Parametric Analysis and Targeting Capabilities for the Planetary Entry Systems Synthesis Tool

Space Systems Design Lab
Georgia Tech Aerospace Eng.

AE8900 MS Special Problems Report
Space Systems Design Lab (SSDL)
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA

Author:
Patrick J. Smith

Advisor:
Dr. Robert D. Braun

December 5, 2008

Parametric Analysis and Targeting Capabilities for the Planetary Entry Systems and Synthesis Tool

Patrick J. Smith
Georgia Institute of Technology, Atlanta, GA 30332

Modeling and simulation has led to major advances in the design of complex systems largely because it provides designers with an affordable method of testing new ideas. This report describes recent improvements to a modeling and simulation tool, known as the Planetary Entry Systems and Synthesis Tool or PESST, that allow a designer to quickly conduct parametric and targeting studies. PESST has been used in several conceptual design studies and the improvements to this tool allow a user to complete several cases quickly and gain valuable insight to a larger region of the design space. It would be impossible for designers to create truly robust systems without the ability to fully grasp the design space. By testing the effect of many different input variable values, the designer gains valuable insight to overall system response. As an example of the improvements added to PESST, hypothetical parametric and targeting studies have been completed for the Orion Crew Entry Vehicle.

Acronyms and Nomenclature

CEV	=	Crew Exploration Vehicle	GUI	=	Graphical User Interface
DSM	=	Design Structure Matrix	JPL	=	Jet Propulsion Laboratory
EDL	=	Entry, Descent and Landing	NASA	=	National Aeronautics and Space Administration
ESAS	=	Exploration Systems Architecture Study	PESST	=	Planetary Entry Systems and Synthesis Tool
g	=	acceleration due to Earth's gravity	SSDL	=	Space Systems Design Laboratory
GNC	=	Guidance, Navigation, and Control	TPS	=	Thermal Protection System

I. Introduction

Vehicle design for interplanetary atmospheric entry, descent and landing (EDL) systems involves many input variables, which may be adjusted by a designer to positively (or negatively) impact the design. For EDL systems, a modeling and simulation tool, known as the Planetary Entry Systems and Synthesis Tool or PESST, has been in development for several years under the direction of faculty and staff at the Georgia Institute of Technology Space Systems Design Lab (SSDL). The goal of PESST is to provide a toolset that increases the user's knowledge and understanding of the design space. PESST focuses on estimating the weight and various performance aspects of an entry system and is currently being used to complete conceptual design studies at the NASA Jet Propulsion Laboratory (JPL).

While PESST provided a very capable framework for estimating the mass and performance of an entry system, it was difficult to perform complex trade studies. PESST has several input variables that an analyst may want to vary, not only to gain insight to the current design capabilities, but also to optimize a design. For example, as the entry flight path angle, velocity or mass are modified, the trajectory attributes, including, the maximum heating rate and maximum deceleration are effected. These trajectory attributes are often key drivers in the design of a new entry vehicle. As such, being able to quickly examine the effects various inputs have on these attributes provides analysts much greater knowledge of a given design's capabilities.

The new features added to PESST's framework that will aid users in completing these trade studies include an input variable parameterization capability and a single-input, single-output targeting capability. Both the implementation of these features and instruction on their use are discussed in this report. Lastly, as an example of the value added by these improvements, a conceptual trade study review of the Orion manned capsule on a return trajectory for the Moon has been completed and its results are provided.

II. Summary of Previously Available PESST Capabilities

PESST is an integrated, multidisciplinary analysis framework that examines entry vehicle mission architecture, configuration, and Guidance, Navigation and Control (GNC).[1] PESST can be run from the command line in a Windows or Linux system as well as from a Graphical User Interface (GUI). The following section provides a brief summary of the tool's overall capabilities. For additional information on the PESST framework, please refer to the PESST User's Guide provided in reference 3.

A. PESST Overview

PESST is a conceptual design tool for EDL system analysis. It provides a rapid conceptual design environment that also includes analysis of entry vehicle geometry, hypersonic aerodynamics, aerothermodynamics, flight mechanics, GNC, mass sizing and vehicle synthesis.[2]

PESST has the capability of accepting user-defined entry vehicle geometries to calculate the hypersonic aerodynamics, flight mechanics, thermal response, and mass estimate for that given system. To assist the user in setting up this calculation, the atmospheric properties for the Earth, Mars, and Venus are pre-loaded and available for use along with the ability to input a user-defined or GRAM atmospheric model. PESST can also be used to study a system's ability to meet varied landed precision requirements.[2]

The framework of the tool is broken up into a few core disciplinary analyses or modules. As such, this framework can be described pictorially via a Design Structure Matrix (DSM), shown below in Figure 1. First-order engineering models are used for each disciplinary analysis resulting in fast computation times, and data is shared between the modules to reach overall system convergence.[2]

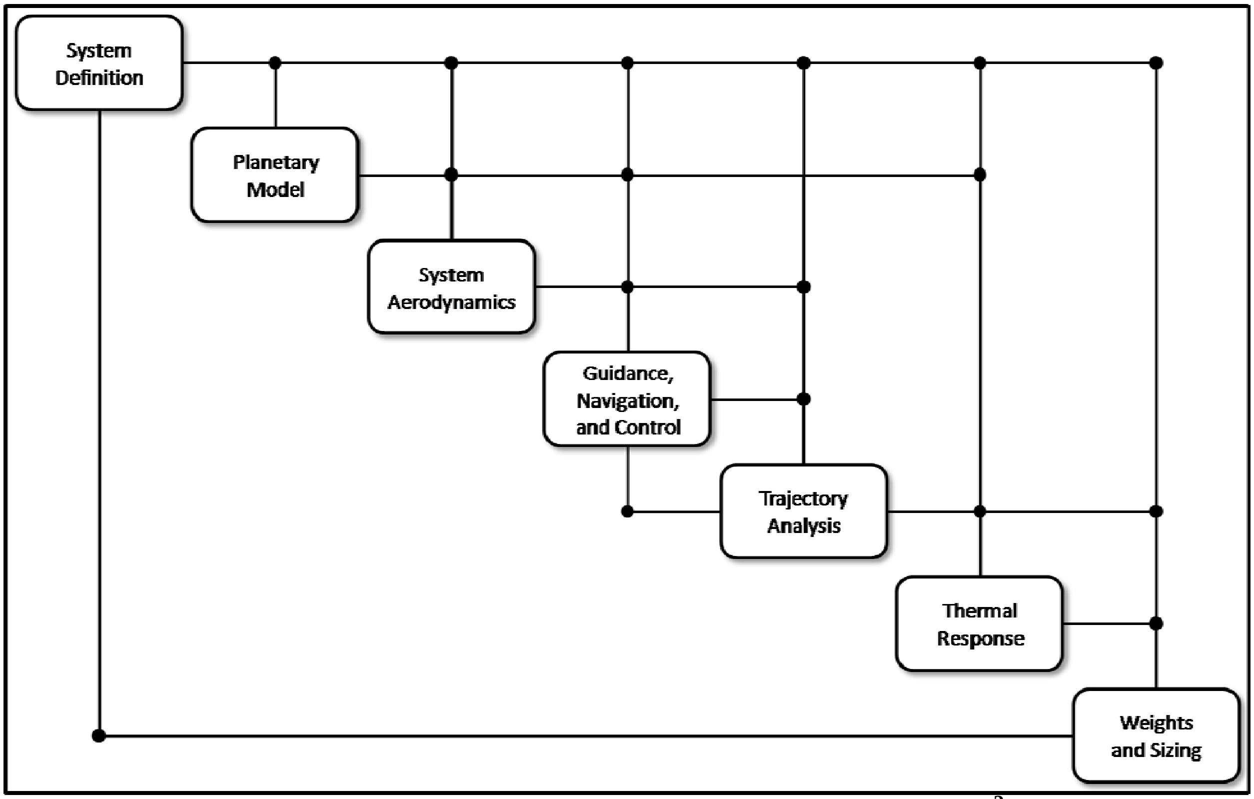

Figure 1: PESST Framework Design Structure Matrix[2]

B. System Definition

The purpose of the system definition module is to assign overall system parameters and define other options. The inputs for this module include the user's selection of the vehicle geometry, the planetary body, the initial or target trajectory conditions, thermal protection system (TPS) materials, trajectory events, and the initial mass estimates.[2]

C. Planetary Model

The planetary model module sets the parameters that define the planetary body to be used for the current study. These parameters include atmospheric properties such as the density, pressure, and temperature at various altitudes, as well as planetary rotation rate and Sutton-Graves constant.[2]

D. System Aerodynamics

The system aerodynamics module provides the approximate hypersonic aerodynamic information for the given entry vehicle's geometry. A mesh of the vehicle's geometry is provided as input along with the user's selection of either Newtonian or modified Newtonian aerodynamics. The coefficients of lift and drag are also provided for the given vehicle shape over a range of angles of attack.[2]

E. Guidance, Navigation, and Control

The guidance, navigation, and control module provides logic to control the flight path of the entry vehicle. This analysis provides two different propulsive terminal descent algorithms; gravity turn and an analytical method. The required guidance control vector is calculated with the current and desired state vectors (i.e., position, velocity, mass, and heating information), available control options, and active guidance mode provided as inputs.[2]

F. Trajectory Analysis

The trajectory analysis module accepts as input key vehicle geometric parameters, the initial state vector, and any possible trajectory events. The module then numerically integrates the inertial equations of motion to obtain the flight path for the upcoming time step and provides the updated state vector along with other standard atmospheric trajectory data as output.[2]

G. Thermal Response

The thermal response module sizes the entry vehicle's thermal protection system (TPS) based on the current state vector information as it relates to the convective and radiated heating environment, the desired TPS material, and the bondline temperature requirement. The TPS sizing information is provided as output from the module.[2]

H. Weights and Sizing

The weights and sizing module iteratively computes an estimate of the subsystem masses given the TPS sizing information and parachute deployment condition. The computed subsystem estimates include the structural, propulsion, parachute, and thermal system masses. These estimates are then fed back into the system definition module for recalculation and further iteration.[2]

III. Planetary Entry Systems and Synthesis Tool Enhancements

Improvements to the PESST framework provide two significant capabilities: 1) the parameterization of inputs allowing for multiple cases to be run automatically and then compiled into an easily analyzed dataset, and 2) a single-input, single-output "targeting" capability.

A. Parameterization Capability

The first of the two main capabilities added to the PESST framework is the ability for a user to set up a parametric trade study that will step sequentially over each of the core inputs to the tool, eventually completing a full-factorial analysis of the parameterized inputs. Among other variables, some of the inputs that could be parameterized include the entry vehicle mass, the landed mass, the entry velocity, and the entry flight path angle. For the full list of input variables that can be parameterized, see Appendix A: Orion CEV Post-Skip PESST Parametric Study Input File. Since PESST can be run from either a command line interface or the PESST GUI, there is more than one way a user might set up a parametric study. Each method for setting up a parametric study will be briefly discussed below. For a more complete description of how to set up a parametric study, please refer to the PESST User's Guide.

1. Parametric Study Implementation

a. Command Line Interface

In order for the user to set up a parametric study from the command line, the user must modify the main PESST input file that is read in during the system definition portion of analysis. For example, if a user wanted to set up the file for multiple points to be evaluated for the entry velocity, the user would modify the line from the input file specifying the value for the entry velocity from

“`entryVel                = r, [1, 7479.0, 7479.0]`”

to

“`entryVel                = r, [`***`n`***`, `***`lower_bound`***`, `***`upper_bound`***`]`”

where ***`n`*** represents the number of points desired and ***`lower_bound`*** and ***`upper_bound`*** represent the lower and upper bounds of the range to be examined, respectively. The increment of each step from the lower to the upper bound will be equally spaced, and if the user specifies more than one input variable with multiple points desired to be analyzed, a full-factorial analysis will be completed. In each case specified above, the “`r`” specifies that the input data type is of type “real.” Other data types can be parameterized as well but require that the user explicitly specify each step to be taken. For more information on the parameterization of variables of other data types, please refer to the PESST User’s Guide.

b. Graphical User Interface

Thanks almost entirely to the assistance of Richard Otero, SSDL Ph.D. Candidate, parametric trade studies can also be set up from the PESST GUI. As is normally the case with a GUI, this task requires less knowledge as to the inner workings of the PESST tool and provides the user a very intuitive window for applying values to parameterized variables. Figure 2 shows how a user can set up a parametric study from the PESST GUI. In the screenshot, a purple box highlights common fields that are used for both Parametric and Targeting studies, and a red box highlights the parametric-only portion of the GUI tab and provides various tools for the parameterization of input variables.

It should be noted that no GUI capability currently exists to parameterize discrete input variables. That is, only real-valued variables may be parameterized from the PESST GUI. Again, more information on how to set up a parametric study from the PESST GUI can be found in the PESST User’s Guide.

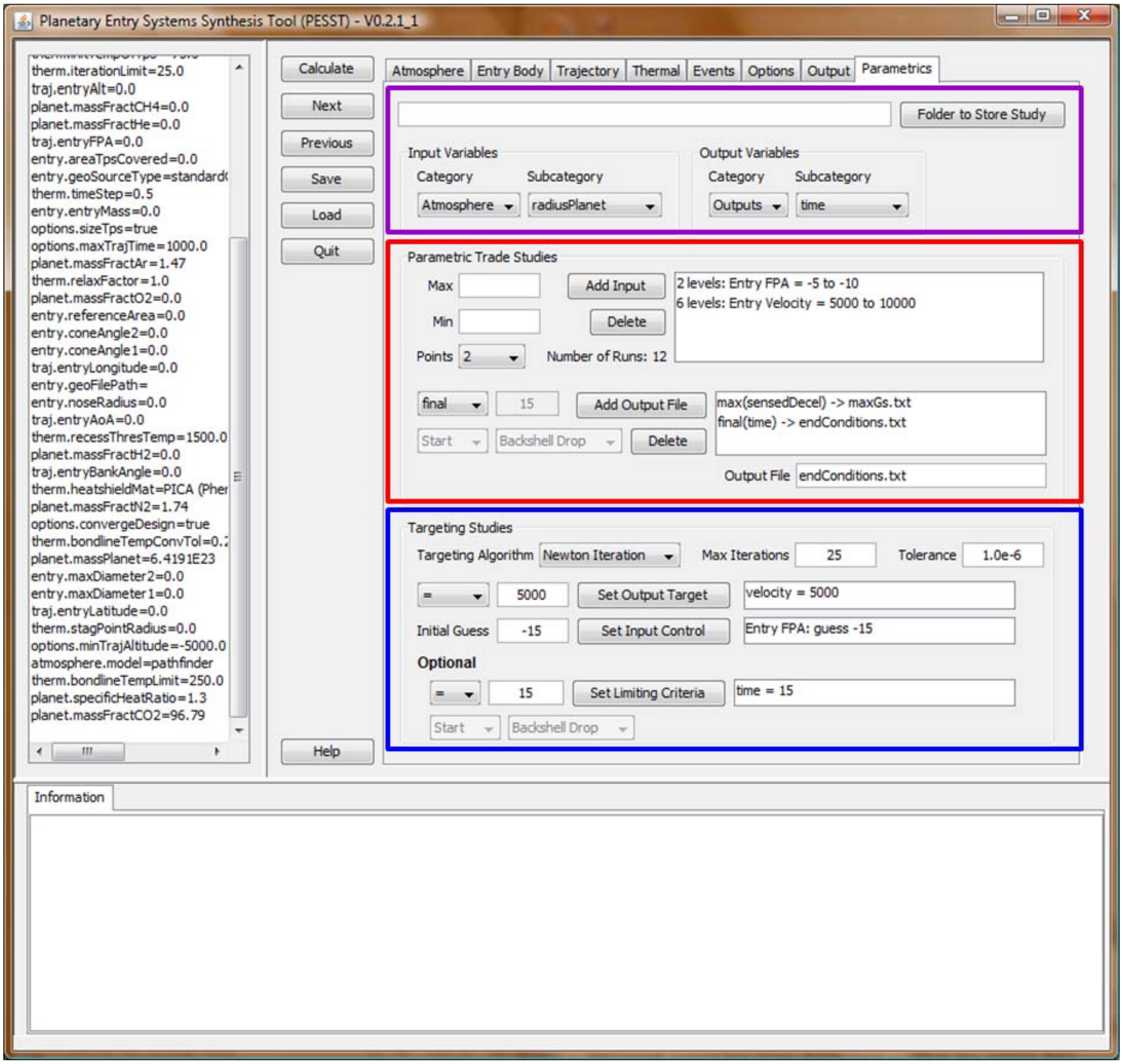

Figure 2: "Parametrics" Tab of PESST GUI

2. Post-Run Analysis and Querying of Output Data

Once a parametric study is completed, the resulting output can be easily plotted and analyzed with a newly developed tool, commonly referred to as "the consolidator." This tool allows the user to query the output data to retrieve a filtered set that includes only the initial, final, maximum, minimum, or average value for a given input or output variable. The tool also allows a user to filter the data based on specific outputs and events.

For example, if a user set up and ran a parametric trade study which completed several hundred cases or more, but was only interested in viewing output data when the vehicle sensed deceleration was greater than or equal to some critical value, the consolidator would supply all of this information in a single file for review. For a detailed discussion of how one might retrieve this data, please refer to the PESST User's Guide.

B. Targeting Capability

In addition to the parametric study capability already discussed, a second capability has been added to the PESST framework that allows a user to "target" an output variable by specifying an input variable that can be changed to meet that target. When evaluating a new design or performing a trade study to evaluate various mission-level concepts, an analyst may be interested in "hitting" certain simulated targets. There may be a system- or component-level requirement that the peak g-load not exceed some upper limit or that the peak heating rate not exceed some maximum value. This section discusses the methodology used to meet these types of goals within the PESST framework and how a user would go about adding these targets.

The targeting capability is achieved by applying a Newton-Raphson solver that takes an initial guess for an input variable and a convergence tolerance, and iterates until a solution is found or a maximum number of iterations has been reached. A key note to be made is that any real output variable can be targeted by changing any real input variable. No current functionality exists with the PESST framework to target a non real-valued output variable; or, furthermore, to target a real-valued output variable by changing a non real-valued input variable.

As with creating a parametric study, a targeting study can be set up via the command line interface or from the PESST GUI. Each method for setting up a targeting study is briefly discussed below.

1. Targeting Study Implementation

a. Command Line Interface

In order to set up and run a targeting study from the command line for PESST, a user must modify the main PESST input file to be read in during the system definition portion of the tool's computation. There are 8 input variables that must be specified within the main PESST input file to set up a targeting study. These input variables are listed below.

```
targetingStudy      = l, [1, true]
targetingMethod     = s, [1, Newton-Raphson]
targetingMaxIters   = i, [1, 25]
targetingTolerance  = r, [1, 1e-6, 1e-6]
controlVariable     = s, [1, entryFPA]
targetingCriteria   = s, [1, mass_heatshield=71.0]
limitingCriteria    = s, [1, null]
limitingInstance    = i, [1, 0]
```

The first variable, "`targetingStudy`" is meant to be a true/false indication of whether or not a targeting study will be completed during this PESST run. The "`targetingMethod`" variable refers to the type of methodology to be used for the targeting study. Currently, the Newton-Raphson algorithm is the only targeting methodology. The "`targetingMaxIters`" and "`targetingTolerance`" variables allow the user to specify the maximum number of iterations allowed for the targeting study and the tolerance required for convergence, respectively. The "`controlVariable`" refers to the real-valued input variable to be "controlled" or adjusted during the targeting study to meet the targeting criteria. The "`targetingCriteria`" variable specifies the name of the output variable to be targeted along with the value that it should meet. The name of the variable can be any of those listed in the trajectory or mass output files. The "`limitingCriteria`" and "`limitingInstance`" variables may be optional, but, when required, will be used in parallel to determine at what point during the trajectory the targeted output variable's value should be evaluated. These input variables are all considered somewhat self-explanatory except for the limiting criteria and limiting instance variables. As such, further discussion of the "limiting" variables has been provided.

One example of a targeting study might stipulate that the sensed deceleration be set to some desired value at the point along the trajectory when the sensed deceleration is at a maximum. However, it could occur that the maximum sensed deceleration occurs at more than one time step along the trajectory's path. In that case, the user might also want to specify a value for the "`limitingInstance`" variable. The "`limitingInstance`" variable would then serve as the specification of the valid instance when the limiting criteria is met that should be returned from the consolidate function. For this case, to achieve the trajectory data point when the maximum sensed deceleration is achieved, the user should specify "`max(sensedDecel)`" for the limiting criteria variable and, if the user only want to return the first instance of the maximum sensed deceleration, he should specify "`1`" for the limiting instance variable.

In addition to just the maximum of a variable, the "limiting criteria" could also be a number of other various criteria applied to a variable from the trajectory output file. For example, the user could specify similar criteria as above to obtain the initial, final, minimum, or average trajectory time step for a given variable. Also, the user could specify a value or range for a given output variable. The description provided here is not meant to give full detail as to the numerous ways a user might use the tool, but only to show some quick examples that might be useful. For a more detailed description of the how to make full use of the tool's capabilities, please refer to the PESST User's Guide.

b. Graphical User Interface

As stated earlier, the PESST GUI provides the user a simplified interface for setting up a PESST run. This also holds true for setting up a targeting study within the tool. Referring back to Figure 2 above, a blue box highlights the targeting study-specific portion of the "Parametrics" tab in the PESST GUI. This blue-highlighted area, along with the purple area, allows a user to set up a targeting study from the PESST GUI by defining the inputs necessary to set up the run. All of the previously described input variables needed to set up a run from the command line can be manipulated from the PESST GUI. Once again, for instructions of how to set up a study of this type, refer to the PESST User's Guide.

2. Targeting Output Data Summary

Summary information is provided from each targeting study completed that provides the values at each intermediate step for the control (or input) variable and targeted (or output) variable. Information at each step is also provided for the error from the previous step and the derivative approximation for the current step. This information is provided so that a user can further validate his or her solution. The output is provided in a file named targetingSummary.txt and is captured in the current study output folder. More information is provided in the PESST User's Guide.

C. Targeting as a Sub-Problem of Parameterization

One of the functionalities that might be overlooked but is also a valuable addition to the PESST framework is the added capability of the tool being able to handle a targeting study as a sub-problem of a parametric study. For example, if a user would like to target a specific value for the heat rate while parametrically sweeping through a range of nose radii, he could perform this analysis in the current framework. When a targeting study is completed multiple times during the course of a parametric trade study, the summary information for each targeting study will still be captured in the targetingSummary.txt output file.

V. Parametric and Targeting Study Results Completed During an Orion Vehicle Design Review

Parametric and targeting study analyses have been completed for the Orion Crew Exploration Vehicle's (CEV) post-skip phase, Earth entry upon return from a lunar mission. In an effort to focus on key entry parameters, the parametric study is limited to varying the entry velocity and flight path angle. These variables will be varied over a wide range of values at relatively small increments to capture local effects and disturbances to the simulated PESST output variables studied. For the targeting study, the entry velocity will be varied to target the maximum sensed deceleration value to a lower-than-nominal value. These analyses were completed not only to showcase the value of the new parametric and targeting capabilities of PESST, but also to examine the effects of the vehicle's design on key trajectory attributes.

A. Orion Vehicle Design Definition and Entry Trajectory Conditions

The geometric design of the Orion CEV was needed in order to conduct the parametric and targeting studies for this report. The specifics of the design used for this study are provided in Table 1. The main source used for the Orion design information was the National Aeronautics and Space Administration (NASA) Exploration Systems Architecture Study (ESAS). Some values for the current study were approximated as the current design status of the Orion CEV is both in flux and proprietary.

Table 1: Orion CEV Design Parameters

Design Characteristic	Value
Approximated Body Type	Sphere Cone†
Nose Radius	6.476 meters*
Cone Angle	65.0 degrees*
Maximum Diameter	5.5 meters*
Approximated Parachute Type	Disk Gap Band†
Number of Parachutes	3*
Parachute Diameter	34.0 meters*

* Actual data as referenced in the NASA ESAS report
† Approximated for current PESST study

The nominal trajectory entry conditions were derived from the ESAS report. Since the study to be conducted focuses on the post-skip phase for the CEV entry, the post-skip set of entry condition values will be used for the nominal case. The list of all of the nominal entry conditions can be found in Appendix A: Orion CEV Post-Skip PESST Parametric Study Input File. The nominal entry mass was given as 10,000 kg, the nominal entry velocity was 8000.0 km/s, and the nominal flight path angle was -2.0 degrees. The entry mass, velocity, and flight path angle variables will be discussed more below, but since they were used for the parametric and targeting studies, the values are noted here.

The trajectory conditions of parachute deployment were also derived directly from the ESAS report, and, as such, the simulated deployment of the parachutes occurred at an altitude of 6000 meters. The type of parachute used for the study was a disk gap band as that is the only standard PESST parachute type. The details of the heat shield design were also approximated based on standard PESST heat shield materials because of the sensitive nature of the Orion heat shield design.

B. Parameterized Input Variable Definitions

The parametric study completed focused mainly on the entry conditions for the Orion CEV. As mentioned before, the entry mass, velocity, and flight path angle were varied over a range of values with a relatively small increment to capture as many local fluctuations in the responses of the output variables as reasonable. The entry mass, velocity, and flight path angle were each varied by a large enough range to capture interesting results but a small enough range to allow for a valid PESST solution set. The settings for these entry conditions are provided in Table 2.

Table 2: Parametric Settings for Entry Conditions

	Entry Mass	Entry Velocity	Entry Flight Path Angle
Lower Bound	10,000 kg	7,000 km/s	-2.2 degrees
Upper Bound	40,000 kg	8,400 km/s	-1.8 degrees
Increment	1,000 kg	100 km/s	-0.04 degrees
Number of Points	31	15	11

As has been noted, a parametric study for PESST applies a full-factorial analysis. As such, 5,115 cases were completed for this parametric study.

C. Targeting Study Definitions

The targeting study completed for this report focused on looking at the maximum sensed deceleration during the entry vehicle's trajectory and varying the entry velocity to reduce that deceleration to a lower value than experienced during entry with nominal ESAS entry conditions. It is discussed later, but the maximum sensed deceleration at nominal entry conditions was found to be approximately 7.6 g's. For this targeting study, a maximum deceleration of 7.2 g's is targeted. The full targeting study PESST input file has been provided in Appendix B: Orion CEV Post-Skip PESST Targeting Study Input File.

D. Results

The output data from the parametric and targeting studies has been summarized and is provided below.

1. Parametric Study Results

As mentioned earlier, the ability to complete a full-factorial parametric study provides the analyst much more detailed insight to the design space and allows greater knowledge to be attained. The data captured from the parametric analysis completed for this report was post-processed with "the consolidator" tool referred to in the "Post-Run Analysis and Querying of Output Data" section above. The post-processing of the parametric data included capturing information from each run at points along the trajectory's path that were considered interesting or of particularly important concern. For example, the points along the trajectory at which the maximum heat-transfer rate and sensed deceleration were experienced, along with the final trajectory values, were captured for further investigation. The calls to the consolidate program used to capture these values and others, have been provided in Appendix C: Post-Processing "consolidate" Program Calls. Once the data from the parametric runs had been post-processed to capture interesting subsets of the output, the information was read into MATLAB® data structures and added to a JMP® data table for further analysis. The MATLAB® scripts used to process the consolidated output data are provided in Appendix D: Post-Processing MATLAB® Scripts and additional plots from the JMP® analysis are provided in Appendix E: Additional JMP® Statistical Analysis. There were three main output variables studied: the maximum sensed deceleration, the total integrated heat load, and the maximum heat-transfer rate. The most interesting data from these output variables are provided here for review.

As a quick overview of the output data studied, the JMP® prediction profiler plot with each of the three input variables set to its approximate nominal ESAS value has been provided below in Figure 3. As discussed earlier, the nominal entry conditions as defined by the ESAS report are given by an entry mass of 10,900 kg, an entry flight path angle of -2.0 degrees, and an entry velocity of 8,000 m/s. The prediction profiler gives some quick insight to the relationships between the output variable responses to various input settings. From the profiler, it can be noted that the Entry Mass value has a relatively significant impact on the maximum heat-transfer rate while it does not seem to impact the total integrated heat load or maximum sensed deceleration as much. Also, the profiler shows a fair bit of variability in the output responses due to the value of the entry velocity. More detailed analysis of each of the response variables is to follow, but the prediction profiler allows for quick orientation to the nature of the responses.

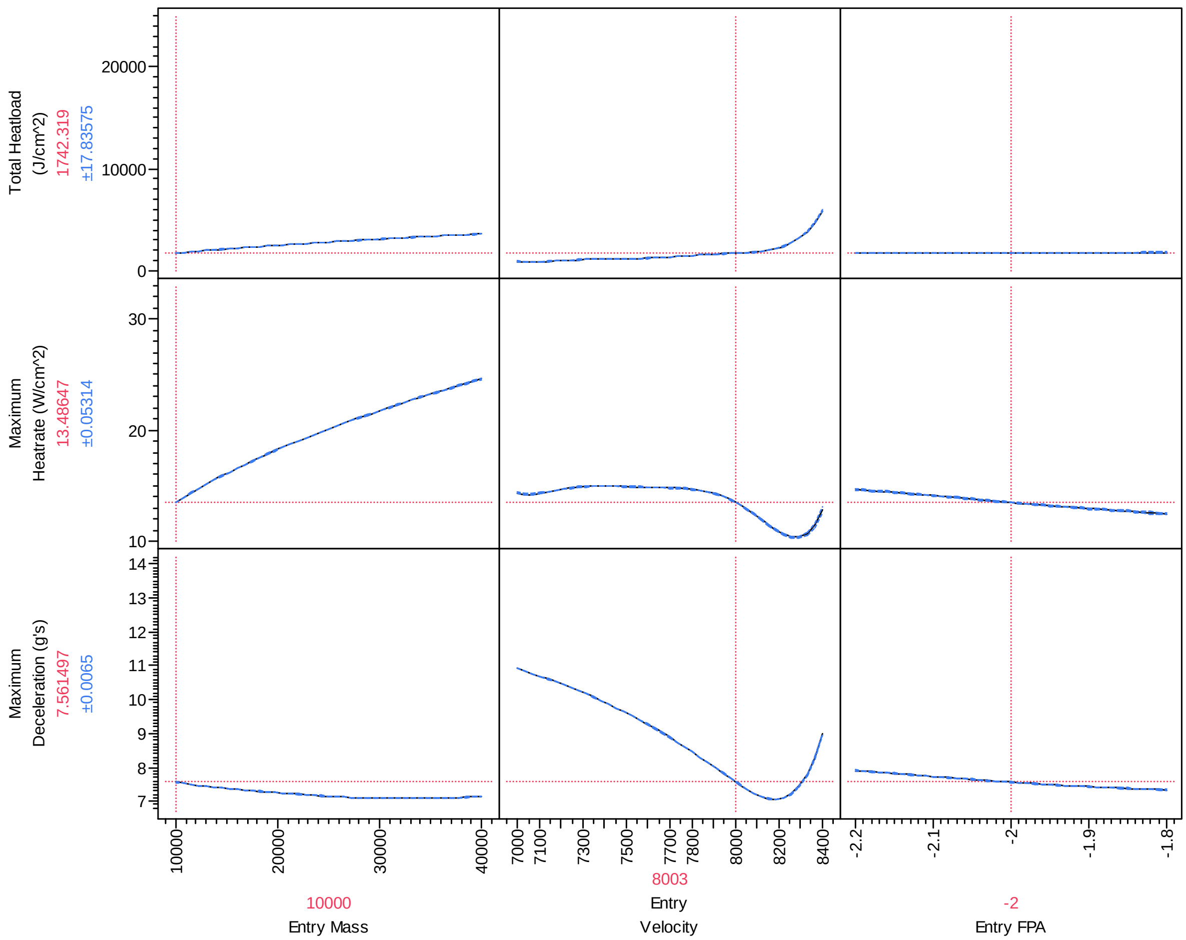

Figure 3: Prediction Profiler Set to the ESAS Nominal Entry Conditions

The surface and contour plots depicting the maximum sensed deceleration at the nominal entry velocity are provided in Figure 4 and Figure 6. Both the surface and contour plots are provided to allow the reader to refer to the plot that he or she prefers. From the figures, the maximum sensed deceleration for the nominal ESAS entry conditions can be located and was found to be approximately 7.6 g's. It can also be noted, as expected, that as the magnitude of entry flight path angle is decreased (i.e., brought toward zero), or, as the entry mass is increased until around 30,000 kg, the maximum sensed deceleration decreases. Once the entry mass is increased beyond 30,000 kg, the maximum sensed deceleration begins to increase. The decrease in maximum sensed deceleration as the flight path angle magnitude is decreased can be explained physically by noting that the trajectory's path is not as "steep" and the entry vehicle slows more gradually in the Earth's atmosphere. To a certain degree, a corollary to the vehicle's ballistic coefficient can be made in explanation of the decrease in maximum sensed deceleration as the entry mass is increased. The ballistic coefficient is a measure of a vehicle's ability to overcome air resistance and can be calculated as

$$BC = \frac{M}{C_d A} \tag{1}$$

where M is the vehicle's mass, C_d is the vehicle's drag coefficient, and A is the entry vehicle's wetted area. For this study of the Orion CEV, the vehicle's drag coefficient and wetted area remained constant while the entry mass was varied. As a result, the ballistic coefficient varied proportionally with the entry mass from 25.3 to 101 kg/m^2. If the reader would like to compute the ballistic coefficient for the cases completed, the value of C_dA from this study was 395.8533 m^2. At a certain point, the size of the entry vehicle would need to increase to accommodate an increase in the entry mass, and, provided the mass remained constant during the increase in vehicle size, the ballistic coefficient would be reduced.

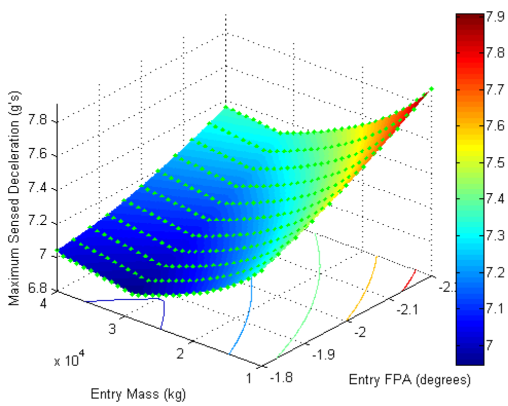

Figure 4: Surface Plot Maximum Sensed Deceleration at Nominal Entry Velocity of 8,000 m/s

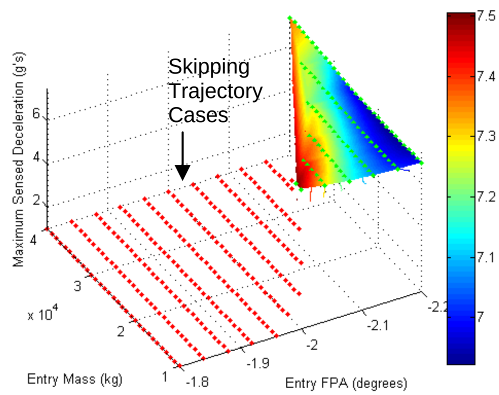

Figure 5: Surface Plot Maximum Sensed Deceleration at Entry Velocity of 8,300 m/s

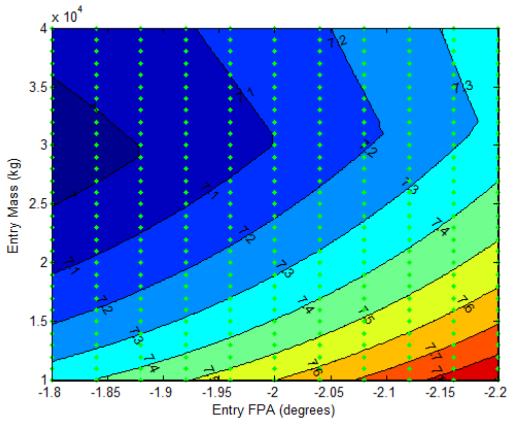

Figure 6: Contour Plot for Maximum Sensed Deceleration (g's) at Nominal Entry Velocity of 8,000 m/s

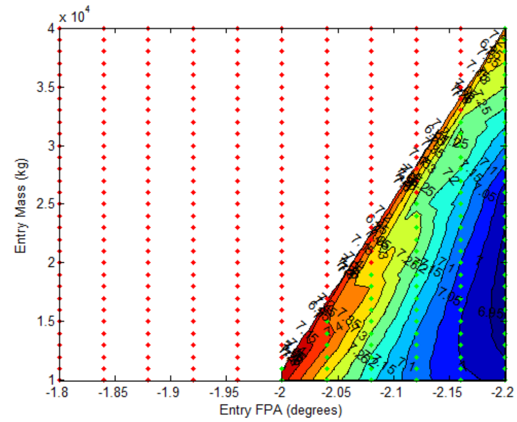

Figure 7: Contour Plot for Maximum Sensed Deceleration (g's) at Entry Velocity of 8,300 m/s

An additional trend that could be easily noted from the plotted output data was that of the range of input variable values that caused the entry capsule to "skip out" of the atmosphere. As would be expected, as the entry velocity increased, skip trajectories became more prevalent. The surface and contour plots of the maximum sensed deceleration at an entry velocity of 8,300 m/s are provided in Figure 5 and Figure 7, and the "skip out" trajectories are noted with red points on each plot. It is easily noticed that the vast majority of parametric runs caused the entry capsule to skip out of the atmosphere and not land. These "skip" trajectories would most likely result in either an aerobraking or aerocapture of the entry capsule. That is, as opposed to the entry capsule being inserted into a non-atmospheric interfacing orbit as the velocities are too low for that type of orbit to occur.

It should be noted that all of the surface and contour plots provided in this study indicate successfully landed parametric cases with green points and non-landed cases with red points. The non-landed cases were neglected when creating surface and contour plots as those values skewed the interpolated data points and provided results that proved to be difficult to interpret. The criterion for a successful landing was that the final altitude of the entry vehicle be below 100 meters. It was noted that some cases were approaching a landed state more slowly than others, but the maximum trajectory time of 2,000 seconds was reached before the landed state could be achieved. While these slow descent cases did not result in skipping trajectories, they were still designated as non-landed and were neglected for the surface and contour plotting interpolation since the trajectory did not bring the entry vehicle to a low enough altitude in the allotted time. Also, the points depicted on the plots provide a representation of the actual cases that were solved with PESST. All points in between are linearly interpolated to create the continuous surface for visualization.

The total integrated heat load and maximum heat-transfer rate were also examined at various entry masses, flight path angles, and velocities. A summary of the results of the total integrated heat load is provided in Figure 8 through Figure 11. Some trends that can be noted from these plots are the relationships between the input variables and total integrated heat load. For example, as the entry mass is increased from 10,000 kg in Figure 8 and Figure 10 to 30,000 kg in Figure 9 and Figure 11, the general shape of the surface does not change much until the region of the graph with higher entry velocity is reached. This moderate shape change as the entry velocity increases indicates that there are some coupling effects between the input variables for this upper range of entry velocities. This coupling, or interaction, effect can also be seen in Figure 12. Figure 12 is the JMP® interaction profile for the total integrated heat load. It shows that as the entry velocity increases, the changes in the total heat load become more exaggerated. Another way one might describe this exaggerated change in the heat load would be to say that the derivative of the heat load with respect to the entry velocity increases as the entry velocity increases. Also of note in Figure 12 is the relative slope of the lines on the entry velocity row for the entry mass and entry flight path angle. In both cases, those lines indicate that as the velocity is increased, the heat load increases more rapidly as either the entry mass or the entry flight path angle is increased. This phenomenon can also be seen in the prediction profile plots provided in Appendix E: Additional JMP® Statistical Analysis. Several prediction profiles were created changing the settings for the input variables to show the relative effects on the outputs due to these changes.

Also from Figure 8 through Figure 11, it can be noted that as the value of the entry mass is increased, the overall height of the surface increases indicating higher values for the total heat load. This fact is also supported by the JMP® plots provided.

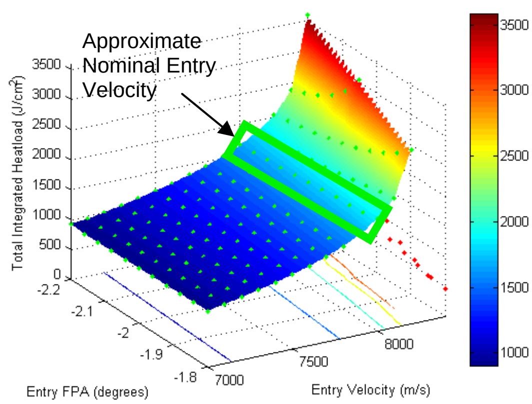

Figure 8: Surface Plot of Total Integrated Heat Load at an Entry Mass of 10,000 kg

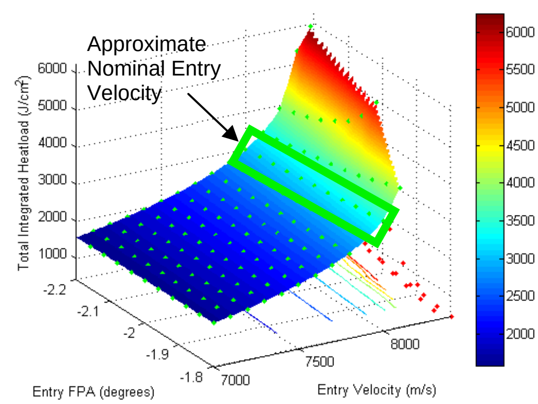

Figure 9: Surface Plot of Total Integrated Heat Load at an Entry Mass of 30,000 kg

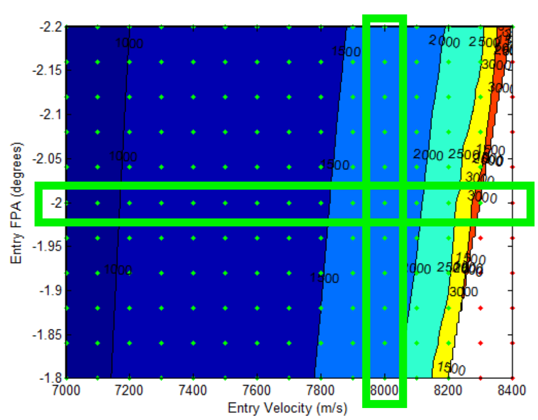

Figure 10: Contour Plot of Total Integrated Heat Load at an Entry Mass of 10,000 kg

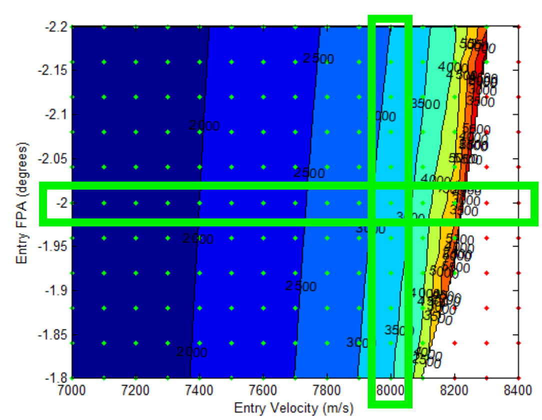

Figure 11: Contour Plot of Total Integrated Heat Load at an Entry Mass of 30,000 kg

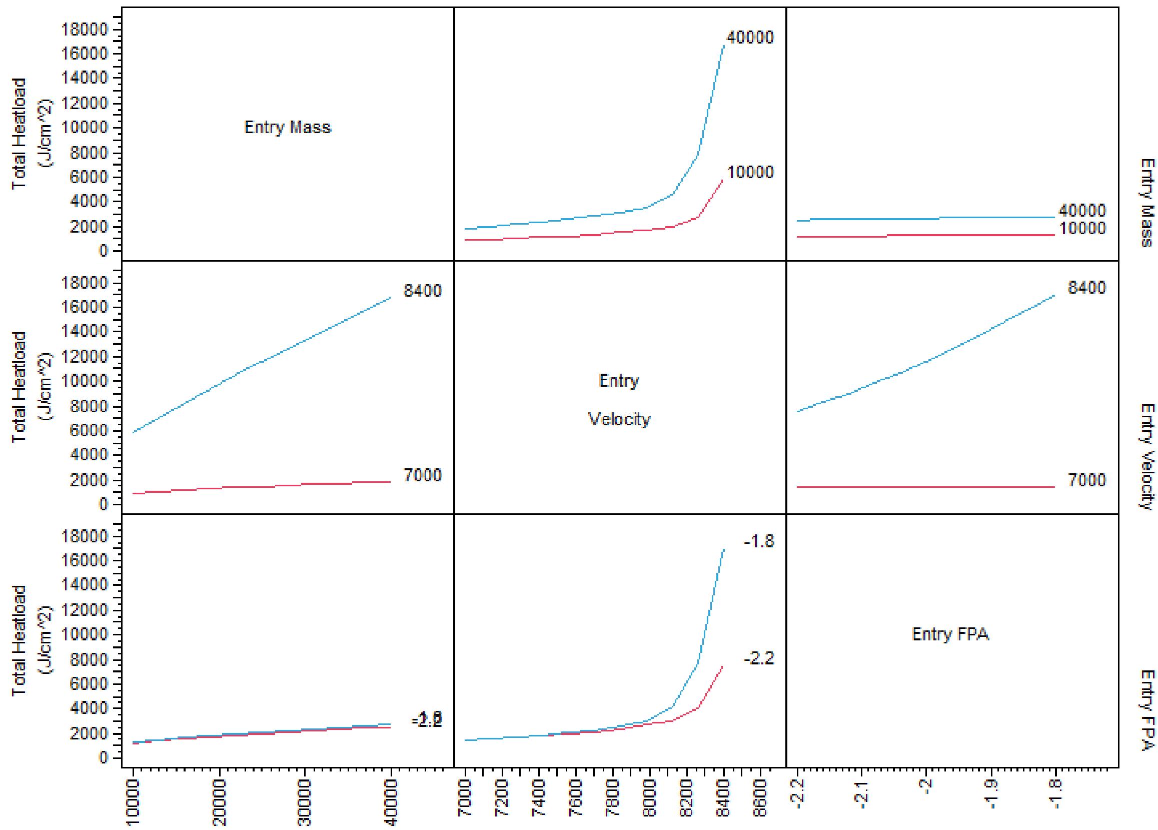

Figure 12: Interaction Profile for the Total Integrated Heat Load

It can also be noted from Figure 8 through Figure 12 that, at the nominal ESAS entry velocity, the flight path angle has little impact on the total integrated heat load. So, if the CEV enters the atmosphere slightly off of the nominal entry flight path angle condition the impact to the total heat load should be minimal. This fact could be a good indication as to why that entry velocity was chosen for the CEV entry.

Figure 9 provides the total integrated heat load when the entry mass is increased to 30,000 kg. This value for entry mass was chosen because of the minimums achieved in Figure 4 for the maximum sensed deceleration at this value of entry mass. In Figure 9, it can be noted that the total heat load is somewhat higher than in the case when calculated for the nominal case with an entry mass of 10,000 kg. Therefore, from a mission design standpoint, a trade would have to be completed that would include either a change to the heat shield design allowing it to withstand the higher heat load or some method to slow the entry vehicle prior to atmospheric interface. In either case, a portion of the increase in mass would likely be attributed to the subsystems allowing for entry with the increased mass and would therefore negatively impact the payload mass that could be used for scientific or life-support mission requirements. These are a few examples of the trades that a mission designer would be able to make more educated decisions about with the new PESST parameterization capability.

The maximum heat-transfer rate is another important entry parameter to consider when designing a new mission. Figure 13 through Figure 16 below provide the surface plots for the maximum heat-transfer rate at entry masses of 10,000 kg and 30,000 kg.

Figure 8 through Figure 11 and Figure 13 through Figure 16 also show several red points that indicate non-landed cases as noted earlier. These non-landed cases occur at higher entry velocities which agrees with the previous analysis that cases with higher entry velocities tend to skip out of the atmosphere.

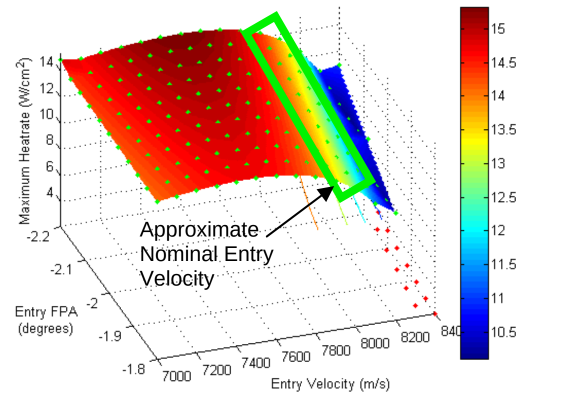

Figure 13: Surface Plot of Maximum Heat-Transfer Rate with an Entry Mass of 10,000 kg

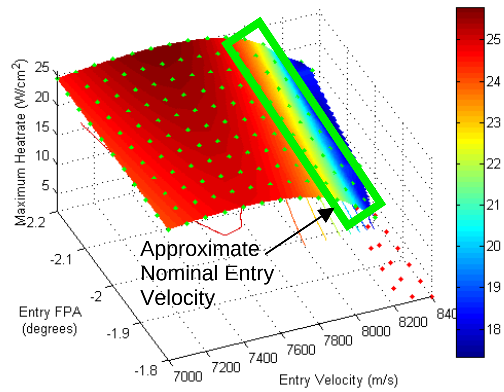

Figure 14: Surface Plot of Maximum Heat-Transfer Rate with an Entry Mass of 30,000 kg

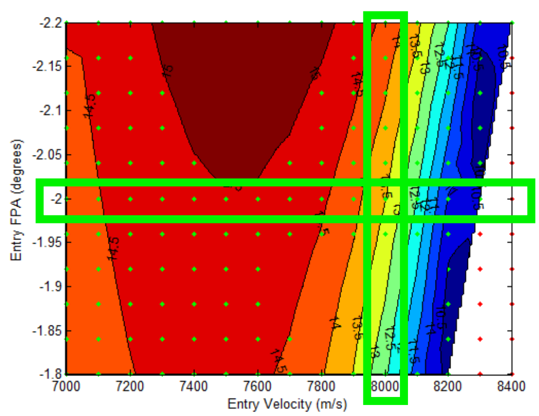

Figure 15: Contour Plot of Maximum Heat-Transfer Rate with an Entry Mass of 10,000 kg

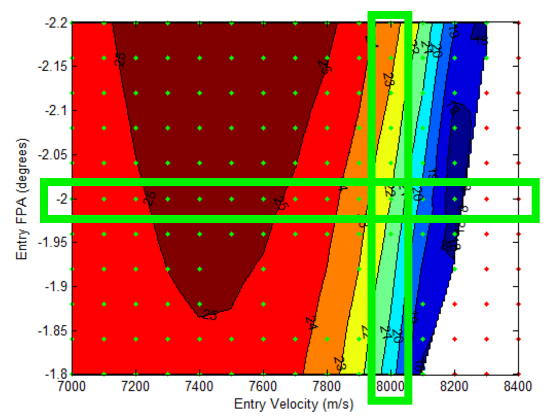

Figure 16: Contour Plot of Maximum Heat-Transfer Rate with an Entry Mass of 30,000 kg

Figure 13 through Figure 16 once again show that there is coupling between the entry mass, entry velocity, and entry FPA when it comes to the effect on the maximum heat-transfer rate. As with the total integrated heat load though, the net effect of the entry flight path angle and the entry velocity seems to be more pronounced than the effect due to the entry mass. As the entry velocity increases, the maximum heat-transfer rate decreases more rapidly as the magnitude of the flight path angle is decreased. These rates do not seem to be as effected by the entry mass in this case. The JMP® interaction profile is provided for the maximum heat-transfer rate in Figure 20 of the appendix for the interested reader.

Another interesting note about the maximum heat-transfer rate is that it appears to level off for a wide range of entry velocities. That is, as the entry velocity is decreased, the maximum heat-transfer rate increases to a point, and then stops increasing and remains near the same value over a wide range of entry velocities. The nominal ESAS entry conditions are located below this leveled-off region of the maximum heat-transfer rate surface plot. This range could have been chosen due to heat shield material properties limitations.

Again, Figure 14 and Figure 16 provide the maximum heat-transfer rates for an entry mass of 30,000 kg, which was chosen to be displayed because of the minimums that occurred in the maximum sensed deceleration plots above. In the case of the maximum heat-transfer rates, the values are significantly higher than for the nominal case. Obviously, the desire to increase the entry mass is hypothetical for the purposes of this study; however, if the mass requirement were critical for the design in question, the heat shield material and thickness choices would be critical aspects of the design. In the case of the heat-transfer rate, decreasing the entry velocity would not improve the entry conditions for the design since the heat-transfer rate increases as the entry velocity decreases.

2. *Targeting Study Results*

The targeting study to target a new value for the maximum sensed deceleration by varying the entry velocity required eight Newton-Raphson iterations. The summary of these iterations is provided in Figure 17 and Table 3.

As can be seen in Table 3, the entry velocity value that resulted in the maximum sensed deceleration of 7.2 was found to be 8,096.696 m/s.

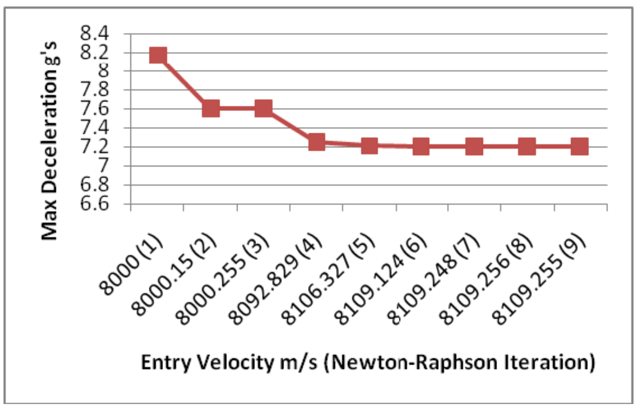

Figure 17: Targeting Iteration History

Table 3: Targeting Iteration History

Iteration	Entry Velocity (m/s)	Maximum Deceleration (g's)
1	8,000	7.557712
2	8,000.15	7.557067
3	8,083.188	7.243398
4	8,094. 677	7.206318
5	8,096.635	7.200153
6	8,096.683	7.200032
7	8,096.696	7.199998
8	8,096.696	7.2

VI. PESST Enhancement Summary and Recommendations for Future Work

The implementation of a PESST input variable parameterization capability allows a user to quickly and easily set up a parametric trade study. In doing so, the user can run multiple cases over a range of values for various input variables of importance. This parameterization capability should greatly aid in increasing the user's knowledge about his or her design early on in the process.

The targeting capability added to the PESST framework will give users the ability to find specific values that provide a valid solution to a given design criteria he or she may be trying to satisfy. This capability should prove invaluable when a show-stopper requirement exists and analysts are required to meet a specific requirement in the design space. In addition, the targeting capability will also give users an additional tool to gain more insight to designs within an already mature conceptual design framework.

These capabilities were added such that future work may further extend the existing functionality adding even more to the capability suite. Some possible areas for improvement for the parameterization functionality might include adding a Monte Carlo analysis capability or a set of Design of Experiments (DoE) options as opposed to always running a full-factorial parametric study. For the targeting functionality, an optimization toolbox could be added to the PESST framework allowing for various numerical engineering analyses to be conducted. A genetic and/or gradient search algorithm would both be valuable assets to the PESST framework, and all of these functionalities should be able to be built on the groundwork that has been laid with this project.

An optimization toolbox would have been very useful for when conducting the trade study for this report. The parametric study capability provides the user with the ability to run hundreds, thousands, or even more cases if desired and if system memory permits; however, the running of these cases could take hours and possibly even days to complete. Due to the significant amount of time required to complete these cases, it is best that the user have a well-defined problem set before committing to that time requirement. If the user were only interested in a sub-domain of the trade space, it would be best if there were a method for the user to find that local area with some optimization technique, and then possibly conduct a parametric study with much smaller bounds on the input variables of interest. This would not only allow for possible time savings, but would provide a much finer detail of the design space region of interest. Time savings would also be another advantage of adding a DoE toolbox to the PESST framework. For example, instead of needing to complete a full-factorial experiment, an analyst could set up a DoE that would require less runs but would not necessarily sacrifice a lot of fidelity due to prior knowledge about the relationships between variables.

With respect to the parametric study completed, as more information about the design of the Orion CEV is made available, the heat shield could be evaluated with respect to its ability to withstand the total heat load and maximum heat rates experienced. With this information, additional trades could be completed on the maximum deceleration, maximum heat-transfer rate, and total integrated heat load among other trajectory characteristics.

VII. Concluding Remarks

The addition of the capabilities described above has been shown to increase the capability of PESST while providing a simple and intuitive user interface through which a user can set up a study. In addition, an analysis of the Orion CEV has been conducted showcasing the full capability set that parametric and targeting studies has added to the tool. Some interesting trends were easily noted in the plots that were created with the large amount of data generated through the parametric study. For example, the relationship between the entry velocity and the frequency of “skip out” trajectories was easily noticed. Also, the coupled relationship between the entry condition variables on the total integrated heat load and the maximum heat-transfer rate could be seen. These relationships should prove useful for designers when trying to determine entry conditions for a new vehicle or when trading other aspects of the overall entry vehicle design.

VIII. Acknowledgements

First and foremost, I would like to thank my wife, Julie, for always having been willing to suffer what must have seemed like excruciatingly boring weekends and time spent without me while I was busy working on this project. She always managed to keep a positive attitude and provided me with infinite support when it came to allowing me to have time to work on this project.

I would also like to thank a few very close friends who were always willing to offer assistance or be my own personal cardboard programmers when I could not, for the life of me, figure out what was wrong with my code. Most prevalent from this group are Ravi Prakash, Steve Ortiz and, Lorraine Jong.

Also, a lot of gratitude goes out to the numerous people who have worked on PESST long before I knew it existed because, without their collective efforts, I would not have had a project on which to work. Special acknowledgement should be given to several team members who have made significant contributions to the tool. Brad Steinfeldt and Michael Grant for their work on the guidance and trajectory portions of the tool, John Dec for his thermal protection system sizing work, John Christian for his module for additional sizing analysis, and Richard Otero, who has not only probably done most of the work to get PESST to its current state, but who has also provided me with literally days, if not months, of his personal time in helping me to get this project completed. I would also like to thank Dr. Braun for allowing me to be a part of the PESST team and for providing me with solid direction and always having an excellent memory when it came to recalling exactly what that direction was.

References

1. Braun, Robert D., "Development of a Conceptual Design Mission Analysis System for Guided Entry Systems," Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, 2007.
2. Otero, Richard E., Grant, Michael J., Steinfeldt, Bradley A., Braun, Robert D., "Introducing PESST: A Conceptual Design and Analysis Tool for Unguided/Guided EDL Systems," Poster, 6th International Planetary Probe Workshop, IPPW-6, June 21-27, 2008, http://conferences.library.gatech.edu/ippw/index.php/ippw6/1/paper/view/90/32.
3. Otero, Richard E., Grant, Michael J., Steinfeldt, Bradley A., "Planetary Entry System Synthesis Tool V0.1.2_2 Users Guide," Space Systems Design Lab, Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, October 8, 2008.
4. Stanley, Dr. Doug, et. al., "NASA Explorations System Architecture Study," NASA-TM-2005-214062, November, 2005.

Appendix A: Orion CEV Post-Skip PESST Parametric Study Input File

```
!----------------------------------------
! PESST input file
! Autogenerated by PESST GUI -  - V0.1.2_2
! One should be aware when using one PESST input file on
! a different version of PESST that input variables and
! placement sometimes change between versions.
!----------------------------------------
!
! Targeting Inputs
!
targetingStudy       = l, [1, false]
targetingMethod      = s, [1, null]
targetingMaxIters    = i, [1, 25]
targetingTolerance   = r, [1, 1e-6, 1e-6]
controlVariable      = s, [1, null]
targetingCriteria    = s, [1, null]
limitingCriteria     = s, [1, null]
limitingInstance     = i, [1, 0]
!
! Consolidate Inputs
!
consolidateFolder    = s, [1, studies]
consolidatePrefix    = s, [1, output]
consolidateFileOut   = s, [1, consolidatedOutput.txt]
consolidateLimiter   = s, [1, final(time(s))]
!
! External GRAM Information
!
gramExecutable       = s, [1, null]
dataDirectory        = s, [1, null]
gcmDirectory         = s, [1, null]
entryDay             = s, [1, null]
entryMonth           = s, [1, null]
entryYear            = s, [1, null]
!
! Planet and atmospheric information
!
attractingBody       = s, [1, earth]
atmoModel            = s, [1, standard_1976]
planetAtmoFile       = s, [1, data/aero/earth1976Atmosphere.csv]
radiusPlanet         = r, [1, 6378136.0, 6378136.0]
massPlanet           = r, [1, 5.9742E24, 5.9742E24]
rotationRate         = r, [1, 0.004178008, 0.004178008]
specificHeatRatio    = r, [1, 1.4, 1.4]
massFractN2          = r, [1, 75.52, 75.52]
massFractO2          = r, [1, 23.2, 23.2]
massFractH2          = r, [1, 0.0, 0.0]
massFractHe          = r, [1, 0.0, 0.0]
massFractCO2         = r, [1, 0.0, 0.0]
massFractAr          = r, [1, 1.28, 1.28]
massFractCH4         = r, [1, 0.0, 0.0]
!
! Initial mass estimate information
!
```

```
entryMass            = r, [31, 10000.0, 40000.0]
landedMass           = r, [1, 10000.0, 10000.0]
fixEntryMass         = l, [1, true]
!
! Geometry
!
bodyType             = s, [1, sphereCone]
noseRadius           = r, [1, 6.476, 6.476]
coneAngle1           = r, [1, 65.0, 65.0]
maxDiameter1         = r, [1, 5.5, 5.5]
coneAngle2           = r, [1, 0.0, 0.0]
maxDiameter2         = r, [1, 0.0, 0.0]
areaTpsCovered       = r, [1, 0.0, 0.0]
!
! Trajectory Tab variables
!
! entryAlt(m)   entryVel(m/s) entryLat(deg N) entryLong(deg E)
! entryFPA(deg) entryAzi(deg) entryAoA(deg)   entryBankAng(deg)
!
entryAlt             = r, [1, 128000.0, 121920.0]
entryVel             = r, [15, 7000.0, 8400.0]
entryFPA             = r, [11, -2.2, -1.8]
entryAzimuth         = r, [1, 0.0, 0.0]
entryLatitude        = r, [1, 0.0, 0.0]
entryLongitude       = r, [1, 0.0, 0.0]
entryAoA             = r, [1, 0.0, 0.0]
entryBankAngle       = r, [1, 52.0, 52.0]
!
! Instantaneous Event Information
!
instEventNum         = i, [1, 0]
numEventVars         = 5
label                = s,
startTriggerSlope    = s,
startTriggerVariable= s,
startTriggerValue    = r,
massDrop             = r,
!
! Parachute Event Information
!
paraEventNum         = i, [1, 1]
numEventVars         = 12
label                = s, [1, Main DGB Parachute]
startTriggerSlope    = s, [1, D]
startTriggerVariable= s, [1, altitude]
startTriggerValue    = r, [1, 6000.0, 6000.0]
endTriggerSlope      = s, [1, D]
endTriggerVariable   = s, [1, altitude]
endTriggerValue      = r, [1, -2410.0, -2410.0]
massDrop             = r, [1, 224.73, 224.73]
areoCdMachFile       = s, [1, data/aero/vikingDGB.dat]
numParachutes        = i, [1, 3]
paraType             = s, [1, DGB]
paraDiameter         = r, [1, 34.0, 34.0]
!
! Guidance Event Information
!
```

```
guidEventNum        = i, [1, 0]
numEventVars        = 19
label               = s,
startTriggerSlope   = s,
startTriggerVariable= s,
startTriggerValue   = r,
endTriggerSlope     = s,
endTriggerVariable  = s,
endTriggerValue     = r,
massDrop            = r,
guidanceMode        = i,
propType            = s,
propTankMat         = s,
totThrustEngines    = r,
ispEngines          = r,
targetAlt           = r,
targetVel           = r,
targetLat           = r,
targetLong          = r,
targetFPA           = r,
targetHeading       = r,
!
! Thermal Tab variables
!
! heatshieldMat(string) insulMat(string)
! stagPointRadius(m) initTempOfTPS(C)
! bondlineTempLimit(C) recessThresTemp(C)
!
heatshieldMat       = s, [1, PICA]
stagPointRadius     = r, [1, 6.476, 6.476]
initTempOfTPS       = r, [1, 75.0, 75.0]
bondlineTempLimit   = r, [1, 250.0, 250.0]
recessThresTemp     = r, [1, 1500.0, 1500.0]
!
! Thermal Convergence Settings
!
! thickGuess(cm) maxGridSize(cm) timeStep(s) bondTempConvTol(C)
! relFactor, iterLimit
!
thicknessGuess      = r, [1, 1.0, 1.0]
maxGridSize         = r, [1, 0.1, 0.1]
timeStep            = r, [1, 0.5, 0.5]
bondlineTempConvTol = r, [1, 0.25, 0.25]
relaxFactor         = r, [1, 1.0, 1.0]
iterLimit           = i, [1, 25]
!
! Options Tab - Integration Settings
!
maxTrajTime         = r, [1, 2000.0, 2000.0]
minTrajAltitude     = r, [1, 0.0, 0.0]
maxTrajAltitude     = r, [1, 130000.0, 130000.0]
!
! Options Tab - General Settings
!
useModNewtonian     = l, [1, true]
convergeDesign      = l, [1, true]
sizeTPS             = l, [1, false]
```

Appendix B: Orion CEV Post-Skip PESST Targeting Study Input File

```
!----------------------------------------
! PESST input file
! Autogenerated by PESST GUI -  - V0.1.2_2
! One should be aware when using one PESST input file on
! a different version of PESST that input variables and
! placement sometimes change between versions.
!----------------------------------------
!
! Targeting Inputs
!
targetingStudy      = l, [1, true]
targetingMethod     = s, [1, Newton-Raphson]
targetingMaxIters   = i, [1, 25]
targetingTolerance  = r, [1, 1e-6, 1e-6]
controlVariable     = s, [1, entryVel]
targetingCriteria   = s, [1, sensedDecel=7.2]
limitingCriteria    = s, [1, max(sensedDecel)]
limitingInstance    = i, [1, 1]
!
! Consolidate Inputs
!
consolidateFolder   = s, [1, studies]
consolidatePrefix   = s, [1, output]
consolidateFileOut  = s, [1, consolidatedOutput.txt]
consolidateLimiter  = s, [1, final(time(s))]
!
! External GRAM Information
!
gramExecutable      = s, [1, null]
dataDirectory       = s, [1, null]
gcmDirectory        = s, [1, null]
entryDay            = s, [1, null]
entryMonth          = s, [1, null]
entryYear           = s, [1, null]
!
! Planet and atmospheric information
!
attractingBody      = s, [1, earth]
atmoModel           = s, [1, standard_1976]
planetAtmoFile      = s, [1, data/aero/earth1976Atmosphere.csv]
radiusPlanet        = r, [1, 6378136.0, 6378136.0]
massPlanet          = r, [1, 5.9742E24, 5.9742E24]
rotationRate        = r, [1, 0.004178008, 0.004178008]
specificHeatRatio   = r, [1, 1.4, 1.4]
massFractN2         = r, [1, 75.52, 75.52]
massFractO2         = r, [1, 23.2, 23.2]
massFractH2         = r, [1, 0.0, 0.0]
massFractHe         = r, [1, 0.0, 0.0]
massFractCO2        = r, [1, 0.0, 0.0]
massFractAr         = r, [1, 1.28, 1.28]
massFractCH4        = r, [1, 0.0, 0.0]
!
! Initial mass estimate information
!
```

```
entryMass            = r, [1, 10900.0, 10900.0]
landedMass           = r, [1, 10900.0, 10900.0]
fixEntryMass         = l, [1, true]
!
! Geometry
!
bodyType             = s, [1, sphereCone]
noseRadius           = r, [1, 6.476, 6.476]
coneAngle1           = r, [1, 65.0, 65.0]
maxDiameter1         = r, [1, 5.5, 5.5]
coneAngle2           = r, [1, 0.0, 0.0]
maxDiameter2         = r, [1, 0.0, 0.0]
areaTpsCovered       = r, [1, 0.0, 0.0]
!
! Trajectory Tab variables
!
! entryAlt(m)    entryVel(m/s) entryLat(deg N) entryLong(deg E)
! entryFPA(deg) entryAzi(deg) entryAoA(deg)   entryBankAng(deg)
!
entryAlt             = r, [1, 128000.0, 128000.0]
entryVel             = r, [1, 0000.0, 8000.0]
entryFPA             = r, [1, -2.0, -2.0]
entryAzimuth         = r, [1, 0.0, 0.0]
entryLatitude        = r, [1, 0.0, 0.0]
entryLongitude       = r, [1, 0.0, 0.0]
entryAoA             = r, [1, 0.0, 0.0]
entryBankAngle       = r, [1, 52.0, 52.0]
!
! Instantaneous Event Information
!
instEventNum         = i, [1, 0]
numEventVars         = 5
label                = s,
startTriggerSlope    = s,
startTriggerVariable= s,
startTriggerValue    = r,
massDrop             = r,
!
! Parachute Event Information
!
paraEventNum         = i, [1, 1]
numEventVars         = 12
label                = s, [1, Main DGB Parachute]
startTriggerSlope    = s, [1, D]
startTriggerVariable= s, [1, altitude]
startTriggerValue    = r, [1, 6000.0, 6000.0]
endTriggerSlope      = s, [1, D]
endTriggerVariable   = s, [1, altitude]
endTriggerValue      = r, [1, -2410.0, -2410.0]
massDrop             = r, [1, 224.73, 224.73]
areoCdMachFile       = s, [1, data/aero/vikingDGB.dat]
numParachutes        = i, [1, 3]
paraType             = s, [1, DGB]
paraDiameter         = r, [1, 34.0, 34.0]
!
! Guidance Event Information
!
```

```
guidEventNum         = i, [1, 0]
numEventVars         = 19
label                = s,
startTriggerSlope    = s,
startTriggerVariable= s,
startTriggerValue    = r,
endTriggerSlope      = s,
endTriggerVariable   = s,
endTriggerValue      = r,
massDrop             = r,
guidanceMode         = i,
propType             = s,
propTankMat          = s,
totThrustEngines     = r,
ispEngines           = r,
targetAlt            = r,
targetVel            = r,
targetLat            = r,
targetLong           = r,
targetFPA            = r,
targetHeading        = r,
!
! Thermal Tab variables
!
! heatshieldMat(string) insulMat(string)
! stagPointRadius(m) initTempOfTPS(C)
! bondlineTempLimit(C) recessThresTemp(C)
!
heatshieldMat        = s, [1, PICA]
stagPointRadius      = r, [1, 6.476, 6.476]
initTempOfTPS        = r, [1, 75.0, 75.0]
bondlineTempLimit    = r, [1, 250.0, 250.0]
recessThresTemp      = r, [1, 1500.0, 1500.0]
!
! Thermal Convergence Settings
!
! thickGuess(cm) maxGridSize(cm) timeStep(s) bondTempConvTol(C)
! relFactor, iterLimit
!
thicknessGuess       = r, [1, 1.0, 1.0]
maxGridSize          = r, [1, 0.1, 0.1]
timeStep             = r, [1, 0.5, 0.5]
bondlineTempConvTol = r, [1, 0.25, 0.25]
relaxFactor          = r, [1, 1.0, 1.0]
iterLimit            = i, [1, 25]
!
! Options Tab - Integration Settings
!
maxTrajTime          = r, [1, 2000.0, 2000.0]
minTrajAltitude      = r, [1, 0.0, 0.0]
maxTrajAltitude      = r, [1, 130000.0, 130000.0]
!
! Options Tab - General Settings
!
useModNewtonian      = l, [1, true]
convergeDesign       = l, [1, true]
sizeTPS              = l, [1, false]
```

Appendix C: Post-Processing “consolidate” Program Calls

```
:: Consolidate call to capture all parametric data in studies folder from
:: files with prefix “output”, saving to “initials.txt”, where the initial
:: time is met and return only the first instance of each initial time.
consolidate studies output initials.txt init(time) 1

:: Consolidate call to capture all parametric data in studies folder from
:: files with prefix “output”, saving to “finals.txt”, where the final time
:: is met and return only the first instance of each final time.
consolidate studies output finals.txt final(time) 1

:: Consolidate call to capture all parametric data in studies folder from
:: files with prefix “output”, saving to “maxConvHeatrates.txt”, where the
:: maximum convective heatrate is found and return only the first instance of
:: each maximum convective heatrate.
consolidate studies output maxConvHeatrates.txt max(convHeatrate) 1

:: Consolidate call to capture all parametric data in studies folder from
:: files with prefix “output”, saving to “maxDecels.txt”, where the maximum
:: sensed deceleration is found and return only the first instance of each
:: maximum sensed deceleration.
consolidate studies output maxDecels.txt max(sensedDecel) 1

:: Consolidate call to capture all parametric data in studies folder from
:: files with prefix “output”, saving to “maxMachs.txt”, where the maximum
:: mach values is found and return only the first instance of each maximum
:: mach value.
consolidate studies output maxMachs.txt max(mach) 1

:: Consolidate call to capture all parametric data in studies folder from
:: files with prefix “output”, saving to “maxHeatrates.txt”, where the
:: maximum total heatrate is found and return only the first instance of each
:: maximum total heatrate.
consolidate studies output maxHeatrates.txt max(totHeatrate) 1

:: Consolidate call to capture all parametric data in studies folder from
:: files with prefix “output”, saving to “maxDynPress.txt”, where the maximum
:: dynamic pressure is found and return only the first instance of each
:: maximum dynamic pressure.
consolidate studies output maxDynPress.txt max(dynPressure) 1
```

Appendix D: Post-Processing MATLAB® Scripts

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Main PESST Plotter Script
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all; clc;

% Plotting with FPA and Velocity on X and Y Axes
clear field;
field=pesstplotter('finals.txt','entryFPA','entryVel','totHeatload',0);

clear field;
field=pesstplotter('maxHeatrate.txt','entryFPA','entryVel','totHeatrate',0);

clear field;
field=pesstplotter('maxDecels.txt','entryFPA','entryVel','sensedDecel',0);

% Plotting with Mass and Velocity on X and Y Axes
clear field;
field=pesstplotter('finals.txt','entryMass','entryVel','totHeatload',0);

clear field;
field=pesstplotter('maxHeatrate.txt','entryMass','entryVel','totHeatrate',0);

clear field;
field=pesstplotter('maxDecels.txt','entryMass','entryVel','sensedDecel',0);

% Plotting with Mass and FPA on X and Y Axes
clear field;
field=pesstplotter('finals.txt','entryMass','entryFPA','totHeatload',0);

clear field;
field=pesstplotter('maxHeatrate.txt','entryMass','entryFPA','totHeatrate',0);

clear field;
field=pesstplotter('maxDecels.txt','entryMass','entryFPA','sensedDecel',0);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pesstplotter function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fields=pesstplotter(filepath,xaxis,yaxis,zaxis,numpoints,varargin)
  % Read in data from file
  fields = readpesstdata(filepath);

  % Determine the odd axis out
  i = checknames(fields, xaxis);
  j = checknames(fields, yaxis);
  k = getremaininginputfield(i,j);

  if ~isempty(varargin)
    values = cell2mat(varargin);
  else
    values = sort(unique(fields(k).data));
    mins = min(values); maxs = max(values); lens = length(values)-1;
```

```
    if numpoints == 0
      incs = (maxs - mins) / (length(values) - 1);
    else
      incs = (maxs - mins) / (numpoints - 1);
    end
    values = mins:incs:maxs;
  end

  for i = 1:length(values)
    indices = find(abs(fields(k).data - values(i)) < 1e-6);

    landed = intersect(fields(1).landed, indices);
    noland = intersect(fields(1).noland, indices);
    for j = 1:length(fields)
      filteredfields(j).name = fields(j).name;
      filteredfields(j).data = fields(j).data(indices);
      filteredfields(j).landed = fields(j).data(landed);
      filteredfields(j).noland = fields(j).data(noland);
    end

    if ~isempty(filteredfields(1).landed)
      makecontour(filteredfields, xaxis, yaxis, zaxis, k, values(i));
    end
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% getremaininginputfield function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function k = getremaininginputfield(i, j)
  if i == 2
    if j == 3
      k = 4;
    else
      k = 3;
    end
  elseif i == 3
    if j == 2
      k = 4;
    else
      k = 2;
    end
  else
    if j == 2
      k = 3;
    else
      k = 2;
    end
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% readpesstdata function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fields = readpesstdata(filepath)
  fields = readdata(filepath);
```

```
  % Get final altitude data to determine whether run completed or not
  finalspath = filepath;
  finalspath = strrep(finalspath, '\', '/');
  [finalspath, remain] = strtok(finalspath, '/');
  while ~isempty(remain)
    remain = remain(2:length(remain));
    [finalspath, remain] = strtok(remain, '/');
  end
  len = length(filepath)-length(finalspath);
  finalspath = strcat(filepath(1:len), 'finals.txt');
  finals = readdata(finalspath);

  % Determine landed/non-landed run numbers
  landed = find(finals(checknames(finals, 'altitude')).data <= 1000);
  landed = finals(1).data(landed);
  landed = unique(landed);
  j = 1;
  for i = 1:length(landed)
    if max(fields(1).data == landed(i))
      % Change landed indices from zero-based to one-based
      fields(1).landed(j, 1) = landed(i)+1;
      j = j + 1;
    end
  end

  noland = find(finals(checknames(finals, 'altitude')).data > 1000);
  noland = finals(1).data(noland);
  noland = unique(noland);
  j = 1;
  for i = 1:length(noland)
    if max(fields(1).data == noland(i))
      % Change non-landed indices from zero-based to one-based
      fields(1).noland(j, 1) = noland(i)+1;
      j = j + 1;
    end
  end

  % Capture landed/non-landed data from each field
  for j = 2:length(fields)
    fields(j).landed = fields(j).data(fields(1).landed);
    fields(j).noland = fields(j).data(fields(1).noland);
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% readdata function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fields = readdata(filepath)
  % Open file of interest
  fid = fopen(filepath, 'r');

  line = fgetl(fid);
  i = 1;
  while ~isempty(line)
    fields(i).name = strtok(line);
```

```
    line = strtrim(line(length(strtok(line))+1:length(line)));
    i = i + 1;
  end
  i = i - 1;

  str = strcat(strrep(blanks(i-1), ' ', '%f'), '%s');
  data = textscan(fid, str);

  for j = 1:i
    if j == i
      fields(j).data = char(data{j});
    else
      fields(j).data = data{j};
    end
  end

  % Close file
  fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% readdata function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function makecontour(fields, xaxis, yaxis, zaxis, third, val)
  % Define axes indices
  i = checknames(fields, xaxis);
  j = checknames(fields, yaxis);
  k = checknames(fields, zaxis);

  % Define axes data arrays neglecting the non-landed points
  x = fields(i).landed;
  y = fields(j).landed;
  z = fields(k).landed;

  try
    xx = min(x):(max(x)-min(x))/200:max(x);
    yy = min(y):(max(y)-min(y))/200:max(y);
    [XX, YY] = meshgrid(xx,yy);
    ZZ = griddata(x, y, z, XX, YY, 'linear');

    % Make surface plot
    figure;
    surfc(XX, YY, ZZ);
    shading interp;
    colorbar;
    axis tight;

    xlabel(fields(i).name);
    ylabel(fields(j).name);
    zlabel(fields(k).name);

    strTitle=strcat(fields(k).name,'@',fields(third).name,'=',num2str(val));
    title(strTitle);

    xg = fields(i).landed;
```

```
    yg = fields(j).landed;
    zg = fields(k).landed;
    xr = fields(i).noland;
    yr = fields(j).noland;
    zr = fields(k).noland;

    hold on;
    plot3(xg, yg, zg, 'g.', xr, yr, zr, 'r.');
    axis tight;

    saveimgs(strTitle);

    % Make contour plot
    figure;
    [c, h] = contourf(XX, YY, ZZ);
    clabel(c, h);
    view(-90, 90);
    axis tight;

    xlabel(fields(i).name);
    ylabel(fields(j).name);
    zlabel(fields(k).name);
    title(strTitle);

    hold on;
    plot3(xg, yg, zg, 'g.', xr, yr, zr, 'r.');
    axis tight;

    saveimgs([strTitle, 'contour']);
  catch
    err = lasterror;
    fprintf('  Message: %s\n', err.message);
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% saveimgs function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function saveimgs(fname)
  h = gcf;
  pat = {'/';'\';'<';'>';':';'*';'?';'"';'|';' '};
  fname = regexprep(fname,pat,'');

  [s,msg,msgid] = mkdir('img/fig');
  saveas(h,strcat('img/fig/',fname,'.fig'),'fig');
  [s,msg,msgid] = mkdir('img/png');
  saveas(h,strcat('img/png/',fname,'.png'),'png');

  close(h);
end
```

Appendix E: Additional JMP® Statistical Analysis

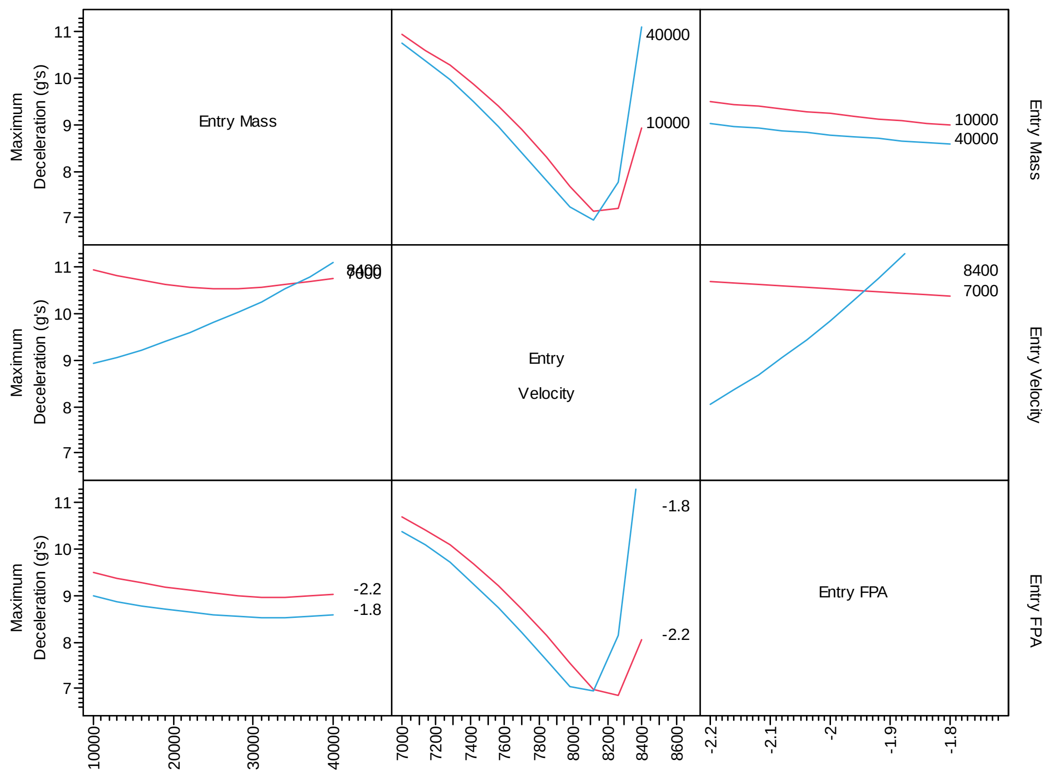

Figure 18: Interaction Profile for the Maximum Sensed Deceleration

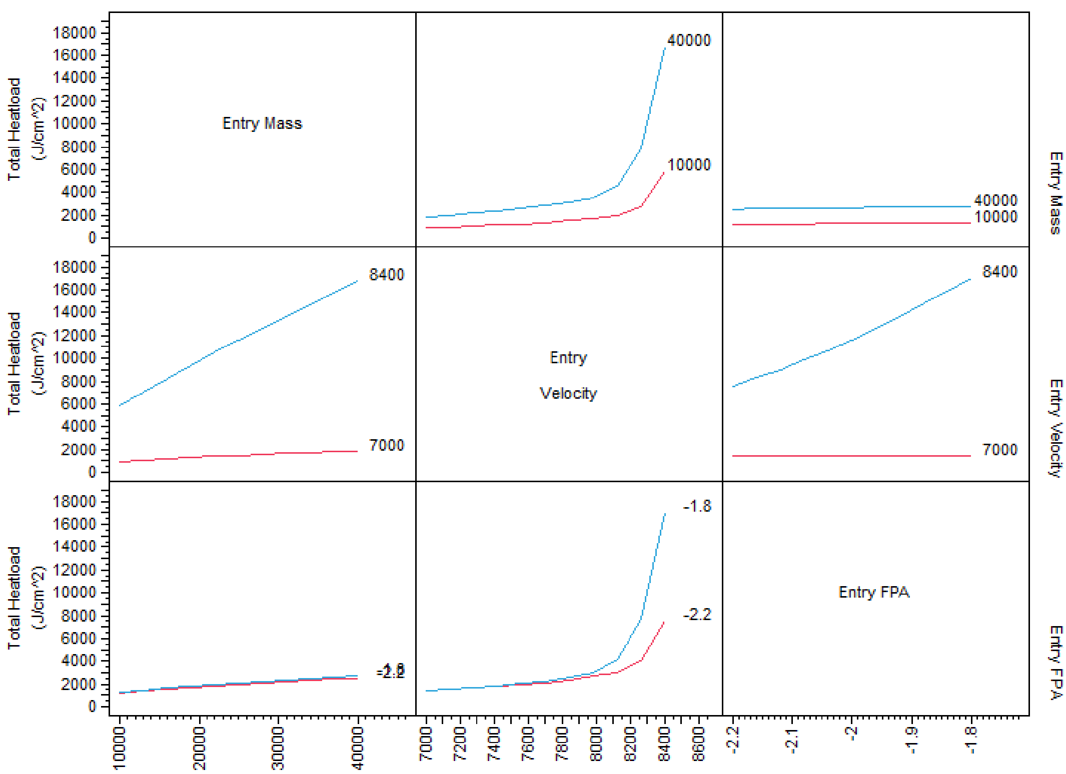

Figure 19: Interaction Profile for the Total Integrated Heat Load

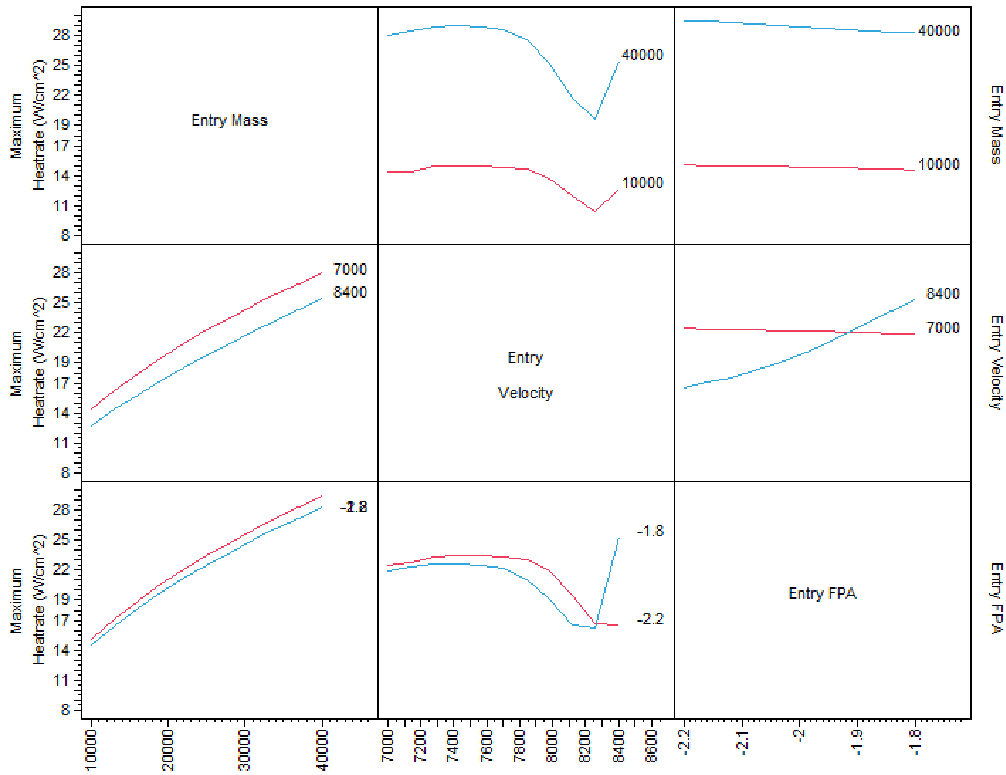

Figure 20: Interaction Profile for the Maximum Heat-Transfer Rate

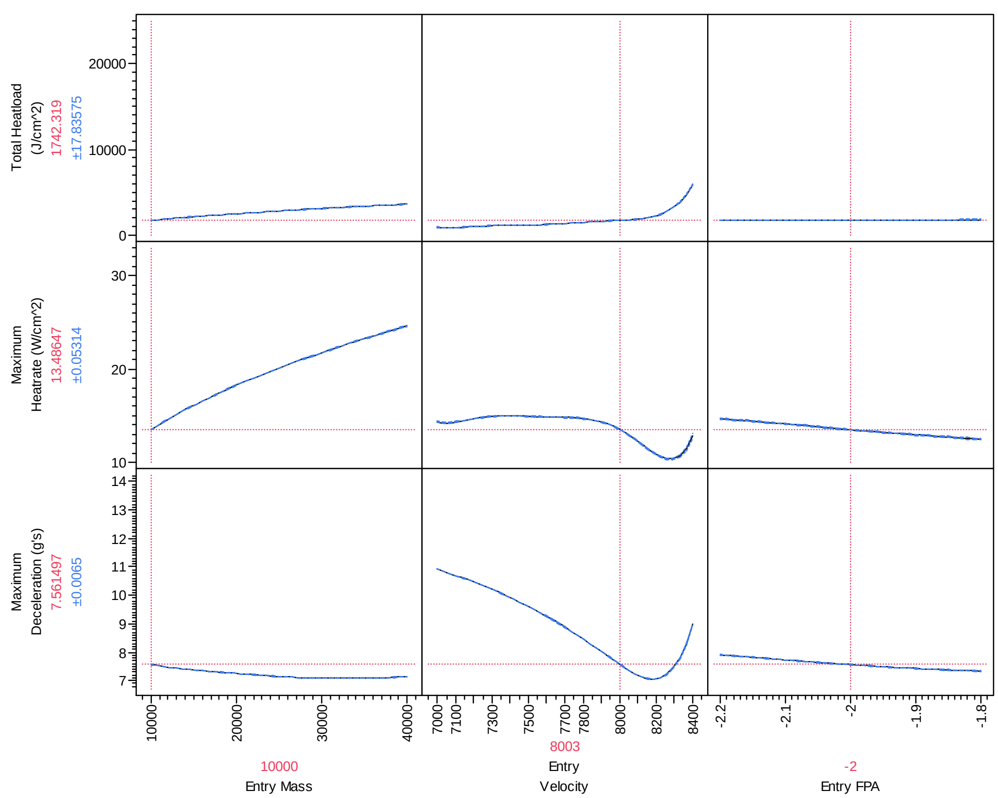

Figure 21: Prediction Profiler Set to the ESAS Nominal Entry Conditions

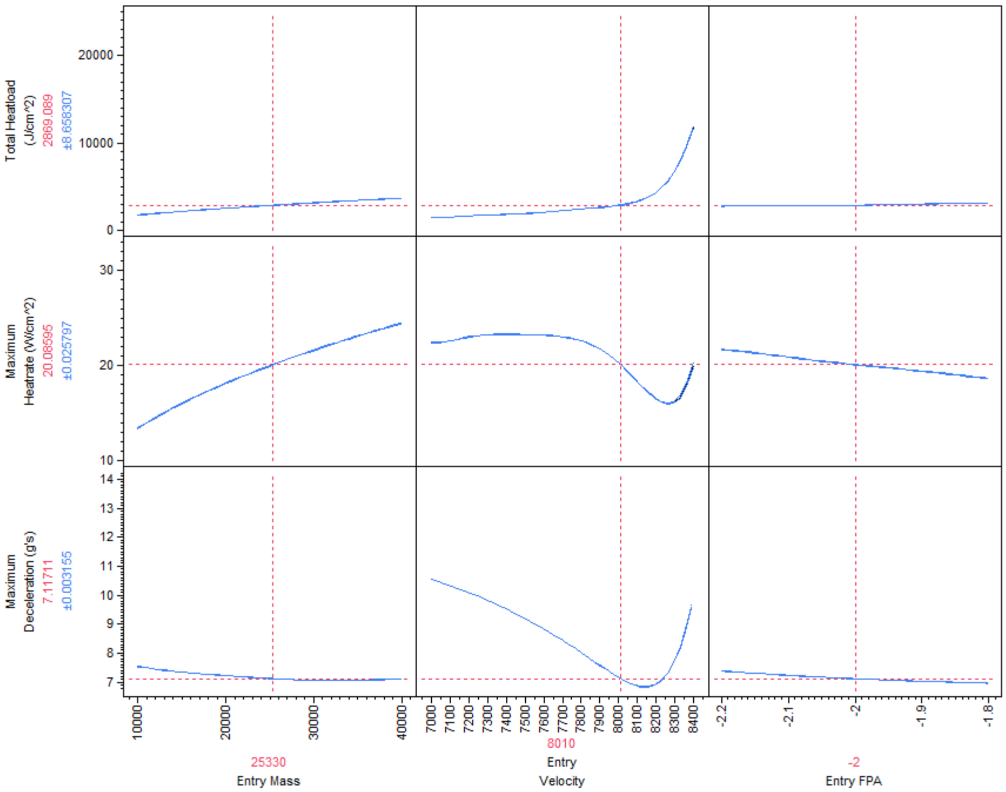

Figure 22: Prediction Profiler with Entry Mass set to 25,330 kg

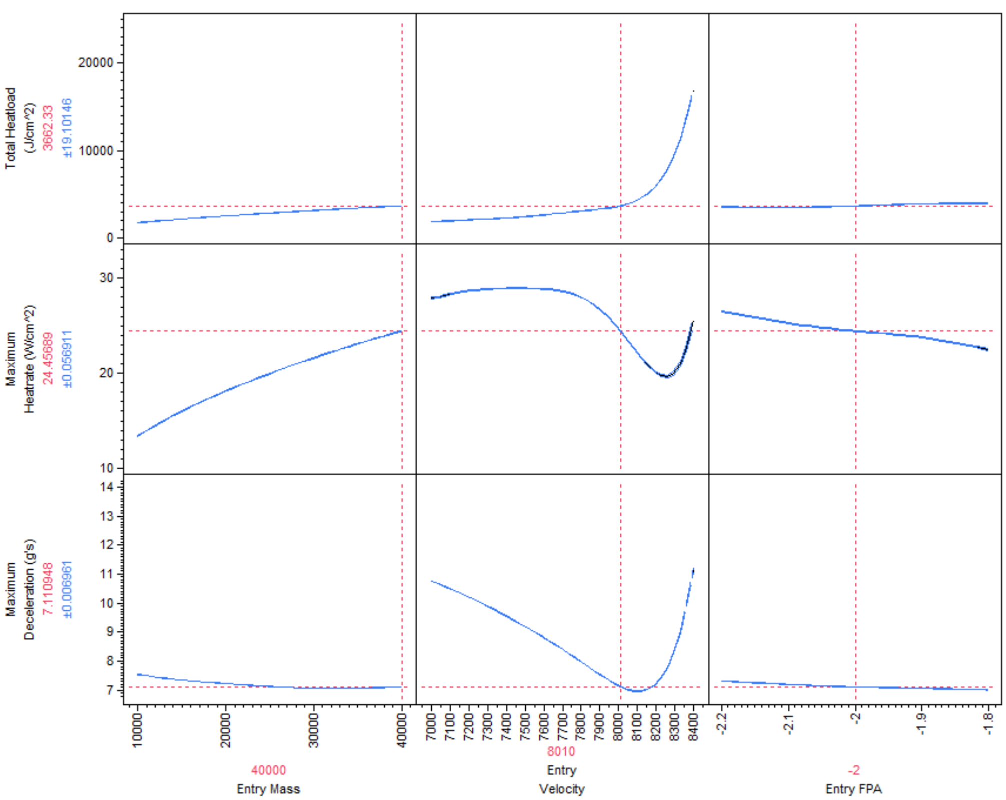

Figure 23: Prediction Profiler with Entry Mass set to 40,000 kg

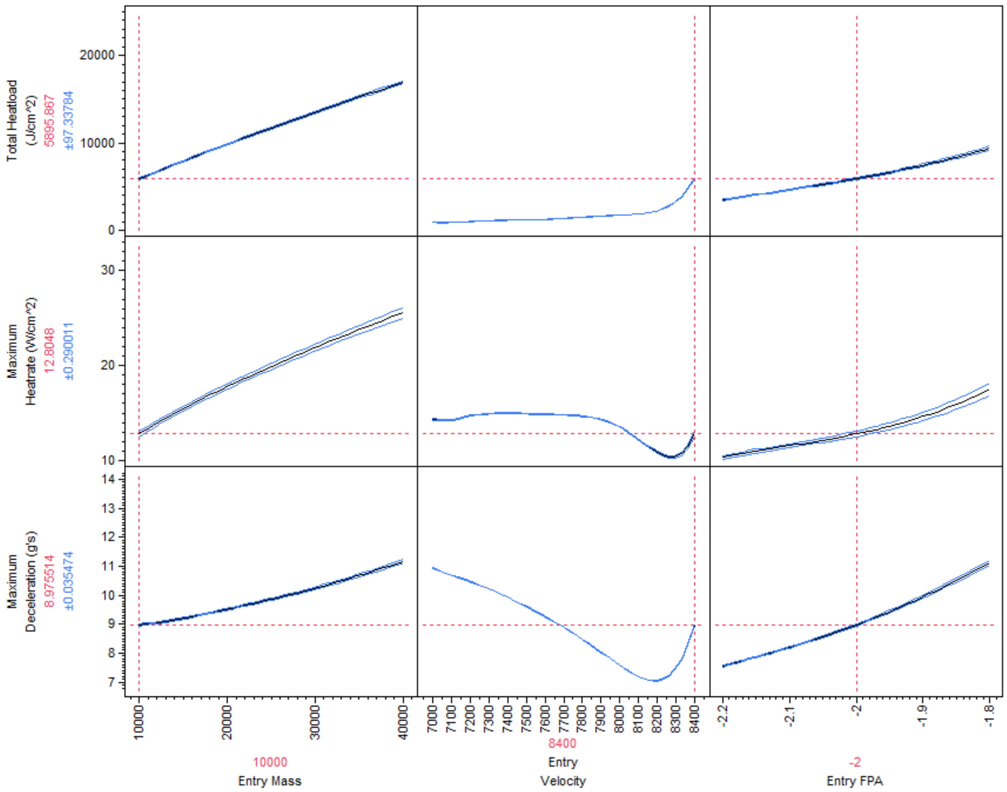

Figure 24: Prediction Profiler with Entry Velocity set to 8400 m/s

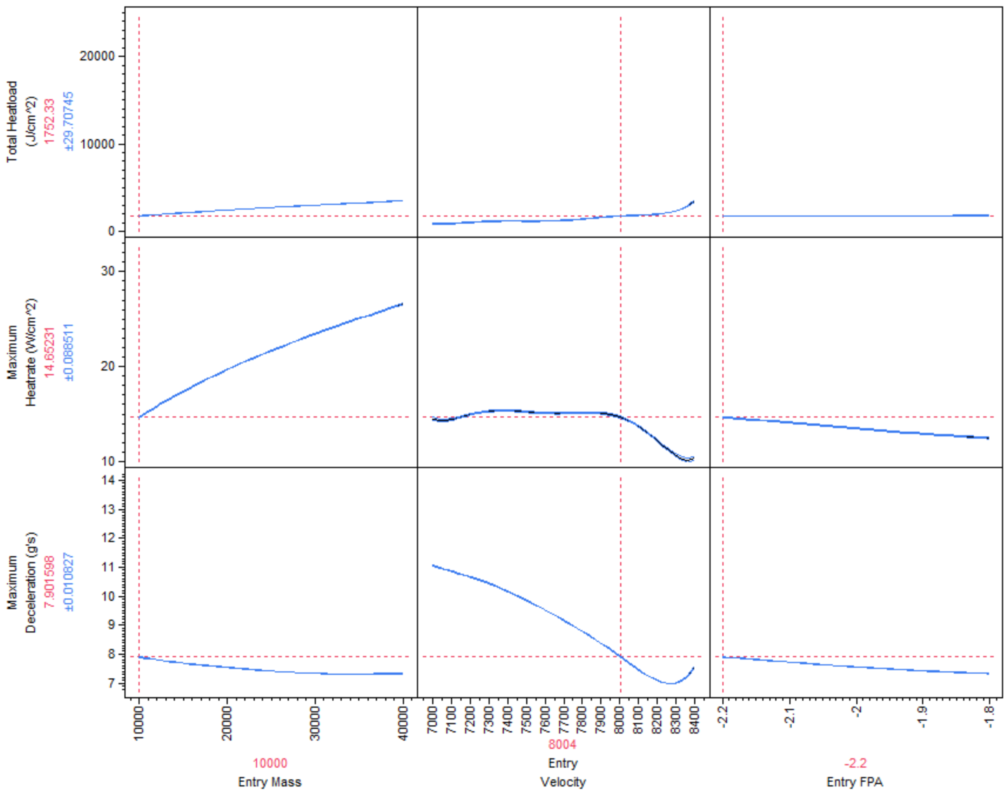

Figure 25: Prediction Profiler with Entry Flight Path Angle set to -2.2 degrees

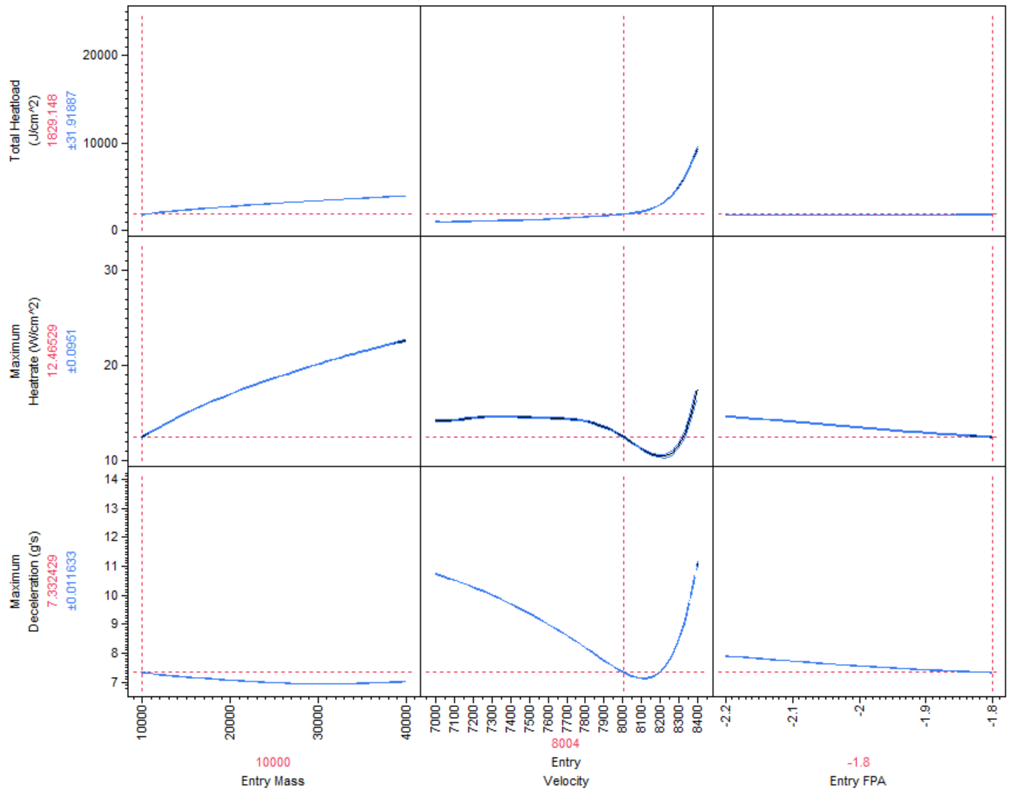

Figure 26: Prediction Profiler with Entry Flight Path Angle set to -1.8 degrees