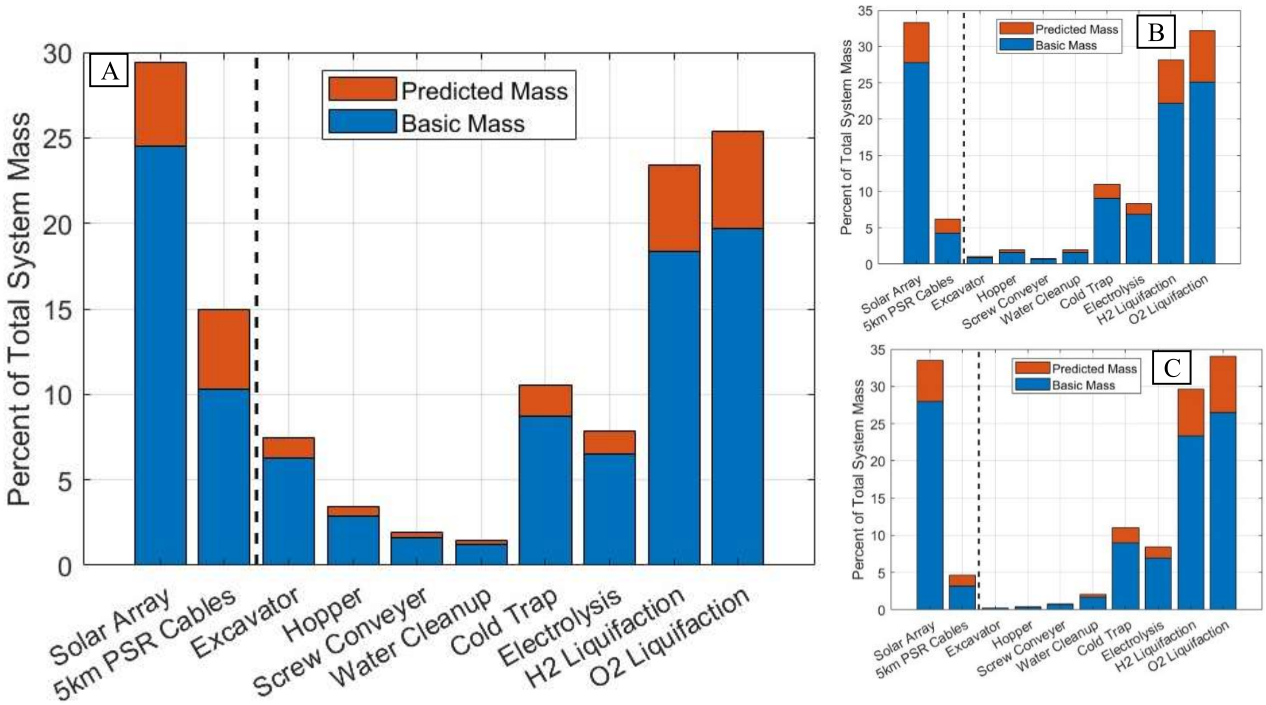

Figure 5: Individual subsystem percent mass makeup at various production targets. Columns represent the subsystem mass as a percent of the total system mass for the optimal ESM iteration for production of 1 mt/year (A), 10 mt/year (B), 50 mt/year (C). The power generation systems (left of dotted line) and the ISRU water processing unit processes (right) are shown. See AIAA standards for definitions of basic and predicted mass including MGA.

B. Individual Variable Contribution to ESM

Beyond simple ESM calculations, MAIT can also be used to investigate which local variables are the most influential and what magnitude they have on system mass, power, or volume post-simulation. During the parametric sweep, a known value range is used for local variables of interest, which can then be compared to the percent change measured in the system ESM for specific iterations. If two or more values are used for a variable in a parametric sweep, an analysis around ΔESM on a 10% deviation from the average of those variables reveals the effect they have on the overall system. For example, Figure 6 shows four variables examined over the parametric sweep. Ice density has been experimentally measured either as a thin frozen layer or dense ice sheet, but must be assumed in different versions of a cold trap model.[12,13] Using the results from parametric sweeps, the ESM analysis projected a nearly 0.9% increase to the system mass per 10% decrease to the assumed ice density, i.e., the properties of a lower frozen layer density were less optimal to the system ESM. Suboptimal water deposition factored into the cold trap installation mass, but also radiator size, and ultimately had downstream impacts to electrolysis and storage. Its effect on the total system was shown to be significant—a more than 8% system mass reduction was achieved by assuming the value of a single variable. This local variable-level analysis exemplifies where model outcomes can more effectively drive funding decisions, as additional research to validate ice density models in a relevant environment will improve the precision of the technology and benefit the overall ISRU system. Other variables, such as the electrolysis module radiator temperature, the number of electrolysis modules, and the number of MLI layers, had specific impacts on system power demands and/or thermal rejection, which is reflected more so in the power generation & distribution contributions (see Figure 4).

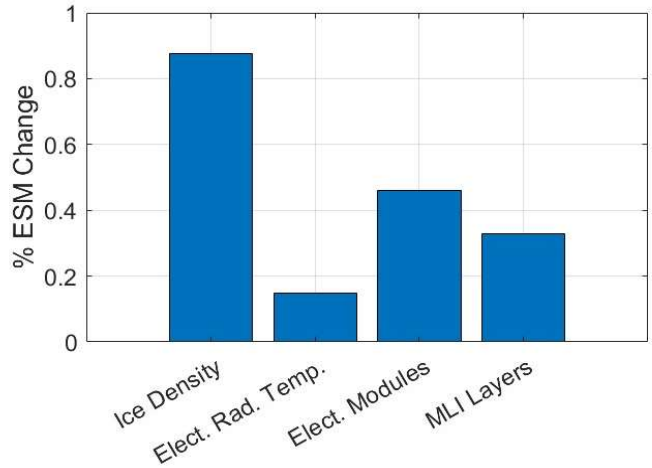

Figure 6: Individual Variable Contribution to ESM Increases. A 10% deviation to an individual variable (column) carries forward a certain % increase to overall system ESM

V. Future Work

A. Improvements to MAIT Capabilities

Despite the significant advancements made to ISRU system modeling capabilities, there are still numerous improvements that can be made to make MAIT a more effective tool. The first improvement is to expand the suite of software packages that MAIT can integrate with. Some software tools that may be particularly relevant for ISRU applications are COMSOL (a multiphysics modeling software), Aspen Plus (a chemical process simulator), and ANSYS Fluent (a Computational Fluid dynamics tool). Expanding the API capabilities will allow MAIT to interface with the most common subsystem models developed by ISRU stakeholders. The second improvement is to add an optimization capability into MAIT's existing logic. Many of the capabilities built into Simulink may allow for the automation of optimization on different systems. Having a way to appropriately size all components in a full ISRU system design based on the ESM can provide meaningful data to the designers of each specific subsystem. Establishing an estimate start point would reduce the parametric space required when investigating the impacts of different variables. Another potential improvement to the MAIT tool is to allow for parallel computing. Currently, MAIT runs in series, where a single parametric case is processed at a time. Adjusting the framework of MAIT to execute in parallel could vastly reduce the computational time.

B. Additional ISRU Architectures

In addition to improving MAIT's core processing ability, numerous ISRU architectures will be investigated, beyond water mining described in the section above. In the short term, there are plans to investigate Molten Regolith Extraction (MRE) technologies[14] and Mars propellent production[7] in FY24. While the systems will involve different subsystem models and architectures, the backbone provided by MAIT will not need significant changes to allow for these investigations. MAIT can also be used to investigate power generation in ISRU systems. This would allow for a more accurate calculation of mission launch mass than using an ESM calculation based on generic scaling factors.

VI. Conclusions

MAIT is a framework produced under the newly formed SIMA project that allows for consolidation of developer-driven subsystem models that will address specific STMD shortfalls involving ISRU production of oxygen on the moon and Mars. The major design elements are summarized below:

- MAIT is an early-stage tool for managing complex & often novel unit processes through a combination of Excel (for variable organization) & MATLAB/Simulink "drag-and-drop" process flow environment.
- A generic modular element was created to standardize inputs and outputs allowing for a more streamlined data/information passage between subsystems. Mass, power, and heat of relevant material flow between subsystems occurs within the framework, as input/output data cascade from one subsystem to the next.
- The model framework allows access to multiple types of software for the convenience of technology developers, as well as parametric sweep and recycling functionality, and communication/organization with downstream requirements management tools.
- Model outcomes are filtered through an ESM calculator that factors AIAA standard mass growth allowances and mass margins into system mass estimates. Additionally, the impacts of power & volume are converted into physical mass via power and distribution infrastructure, or deployment modules. Individual local variables can be assessed for importance in defining system mass & function, and highly influential variables can be adequately addressed through research.
- This is an ongoing project that is soliciting the ISRU community for higher fidelity models. Trade studies with alternative architectures will be produced in the coming year.

Acknowledgments

The FY24 ISRU System Analysis work discussed in this paper is part of JSC's ISRU portfolio under the ISRU TRL Maturation project funded by the Game Changing Development Program (GCDP) within the Space Technology Mission Directorate (STMD).

References

[1] International Space Exploration Coordination Group, "In-Situ Resource Utilization Gap Assessment Report", 2021.

[2] NASA Space Technology Mission Directorate, "LIVE: In Situ Resource Utilization (ISRU)", 2023.

[3] National Aeronautics and Space Administration, "NASA's Moon to Mars Architecture", 2023.

[4] MathWorks, "Model-Based Design for Embedded Control Systems", URL: https://www.mathworks.com/campaigns/offers/model-based-design-embedded-control-systems.html, 2023.

[5] American National Standards Institute (ANSI). "Mass Properties Control for Space Systems", 2015.

[6] Sudholt, W., Baldridge, K. K., Abramson, D., Enticott, C., Garic, S., Kondric. C., & Nguyen, D. "Application of grid computing to parameter sweeps and optimization in molecular modeling." *Future Generation Computer Systems*, 2005, pp., 27-35.

[7] Kleinhenz, J. E., & Paz, A. "Case Studies for Lunar ISRU Systems Utilizing Polar Water." *ASCEND 2020*. AIAA 2020-4042, 2020.

[8] Kleinhenz, J. E., & Paz, A. "An ISRU propellant production system for a fully fueled Mars Ascent Vehicle." *10th Symposium on Space Resource Utilization*. AIAA, 2017.

[9] Colaprete, A., Schultz, P., Heldmann, J., Wooden, D., Shirley, M., Ennico, K., et al. "Detection of water in the LCROSS Ejecta Plume." *Science*, Vol. 330, No. 6003, pp. 463-468, 2010.

[10] Collins, J., Michel, J., Araghi, K. "Lunar water extraction via Lunar Auger Dryer ISRU (LADI)." *ASCEND 2023*. AIAA 2023-4078, 2023.

[11] Paz, A. "Technology Assessment for Producing Propellant from Lunar Water." *NASA Exploration Science Forum*. NTRS - NASA Technical Reports Server [online database], URL: https://ntrs.nasa.gov/citations/20230010039, [cited January 2023].

[12] Compton, B., Diane, L., & Ring, A. "Ice Growth Model for Polar Water Capture Systems." ASCEND 2021. AIAA 2021-4043, 2021.

[13] Connor, J., Holquist, J., Ruble, A., Rivera, R., & Moeller, T. "Demonstration and Model Validation of Freeze Distillation as a Purification Step for Lunar Water Processing." 52nd International Conference on Environmental Systems. ICES-2023-136, 2023.

[14] Schreiner, S. S., Sibille, L., Dominguez, J. A., & Hoffman, J. A. "A parametric sizing model for Molten Regolith Electrolysis reactors to produce oxygen on the Moon." *Advances in Space Research*, Vol. 57, pp., 1585-1603, 2016.