

System Testing of Ground Cooling System Components

Tyler Ensey

Kennedy Space Center

Major: Electrical Engineering

KSC FO Spring Session

Date: 02 04 2014

System Testing of Ground Cooling System Components

Tyler S. Ensey¹

University of Central Florida, Orlando, FL, 32828

This internship focused primarily upon software unit testing of Ground Cooling System (GCS) components, one of the three types of tests (unit, integrated, and COTS/regression) utilized in software verification. Unit tests are used to test the software of necessary components before it is implemented into the hardware. A unit test determines that the control data, usage procedures, and operating procedures of a particular component are tested to determine if the program is fit for use. Three different files are used to make and complete an efficient unit test. These files include the following: Model Test file (.mdl), Simulink SystemTest (.test), and autotest (.m). The Model Test file includes the component that is being tested with the appropriate Discrete Physical Interface (DPI) for testing. The Simulink SystemTest is a program used to test all of the requirements of the component. The autotest tests that the component passes Model Advisor and System Testing, and puts the results into proper files. Once unit testing is completed on the GCS components they can then be implemented into the GCS Schematic and the software of the GCS model as a whole can be tested using integrated testing. Unit testing is a critical part of software verification; it allows for the testing of more basic components before a model of higher fidelity is tested, making the process of testing flow in an orderly manner.

Nomenclature

<i>COTS</i>	= Commercial off the Shelf
<i>DPI</i>	= Default Physical Interface
<i>GCS</i>	= Ground Cooling System
<i>GIS</i>	= Ground Integrated Schematic
<i>GSE</i>	= Ground Support Equipment
<i>GUI</i>	= Graphical User Interface
<i>SCCS</i>	= Spaceport Command and Control System
<i>MTR</i>	= Model Test Report
<i>MRDD</i>	= Model Requirements and Design Document
<i>PRSD</i>	= Power Reactant Supply and Distribution
<i>SMS</i>	= System Mechanical Schematic
<i>UCTS</i>	= Universal Coolant Transporter System

I. Introduction

THE testing of software before it is implemented into the hardware is crucial. Since the software cannot be tested on the hardware immediately, because of possible issues that could arise, such as, in extreme cases, the possibility of breaking the hardware in question or other hardware as a result of the effects the software has on the hardware. Instead, a model of the hardware is made in an environment, so that the effectiveness of the software can be tested in a safe and feasible environment.

Each model is first made independently and tested, before it is incorporated into either another model, or into the larger scale model, known as the Ground Integrated Schematic (GIS). This takes much cooperation and coordination, something I learned a great deal about during my internship. This is because sometimes a different modeler will create the system test for a model that was developed by a separate modeler. Therefore, the modeler who created the model needs to put in adequate requirements, to make it easier for the system test to be made. Also, the modeler designing the system test needs to be sure to coordinate with the original modeler, to be sure that any changes made to the original model will not change the desired outcome of the component. Another consideration that needs to be taken into account is the effect on the schematic. The modeler needs to be certain of the

¹ KSC FO Intern, NE-C1, Kennedy Space Center, University of Central Florida.

requirements of their model and what should and should not be included as the inputs, outputs, subsystems, parameters, etc.

II. Description

On my first day at NASA Kennedy Space Center, I was informed that I would be working towards testing fluid and electrical components for GCS. I worked with components that ranged from more simple components like a dead head to components that required more thought and time like a diode relay module. It was very important to test the requirements of the component and not what the modeler developed the component to do. This was important because the modeler could be informed of differences between requirements and model.

A. Simulink

Every component is modeled in the Simulink environment. Simulink is a Commercial off the Shelf (COTS) simulation tool, created by the company MathWorks. Simulink is used for multi-domain simulation and developing model based designs. Simulink provides the user with an environment that is both one that provides the user with both a visual and interactive experience. Customizable block libraries are another feature that Simulink graces to the modeler. These libraries are used to design, simulate, and test multiple time-varying systems within the Simulink environment. These systems include communications, controls, signal processing, video processing, and image processing. Many different industries utilize Simulink for the development of mathematical models of physical dynamic systems. In specific, my coworkers and I, used/are using Simulink and multiple libraries, libraries provided within Simulink by MathWorks and also a set of KSC developed custom libraries, to construct models of Ground Support Equipment (GSE).^{1,2}

There are many ways of doing mathematical calculations. In this case, there are five recognizable ways designated by the Simulation Product Group: Simulink blocks, embedded MATLAB functions, stateflow diagrams, Simulink look-up tables, and truth tables. There are certain requirements for each of these. Simulink blocks should contain equations that are broken up into logical, contained units, and the equation should be presented in a readable way. This is often achieved using the Simulink “Subsystem” block. Simulink “Integrator” blocks are typically used within the developing community to represent physical quantities, such as pressure, temperature, etc. Also, the parameters “Upper Saturation Limit” and “Lower Saturation Limit,” of the “Integrator” block, should be populated with acceptable limits of the physical quantity. For example, most negative quantities would not be expected in a real world situation, so most “Integrator” blocks would have a “Lower Saturation Limit” of “0.” Lookup tables can be helpful, but the developer should not import data into lookup tables using Excel files. Instead, the file should be converted into a .csv file and then “csvread” should be used in the callback section of the model to import data. Truth tables are useful as decoders and controllers, especially when used in multiple places.^{1,3}

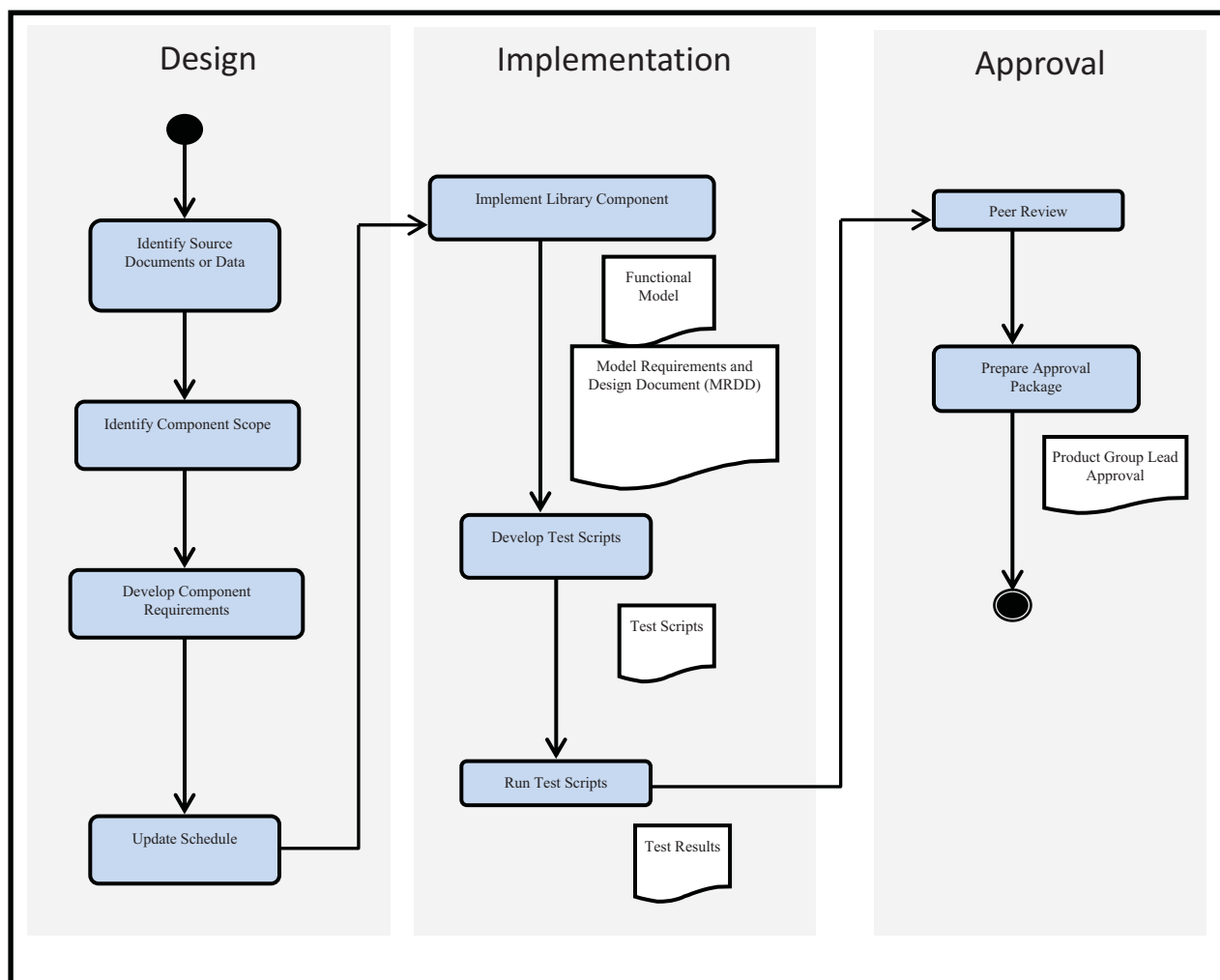


Figure 1. GSE Library Component Lifecycle. This figure represents all of the planning and execution during the lifecycle of a GSE Library Component.

B. GSE Library Component Lifecycle

Model Development for a GSE model is broken into three phases: Design, Implementation, and Approval. A visual representation of this process is shown in Fig. 1.

The Design phase is focused on establishing structure and external interfaces. These models represent components of the GSE Subsystem and they, without exception, must coincide with the Simulation Computer Software Configuration Item (CSCI) Modeling Standard. Design is begun once the necessity for a GSE component is recognized, and a schedule for the development has been created. Source documents must first be identified in the Design phase. These documents may include: GIS/System Mechanical Schematic (SMS), which identifies the manufacturer and model of a component and its functional parameters; Design Dictionary, which includes cutsheets of the specific component type and function parameters; and/or, ConOps, which describes the function of the component. Component scope is then identified, which includes layout and unique feature identification. For layout, the user must identify internal structures that will help in easily showing the complexity of certain interactions. Some great things to consider are grouping blocks, within Subsystems, based on their function, and also the reusability of the component, as simplicity is the key if it will be used in multiple other components. For unique feature identification, the user must identify what internal functions are unique, meaning the component cannot be developed by using readily available library blocks or components. Requirements must be developed within the Design phase. There are three types of requirements that should be concluded before moving to next phase: Test Scenario Requirements, which establishes points of verification, to ensure that the component being tested functions

as desired; Training Scenario Requirements, which establishes fail-safe requirements; and, finally, Functional Requirements, which are the requirements that the component must meet in order to be of use to the GSE Subsystem Model. The schedule is then updated, as needed, and the modeler moves on to the next stage, known as Implementation.

The Implementation stage, the stage that I worked primarily with during my internship, is focused on developing the component and the test scripts. Common-use components, subsystem-specific components, and Simulink Primitive blocks should be arranged into Simulink approximations. Test Scripts must then be developed. Draft steps are created to verify the software is behaving accurately. Draft steps must also be produced to test the interface function, given that there are explicit interfaces with another block. The Test Script is then run, and the test data is recorded in a Model Test Report (MTR). A Model Advisor check will verify that Simulation Standards are met. Finally, a Model Requirements and Design Document (MRDD) will show the requirements, structure, parameters, and referenced libraries within a single document, and the user may now move on to the Approval phase.

The Approval phase is focused on reviewing the GSE Component model and approving it for use within the schematic. In the Approval phase, peer review will commence. The MRDD, MTR, and Test reports are distributed to the Simulation team, through a Graphical User Interface (GUI), known as AccuRev, so that peer review can begin. Comments are gathered pertaining to the validity of the requirements, implementing logic, and test results. Once a response from two SIM Team members and a Subject Matter Expert, appropriate changes are made to the component, and the MRDD, MTR, and Model Advisor Report are regenerated. The final step is to prepare and submit the Approval Package. This package is objective evidence of the successful completion of the Library Component Peer Review. Once the SIM Product Group Lead approves the model, the development of that component is officially complete, and can now be used in the GSE Subsystem model.⁴

C. Universal Coolant Transporter System (UCTS)

Many of the components that I supported in testing were a part of the Universal Coolant Transporter System

(UCTS) (part of GCS), which is a Space Shuttle Orbiter support system. The UCTS is used for two main services, coolant servicing and Power Reactant Supply and Distribution (PRSD) servicing. Coolant servicing is used once the Orbiter lands; the Ammonia Boiler, which cools the avionics and payloads on board the orbiter, is shut off upon landing, so the Orbiter is connected to the UCTS in order to keep the avionics and payloads from getting damaged by overheating. PRSD servicing allows for the deservicing of propellants.

¹ The operating system of the UCTS is shown in Fig. 2. The four main parts of the system, the primary loop, the secondary loop, the refrigeration subsystem, and the pump/reservoir operation are described below:

PRIMARY LOOP

R-124 in the primary loop can exit the Orbiter at 30 to

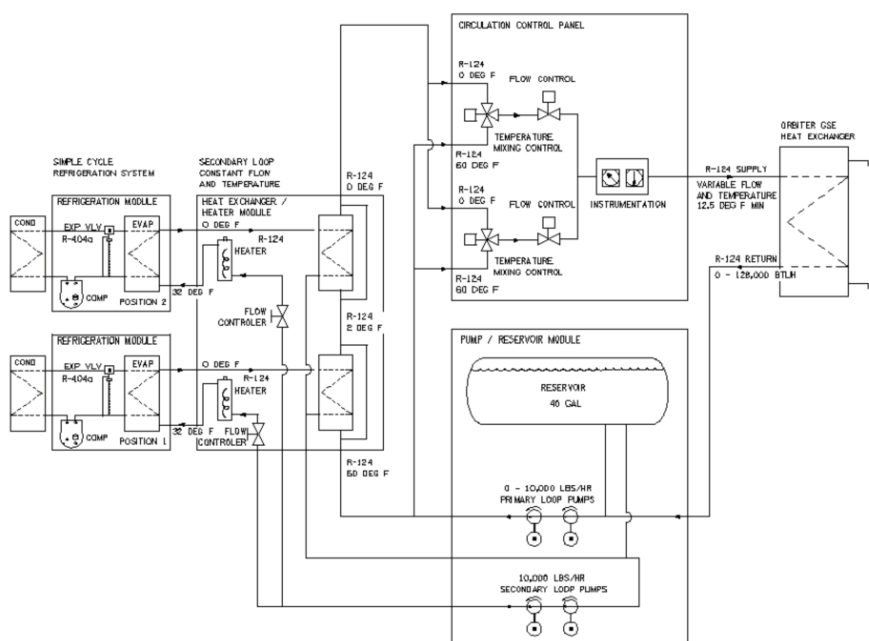


Figure 2. The Universal Coolant Transporter System. This is the model for the operating system of the UCTS.

100 degrees F with a maximum heat load of 128,000 Btu/h. Experience in operations has shown that the Orbiter return temperature is maintained at 60 °F. The R-124 return fluid first encounters the systems' primary circulation pump, located on the reservoir / pump subsystem. After which the flow is split into two paths. A portion of the primary loop fluid is routed to the temperature controlled mixing valves, located on the circulation control subsystem. This path is the source of the warm fluid that is mixed with the cold fluid which is conditioned by the secondary loop. The remaining portion of the primary loop is routed to the secondary loop heat exchangers where it is conditioned to 10 °F. From the heat exchangers the fluid proceeds direct to the temperature controlled mixing valves. At the temperature controlled mixing valves all or none of the fluid can be directed to flow through the secondary loop heat exchangers.

SECONDARY LOOP

The secondary loop is simple. The PLC monitors only one variable (temperature) and adjusts it through a single device, the heater, into a constant. Flow in the secondary loop is preset and fixed. Empirical data taken at OPF-3 has proven the criticality of flow rate in order to make the system work. The secondary loop eliminates all variables ensuring the greatest reliability and ease in troubleshooting. (...) As a result of the temperature controlled mixing valves, all or part of the primary loop flow can be directed to the secondary loop heat exchangers. The portion of primary loop return fluid directed to the secondary heat exchangers, flow varies, causing changes in the heat load and secondary fluid temperature; however, the thermostatically controlled heaters in the secondary loop maintain the fluid going to the refrigeration subsystems at a constant design temperature.

REFRIGERATION SUBSYSTEM

The UCTS has two identical refrigeration subsystems, piped in series into the primary loop, via the secondary loops. Each refrigeration subsystem is sized to handle 87,000 Btu/h at 110 degrees F desert conditions. Experience has shown that the maximum Orbiter heat load generated during post landing operations is 60,000 Btu/h, and averages 40,000 Btu/h. The refrigeration subsystem is basic and consists of a positive displacement disc compressor; refrigerant receiver, air-cooled condenser; suction heat exchanger / accumulator; an expansion valve and chiller barrel.

PUMP / RESERVOIR OPERATION

The primary pumps are configured in series. One pump is required for normal operation, and the second is provided for back up. Series configuration was chosen to meet any future pressure increase requirement. The pump operation is locked out by software preventing undesired operation of two pumps simultaneously. The secondary pumps are also configured in series to accommodate space allocation. (...) The reservoir maintains both the primary and secondary loops. This decision was made to accommodate space allocations. The reservoir was piped as an accumulator (out of the flow path) vs. in line to allow for system temperature stabilization of the secondary loops.⁵

III. Component Testing

A. The Test

The requirements of the component are studied in depth until the system test creator has adequate information about the components functionality. A MATLAB script is written within the test harness, in order to test if the desired outputs are indeed being calculated. The test is then run, and the expected output values are compared to the actual output values, and a pass or fail prompt is returned to the user. A portion of a written test script is shown in Fig. 5. Once the test script passes on every iteration (the number of iterations is based on the number of user designated expressions for the length of each input, and the program determines the calculated number of possible iterations), an autotest is run. The autotest will run Model Advisor and the system test, and then produce the MTR, MRDD, System Test, and Model Advisor documents. These documents (only small portions, per their large sizes) are represented in the Appendix, Fig. 6. After these documents have been produced, the library component, the test model, the test script, and the documents are uploaded to AccuRev, and when ready, will be peer reviewed for acceptance.

For $t = 1:\text{numSteps}$

```

Ve10exp(i) = ((r1)^2 * h1 * Ve1In) / ((r2)^2 * h2);
P0exp(i) = (ro * 32.2 / 144) * ((z1 - z2) + (144 * PIn / (ro * 32.2)) + (((Ve1In)^2 - (Ve10exp(i))^2) / (2 * 32.2)) - hL);

```

Figure 5. MATLAB Test Script. This is a representation of a small portion of the Test Script that is used to test the functionality of the Discrete Pressure Switch.

Appendix

a)

b)

c)

d)

Report name: Model Advisor
 Simulink version: 7.4 Model version: 1.3990
 System: F_Pressure_Switch_Test Current run: 16-Jul-2013 16:25:42
 67 of 67 Passed

Check model, local libraries, and referenced models for known upgrade issues
 Passed

a) The Model Advisor Document

Limit Check

Test Variable	Operator	Limit	Evaluates To
OT	==	OTexp 1	TRUE
VelQ	==	VelQexp 400	TRUE
TO	==	TOexp 1	TRUE
KO	==	KOexp 1	TRUE
QO	==	QOexp 1	TRUE

Test Variable	Expected Value	Tolerance Type	Tolerance Limit	Evaluates To
PO	POexp 549.96	Absolute	POexp_tol 0.54996	TRUE

Iteration 18 Passed

Post Test

F_Pressure_Switch_Test Done

Generated Files

The following files were generated in \workspace\may\accurev\SM_MODEL_BasicModel_Development\lib\subtest\F_Pressure_Switch_Test\

Open	Filename
Test Results	F_Pressure_Switch_Test_results.mat
Test Report	F_Pressure_Switch_Test_report\F_Pressure_Switch_Test_report.html

Final Test Status

Property	Value
Start Time	16-Jul-2013 16:26:01
Stop Time	16-Jul-2013 16:26:26
Iterations Completed	18
Iterations Passed	18
Iterations Failed	0

Final Status

c) The Test Script Document

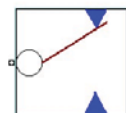
Library Component: F_Pressure_Switch

Model Requirements and Design Document (MRDD)

Date: 27-Jun-2013

Chapter 1. Root System

Diagram



F_Pressure_Switch_3210

Mask Parameters

Mask Parameter	Initial Value
Callout	3210
Max Pressure	500
Input Elevation	5
Output Elevation	5
Density	62.4
Length	10
Input Radius	4
Output Radius	4
Mu	2.25e-4
Input Length	2
Output Length	2

Requirements

<F_Pressure_Switch_R_001>

The F_Pressure_Switch_3210 block shall implement the equations for VelOut, POut, TOut, KOut, and QOut.

Parameters:
 $r1$ = input radius; $h1$ = input length; $r2$ = output radius; $h2$ = output length
 ρ = density; $e1$ = input elevation; $e2$ = output elevation

These equations are as follows:
 $VelOut = ((r1^2 * h1 * VelIn) / (r2^2 * h2))$
 $e2 = density; e1 = input elevation; e2 = output elevation; L = Length; mu = Viscosity$
 $POut = ((\rho * 32.2 / 144) * ((e1 - e2) * (144 * Pin / (\rho * 32.2)) + ((VelIn^2 - VelOut^2) / (2 * 32.2)) - h1))$
 $h2 = Head Loss = ((f * (L / 12)) * (Dia / 12)) * VelIn^2 / (2 * 32.2)$
 $f = (64 * mu) / (Dip * Dia)$
 $Dip = (Dia / 12) * VelIn * \rho$
 $Dia = 2 * r1$

TOut = TIn

KOut = KIn

QOut = QIn

If $Pin > P_max$ (cutoff pressure), then the block will generate a boolean integer output of 1.

Otherwise, the block will generate a boolean integer value of 0.

b) The MRDD Document

Library Component: F_Pressure_Switch

Math Model Test Report (MTR)

Date: 16-Jul-2013

Chapter 1. Report Location

Accurev Location : ..\lib\fluid\test\F_Pressure_Switch\Test\F_Pressure_Switch_MTR.html

Chapter 2. General Software Environment and Assumptions

1.1 Test Environment

Testing will be performed on an individual basis at dedicated workstations, such as an office environment.

Whenever possible, testing will be performed with Mathworks Simulink/Matlab suite of programming and simulation tools/environment.

It is recognized that testing may be required that exceeds the scope of Simulink test, such as TRICK interface. In these cases, the appropriate testing tools are permitted.

1.2 Assumptions

All members of the test team are familiar with testing tools, such as the Mathworks Simulink/Matlab suite of software tools or the TRICK environment.

All members of the test team have compatible versions of the testing tools.

All members of the test team have access software same repository, such as Accurev, KDDMS, etc.

All pathname locations in this report are relative to the Accurev path :

..\lib\fluid\test\F_Pressure_Switch\Test

1.3 Scope Purpose

This test report provides a summary of the tests executed against a particular component for the purpose of verification.

The Test Summary section identifies the tests that have been run, the requirements satisfied with each test, any required input files for the test, the location of the detailed test report in accurev, test execution information and a Final Status for the Pass/Fail criteria for the test.

Chapter 3. Test Summary

Location: slprj\modeladvisor\F_Pressure_Switch_Test\report.html
 Final Status: passed

Acknowledgments

I would like to thank a few people who were crucial to my success during my internship at KSC. First, I owe Cheryle Mako my greatest gratitude for giving me this opportunity, and giving me a chance to succeed, while being able to see what my future has in store. Next, I would like to thank Lien Moore for all her hard work and diligence

towards everything she does. She undoubtedly gave me the feeling that she genuinely was concerned with my well-being, something that I cherish deeply. Finally, I would like to thank Camiren Stewart; Cam was perhaps one of, if not the, most helpful member of NE-C1. I felt that he could relate to me in a way that no one else could, because he was fairly recently an intern at KSC, but at the same time he is intelligent far beyond his years.

References

- ¹MATLAB, Ver. 7.9.0 (R2009b), The MathWorks Inc., Natick, MA, 2009.
- ²Grant, K., “Simulation CSCI Software Design Description,” Document Number: K0000111998-GEN, Archive URL: <https://sp.ksc.nasa.gov/ne/simulation/Shared Documents>, NASA Kennedy Space Center, Merritt Island, FL, 2013.
- ³“Simulation Computer Software Configuration Item (CSCI) Modeling Standard,” Document Number: K00000141065-GEN, Archive URL: <https://sp.ksc.nasa.gov/ne/simulation/Shared Documents>, NASA Kennedy Space Center, Merritt Island, FL, 2013.
- ⁴Moore, L., “E2E CC SDP Volume 4,” Document Number: K0000143174-PLN, Archive URL: <https://sp.ksc.nasa.gov/ne/simulation/Shared Documents>, NASA Kennedy Space Center, Merritt Island, FL, 2013.
- ⁵Katz, M., “Universal Coolant Transporter System Technical Manual & Operating Criteria,” URL: https://usa2.usa-spaceops.com/fl/ucts/a80k59593_index.htm, NASA Kennedy Space Center, Merritt Island, FL, 2013.