A Methodology for Determination of Entry, Descent, and Landing Design Performance Margins of Spacecraft Landing on Planetary Bodies

Myron R Grover III

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Adam P Bruckner, Chair

Robert D Braun

Christina M Mastrangelo

Mehran Mesbahi

Program Authorized to Offer Degree:
Aeronautics and Astronautics
University of Washington

The research in this dissertation was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, and sponsored by the National Aeronautics and Space Administration (80NM0018D0004)

University of Washington

Abstract

A Methodology for Determination of Entry, Descent, and Landing Design Performance Margins of Spacecraft Landing on Planetary Bodies

Myron R Grover III

Chair of the Supervisory Committee:

Professor Adam P. Bruckner

William E. Boeing Department of Aeronautics & Astronautics

Design of a spacecraft and mission architecture that provide adequate performance margin is essential to successful spaceflight. This is particularly true for the Entry, Descent and Landing (EDL) phase of a landed mission. The quantification and management of performance margins is a key aspect for design of a successful EDL phase. In this research, a methodology for quantifying design performance margin, called Target Margin, is developed for EDL systems for planetary landing, using as a starting point a seed methodology utilized by NASA's InSight Project, a Mars lander that performed a landing in 2018. The methodology presented here utilizes Monte Carlo analysis, Monte Carlo filtering and Design of Experiments (DOE) techniques to identify modeling input parameters that most influence EDL performance metrics. The methodology then provides a framework for assessing confidence in modeling parameters to feed a quantification of Target Margin that utilizes second-order, DOE-based regression modeling. Assessment of epistemic and aleatory (reducible and irreducible) uncertainty is also included, to reveal where uncertainty in modeling might be reduced during a project lifecycle. This approach is applied to the InSight EDL system and results are compared to the originating InSight seed methodology, showing that the methodology results in more conservative, potentially more robust quantification of Target Margin than the seed methodology. Using InSight landing flight data, the Target Margin resulting from application of the methodology to InSight is compared to as-flown EDL performance margin revealing InSight landed with healthy margin, and EDL simulation modeling confidence is assessed in light of flight observations. Three sensitivity analyses were performed, the first using the methodology to establish bounding Target Margin based on low and high modeling confidence; the second is an examination of Target Margin quantification resulting from applying the methodology to a simulated InSight early project lifecycle EDL system; and the final sensitivity analysis performs a simple assessment of two modeling confidence weighting designs, showing generally low sensitivity to the designs. The methodology is successfully applied to the InSight EDL system, and in the process, findings identified for future work to improve the methodology for application to other planetary landings.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to acknowledge and thank my supervisor, Professor Adam Bruckner, who has been exceptionally supportive and patient as I have worked to complete this dissertation. He has been a superb mentor and guide, and I cannot thank him enough for all he has done to support my education and ambitions through the years. I would also like to thank my excellent committee, who provided expert feedback and guidance during my research. Additionally, I would like to thank my colleagues at the Jet Propulsion Laboratory, who provided great comments and encouragement, and especially Miguel San Martin and David Skulsky, who were collaborators on earlier efforts that seeded the research contained in this dissertation. A huge thanks and acknowledgement to Robert Maddock, at NASA's Langley Research Center, for his help with running many high-fidelity simulations required for this research. And finally, thank you to my wonderful wife Meleen Grover and my family for their support and patience as I have disappeared many nights and weekends working to complete this dissertation.

1 Introduction

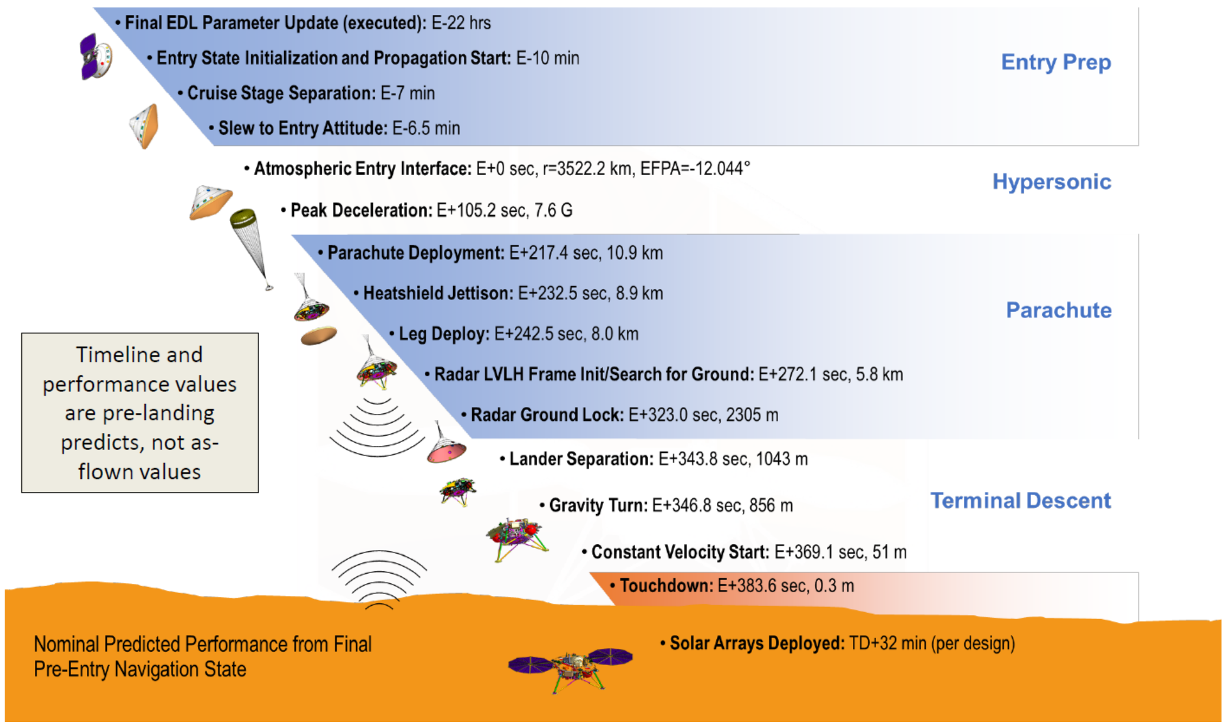
Designing a spacecraft and constructing a mission architecture that provides adequate performance margin is an essential element to successful spaceflight. One particularly challenging area of planetary spaceflight is landing on a planetary body, called the Entry, Descent and Landing (EDL) phase of a landed mission. The EDL phase is typically a very dynamic event, often executed without or with little fault protection, and with minimal system redundancy. It is often executed in an environment that can have significant variability or a low level of characterization. As such, designing EDL spacecraft systems with adequate design and performance margin in the presence of uncertainty is a challenge.

This dissertation describes the development of a rigorous, methodical approach to quantifying adequate performance margins for EDL systems. The methodology includes understanding the level of margin to be carried at different points in the lifecycle of a spaceflight project, including during the development, implementation and flight phases. It also includes tracking the fraction of margin that is in response to two types of uncertainty: epistemic (knowledge-based and reducible) and aleatory (stochastic and irreducible). The methodology is seeded by a desired performance margin sizing method implemented by NASA's InSight Project that successfully landed on Mars in November 2018. A description of this Seed Methodology is included as a starting point, followed by a detailed discussion of the evolution to a more rigorous methodology, referred to as the Evolved Methodology. Data from the development of the InSight mission, as well as InSight landing day flight data are used to assess the validity of the Evolved Methodology that is the topic of this dissertation.

1.1 Entry, Descent and Landing Systems Engineering

The design and implementation of EDL systems is fundamentally a problem of energy management, reducing the very high planet-relative kinetic energy of a spacecraft to zero at landing. A typical EDL architecture represented by the InSight architecture is shown in Figure 1-1. Shown are example elements used to dissipate the energy of a spacecraft system from entry into the atmosphere to the planet surface. The dissertation research is focused on EDL for Mars where the EDL phase of a mission is typically defined as starting at the top of the atmosphere (called Entry Interface, or EI). It includes an aeroshell for aerodynamics and aerothermodynamics, and spans the descent of a spacecraft through the atmosphere until landing. Whether entering at hyperbolic velocity or less velocity, the upper part of the EDL phase is characterized by high aerothermal heating as the very high kinetic energy of the spacecraft is dissipated by interaction with the atmosphere. The majority of the vehicle energy at entry is dissipated during hypersonic and supersonic flight. For a state-of-the-art robotic mission to Mars, as illustrated by the InSight EDL architecture in Figure 1-1, once near Mach 2 a parachute augments deceleration, further reducing the energy of the entry system. Historically, for Mars applications the final energy is removed by either a descent propulsion system or airbag system. In the case of InSight, 12 descent rocket engines were used to remove the residual energy in the last kilometer to touchdown. Other examples of Mars EDL architectures are provided in [1], [2], [3], and [4].

The discipline of EDL systems engineering is multifaceted and involves managing and optimizing performance and risk across the EDL phase of flight. For an EDL system designed for another planetary environment, it is usually not practical or possible to test an EDL system end-to-end in a flight relevant environment, and therefore high-fidelity simulation is used to model and assess EDL system performance. To provide a statistical assessment, EDL systems engineering frequently relies on Monte Carlo simulation methods to assess performance.



Note: The acronym EFPA refers to entry flight path angle, the angle with respect to the local horizonal with which the vehicle enters the atmosphere.

Figure 1-1 An example EDL architecture: InSight Mars Lander. [5]

1.2 Monte Carlo Methods for Assessing Performance

1.2.1 History of Monte Carlo Methods

Monte Carlo techniques are statistical methods of modeling and analyzing complex systems, first named as such shortly after the end of World War II. The method was suggested by mathematician Stanislaw Ulam as a rebirth of the mathematical technique of statistical sampling enabled by modern computers, and it was co-developed by John von Neumann [6]. Fellow physicist and mathematician Nicholas Metropolis suggested the name Monte Carlo to describe the method. It was based on an account of Ulam's uncle who was a gambler and enthusiastic visitor to Monte Carlo on the French Riviera, famous for games of chance. The method generally employs pseudo-random numbers to sample random variable distributions in determining modeling outcomes. Early use of Monte Carlo methods was facilitated by the first electronic computers, including the ENIAC (Electronic Numerical Integrator and Computer), to model nuclear chain reactions in nuclear weapons development. The method quickly gained in popularity and is used today in a wide array of fields, including as an important tool of EDL systems engineering.

1.2.2 Monte Carlo Methods for EDL System Performance

The technique of choice for evaluating EDL system performance is the Monte Carlo flight dynamics simulation, which consists of a high-fidelity, six-degree-of-freedom simulation that models the most important dynamics of the EDL event, and runs from prior to Entry Interface to landing. Uncertainties in input parameters that comprise the modeling of the flight dynamics are included in the modeling by treating the modeling parameters as random variables. These modeled uncertainties can be aleatory and/or epistemic, which will be discussed further in Section 1.3.1.2. In the Monte Carlo simulation, the EDL event is modeled many thousands of times (each individual modeled EDL event is often called a Monte Carlo case or run), with each case simulated with independently randomized (or sampled) values for the input parameters. NASA's InSight project used 8001-case Monte Carlo simulations to assess EDL system performance. An example of InSight Monte Carlo output is shown in Table 1-1 [7]. The results show the statistical performance of various EDL performance metrics. An important EDL systems engineering task is the identification of critical EDL performance metrics to use in constructing a vehicle and architecture that operates safely within the performance bounds levied by constraints such as material limits, sensor limits, aerodynamic and aerothermal limits, and vehicle resource limits.

The first column in Table 1-1 lists the Metric Name tracked by the InSight Project. The second column lists the Requirement value for the metric. Column 3 shows the Desired Margin (or Target Margin) against the Requirement, and the fourth column shows the Target Value which is the performance limit, or the Target Margin subtracted from the Requirement. The last four columns show the 99% or 1% performance derived from the Monte Carlo statistics for four different InSight Monte Carlo simulations, labeled MKB, MKR, MKD, and MKG (these acronyms refer to atmosphere models used by the InSight Project, where in each MK refers to the atmosphere model creators, McDunn and Kass, and the last letter refers to the type of atmosphere: B for background, R for regional dust storm, D for decaying dust storm, and G for global dust storm [7]). Other examples of Monte Carlo methods used to manage EDL system performance for NASA landed missions are provided by [8], [9], [10], and [11]. Managing the system performance margin relative to performance metrics is a fundamental task of EDL systems engineering.

1.3 Assessing and Managing Margin

Modeling of complex systems is a powerful engineering tool. Theoretically, if a model of a complex system completely and accurately captures all aspects of the modeled system behavior, there is no need for performance margin. In short, the model completely reflects and predicts reality. In engineering EDL systems, margin is important because there will always be uncertainty in modeling and therefore margin against system performance limits is required to account for the potential the actual system will perform outside the bounds of the modeled system behavior. Working in an opposite direction, requiring excessive margin to accommodate uncertainty and “unknown unknowns” can be and often is costly and can drive risk from one area to another. It is therefore important to quantify and size design performance margin as efficiently and accurately as possible, sizing sufficient margin while not specifying excessive margin.

Table 1-1 Example Monte Carlo Results. [7]

EDL Metric		Requirement Value	Desired Margin	Target Value	Units	Type	MKB	MKR	MKD	MKG
Monte Carlo										
Number of Successful Cases	>=	8001	1%	7920	[]	Size	8001	8001	8001	8000
Propellant Consumption										
Propellent Used During Hypersonic	<	1.0	1.0	0.0	%	% of Cases	0.04	0.09	0.00	0.00
Usable Propellent Remaining @ TD	>	0	1.6	1.6	kg	1%-tile	8.90	7.83	8.18	8.02
Aeroheating (Sutton-Graves)										
Heat Rate Indicator	<	51.8	0	51.8	W/cm^2	99%-tile	48.32	48.99	45.25	45.25
Total Angle of Attack at Peak Heating	<	10	0	10	deg	99%-tile	2.03	2.03	2.09	2.07
Integrated Heat Load Indicator	<	3200	0	3200	J/cm^2	99%-tile	2813.38	2962.51	2893.72	2988.13
Loads										
Peak Deceleration	<	13	0.35	12.65	Earth g's	99%-tile	8.18	8.01	7.79	7.00
Parachute Inflation Load Indicator	<	15	0	15.0	1000 lbs	99%-tile	14.19	14.18	14.40	14.24
(Alt) Parachute Inflation Load Indicator	<	15	0	15.0	1000 lbs	99%-tile	12.39	12.27	12.41	12.28
Parachute Deploy Conditions										
High Deploy Mach	<	2.3	0.115	2.185	[]	99%-tile	1.93	1.82	1.77	1.71
Low Deploy Mach	>	1.1	0.055	1.155	[]	1%-tile	1.47	1.46	1.44	1.42
High Deploy Dynamic Pressure	<	750	37.5	712.5	Pa	99%-tile	573.16	569.66	574.45	568.61
Low Deploy Dynamic Pressure	>	300	15	315	Pa	1%-tile	479.56	442.52	439.56	443.91
Total Angle of Attack	<	9.7	2.2	7.5	deg	99%-tile	6.46	2.78	2.48	2.50
Low Altitude at Parachute Deploy						1%-tile	7620.07	7148.79	6736.60	6322.34
High Altitude at Parachute Deploy						99%-tile	14898.97	14616.59	13758.71	13242.06
Heatshield Separation										
Attitude Rate Amplitude	<	100	15	85	deg/s	99%-tile	91.74	74.63	73.54	72.28
Mach	<	0.8	0	0.8	[]	99%-tile	0.67	0.63	0.61	0.60
Leg Deploy										
Attitude Rate Amplitude	<	100	15	85	deg/s	99%-tile	73.69	63.37	62.05	60.88
MRD Init										
High Altitude	<	10051.0	631.5	9419.5	m	99%-tile	8931.20	9720.92	9326.18	8919.23
Mean Altitude						Mean	6236.60	6532.36	6408.74	6192.61
Low Altitude	>	2496.9	1263.0	3759.9	m	1%-tile	4015.13	3828.04	3584.17	3301.76
MRD Init Altitude Spread (99%-1%)	<	7554.1	1894.5	5659.6	m	Nominal	4916.07	5892.89	5742.01	5617.47
Number of cases with MRD Init @ 35 sec						Size	796	1276	1628	2389
Low Time from Chute Deploy to MRD Init						1%-tile	35.09	35.09	35.09	35.09
High Time from Chute Deploy to MRD Init						99%-tile	89.29	60.29	53.59	51.69
Lander Separation										
High Altitude						99%-tile	1419.13	1427.29	1426.98	1429.95
Low Altitude						1%-tile	937.05	935.88	935.33	934.98
Attitude Rate Amplitude	<	60	5.3	54.7	deg/s	99%-tile	41.41	37.80	39.95	37.52
Landing Accuracy										
99%-tile Landing Ellipse Major Axis	<	150	0	150	km	Nominal	110.94	118.50	118.50	119.54
99%-tile Landing Ellipse Minor Axis	<	35	0	35	km	Nominal	24.14	24.53	24.33	25.05
Touchdown Conditions										
High Vertical Velocity (wrt ellipsoid)	<	3.4	0.1	3.3	m/s	99%-tile	2.82	2.83	2.83	2.84
Low Vertical Velocity (wrt ellipsoid)	>	1.4	0.1	1.5	m/s	1%-tile	1.96	1.93	1.94	1.93
Horizontal Velocity (wrt ellipsoid)	<	1.4	0.08	1.32	m/s	99%-tile	0.60	0.82	0.75	0.82
Attitude Rate (RSS pitch/yaw)	<	11.3	1.64	9.66	deg/s	99%-tile	6.99	7.13	7.17	7.29
Overall Probability of Safe Landing					%	Mean	99.00	99.09	98.96	98.65

Note: The columns MKB, MKR, MKD and MKG refer to Monte Carlos using the four types of atmosphere models used by the InSight Project.

1.3.1 Existing Methods for Characterizing & Managing Uncertainty

Methods for characterizing uncertainty fall generally under the analysis category of Uncertainty Quantification, or UQ. Vincent Romero of Sandia National Laboratories [12] defines UQ as

Uncertainty Quantification is the process of characterizing all significant uncertainties in a model, simulation, or experiment and of quantifying their effect on computed or experimental results.

Uncertainty Quantification techniques are used in a variety of spaceflight applications, from conceptual design to flight reconstruction. One example of UQ applied to conceptual design is provided by Noyes*, et al*, [13] in a discussion of UQ applied to modeling of an early concept Mars Ascent Vehicle. An example of UQ applied to flight reconstruction is provide by Dutta, *et al*, in *Uncertainty Quantification for Mars Entry, Descent, and Landing Reconstruction Using Adaptive Filtering* [14]. Techniques of UQ can also be applied to optimizing uncertainty in EDL systems to yield desired performance, as was investigated by Olds & Way in *Uncertainty Optimization Applied to the Monte Carlo Analysis of Planetary Entry Trajectories* [15].

In addition to quantifying uncertainty, there exist methods for managing uncertainty and performance margin, one in particular being similar to the Seed Methodology called Quantification of Margin and Uncertainty, or QMU [16].

1.3.1.1 Sandia National Laboratory & Quantification of Margin & Uncertainty

Making decisions in the presence of uncertainty is a reality faced in a broad range of disciplines, from financial planning to nuclear engineering to aerospace engineering. In the case of nuclear weapons development and management, the nuclear engineering community faces a similar challenge shared with EDL systems engineering: the inability to do full systems testing resulting in a reliance on high-fidelity simulation of system behavior. In the case of EDL systems engineering, the inability to do full systems tests is driven mainly by the lack of the correct environment (correct atmosphere, correct gravity field) in which to conduct the test. For nuclear weapons development, it is the existence of nuclear test ban treaties that restricts full system tests. To address this challenge, the National Nuclear Security Administration (NNSA), in conjunction with the Sandia National Laboratories (SNL) and Los Alamos National Laboratory, developed a methodology, called Quantification of Margins and Uncertainty, in 2001 [16]. QMU was developed for nuclear weapons stockpile decision-making, and is described by Pilch, *et al* as "a decision-support methodology for complex technical decisions centering on performance thresholds and associated margins for engineered systems that are made under conditions of uncertainty." [17] As described by Pilch, QMU concerns itself with the following:

- Element 1: Identification and specification of performance threshold(s)
- Element 2: Identification and specification of associated performance margin(s), that is, measure(s) of exceeding performance thresholds
- Element 3: Quantified uncertainty in threshold and margin specifications

Because QMU has been developed and inserted into nuclear weapons management critical to national security, the methodology has received a high level of scrutiny, including a positive review by the National Research Council (requested by Congress) that concluded:

> "QMU *is a sound and valuable framework that helps the national security laboratories carry out the Department of Energy's responsibility to maintain the nation's nuclear weapons capabilities. ... The national security laboratories and NNSA should expand their use of QMU while continuing to develop, improve, and increase application of the methodology."* [18]

At its core, QMU provides a framework for assessing the confidence that a nuclear weapon system will function properly. As part of the assessment, system performance "gates" are identified that are required to be passed or met in the sequence of events that leads a weapon system to a nuclear detonation. The level of margin and uncertainty in a specific performance gate is assessed to provide a measure of confidence that a gate will be successfully achieved by a design. The QMU concepts of performance margin and uncertainty are schematically illustrated in Figure 1-2, called a cliff chart. The thick shaded curve is a conceptual representation of system performance for a performance gate. A government science and technology advisor body named for Jason in Greek mythology [19], JASON provides a useful performance gate example in a review of QMU. The example used is an internal combustion engine, where the System Performance is the energy released by gasoline combustion in an engine cylinder, and the Metric is the spark ignition energy required to initiate combustion. In this example, a threshold spark energy, $Y_{p,min}$, is required to achieve combustion. In a particular engine, an Operating Range of ignition spark energy is designed with the lower end of ignition spark energy represented by Y_{BE}. There is both an uncertainty, U_1, in the lower end of the design Operating Range, and an uncertainty in the threshold spark ignition energy, U_2. The total Margin, M, is the difference between Y_{BE}, and $Y_{p,min}$, or

$$M = Y_{BE} - Y_{p,min} \tag{1-1}$$

The total uncertainty is given by

$$U = U_1 + U_2 \tag{1-2}$$

The QMU methodology then uses the simple ratio of margin to uncertainty, M/U, to compute a scoring of the confidence that a particular gate will be met. At a minimum, M/U > 1 is required for confidence, and depending on the criticality of the performance gate, M/U >> 1 may be required. The Evolved Methodology described in this dissertation shares elements with Sandia's QMU methodology, but is tailored to established EDL analysis techniques.

1.3.1.2 Epistemic and Aleatory Uncertainty

Uncertainty is commonly classified into two categories: epistemic and aleatory. Epistemic

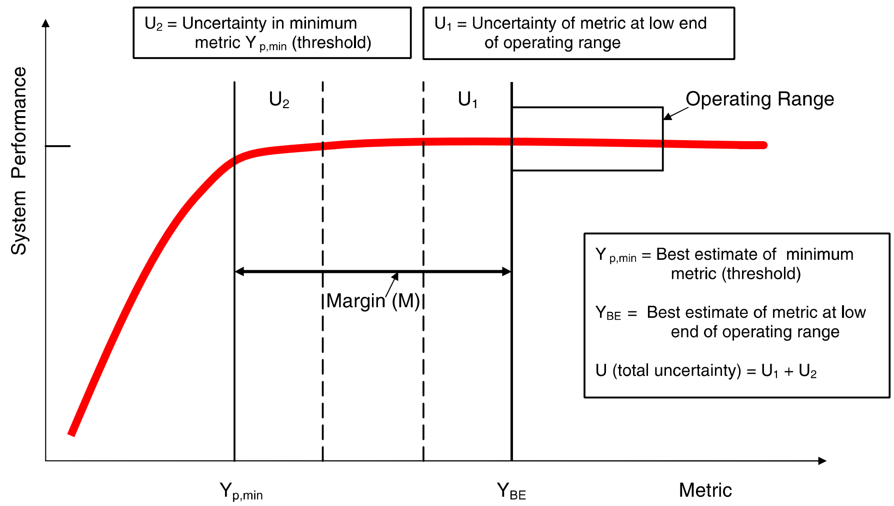


Figure 1-2 Generalized QMU cliff chart. [17]

uncertainty is uncertainty due to lack of knowledge and is reducible uncertainty. In the case of EDL, an example of epistemic uncertainty is spacecraft entry mass, which has uncertainty at the beginning of a project but has very little uncertainty on the day of landing. Aleatory uncertainty is sometimes referred to "stochastic variability" and is uncertainty due to random variation, such as the atmospheric density variation in the case of EDL. QMU has techniques for handling aleatory and epistemic uncertainty, and the Evolved Methodology described in this dissertation likewise will incorporate epistemic and aleatory uncertainty. Identifying what forms of uncertainty are present is important because epistemic uncertainty, theoretically, can be reduced while aleatory uncertainty is irreducible. Identification of epistemic uncertainty early in a project can guide investments to reduce uncertainty, and is an important component of managing performance margin over the lifecycle of a project.

1.3.1 Seed Methodology

A margin methodology developed at NASA's Jet Propulsion Laboratory (JPL) by the author and collaborators E. David Skulsky and Alejandro Miguel San Martín serves as the seed of this dissertation. The methodology and terminology were developed independent of the QMU method described in Section 1.3.1.1, but striking similarities to QMU serve to validate the Seed Methodology. Described in the following subsections, the Seed Methodology is documented in the internal JPL white paper *InSight Entry, Descent and Landing Margins Policy* [20].

1.3.1.1 Selecting Critical Metrics

While many metrics are assessed during EDL systems design, the initial JPL Seed Methodology takes an approach of identifying a subset of EDL metrics called Critical Metrics. Critical Metrics are defined as metrics that provide visibility into significant design performance goals and thus protect the most significant objectives of the design effort and flight. It is the Critical Metrics to which the margin methodology is applied, and the term Critical Metric will be used throughout the remainder of the methodology discussion.

1.3.1.2 Margin Sizing Definitions

For each Critical Metric, a margin methodology is applied to establish a Target Performance threshold relative to the performance limit determined for the metric. The Target Performance threshold is the Target Limit, and the performance limit is called the Metric Limit. The difference between the Metric Limit and Target Limit is the Target Margin, as shown in Figure 1-3. The fundamental goal of the Seed Methodology is to quantify the Target Margin as rigorously as possible, which is the desired minimum difference between the Metric Limit and the actual system performance. As has been applied to date, performance in the green region on the left end of the performance axis is called Target Performance, which is the area of preferred performance. Performance in the Target Margin region is deemed Acceptable Performance, and performance that exceeds the Metric Limit, the red region, is deemed Unacceptable Performance. Figure 1-4 provides a comparison of the terminology from the Seed Methodology with the QMU terminology described in Section 1.3.1.1.

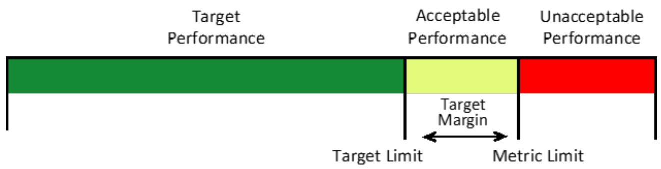


Figure 1-3 Seed Methodology terminology. [20]

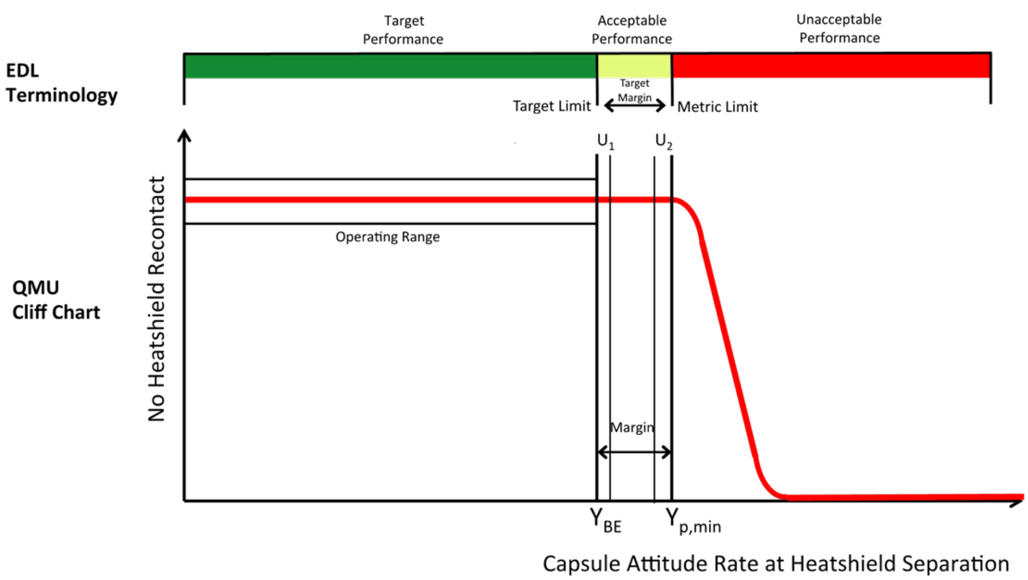


Figure 1-4 Comparison of QMU and Seed Method terminologies.

1.3.1.3 Three Methods of Sizing Target Margin

The Target Margin sizing Seed Methodology consists of three methods for determining the size of the desired Target Margin. While all three methods are described in this section, the research documented in this dissertation focuses on development of Method 2.

1.3.1.3.1 Method 1: Margin Based on Previous Mission

As currently implemented, Method 1 for determining the desired size of a Target Margin is based on past mission implementations. The method calls for adopting the same Target Margin carried by a previous high-similarity mission, or alternately it calls for adopting a desired Target Margin based on the EDL system performance seen in a previous mission's flight. Method 1 allows for historical precedence as a method for determining a desired Target Margin.

1.3.1.3.2 Method 2: Margin Based on System Performance Distribution

The Seed Methodology Method 2 approach for sizing Target Margin is based on the performance distribution of an EDL Critical Metric and offers a method that is based on Monte Carlo-derived span of EDL performance. In this method, the value for one standard deviation is determined from the Monte Carlo performance distribution for the metric, assuming the resulting distribution in performance is near Gaussian (or normal). The Target Margin for the metric is then sized based on one standard deviation, as recognition the performance distribution is a representation of the span of expected performance. For a metric whose performance is driven in simulation by modeling that has higher confidence, the margin is set based on a half-standard deviation. Alternately, the margin of a metric whose performance in simulation is driven by modeling with lower confidence is sized equivalent to a full standard deviation. Accurately characterizing the confidence in modeling system behavior becomes key for this method because the more confidence there is in modeling, the smaller the Target Margin required. Also, key in applying this method is a clear understanding of what simulation models are most affecting the system performance of a particular EDL Critical Metric. Figure 1-5 depicts a graphical representation of the Method 2 sizing methodology. As mentioned earlier, a more rigorous development of Method 2 is the key focus of the research outlined in this dissertation.

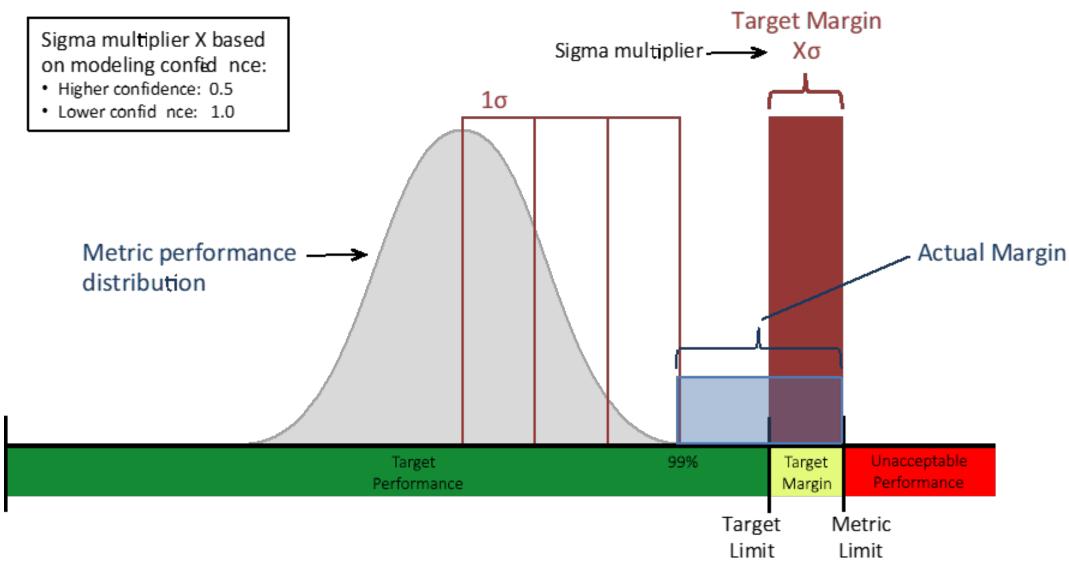


Figure 1-5 A graphical representation of Method 2 Target Margin sizing process. [20]

1.3.1.3.3 Method 3: Margin Based on Engineering Judgment

In cases where engineering judgment and discretion come into play, Method 3 is employed to set Target Margin to a fixed value. In this method, simple engineering judgment, based on experience and in-depth knowledge of the system, is used to set a desired Target Margin to provide a prudent and reasonable margin against the Metric Limit.

The goal of the research conducted in this dissertation, as described in Section 2, is to expand and develop the Seed Methodology Method 2 into a more rigorous method for determining Target Margin for EDL Critical Metrics. Research objectives and validation approach follow in the next section.

2 Research Objectives

The initial JPL Seed Methodology described in Section 1.3.2 for determining entry, descent and landing (EDL) Target Margin was well received within the Jet Propulsion Laboratory (JPL) EDL community. However, in the application of the methodology, a number of issues were identified that serve as foundation pieces for the objectives of the dissertation research. These issues led to the following research objectives:

Objective 1: Input Model Influence

Develop a more rigorous method for identifying simulation models that most influence performance distributions of Critical Metrics and quantify their level of influence.

Objective 2: Mapping Input Model Confidence to Target Margin

Develop a more rigorous, higher resolution technique to map input model confidence to Target Margin sizing via metric performance distribution.

Objective 3: Technique for Epistemic and Aleatory Uncertainty

Develop a method to handle epistemic and aleatory uncertainty appropriately within the methodology.

Objective 4: Evolvable Methodology

Develop an evolvable methodology that can be applied throughout the lifecycle of a project.

The research and resulting Evolved Methodology described in the dissertation address each of these four research objectives.

2.1 Validation Approach

Validation of the Evolved Methodology is key to establishing its legitimacy for sizing Target Margin. The validation approach taken in this dissertation will use the following:

- Comparison to existing Seed Methodology, discussed in Section 11.7
- Validation using InSight EDL flight data, which is the subject of Section 11.8

3 Fundamental Reformulation of Methodology

As outlined in the previous section, the Seed Methodology uses the distribution of Critical Metric performance resulting from an entry, descent and landing (EDL) Monte Carlo analysis to size Target Margin. Modeling confidence is used to determine what fraction of a standard deviation is used (assuming the Critical Metric performance distribution approximates a normal distribution) in sizing Target Margin. During the course of research and formulation of the Evolved Methodology, addressing Objective 1, identification of input models of highest influence, led to a realization the process of using fractions of uncertainty distributions could be applied to input distributions instead of output performance distributions. Focusing on inputs provides a finer resolution application of the basic principle of combining modeling confidence with uncertainty distribution to obtain a quantified Target Margin.

3.1 Change to Emphasis on Monte Carlo Inputs

A high-level block diagram representing the original Seed Methodology is shown in Figure 3-1. As illustrated in the figure, the fundamental process for quantifying Target Margin begins with the Monte Carlo Simulation, the statistical results of which are used to size Target

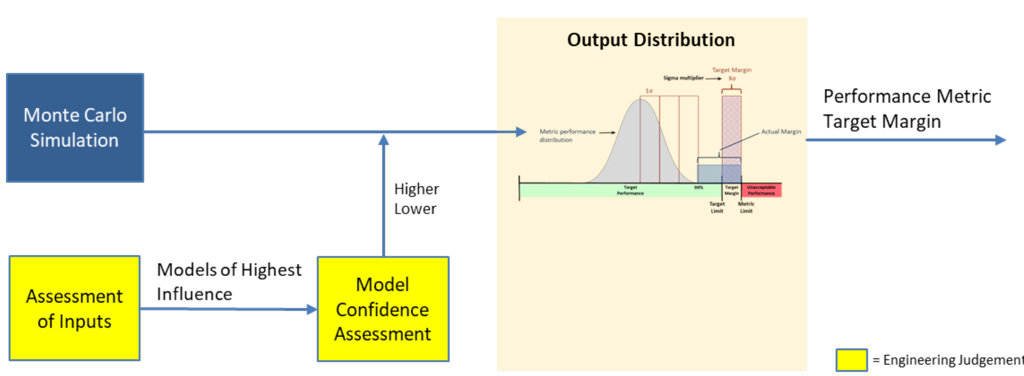


Figure 3-1 High-level representation of Seed Methodology.

Margin. As indicated by yellow shading, engineering judgement is used to identify models of highest influence on a Critical Metric, and again, using engineering judgement, a model confidence rating is applied to determine what fraction of the performance distribution is used for Target Margin for that Critical Metric.

As can be seen in Figure 3-2, a high-level block diagram of the reformulated Evolved Methodology shows that the process of using a fraction of a distribution is applied to the modeling input distributions rather than the performance distribution. This is the fundamental change from the original process. Fractions of the input distributions are then combined via a regression model developed via a Design of Experiments (DOE) process to develop a quantified Target Margin.

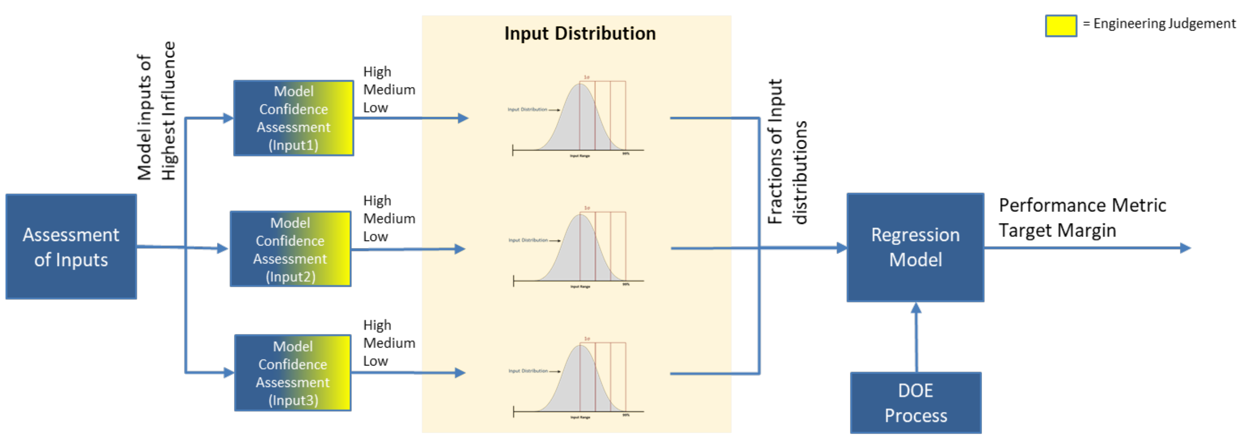


Figure 3-2 High-level representation of the Evolved Methodology.

In the Figure 3-1 and Figure 3-2 block diagrams, yellow shading represents areas in the methodology where engineering judgement is applied. As can be seen, the Evolved Methodology significantly reduces the use of engineering judgement and thus moves the methodology toward a more rigorous, stronger foundation. The following section contains a more detailed discussion of the Evolved Methodology.

3.2 Process Overview

A detailed block diagram of the Evolved Methodology is shown in Figure 3-3. The methodology is introduced in this section via a brief description of each step in the process (or block in the diagram). A more thorough discussion of each step is contained in Sections 4 – 9.

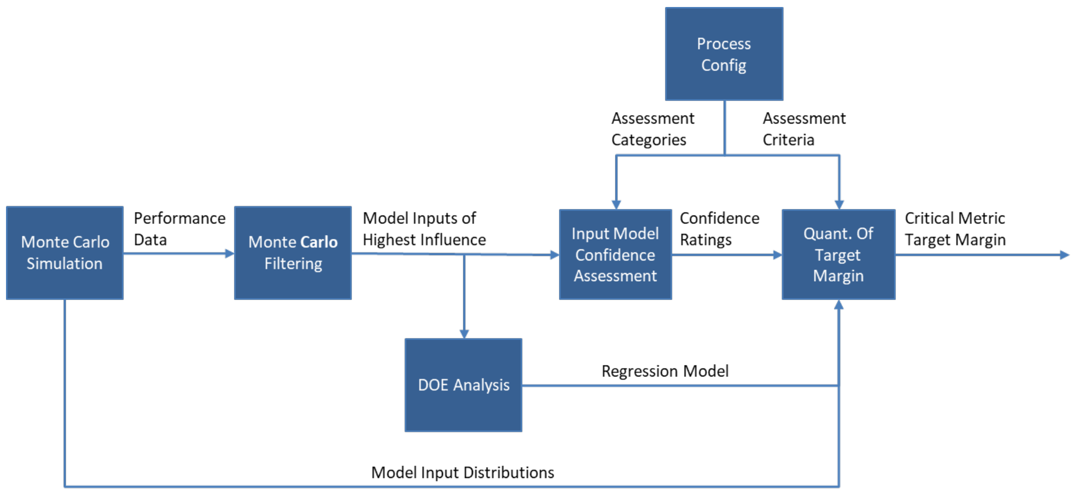


Figure 3-3 Detailed Evolved Methodology process diagram.

3.2.1 Monte Carlo Simulation

The process of generating a quantification of Target Margin begins with a Monte Carlo simulation of the EDL architecture. The Monte Carlo simulation generates statistical performance data for each Critical Metric, which is comprised of the performance output for each of the single simulation runs that comprise the Monte Carlo simulation. The performance data is passed into the next step, which is Monte Carlo filtering of the data. A more detailed discussion of the Monte Carlo simulation step is presented in Section 4.

3.2.2 Monte Carlo Filtering

Monte Carlo Filtering is the process of determining the Monte Carlo inputs of strongest correlation to a given Critical Metric. This process is used in the Evolved Methodology as an initial filtering to identify inputs that have the strongest influence on a Critical Metric, called Model Inputs of Highest Influence. Monte Carlo Filtering in the methodology is accomplished by two filtering approaches: correlation coefficient and mutual information. The Monte Carlo Filtering step identifies the input models of strongest correlation that seed two succeeding steps in the Evolved Methodology: a DOE Analysis step, and an Input Model Confidence Assessment. Establishing strength of correlation does not measure strength of influence, which is accomplished via DOE regression modeling. While Monte Carlo Filtering establishes correlation and not necessarily strength of influence, the terms correlation and influence will be used interchangeable at times in this dissertation. A more detailed discussion of the Monte Carlo Filtering step is presented in Section 5.

3.2.3 Design of Experiments and Regression Modeling

As can be seen in Figure 3-3, the Model Inputs of Highest Influence resulting from Monte Carlo Filtering seed a DOE Analysis that results in a regression model. The top influential model inputs per Critical Metric are used in a DOE analysis that results in a normalized second-order polynomial regression model. The regression model serves two purposes:

- The coefficients of the normalized regression model terms provide a quantification of the level of influence each model input has on a Critical Metric
- The regression model serves as an analytical method by which to map the model input confidence assessment to the sizing of Target Margin

A more detailed discussion of the DOE Analysis and resulting regression model is provided in Section 6.

3.2.4 Process Configuration

As part of the detailed Evolved Methodology, Figure 3-3 shows a Process Configuration step. This step provides the ability to configure a number of process parameters needed to complete the quantification of Target Margin. These parameters include the Critical Metrics being assessed, the Assessment Categories under which to assess modeling confidence, the project epochs to be included in the assessment, and the Assessment Criteria for each level of confidence, or Confidence Rating. These parameters can evolve and change across the lifecycle of a project. A more detailed discussion of the Process Configuration is presented in Section 7.

3.2.5 Input Model Confidence Assessment

For each Model Inputs of Highest Influence, there is an assessment of the modeling confidence by the categories identified during the Process Configuration step. As shown in Figure 3-3, this step is called the Input Model Confidence Assessment. The modeling confidence assessment is based on supporting system engineering evidence. From the confidence assessment, Confidence Scores are determined and feed the Quantification of Target Margin step. Section 8 provides more details on the assessment of modeling confidence.

3.2.6 Quantification of Target Margin

As shown in Figure 3-3, the final step in the Evolved Methodology is the Quantification of Target Margin. This is accomplished by mapping the Confidence Scores from the Input Model Confidence Assessment to Target Margin by using the Confidence Scores as inputs into the DOE regression model to provide a sizing of Target Margin. A detailed discussion of the Quantification of Target Margin step is provided in Section 9.

3.3 TMST: Implementation as a Software Tool

As part of this dissertation, the Evolved Methodology outlined in this section was implemented in a MATLAB-based software tool called Target Margin Sizing Tool, or TMST. The tool takes as input the results of Monte Carlo Filtering and regression model analysis and provides a methodical structure for mapping input model confidence to quantified Target Margin. The Startup Window for TMST is shown in Figure 3-4, and the Assessment Main Window is shown in Figure 3-5. The Assessment Main Window is launched by pressing the Run Analysis button in the green shaded region in Startup Window. The features and function of TMST will be introduced in each relevant section of this dissertation, and the tool will be used to perform the Target Margin assessments detailed in Section 11 and Section 12. A MATLAB code listing for TMST is provided in Appendix A – Appendix C.

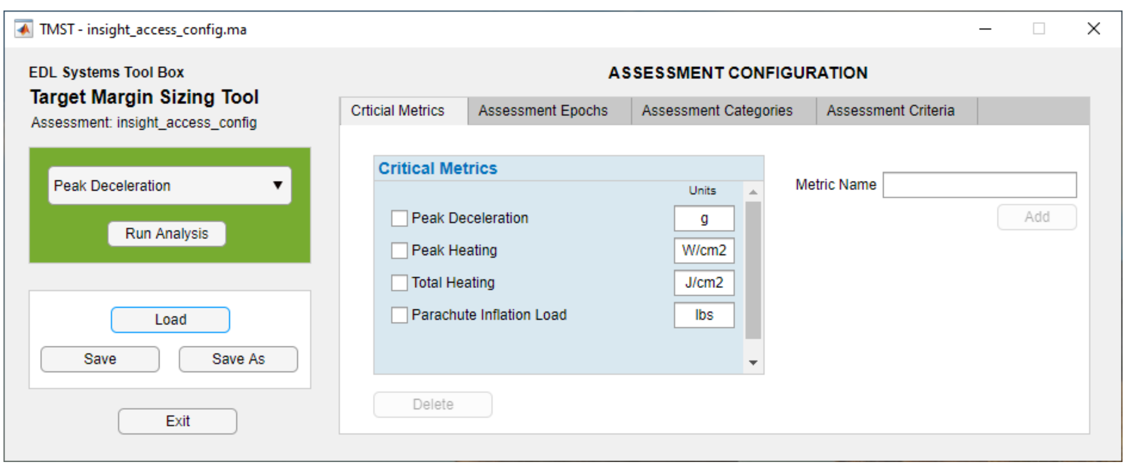


Figure 3-4 Target Margin Sizing Tool Startup Window.

(This space intentionally left blank)

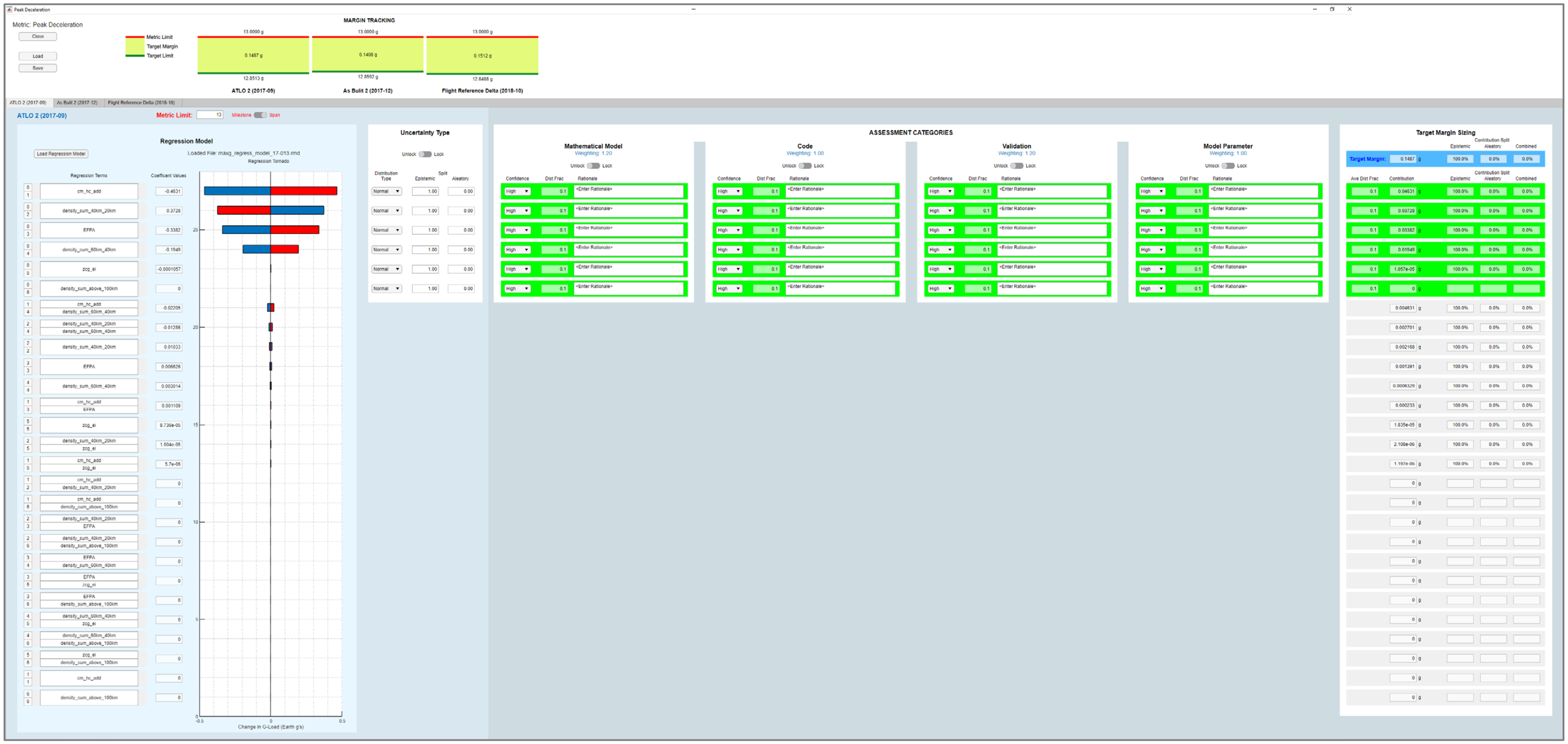


Figure 3-5 Target Margin Sizing Tool Assessment Main Window.

(This space intentionally left blank)

4 Monte Carlo Simulation

This section describes in more detail the entry, descent and landing (EDL) Monte Carlo simulation, which is the first step in the margin methodology process, as highlighted by the dark blue shaded box in Figure 4-1. Additionally, this section discusses in greater detail uncertainty in Monte Carlo modeling inputs and forms of Monte Carlo output distributions.

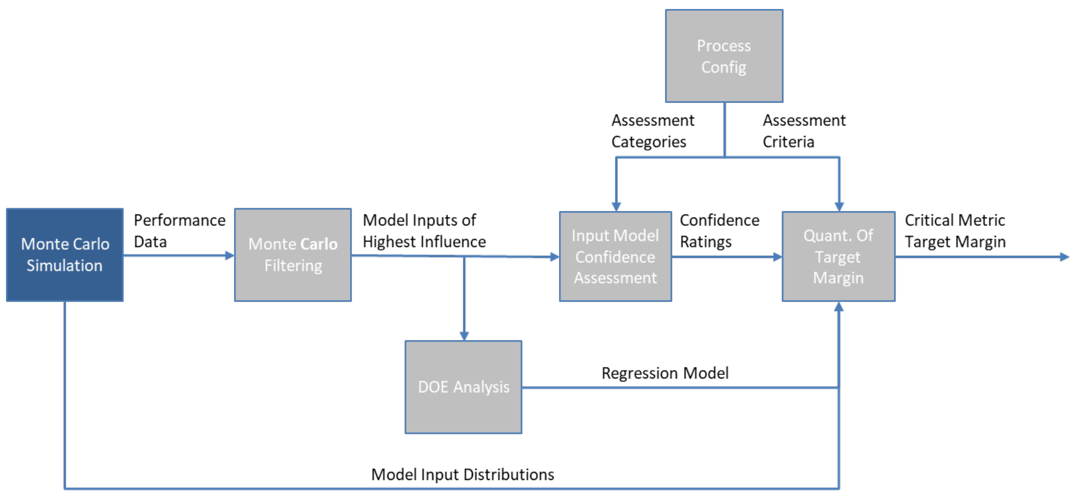


Figure 4-1 Detailed Evolved Methodology process diagram: Monte Carlo Simulation.

4.1 EDL High-Fidelity Monte Carlo Simulations

As described in Section 1.2.2, Monte Carlo simulation is a fundamental tool for EDL systems engineering. A detailed schematic shown in Figure 4-2 illustrates how a typical EDL Monte Carlo simulation is composed. There are a variety of modeling inputs shown in the schematic that fall into the following categories:

- Vehicle Initial State
- Vehicle Models
    - Aerodynamics

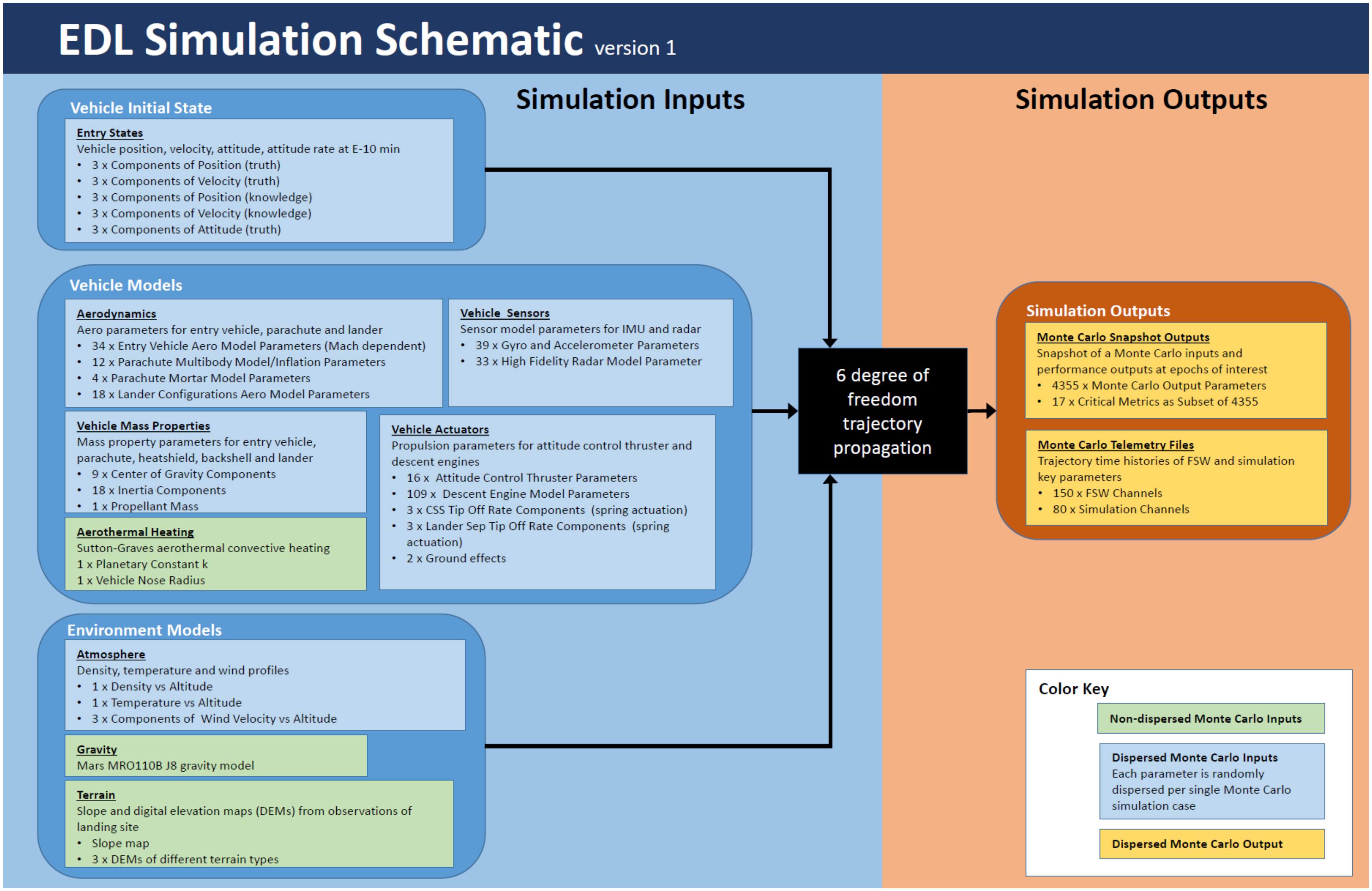


Figure 4-2 High-fidelity Monte Carlo simulation block diagram.

    - Vehicle Mass Properties
    - Aeroshell Heating
    - Vehicle Sensors
    - Vehicle Actuators
- Environment Models
    - Atmosphere
    - Gravity
    - Terrain

The Color Key shows inputs shaded green are non-dispersed, meaning they remain a fixed value from Monte Carlo case to Monte Carlo case, and inputs in blue are dispersed, meaning their values change from Monte Carlo case to Monte Carlo case based on an input distribution. As can be seen, there can be many hundred or many thousands of inputs in a high-fidelity EDL Monte Carlo simulation. The inputs initialize a six degree-of-freedom (three translational and three rotational) simulation of the EDL event that numerically propagates an EDL trajectory based on modeled flight dynamics.

4.2 Uncertainty in Modeling

Monte Carlo analysis provides a method for handling uncertainty while providing a statistical assessment of outcomes. Uncertain inputs in a Monte Carlo model, which may be uncertain because of measurement uncertainty, knowledge uncertainty or random variation, are treated as random variables that are often represented by a normal (Gaussian) distribution or uniform distribution, depending which distribution best represents the random nature of the variable. The span of a distribution is often determined by experiment, test or flight data combined with engineering judgement. For each Monte Carlo case, the input variable distribution representing the input uncertainty is randomly sampled to provide an input value for that case. The uncertainty distribution for each random input variable is sampled each time a Monte Carlo case is initialized.

4.2.1 Input Uncertainty Distributions

As mentioned in the previous section, uncertainty in the value of an input variable is represented by an uncertainty distribution, or probability density function (PDF). While any appropriate representative distribution may be utilized, the two most common uncertainty distributions utilized in EDL Monte Carlo simulation are the normal or Gaussian distribution (here forward referred to as normal), and the uniform distribution. Example distributions from the InSight Project Monte Carlo EDL simulation are shown in Figure 4-3 and Figure 4-4. Figure 4-3 shows the distribution for entry flight path angle, which has a mean of -12.0° and a 3-sigma value of 0.21°. An example of variable uncertainty represented by a uniform distribution is shown in Figure 4-4, which is the input distribution for hypersonic aerodynamics pitching moment derivative *Cmq* adder, where the subscript *m* refers to moment, and *q* is the pitch rate about the pitch axis. An adder is an addition term in the linearized expression for dispersed *Cmq*, and is explained in more detail in Section 11.2.1. The *Cmq* adder distribution shown in Figure 4-4 is a uniform distribution from -1.0 to 1.0. Generally, the type of distribution, whether normal or uniform, is determined by experimental data, where a uniform input distribution with even weighting across the span of the distribution represents greater input uncertainty relative to a normal input distribution that is weighted more heavily around the mean.

4.2.2 Classification of Modeling Uncertainty: Epistemic and Aleatory

As introduced in Section 1.3.1.2, modeling uncertainty can generally be classified as

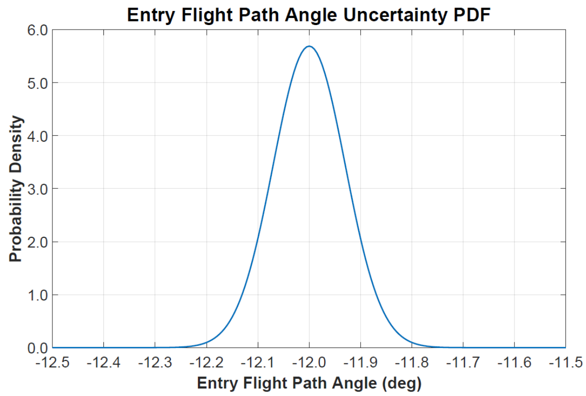


Figure 4-3 Example normal input distribution.

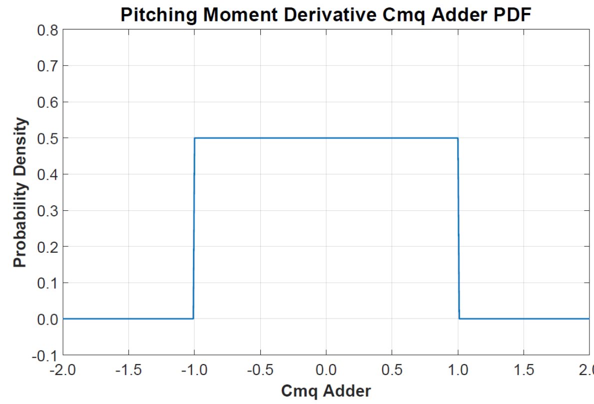


Figure 4-4 Example uniform input distribution.

epistemic or aleatory. Identifying input uncertainty as either epistemic or aleatory is important to assessing the need for current and future performance margin. Because epistemic uncertainty has the potential to be reduced, understanding whether uncertainty is epistemic or aleatory provides insight into where uncertainty may be reduced and how the need for performance margin may be minimized. Because elements of the uncertainty in the value of an input variable may be both epistemic and aleatory (both reducible through better knowledge while also having uncertainty that is truly random and not reducible), input variables may be and often are a blend. The relative amount of epistemic and aleatory uncertainty contained in the uncertainty of an input variable can also change with time through the life cycle of a project. Figure 4-5 illustrates the blended nature of epistemic and aleatory uncertainty, representing the relative amounts of epistemic and aleatory uncertainty in EDL parameters at the beginning of a project. Part of the Target Margin Evolved Methodology process is understanding the level of epistemic and aleatory uncertainty currently present in input variable uncertainty, as well as forecasting epistemic and aleatory uncertainty as an EDL system matures and transitions into flight. The methodology presented in this dissertation will assess to what degree input uncertainty is epistemic and/or aleatory. Further discussion of epistemic and aleatory uncertainty is in Section 9 and Section 10.

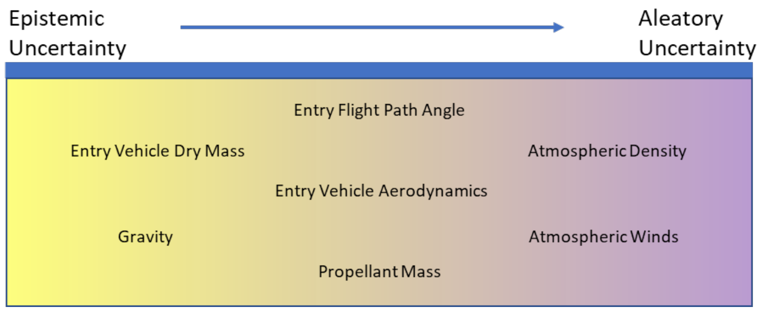


Figure 4-5 Epistemic/aleatory uncertainty spectrum.

4.3 Monte Carlo Performance Output

As alluded to in Section 1.3.2 and Section 3.1, the outputs of Monte Carlo simulations are a statistical set of all outcomes from the individual runs of the Monte Carlo. For each performance metric being used to characterize EDL system performance, the set of output performance outcomes creates a Monte Carlo output distribution. Figure 4-6 and Figure 4-7 show example histograms of Critical Metric Peak Heat Rate and Total Heating from an InSight Project EDL Monte Carlo simulation. The dashed red line in both figures indicates the 99th percentile performance from the Monte Carlo, meaning 99% of the Monte Carlo cases had performance at or to the left of the dashed red line. Generally, EDL performance margin is measured against the 99th

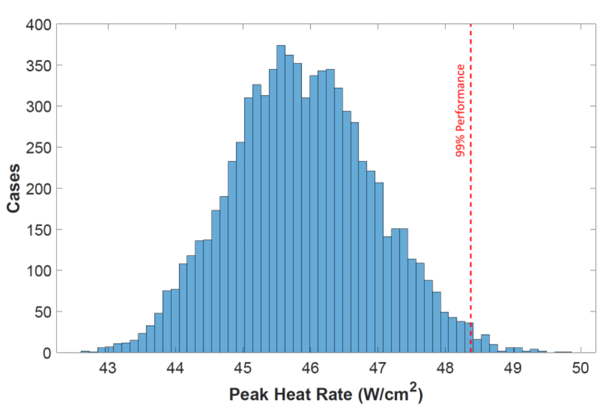


Figure 4-6 Peak heat rate performance output.

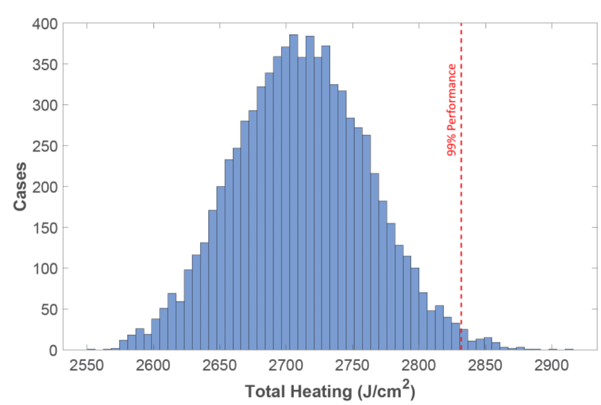


Figure 4-7 Total heating performance output.

percentile performance if performance is to not exceed a specified value, or 1st percentile performance if performance is specified not to drop below a specified value. For some Critical Metrics, both 1st percentile and 99th percentile is monitored to ensure performance is within high and low bounds. The focus of this dissertation is development of a method to quantify how much margin is desired against the 99th percentile performance and the metric limit.

5 Monte Carlo Filtering

Once Monte Carlo entry descent and landing (EDL) analysis is complete, the next step in the process is Monte Carlo Filtering, as shown in Figure 5-1. Monte Carlo Filtering provides an initial identification of input parameters of greatest correlation to Critical Metrics. Two established filtering techniques were chosen to identify parameters of greatest correlation: correlation analysis by computing the correlation coefficient, and mutual information methods by computing the mutual information parameter. In this way, a quantitative measure of correlation between input and output is achieved and is used to provide an initial filtering for Model Inputs of Highest Influence. For both techniques, a Python-based internal Jet Propulsion Laboratory (JPL) tool called Monte Carlo Processing, or MCP, was used to perform the analysis.

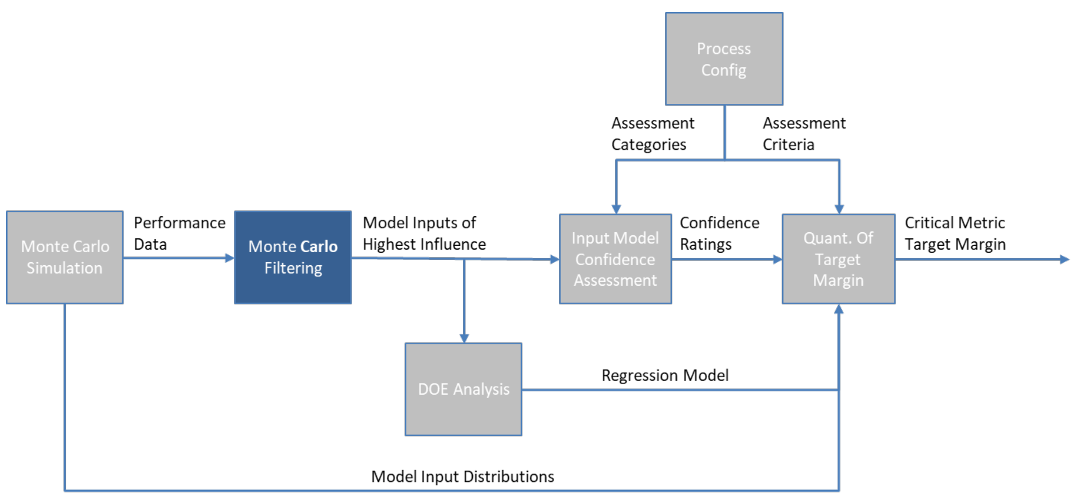


Figure 5-1 Detailed Evolved Methodology process diagram: Monte Carlo Filtering.

5.1 Correlation Coefficient

The correlation coefficient measures the strength of a linear relationship between input parameters and Critical Metrics. JPL's MCP tool employs the Python scientific computing library *numpy* (*num*erical *py*thon) to compute the correlation coefficient via the Pearson product-moment method, which uses a linear best fit through bivariate data points [21]. For input parameter set X and Critical Metric set Y, the correlation coefficient for X and Y is given by the formula

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad [22] \qquad (5\text{-}1)$$

where cov(X,Y) is the covariance[1] of X and Y, and σ_X is the standard deviation of X and σ_Y is the standard deviation of Y [22]. Employing the correlation coefficient provides an initial filtering of Monte Carlo input parameters of highest influence for each Critical Metric.

5.2 Mutual Information

Mutual information was selected as a second check on correlation between input parameters and Critical Metrics. It measures the amount of shared information between data sets, so its usefulness is to detect higher-order relationships between data sets not detected by the correlation coefficient. If X represents the set of inputs for a parameter for an EDL Monte Carlo simulation, and Y is the corresponding set of results for a Critical Metric, then mutual information for the two sets of random variables is given by

$$I(X,Y) = \iint dxdy\mu(x,y)log\frac{\mu(x,y)}{\mu_x(x)\mu_y(y)} \quad [23] \qquad (5\text{-}2)$$

where μ is the joint probability density[2] of x and y, and μ_x is the marginal probability density of X and μ_y is the marginal probability density of Y [23].

In application, the JPL MCP tool uses a Python library called *sklearn* to compute mutual information, which relies on nonparametric methods based on entropy estimation from k-nearest

[1] Covariance is a measure of the joint variability of two random variables and is the expected value of the product of the deviations of two variables from their respective means. [32].
[2] Joint probability density is a probability density function that defines the behavior of multiple random variables simultaneously. [32]

neighbors. A discussion of this method is found in the paper *Estimating Mutual Information* [23]. In applying mutual information to Monte Carlo Filtering, the relative magnitude of the mutual information parameter is used to filter for significant relationships between input parameters and Critical Metrics that are not revealed by the correlation coefficient.

5.3 Challenge of Parameterizing an Atmospheric Density Profile

In a landing on Mars, the atmosphere plays a significant role in the performance of an EDL system. The atmosphere of Mars is fundamentally a continuous distribution of atmospheric density, temperature and pressure, and therefore as an input into a simulation, it is difficult to represent the atmosphere in terms of input parameters. Thus, one of the challenges of correlating Monte Carlo model inputs to Monte Carlo Critical Metrics is representing a continuous density profile parametrically, so that a Critical Metric can be correlated to a parameter or parameters representing an atmosphere profile. For this research, a simple approach was taken by summing density values are each data step within evenly distributed altitude bands. This provides an indicator of the bulk density for each altitude band and thus an indicator of how the density in the atmosphere is distributed, and how higher or lower overall destiny in a given band affects or influences a Critical Metric.

For each case in the Monte Carlo simulation, the density was divided into the following altitude bands:

- 0 km – 20 km
- 20 km – 40 km
- 40 km – 60 km
- 60 km – 80 km
- 80 km – 100 km
- Above 100 km

Two methods for summing the density were considered. The first method sums density along the trajectory in the time domain as the vehicle encounters the atmosphere. The density parameter for a given atmosphere band is given by

$$Density\ Parameter = \sum_{i=1}^{N} d_i * \Delta t \tag{5-3}$$

where N is the number of data time steps along the trajectory in the altitude band, d_i is the value of density at time step i, and Δt is the data time step magnitude. The second method is independent of the trajectory flown and sums density along vertical altitude in the altitude band in the distance domain. For the second method, the density parameter for a given band is given by

$$Density\ Parameter = \sum_{i=1}^{N} d_i * \Delta a \tag{5-4}$$

where N is the number of data steps along the vertical density profile in the altitude band, d_i is the value of density at step i, and Δa is distance between data steps in the density profile.

The two methods for computing the *Density Parameter* differ in that the sum of the first method, Equation 5-3, is a function of the time the vehicle spends in the altitude band while the second method, Equation 5-4 is not. Both provide a parameterization that can be used in the Monte Carlo Filtering process. A correlation analysis was performed to explore the differences between the two methods and the Density Parameters computed using the first method showed a stronger correlation to Critical Metrics, and thus the first method was chosen as the atmosphere parameterization process for Monte Carlo Filtering. Using the first summing method for all altitude bands creates six input variables parameterizing an atmosphere profile by altitude where each altitude band can be correlated to the Monte Carlo Critical Metrics.

While the method outlined in this section serves as an initial technique for parameterizing an atmosphere, future work should include investigating other methods. One area to explore to further the current method is what is the optimal altitude band design for parameterization of the atmosphere? The current method divides the atmosphere into six equal bands, but there is very likely a different band design, with potentially unequal band sizes, that provides a better resolution and understanding of density correlation to Critical Metrics.

6 Design of Experiments (DOE) and Regression Modeling

Design of Experiments (DOE) is a commonly used method for designing experiments in industry and academia. It is a process by which multiple input variables in an experimental design are varied simultaneously and the resulting output response is measured. The mapping from input variation can then be used to derive a regression model of the system, which provides insight into inputs of highest influence on system behavior.

Monte Carlo Filtering discussed in the previous section identifies inputs of highest correlation to Monte Carlo performance metrics. Once identified, a set of the most correlated inputs is fed into the next step in the methodology, the development of a regression model via DOE Analysis, as shown in Figure 6-1. Methods outlined in the book *Statistical Design of Experiments with Engineering Applications* by Rekab and Shaikh [24] are used as a guide to implementing a DOE analysis for this application.

For each input variable, input levels are chosen based on the expected range of input variation. In its simplest implementation, DOE uses two-level input variable variation in

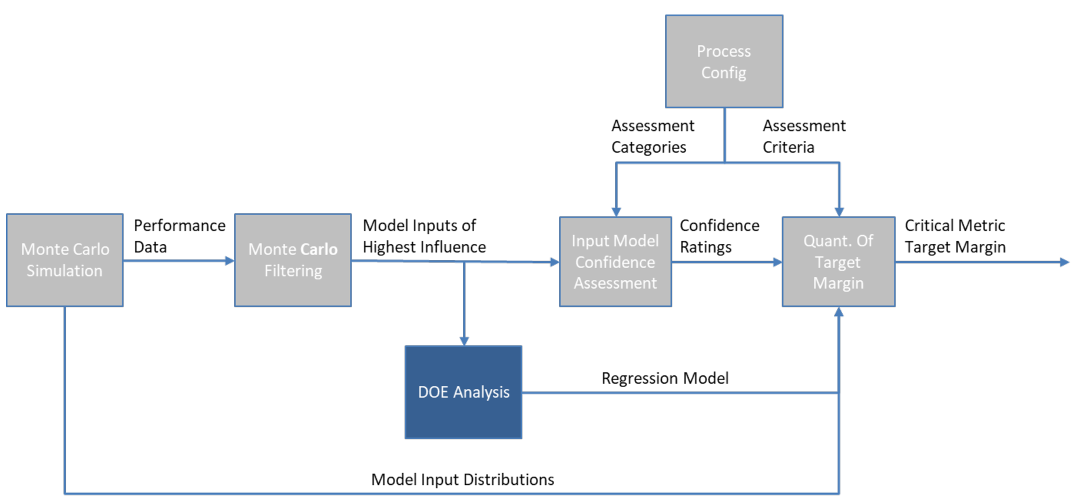


Figure 6-1 Detailed Evolved Methodology process diagram: DOE Analysis.

which the levels represent the "high" and "low" end of the expected input variation. For experiments anticipated as being more complex, representing variable variation by three levels is used, and was chosen for this investigation.

In three-level DOE, the levels represent the "low," "mid", and "high" expected input variable variation. Because of the complex and nonlinear nature of entry, descent and landing (EDL) system dynamics, for this investigation full factorial DOE is chosen in which all combinations of all factor levels are run in the experiment, which leads to a total of 3^p experimental runs to capture the full range of variation. The parameter p is the number of input variables. Other factorial schemes exist to reduce the number of experiments required, but because enough computational power exists for a full factorial experimental design in this investigation, a full factorial design was chosen. From here forward, the three levels will be represented by -1, 0, and 1 (low, mid, high).

Because of the nonlinearity of the physics governing the flight dynamics of an EDL trajectory, a second-order regression model was chosen as a starting point to model metric dependency on the chosen simulation inputs. As outlined in Rekab and Shaikh, the estimated value of the metric $\hat{Y}$ can be found from the combination of linear, cross terms (two-factor interactions) and second-order terms of inputs contained in the matrix X, where X is composed of the values -1, 0, 1 representing normalized values of the 3 levels of the DOE inputs. A vector b contains the coefficients of the terms of the second-order model, with the magnitude of the coefficients serving as a measure of the influence of a particular input on the variation of the output. In long form, the estimate of the metric $\hat{Y}$ is given by

$$\begin{aligned}\hat{Y} &= b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p \\ &+ b_{12} x_1 x_2 + b_{13} x_1 x_3 + \cdots + b_{p-1} x_{p-1} x_p \\ &+ b_{11} x_1^2 + b_{22} x_2^2 + \cdots + b_{pp} x_p^2\end{aligned} \tag{6-1}$$

where p is the number of input variables in the DOE analysis. From Rekab and Shaikh, when n experiments are carried out with Y representing a vector whose elements are the results of each experiment, the resulting matrix equation can be formulated

$$Y = Xb \tag{6-2}$$

Using simple linear algebra to solve for the vector b gives

$$b = (X'X)^{-1}X'Y \tag{6-3}$$

Because the model has been normalized, the coefficients contained in b can be used to quantify the level of influence of linear, cross terms and second-order terms that contribute to the model response. Thus, they provide a quantitative measure of the inputs of most influence.

In expanded matrix form, each element of the matrix equation has the following form

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \tag{6-4}$$

$$X = \begin{bmatrix} 1 & x_{11} \dots x_{1p} & x_{11}x_{12} \dots x_{1,p-1}x_{1,p} \dots x_{11}^2 \dots x_{1p}^2 \\ 1 & x_{21} \dots x_{2p} & x_{21}x_{22} \dots x_{2,p-1}x_{2,p} \dots x_{21}^2 \dots x_{2p}^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} \dots x_{np} & x_{1n}x_{n2} \dots x_{n,p-1}x_{n,p} \dots x_{n1}^2 \dots x_{np}^2 \end{bmatrix} \tag{6-5}$$

$$b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_p \\ b_{12} \\ \vdots \\ b_{11} \\ \vdots \\ b_{pp} \end{bmatrix} \tag{6-6}$$

To determine which terms of the regression model are statistically significant, an inference methodology outlined in Rekab and Shiakh is used. The following is the process used to infer the statistical significance of the second-order regression parameters using a two-tailed t-test, which is an often-used hypothesis test for significance.

The fundamental approach is to test a null hypothesis for the second-order prediction equation

$$\hat{Y} = b_0 + \sum_{i=1}^{L} b_i x_i \tag{6-7}$$

where the inference test determines which terms should be included in the prediction equation, L is the number of terms, and x_i represents linear, cross terms and second-order terms. The null hypothesis test is then the following

$$H_0 : b_k = 0 \ versus \ H_a : b_k \neq 0 \tag{6-8}$$

where H_0 is the null hypothesis and H_a is the alternate hypothesis.

To test the hypotheses, the first step is to compute T_0 for each coefficient in the prediction equation. These values are then compared against the t-statistic, T^*, to test for statistical significance. The value of T_0 is found from the following formulation

$$T_0 = \frac{b_k}{S(b_k)} \tag{6-9}$$

The function $S(b_k)$ is the square root of the *kth* element of the estimated variance-covariance matrix $S^2(b)$. $S^2(b)$ is found from the following

$$S^2(b) = MSE(X^t X)^{-1} \tag{6-10}$$

where *MSE* is the mean squared error and is given by

$$MSE = \frac{\sum_{i=1}^{n} e_i^2}{n - L - 1} \tag{6-11}$$

The variable L is the number of non-constant coefficients in b, n is the number of experiments in Y, and $n-L-1$ is the degrees of freedom. The error, e_i, is given by

$$e_i = Y_i - \hat{Y}_i \tag{6-12}$$

Once T_0 is determined for each of the regression parameters, they must be compared against T^*, where T^* is determined from the t-distribution, which is a distribution used to test for significance when working with small sample size. $T^*$is given by

$$T^* = t(\alpha/2, n - L - 1) \tag{6-13}$$

The variable α is the significance level, which is a selected value. Values often range from 0.01 to 0.1, with 0.05 being the most common used. For this assessment the value of α is chosen to be 0.05, which means there is a 5% risk the significance test conclusion is invalid. This provides an adequate level of significance without being too restrictive. The significance of the regression parameter is then determined with the following comparisons

$If\ |T_0| > T^*, the\ null\ hypothesis\ is\ rejected\ and\ the\ regression\ term\ is\ included$

$If\ |T_0| \leq T^*, the\ null\ hypothesis\ is\ accepted\ and\ the\ regression\ term\ is\ not\ included$

In addition to the t-test, the p-value for each regression term is also utilized. The p-value is the cumulative distribution function (CDF) of the t-distribution computed at $T_{0.}$ It tests the null hypothesis that the regression term is zero. The following is applied to p-values for significance level 0.05: [25]

If the p-value ≤ 0.05, the null hypothesis is rejected and the regression term is included

If the p-value >0.05 the null hypothesis is accepted and the regression term is not included

In implementing the regression modeling, the t-test and p-value are used to identify those terms in the regression model that should be included as statistically significant. Terms that do not fail the null hypothesis are not included. The t-test and p-value are expressions of the same significance test, and thus provide corresponding results.

The process of using DOE to generate a second-order regression model, and then applying a test to determine statistical significance to each term of the second-order model provides a basis for determining which model inputs are most influential in output performance for a given performance metric. Because the regression model is normalized, the coefficients of the regression model provide a direct measure of the strength of influence for each input, which provides identification of which inputs should receive focus for modeling fidelity and validation efforts, and, as discussed in Section 9, a quantitative means to map modeling confidence to Target Margin.

6.1 TMST Implementation of DOE Regression Modeling

In the Evolved Methodology implemented in Target Margin Sizing Tool (TMST) software, the process of generating a regression model is conducted outside the software tool, and the resulting model is then taken as input into TMST. The tool then uses the model to quantify input level of influence and the quantification of Target Margin. The regression model is input into TMST using a configuration file, and the tool provides a regression model tornado plot, which is a horizontal bar chart showing the magnitude of each regression model coefficient, ordered by magnitude, to help visualize the relative strength of influence of inputs. A more detailed discussion of regression model implementation is provided in Section 11.3.

6.1.1 Regression Model Configuration File

The output of the DOE regression model generation process is a list of the coefficients for the second-order regression model, along with the results of t-test and p-value statistical significance test for each coefficient. Appendix D shows an example DOE regression model output file. The last column titled "Significant?" indicates whether the null hypothesis was rejected, thus indicating the coefficient should be included in the regression model. Appendix E shows a corresponding TMST input configuration file where all insignificant coefficients have been set to zero. This file is read into TMST to provide the regression model when a Target Margin assessment is initiated.

6.1.2 Visualization of Regression Importance Parameters

As stated earlier, the magnitude of the regression model coefficients provides a quantification of the level of influence each has on the Critical Metric performance. To aid in the execution of the Evolved Methodology, a regression model visualization was developed via a horizontal bar chart sometimes referred to as a tornado plot (so named because of its resemblance to the profile shape of a tornado's funnel cloud), to illustrate the relative influence each input has on the response of the Critical Metric. Figure 6-2 illustrates a portion of a regression model tornado plot implemented in TMST. On the left half of the figure, the regression model coefficients are listed along with coefficient indices (the names of the coefficients will be discussed in a later section). The linear coefficients are listed first in order of magnitude (the reason for this will be discussed later), followed by nonlinear coefficients listed in order of magnitude. The adjacent bars of the tornado plot are scaled by the magnitude of the coefficient, and the coloring indicates the sign of the coefficient. A positive coefficient has blue shading on the positive half of the tornado, and red on the negative. A negative coefficient has the opposite shading. The shading helps visualize which terms of the regression model add constructively and destructively,

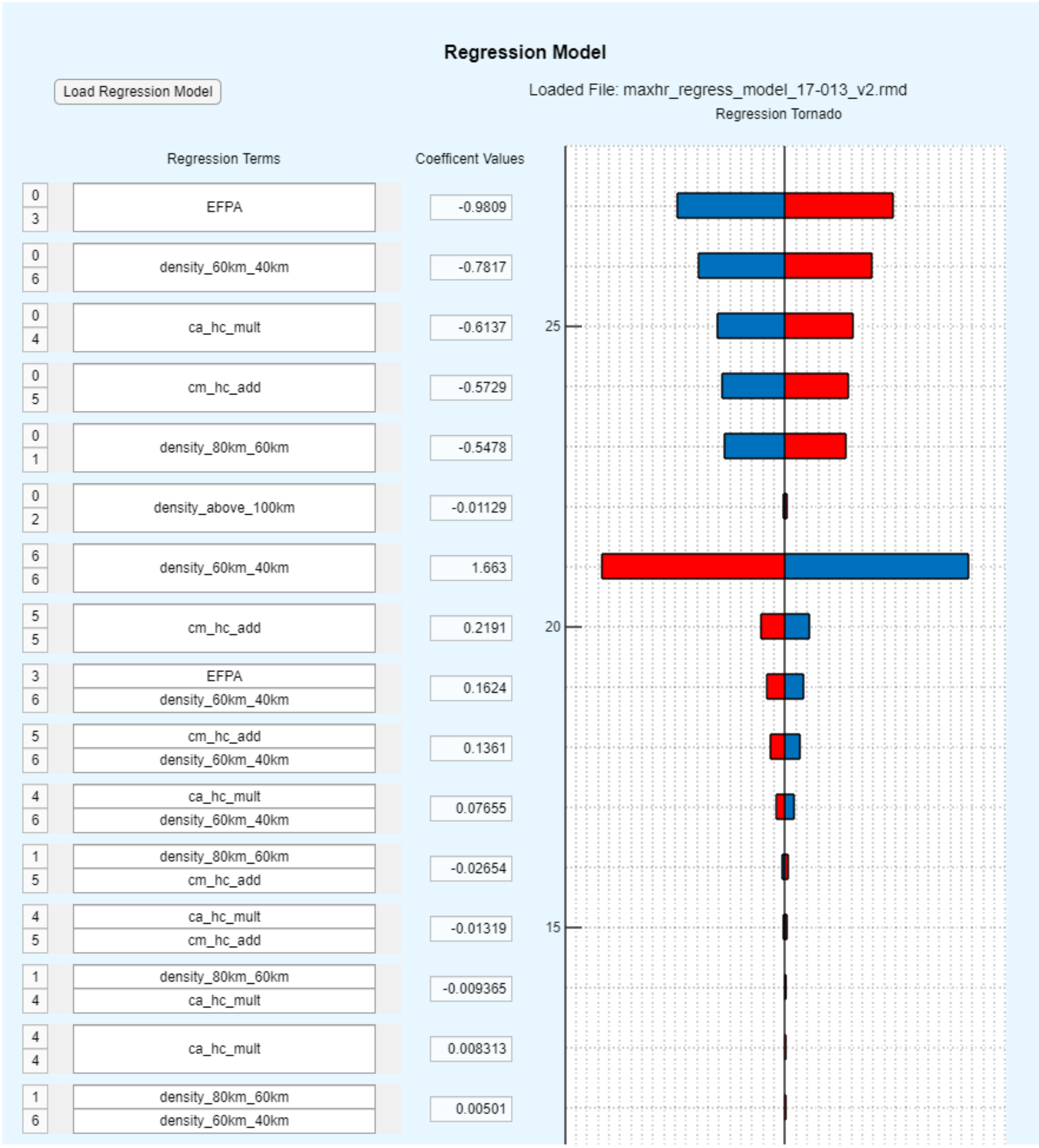


Figure 6-2 TMST regression model visualization.

and is not meant to infer coefficients can change sign. The tornado plot provides an informative visualization of the relative strength of importance of input parameters on Critical Metric response.

A design of experiments 3-level, full factorial analysis provides a second-order regression model that provides a quantification of the influence each Model Inputs of Highest Influence has on Critical Metric response, and thus provides insight into where model confidence efforts should be focused and assessed. It also provides an analytical expression (the regression model) for quantifying Target Margin based on modeling confidence. It therefore is a key element of the Evolved Methodology.

7 Process Configuration

In order to execute the Evolved Methodology process, the process must be configured prior to running the Target Margin sizing analysis. As shown by the blue shaded box in Figure 7-1, the process configuration feeds the Input Model Confidence Assessment by providing Assessment Categories, and the Quantification of Target Margin by providing Assessment Criteria with associated Distribution Fractions.

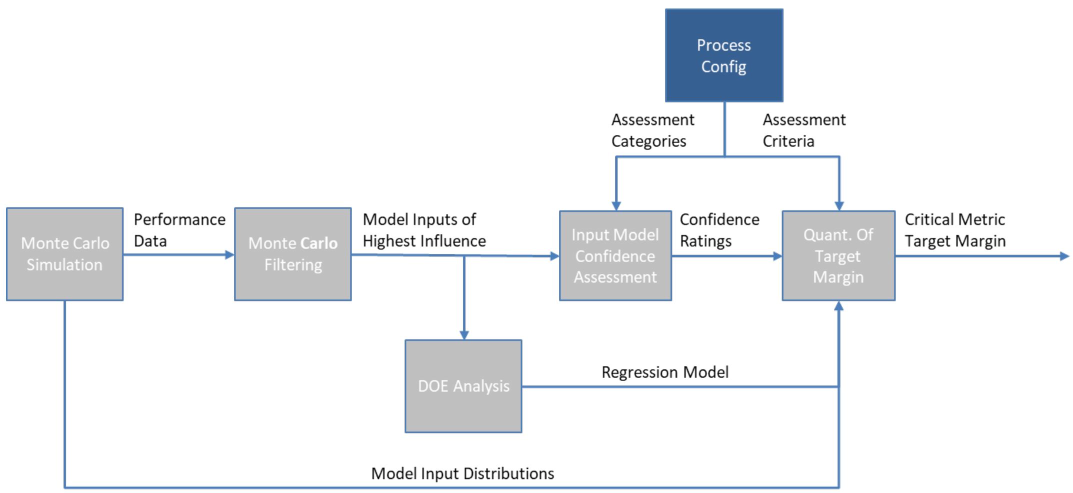


Figure 7-1 Detailed Evolved Methodology process diagram: Process Configuration.

7.1 Assessment Categories

The Assessment Categories are focus areas for the assessment that provide a measure of the confidence in the input modeling and are chosen as part of configuring the assessment. As shown in Figure 7-2, example Assessment Categories are Mathematical Model, Code, Validation and Model Parameter. Other categories can be part of the assessment at the discretion of the systems engineer conducting the assessment.

As can be seen in Figure 7-2, each Assessment Category is assigned a weighting that is used in computing Target Margin, the process of which is detailed in Section 9. Assessment

Categories can be identified and configured based on the unique assessment needs for a particular entry, descent and landing (EDL) system engineering assessment. A more detailed discussion of Assessment Categories is provided in Section 8.

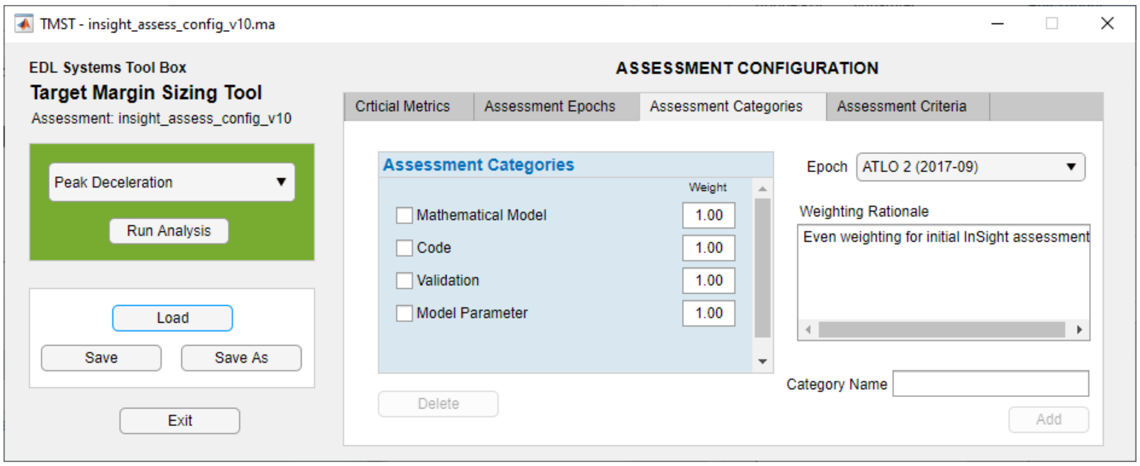


Figure 7-2 Target Margin Sizing Tool Assessment Categories configuration panel.

7.2 Assessment Criteria

As described in Section 3.1, the sizing of Target Margin in the methodology is based in a fraction of the distribution for each of the most influential inputs identified via Monte Carlo Filtering and DOE (design of experiments) Analysis. For each of the most influential inputs, all Assessment Categories discussed in the previous section are given a Confidence Rating of High, Medium or Low, as shown in Figure 7-3. As part of configuring the assessment, Assessment Criteria are established by assigning a Distribution Fraction to each Confidence Rating, and a rationale for the fraction assigned is provided.

The Evolved Methodology as currently developed is limited to inputs that have normal and uniform uncertainty distributions, which are common input distributions in EDL Monte Carlo

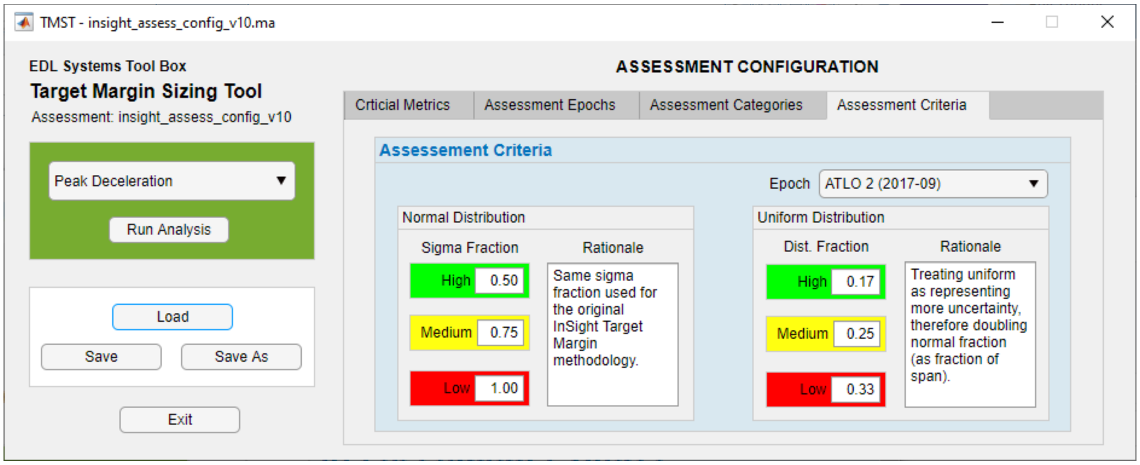


Figure 7-3 Target Margin Sizing Tool Assessment Criteria configuration panel.

simulations. The following are the Distribution Fraction approaches used for normal and uniform distributions:

Normal Distribution: Because standard deviation is a measure of the level of spread in a normal distribution, and thus the uncertainty in input value, Distribution Fraction for an input with normal distribution is based on a fraction of the standard deviation of the distribution.

Uniform Distribution: A uniform distribution, as its name implies, has uncertainty that is uniformly distributed across the range of possible input values, and thus Distribution Fraction for an input with uniform distribution is based on a fraction of the entire span of the distribution.

Because a normal distribution has possible values heavily weighted near the mean, the fractional sigma value assigned to Distribution Fraction for normal distributions can be less conservative. A uniform distribution implies the input value can be equally likely anywhere in the distribution, including near the extremes of the distribution, and therefore a more conservative approach to assigning uniform Distribution Fractions (larger fractions) to Confidence Rating may be appropriate. If the bounds of a uniform distribution are well defined and understood with confidence, then a conservative approach may not be appropriate. In Addition to normal and uniform distributions, developing an approach for sizing Assessment Criteria for other types of input distributions is future work.

7.3 Tool Implementation

The configuration of the Evolved Methodology process is provided by the Target Margin Sizing Tool (TMST) Startup Window. As shown in Figure 7-2, in addition to buttons for loading an existing assessment configuration, and Save and Save As buttons for saving a current assessment configuration, the window has four tabs for configuring the assessment. The first tab, labeled Critical Metrics, provides the ability to input the Critical Metrics being assessed, with buttons for adding and deleting Critical Metrics. For each metric, an area for inputting the physical units of the metric is also provided. The second tab, labeled Assessment Epochs, provides the ability to configure the epochs for which an assessment will be conducted, also utilizing Add and Delete buttons. A more detailed discussion of Assessment Epochs is provided in Section 10. The third tab, shown in Figure 7-2 and labeled Assessment Categories, provides the ability to configure Assessment Categories and category weighting, as described previously in Section 7.1. The fourth tab, shown in Figure 7-3 and labeled Assessment Criteria, provides the ability to configure criteria scoring via specifying Distribution Fractions for each Confidence Rating level. Also provided is an area for a rationale for the chosen numerical scoring choices, which is critical for documenting the critical thinking that led to the choice of Distribution Fraction sizes for each Confidence Rating level. These four tabs provide the ability to configure the Target Margin assessment and perform the function of the Process Config block in the process block diagram shown in Figure 7-1.

7.3.1 Process Configuration File

TMST has the ability to save and load assessment configurations via the Startup Window, as previously mentioned. The configuration is saved to a file with file extension ".ma" (ma for *margin assessment*) and a file name chosen by the user. An example TMST margin assessment configuration file is shown in Appendix F. The file contains all parameters required to save a Target Margin assessment configuration.

8 Input Model Confidence Assessment

As with the original Seed Methodology described in Section 1.3.2, the Evolved Methodology uses confidence in input models as a basis for sizing Target Margin. As shown in Figure 8-1, the Input Model Confidence Assessment block in the methodology block diagram, shown in blue shading, takes as input the Model Inputs of Highest Influence from Monte Carlo Filtering and the identified Assessment Categories that were described in the previous section. The process by which the Input Model Confidence Assessment is performed is detailed in this section.

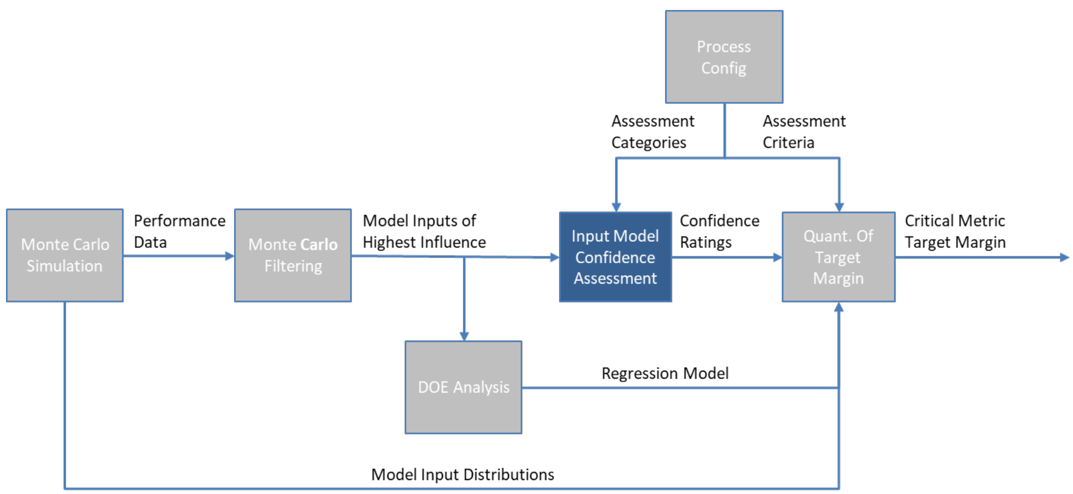


Figure 8-1 Detailed Evolved Methodology process diagram: Input Model Confidence Assessment.

8.1 Assessment Categories

The Assessment Categories play a critical role in assessing confidence in Model Inputs of Highest Influence on performance Critical Metrics. The categories form the basis for a methodical assessment of the confidence in input modeling, which is important for understanding how much performance margin should be held against performance limits. An existing methodology called the Phenomena Identification and Ranking Table, or PIRT, is employed in the practice of QMU (introduced in Section 1.3.1.1) and was developed at Sandia National Laboratories [26] [27]. The PIRT method serves as a basis for the process created for the Target Margin sizing methodology in this dissertation.

8.1.1 Phenomena Identification and Ranking Table as Confidence Assessment Basis

As outlined in the Sandia National Laboratories cited reports, a PIRT is a process by which modeling "quality" is captured in a concise table easily used by engineering decision makers to understand the quality of modeling results. PIRTs provide a way to assess and document model adequacy for intended use by assessing quality of the mathematical model, computer code, level of validation, and quality of model parameters [26]. In the PIRT process, a phenomenon is a physical feature or behavior relevant to the assessment. Figure 8-2 shows an example of a Sandia National Laboratories three-phenomena PIRT table. g

Phenomena	Importance	Adequacy for Intended Use			
		Math Model	Code	Validation	Model Parameter
Phenomena 1	H	H	M	L	L
Phenomena 2	M	H	M	L	L
Phenomena 3	L	H	M	L	L

Figure 8-2 Example three phenomena PIRT. [26]

For a PIRT, Sandia National Laboratories defines Adequacy for Intended Use categories as follows [26]:

- *Mathematical Model* is a physical, conceptual, or phenomenological model that is defined using precise mathematical equations. Math model adequacy represents the pedigree, completeness, and relevance of the math model form for the application.

- *Code* represents the computational modeling and simulation capabilities used in an engineering analysis. Code adequacy represents the status and quality (code verification) of the math model implementation.
- *Validation* is the process of determining the accuracy of a computational simulation to represent the real world as approximated by experimental data. Validation adequacy represents the rigor (solution verification and uncertainty quantification) used in quantifying the math model accuracy and the relevance of the validation comparison for the application.
- *Model Parameter* refers to parameters or functions in the physics or material models that are typically experimentally determined. Model parameter adequacy represents the pedigree, completeness, and relevance of the model parameter values or functions for the application.

As can be seen in Figure 8-2, a PIRT also utilizes a high (H), medium (M) and low (L) rating for the Importance of the phenomena and the Adequacy of Intended Use. As presented in Section 7.2, the Target Margin Evolved Methodology adds the quantification of the rating via Distribution Fractions.

In implementing the Evolved Methodology in this dissertation, the Sandia National Laboratories PIRT Adequacy for Intended Use categories have been adopted as an applicable set of Assessment Categories, although the methodology allows the flexibility for any appropriate categories. As part of adopting the PIRT categories, their use is extended by adding numerical weighting for each of the Assessment Categories used in the computation of Target Margin, as previously mentioned in Section 7.1.

8.2 Category Weighting

With multiple Assessment Categories used in the Evolved Methodology, aggregation of categories is required. A more detailed discussion of the numerical aggregation follows in Section 9, however here is a high-level discussion of category weighting to be employed in aggregation.

The simplest aggregation technique would be a simple average of contribution from each Assessment Category; however, the methodology deploys an option of weighted Assessment Categories if one category is deemed more critical or more relevant to the Target Margin assessment. A weighted average of contribution to Target Margin is then computed using straightforward weighted average arithmetic. In implementation of weighted categories is discussed further in Section 9.1.1. Category weighting also provides the ability to conduct a sensitivity analysis to category weighting, which is examined in Section 12.3.

8.3 Tool Implementation

An important feature of the Target Margin Sizing Tool (TMST) is the interface provided for an Input Model Confidence Assessment. The interface is provided on the Assessment Main Window, shown in Figure 8-3. As seen in the figure, the Category Assessment area is structured much like a PIRT, with each Assessment Category forming a column of the table, and each row a linear regression model term corresponding to a Model Inputs of Highest Influence identified by Monte Carlo Filtering and used in formulating the design of experiments (DOE) regression model. Each cell in the table provides a drop-down menu to select the Confidence Rating for that input variable under the cell's Assessment Category. When a Confidence Rating is selected, the corresponding Distribution Fraction is updated. A Rationale edit box is provided to enter the rationale for the rating, which is important for capturing and documenting the reason for the rating.

Providing a rationale also creates a structured critical thinking process for considering level of confidence in the entry, descent and landing (EDL) simulation modeling. Figure 8-3 illustrates an assessment table interface before a category assessment has been carried out. The table is a section of the larger TMST Assessment Main Window shown in Figure 3-5. A detailed discussion of Category Assessment implementation is presented in Section 11.5.

8.3.1 Assessment Configuration File

TMST saves a Input Model Confidence Assessment to a margin configuration file with extension ".mcf", an example of which is shown in Appendix G. An assessment can be configured using the Assessment Main Window, or by directly editing the configuration file. The file contains all of the configurable parameters for conducting the assessment and computing Target Margin.

(This space intentionally left blank)

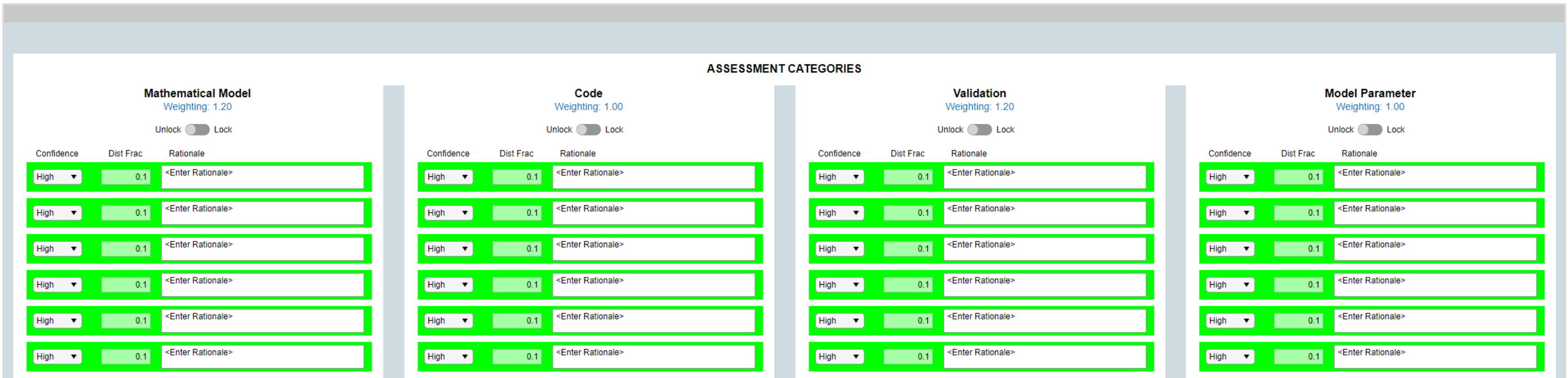


Figure 8-3 Input model assessment table on TMST Assessment Main Window.

(This space intentionally left blank)

9 Quantification of Target Margin

All the Evolved Methodology steps described in the previous sections feed into to the final step in the process, the Quantification of Target Margin, as shown by the blue highlighted block in Figure 9-1. As seen in this process diagram, Target Margin is computed using the Assessment Criteria from the Process Configuration, the Confidence Ratings from the Model Inputs Confidence Assessment, and the Critical Metric Regression Model from DOE (design of experiments) Analysis. These inputs feed the numerical quantification of Target Margin that is described in this section.

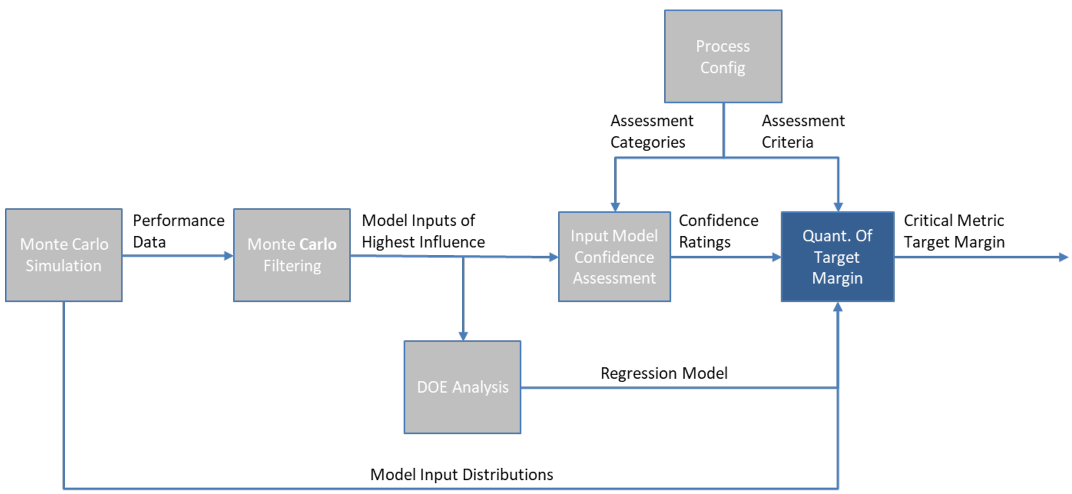


Figure 9-1 Detailed Evolved Methodology process diagram: Quantification of Target Margin.

9.1 Distribution Fractions and Computing a Confidence Score

As introduced in Section 7.1 and Section 7.2, the Input Model Confidence Assessment for each Assessment Category results in a Confidence Rating per Assessment Category for each input. The Confidence Rating per category results in a Distribution Fraction based on the fraction specified in the Process Configuration. A weighted average is then computed, and then mapped into the DOE normalized space described in Section 6, resulting in a Confidence Score used in the DOE regression model to quantify Target Margin.

9.1.1 Computing Weighted Average

For each Assessment Category configured as part of the assessment, the Input Model Confidence Assessment results in a Confidence Rating that determines the Distribution Fraction, or *DF*. As part of the process of computing a Confidence Score, a weighted average of Distribution Fractions, or *WADF*, is computed using the following simple weighted average

$$WADF = \left(\sum_{i=1}^{n} W_i \cdot DF_i\right) \Bigg/ \left(\sum_{i=1}^{n} W_i\right) \qquad (9\text{-}1)$$

where n is the number of Assessment Categories and W_i is the weighting that was assigned to the *ith* category during Process Configuration. The *WADF* provides a single aggregation of the Distribution Fractions that result from the Input Model Confidence Assessment, thereby generating a single metric representative of the confidence in the input parameter assessed. The *WADF* is representative of a fraction of the input distribution; however, in order to utilize the *WADF* in the DOE regression model, the *WADF* must be mapped to the [-1, 1] interval used in the DOE process (described in Section 6). The following subsections discuss the mapping process for a normal distribution and a uniform distribution, which results in a Confidence Score.

9.1.2 Mapping *WADF* to DOE Interval

As a recap of Section 7.2, the Evolved Methodology defines the Distribution Fraction for a normal distribution as a fraction of one standard deviation, whereas the Distribution Fraction for a uniform distribution is defined as the fraction of the span of a uniform distribution. Because of the difference in definition, the mapping of *WADF* to the DOE [-1,1] interval must be handled differently for each distribution type, as discussed in the next two subsections.

9.1.2.1 Normal Distribution

To compute the Confidence Score from *WADF* for a normal distribution, the *WADF* must be mapped to the DOE [-1, 1] interval. Figure 9-2 on the following page shows graphically the mapping of *WADF*, shown as *WADF*σ, to the DOE interval, where *WADF*σ represents a fraction ($WADF$ < 1) or multiple ($WADF \geq 1$) of a standard deviation. As the DOE process was implemented, the -3σ to 3σ input parameter distribution bounds mapped to the -1 to 1 bounds on the DOE interval, as is easily seen in Figure 9-2. Because the interval between -3σ and 3σ spans a total of 6σ, the fraction of the total distribution interval that *WADF*σ spans is given by

$$Interval\ Fraction = \frac{WADF\sigma}{6\sigma} = \frac{WADF}{6} \qquad (9\text{-}2)$$

The total span of the DOE interval [-1, 1] is 2, so that *WADF* on the DOE interval, which is defined by the methodology as the Confidence Score, or CS, is given by

$$CS_{Normal} = Interval\ fraction \times 2 = \frac{WADF}{6} \times 2 = \frac{WADF}{3} \qquad (9\text{-}3)$$

This Confidence Score is used in the DOE regression model to quantify Target Margin, as is shown in Section 9.2.

9.1.2.2 Uniform Distribution

For a uniform distribution, the Confidence Score is derived from the *WADF* again by

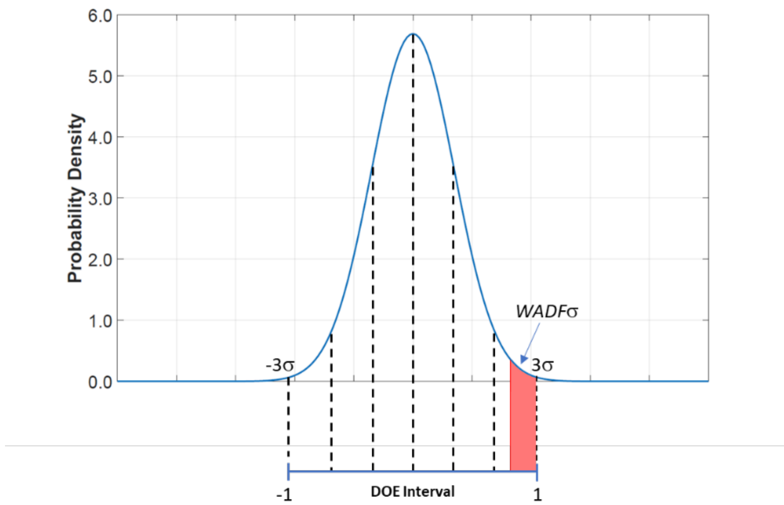


Figure 9-2 Normal input distribution fraction mapped to DOE range.

mapping *WADF* to the DOE confidence interval. The mapping for a uniform distribution is shown in Figure 9-3. As can be seen in the figure, the bounds of the uniform distribution are u_1 and u_2 and *WADF* is indicated in red shading. The uniform distribution bounds u_1 and u_2 map directly to -1 and 1 bounding the DOE interval. Because Distribution Fraction for a uniform distribution is defined as the fraction of the total span, the interval fraction is simply

$$Interval\ Fraction = WADF \quad (9\text{-}4)$$

The total span of the DOE interval [-1, 1] is 2, so that the Confidence Score, CS, for a uniform distribution is given by

$$CS_{Uniform} = Interval\ Fraction\ \times 2 = WADF \times 2 \quad (9\text{-}5)$$

9.2 Computing Target Margin Using DOE Regression Model

The Confidence Scores for model input are used in the DOE regression model to quantify

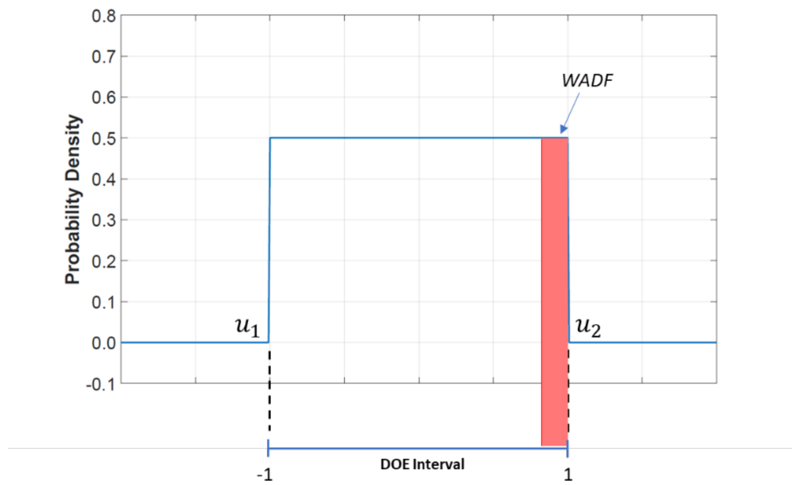


Figure 9-3 Uniform input distribution fraction mapped to DOE range.

Target Margin, the process of which is described in this section. From Section 6, the derived DOE second-order regression model for Critical Metric performance response is given by

$$\begin{gathered}\hat{Y} = b_0 + b_1x_1 + b_2x_2 + \cdots + b_px_p \\ +b_{12}x_1x_2 + b_{13}x_1x_3 + \cdots + b_{p-1}x_{p-1}x_p \\ +b_{11}x_1^2 + b_{22}x_2^2 + \cdots + b_{pp}x_p^2\end{gathered} \tag{9-6}$$

where p is the number of input variables. The Target Margin is computed at the bounds of Critical Metric performance, which is, in summation form, given by

$$\hat{Y}_{max} = b_0 + \sum_{i=1}^{p} |b_i|(1) + \sum_{i=1}^{p-1}\sum_{j=i+1}^{p} |b_{ij}|(1)(1) + \sum_{i=1}^{p} b_{ii}\,(1\ or\ 0)^2 \tag{9-7}$$

where the response model is evaluated at 1 for linear and interactive terms, with the absolute value of the regression model coefficients used to provide the bounding magnitude of $\hat{Y}$. The second-order terms are evaluated at 1 if b_{ii} is positive, or at 0 if b_{ii} is negative, thus ensuring a positive contribution to $\hat{Y}_{max}$. The Target Margin, TM, is then defined by

$$TM = \hat{Y}_{max+CS} - \hat{Y}_{max} \tag{9-8}$$

where CS is the Confidence Score. The term $\hat{Y}_{max+CS}$ extends the bounding response, $\hat{Y}_{max}$, by CS, and the difference between $\hat{Y}_{max+CS}$ and $\hat{Y}_{max}$ provides quantification of Target Margin. It quantifies the margin, based on the Critical Metric regression model, that is needed to protect against modeling uncertainty, as determined by the input model Confidence Score, CS. The expanded expression for Target Margin is

$$\begin{gathered} TM = b_0 + \sum_{i=1}^{p} |b_i|(1 + CS_i) + \sum_{i=1}^{p-1}\sum_{j=i+1}^{p} |b_{ij}|\,(1 + CS_i)\big(1 + CS_j\big) \\ + \sum_{i=1}^{p} b_{ii}\,(1 + CS_i \; or \; 0)^2 - b_0 - \sum_{i=1}^{p} |b_i|(1) - \sum_{i=1}^{p-1}\sum_{j=i+1}^{p} |b_{ij}|\,(1)(1) \\ - \sum_{i=1}^{p} b_{ii}\,(1 \; or \; 0)^2 \end{gathered} \tag{9-9}$$

There are two important notes that go along with this expression for Target Margin:

- The expression for Target Margin in Equation 9-9 assumes Critical Metric performance whose Metric Limit is on the high side of the performance distribution, which is the specific case that is applied in this dissertation. Extending the process to other Critical Metrics that have Metric Limit on the low side of performance distribution is future work.
- The Critical Metric regression model is an interpolation model. By evaluating the Critical Metric regression model at $\hat{Y}_{max+CS}$, the model is being extended outside the [-1, 1] interval over which it was derived, and becomes progressively less valid at larger values of CS. Awareness of this potential limitation should be present when applying the methodology.

9.3 Computing Epistemic and Aleatory Contribution to Target Margin

One feature of the Evolved Methodology is tracking epistemic uncertainty and aleatory uncertainty contribution to Target Margin, both for total Target Margin and the Target Margin contribution from each term in the regression model. This is done by considering the linear, interactive (or cross) and second-order terms in the Target Margin expression in Equation 9-9.

Referring to Equation 9-9, the contribution to Target Margin by the *nth* linear term is given by

$$|b_n|(1 + CS_n) - |b_n|(1) \tag{9-10}$$

Because the uncertainty inherent in the input distribution is cast as being made of epistemic and/or aleatory uncertainty, Equation 9-10 can be written as

$$|b_n|[(EU_n + AU_n) + (EU_n + AU_n)CS_n] - |b_n|(EU_n + AU_n) \tag{9-11}$$

where EU_n is the fraction of total uncertainty that is epistemic, and $AU_n$is the fraction of total uncertainty that is aleatory for the *nth* model input, and

$$EU_n + AU_n = 1 \tag{9-12}$$

Equation 9-11 can be simplified and written as

$$|b_n|CS_nEU_n + |b_n|CS_nAU_n \tag{9-13}$$

The first term in Equation (9-13) is the epistemic uncertainty contribution to Target Margin from the *nth* linear term of the regression model, and the second term is the aleatory uncertainty contribution to Target Margin from the *nth* linear term of the regression model. This expression shows that for the linear terms of the regression model, the contribution from epistemic uncertainty can be separated from contribution from aleatory uncertainty, and the epistemic and aleatory contributions are simply EU_n and AU_n, respectively.

In a similar way, the interaction term for the *nth* and *mth* regression model inputs can be written in terms of epistemic and aleatory uncertainty contribution as

$$|b_{nm}|[(1 + CS_n)(1 + CS_m) - 1][EU_nEU_m + EU_nAU_m + EU_mAU_n + AU_nAU_m] \tag{9-14}$$

As Equation 9-14 shows, for the interactive terms of the regression model, epistemic uncertainty contribution and aleatory contribution can be separated, but not completely. In the second square bracket in Equation 9-14, contribution from epistemic uncertainty is given by the term EU_nEU_m, contribution from aleatory uncertainty is given by the term AU_nAU_m, and contribution from combined uncertainty is given by the expression $EU_nAU_m + EU_mAU_n$, and it can be easily shown that

$$EU_nEU_m + EU_nAU_m + EU_mAU_n + AU_nAU_m = 1 \tag{9-15}$$

The following is an expression for the *nth* second-order regression model term in terms of epistemic and aleatory uncertainty contribution

$$b_{nn}CS_n(2 + CS_n)EU_n^2 + b_{nn}CS_n(2 + CS_n)AU_n^2 + b_{nn}CS_n(2 + CS_n)2EU_nAU_n \tag{9-16}$$

As seen in Equation 9-16, the first term contains epistemic uncertainty contribution only, the second term contains aleatory uncertainty only and the third term has combined epistemic and aleatory uncertainty. Therefore, as with interactive terms, for second-order regression model terms the uncertainty contribution to Target Margin can be partially separated into epistemic and aleatory, but part of the uncertainty can only be considered as combined. The contribution from epistemic uncertainty is EU_n^2, the contribution from aleatory uncertainty is AU_n^2, and the contribution from combined uncertainty is given by $2EU_nAU_n$, where

$$EU_n^2 + 2EU_nAU_n + AU_n^2 = 1 \tag{9-17}$$

Note that when b_{nn} in Equation 9-16 is positive, the second-order contribution to uncertainty type is provided by Equation 9-16, and when b_{nn} is negative, the contributions are set to zero, for the reason discussed in Section 9.2.

The above formulations are used by the Evolved Methodology to compute Target Margin contribution from epistemic uncertainty, aleatory uncertainty and combined uncertainty for each regression model term and for total Target Margin.

9.4 Tool Implementation

As part of the implementation of the Evolved Methodology in the Target Margin Sizing Tool (TMST), an assessment of uncertainty type for each input variable is conducted. As part of the Assessment Main Window, TMST provides an interface, shown in Figure 9-4, that allows for capturing the type of uncertainty distribution (normal or uniform), and an estimation of fractional split between epistemic and aleatory uncertainty in the input uncertainty distribution. A discussion of the value in capturing epistemic and aleatory uncertainty split is presented in Section 10.

The Assessment Main Window in TMST also contains a Target Margin sizing area where the results of the quantification of Target Margin are displayed, shown in Figure 9-5. Internally, TMST uses the expression for Target Margin shown in Equation 9-9 to compute the Target Margin, and also lists the contribution each term in the regression model makes to Target Margin, shown in Figure 9-5 for only the linear terms in the regression model for the sake of brevity, although the complete interface shows all regression model terms. Note that the terms of the regression model that are determined to be statistically not significant via the significance test

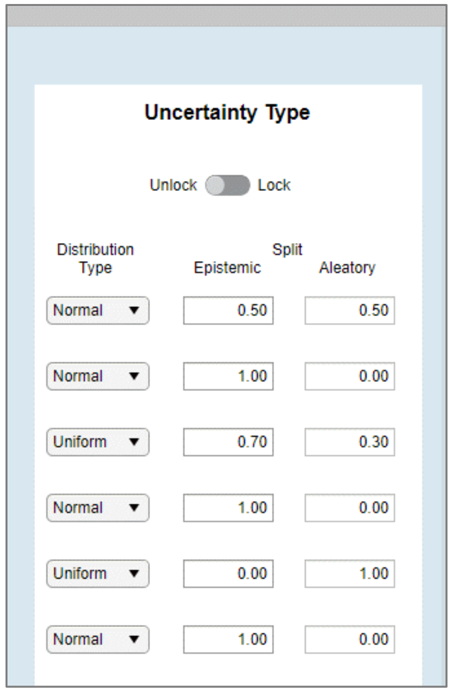


Figure 9-4 TMST input uncertainty specification interface.

described in Section 6 are set to zero. The blue shaded region in the TMST interface shown in Figure 9-5 shows Target Margin, and also the Contribution Split, which is the percent of Target Margin computed to be attributed to epistemic uncertainty, aleatory uncertainty and the combination of both, based on the estimations provided in the Figure 9-4 interface. The green shaded regions of the interface show the Confidence Score for the input variables associated with each of the linear terms of the regression model, along with the term's contribution to Target Margin, and the Contribution Split of percent contribution of epistemic, aleatory and combined uncertainty for each term. Note that the Confidence Score is shown only for the linear terms because it only needs to be computed once for each input variable, although the Confidence Score is used to compute the contribution to Target Margin for each term in the regression model. Application of the TMST tool to the InSight entry, descent and landing (EDL) system is the subject of Section 11.

Figure 9-5 TMST Target Margin sizing results interface.

10 Methodology Applied Across Project Lifecycle

The application of the Evolved Methodology across project lifecycle is accomplished by applying the methodology to chosen epochs or milestones during the lifecycle. By applying the methodology and tracking resulting Target Margin at epochs, the following benefits result:

Tracking Margin Trends Across Epochs

Exercising the Target Margin methodology across epochs and major milestones during the lifecycle of the project allows for determination of Target Margin given the current status of modeling confidence and uncertainty, and provides a record of sequential assessments under which to establish a Target Margin trend.

Identification of Modeling with Low Confidence

At each epoch, and particularly at early epochs, exercising an Input Model Confidence Assessment as part of the Evolved Methodology provides a survey of where low modeling confidence exists in the Model Inputs of Highest Influence. This survey can be used to assess where investments can be made to increase confidence and thus reduce the need for performance margin. In this way, the Evolved Methodology can aid in the strategic use of project resources.

Identification of Areas with High Epistemic Uncertainty

As with identification of areas with low modeling confidence, estimating the amount of epistemic uncertainty content in input uncertainty provides a survey of the reducible uncertainty present in the modeling of entry, descent and landing (EDL) system performance. Again, this knowledge can guide strategic use of project investments to reduce uncertainty.

Forecasting Target Margin and Performing Target Margin Sensitivities

The Evolved Methodology provides forecasting of Target Margin by applying the methodology to future epochs with current regression models, anticipated or expected configuration parameters, anticipated Assessment Category weightings and predicted Assessment Category Confidence Ratings (guided by expected changes in model fidelity, validation, etc). In this way, future needed Target Margin can be predicted and forecast to inform planning. Additionally, by parametrically varying configuration parameters, weightings and ratings, Target Margin sensitivity to variations (partial derivatives) can be established to also inform planning and decision-making. Sensitivity to configuration of the Evolved Methodology is applied to InSight in Section 12.

10.1 Tool Implementation

The Target Margin Assessment Tool (TMST) is designed to apply Target Margin assessments across project lifecycle. This is accomplished primarily by providing a user interface that provides the ability to track Target Margin assessments across all configured epochs. In the TMST Assessment Main Window interface, there is a tab for each epoch which contains all the controls necessary for doing an epoch assessment. The epoch tabs are indicated by the red bracket in the TMST Assessment Main Window interface in Figure 10-1. The TMST Assessment Main Window also has a Target Margin tracking display, shown in Figure 10-2. The display visually shows the Metric Limit, Target Margin and Target Limit side-by-side for each Target Margin assessment on the epoch tabs. This provides a record of Target Margin trending for past and current epochs. The epochs shown in Figure 10-2 are the InSight epochs to be used in Section 11. Figure 10-3 shows the Target Margin Sizing area of the TMST Assessment Main Window. The last three data columns shown in the figure labelled Epistemic, Aleatory and Combined provide a breakdown of the contributions to Target Margin from epistemic and aleatory. The column Combined lists the contribution from combined epistemic and aleatory for the portion of uncertainty where epistemic and aleatory uncertainty are mathematically inseparable (as was discussed in Section 9.3). These columns provide an understanding of fraction of Target Margin that is due to epistemic uncertainty, that has the potential to be reduced in future assessments through project strategic investments. The features outlined here provide for the means to apply the Evolved Methodology across a project lifecycle.

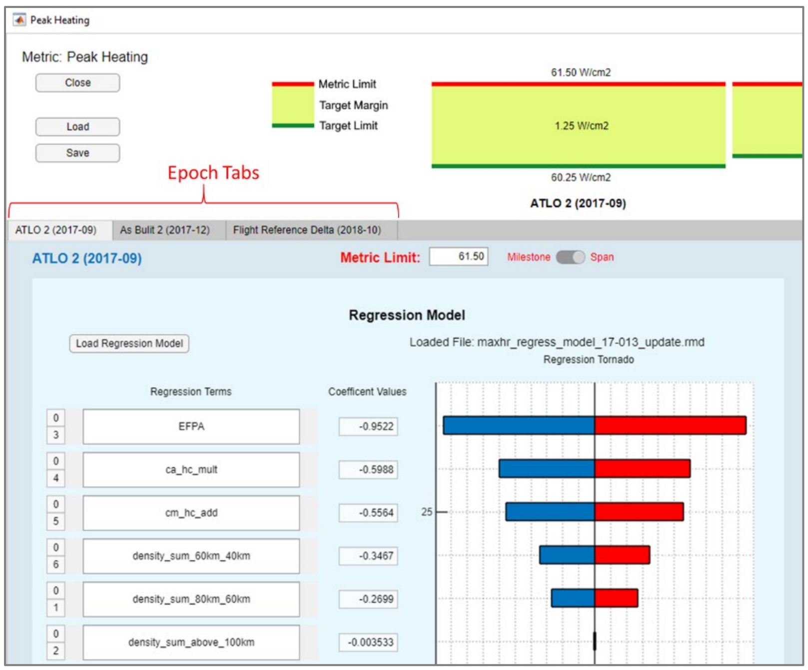


Figure 10-1 TMST Assessment Main Window epoch tabs.

(This space intentionally left blank)

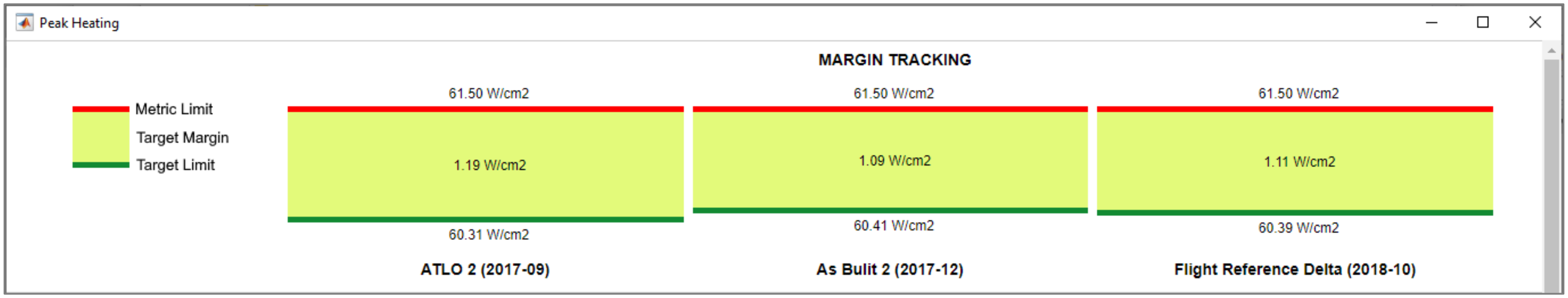


Figure 10-2 TMST Assessment Main Window margin tracking display across analysis epochs.

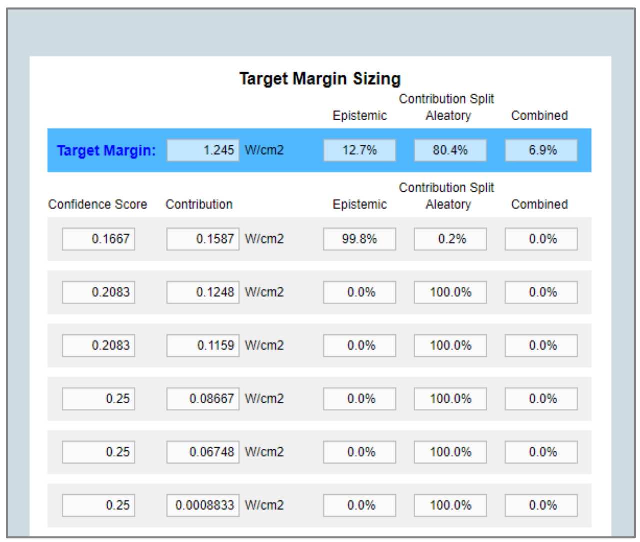


Figure 10-3 TMST Assessment Main Window epistemic and aleatory breakdowns.

11 Application to InSight Landing

The previous sections of this dissertation describe the mechanics of the Evolved Methodology for quantifying Target Margin. The remainder of the dissertation applies the methodology to NASA's InSight Mars landing, and will explore sensitivities to variations in process configuration. It also provides a comparison to the original Seed Methodology described in Section 1.3.2 and provides an assessment of the degree to which the original objectives of the research (outlined in Section 2) are achieved.

11.1 InSight Assessment Epochs

The InSight mission formally became a project in 2012 as part of NASA's Discovery Program [28]. Between its inception in 2012 and its landing on Mars in 2018, the project went through the standard phases of project development, implementation and flight. Ideally, a full application of the Evolved Methodology across all the phases of the InSight Project lifecycle would be most valuable in assessing the methodology. However, due to incomplete required data sets from the early phases of the InSight Project, only epochs from late in the project lifecycle will be used to apply the methodology. Application to a simulated early project epoch is presented in Section 12.2.

In the analysis of entry, descent and landing (EDL) flight dynamics performance, InSight utilized recurring analysis cycles, called Simulation Campaigns, to incrementally mature and ready the EDL system for flight. These analysis cycles included extensive Monte Carlo end-to-end simulation, the data from which is used in this dissertation to apply the methodology to InSight. The Monte Carlo simulation tool used by InSight is Program to Optimize Simulated Trajectories II (POST2), a widely used simulation tool developed by NASA Langley Research Center [29].

Three InSight Simulation Campaign data sets are used in applying the Evolved Methodology to InSight, as described in Table 11-1. The three Simulation Campaigns are ATLO 2.0, As-Built 2.0, and Flight Reference Delta. ATLO is an acronym for Assembly, Test and Launch Operations, and the "2.0" designation refers the fact that due to a launch date slip in 2016, the ATLO and As-Built campaigns were run twice, once prior to the original 2016 launch date, and again prior to the 2018 launch. The Simulation Campaigns described in Table 11-1 serve as the three InSight epochs to which the Evolved Methodology is applied.

Table 11-1 Summary of InSight Simulation Campaigns.

Sim ID	InSight Simulation Campaign	Description	Timing
17-013	ATLO 2.0	A simulation campaign timed to coincide with early Assembly, Test and Launch Operations (ATLO), nine months before launch	Launch – 9 months (2017-09)
17-037	As-Built 2.0	A simulation campaign timed to coincide with the earliest availability of as-built/as-measured vehicle parameters six months before launch	Launch – 6 months (2017-12)
18-563	Flight Reference Delta	A simulation campaign using the best-known configuration of the vehicle as launched, and used for EDL performance reference during flight, run two months before landing	Landing – 2 months (2018-10)

11.2 Monte Carlo Filtering Results

Archived Monte Carlo performance data for the three Simulation Campaigns in Table 11-1 were filtered for the Model Inputs of Highest Influence using the Monte Carlo filtering techniques described in Section 5, using correlation coefficient and mutual information. For each of the Monte Carlo data sets, the top six input variables of highest influence were identified. A count of six inputs was selected as a balance between capturing as many top influence inputs as possible while limiting the dimensions of the design of experiments (DOE) analysis and the number of terms of the resulting regression model. As was discussed Section 5, the results of the Monte Carlo Filtering flow into DOE Analysis in Section 11.3

The subsections that follow provide the results of the Monte Carlo Filtering process for the InSight data for the four Critical Metrics described in Table 11-2. During the formulation phase of the dissertation, to achieve reasonable research scope, the application of the Evolved Methodology to InSight Critical Metrics was limited to the atmospheric flight phase of InSight EDL from entry into the atmosphere to parachute deployment. This results in application to the Critical Metrics Peak Deceleration, Peak Heat Rate, Total Heating and Parachute Inflation Load. To better define the cryptic InSight Monte Carlo input names, Table 11-3 provides the definitions for the model input variables in the Monte Carlo Filtering results presented in this section. Note that Monte Carlo filtering for four Critical Metrics, over three epochs and limiting inputs of highest influence to six, results in a total of 72 Model Inputs of Highest Influence, although only 12 inputs are unique with different combinations of the 12 resulting in a total of 72. This will become apparent in the subsections that follow.

Table 11-2 Critical Metric descriptions and units[3].

Critical Metrics	Description	Units[3]
Peak Deceleration	The maximum deceleration experienced by the vehicle during entry	$Earth\ g's$
Peak Heat Rate	The peak heat rate experienced by the vehicle, measured at the stagnation point of the vehicle's heatshield	W/cm^2
Total Heating	The integrated heat load accumulated from entry to parachute deployment, measured at the stagnation point of the vehicle's heatshield	J/cm^2
Parachute Inflation Load	The maximum force experienced by the vehicle during the parachute inflation event	lbf

[3] For margin tracking for EDL, the InSight Project used both metric and English units, including pound-force for parachute inflation load. For consistency, InSight unit convention is adopted for this dissertation.

Table 11-3 Definitions of InSight Monte Carlo input names.

Input Name	Definition
ca_hc_mult	Entry vehicle axial force coefficient multiplier in hypersonic regime.[1]
ca_mc_mult	Entry vehicle axial force coefficient multiplier in supersonic regime.[1]
chute_aero_disp	Dispersed parachute drag coefficient.
cm_hc_add	Entry vehicle pitch moment coefficient adder in the hypersonic regime.[1]
cm_mc_add	Entry vehicle pitch moment coefficient adder in the supersonic regime.[1]
cn_hc_add	Entry vehicle yaw moment coefficient multiplier in hypersonic regime.[1]
density_sum_40km_20km	Sum of atmospheric density between 40 km and 20 km.[2]
density_sum_60km_40km	Sum of atmospheric density between 60 km and 40 km.[2]
density_sum_80km_60km	Sum of atmospheric density between 80 km and 60 km.[2]
density_sum_above_100km	Sum of atmospheric density above 100 km.[2]
EFPA	Entry vehicle flight path angle at entry interface at the top of the atmosphere.
zcg_ei	Entry vehicle aero frame z-axis center of gravity offset at entry interface.[3]

[1] Further defined in Section 11.2.1
[2] Discussed in Section 5.3
[3] Further defined in Section 11.2.2

11.2.1 InSight Entry Vehicle Aerodynamic Model

The input variable definitions listed in Table 11-3 contain aerodynamic coefficients defined as multipliers and adders. Aerodynamic coefficient multipliers and adders are used to disperse the local slope and intercept of an aerodynamic coefficient linear uncertainty function defined by M. Schoenenberger [30] by the following expressions:

Axial Force Coefficient

$$C_{A_{Disp}} = C_A(\alpha,\beta)\left(1 + U^M_{C_A}\right) \tag{11-1}$$

Pitch Moment Coefficient

$$C_{m_{Disp}|cg} = \left[C_m(\alpha,\beta)|_{MRP} + \frac{\Delta x}{d}C_N(\alpha,\beta) - \frac{\Delta z}{d}C_A(\alpha,\beta) + U^A_{C_m}\right]\left(1 + U^M_{C_m}\right) \tag{11-2}$$

Axial Moment Coefficient

$$C_{n_{Disp}|cg} = \left[C_n(\alpha,\beta)|_{MRP} + \frac{\Delta x}{d}C_Y(\alpha,\beta) - \frac{\Delta y}{d}C_A(\alpha,\beta) + U_{C_n}^A\right]\left(1 + U_{C_n}^M\right) \tag{11-3}$$

In Equation 11-1, $U_{C_A}^M$ is the axial force coefficient multiplier (ca_hc_mult & ca_mc_mult); in Equation 11-2, $U_{C_m}^A$ is the pitch moment coefficient adder (cm_hc_add & cm_mc_add); and in Equation 11-3, $U_{C_n}^A$ is the yaw moment adder (cn_hc_add). Other parameters in these equations are defined in Figure 11-1, from *Aerodynamic Performance of the 2018 InSight Mars Lander* by A. Korzun, et al [31].

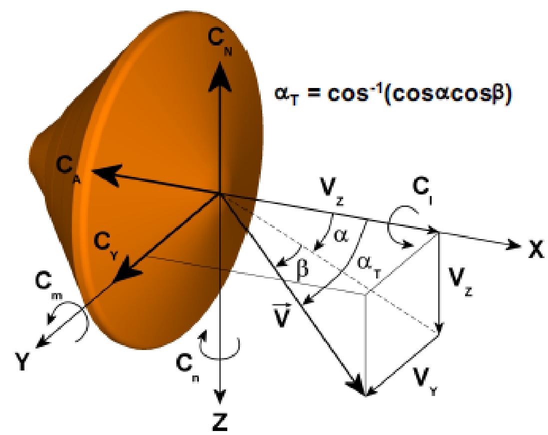


Figure 11-1 InSight aerodynamic coordinate frame. [31]

11.2.2 Entry Vehicle Center of Gravity Location Definition

The center of gravity location input parameter zcg_ei from Table 11-3 is the distance of the center of gravity from the origin of the spacecraft aerodynamic coordinate frame along the z-axis at entry interface, the moment of entry into the atmosphere at the beginning of the EDL event. From Blanchard, et al [32], the spacecraft Cruise Frame coordinate system is shown in Figure 11-2 for the Phoenix Mars Lander spacecraft, which had identical coordinate frame definitions as InSight. The Cruise Frame is shown in red, with X_C, Y_C, and Z_C defining the Cruise Frame axes. Figure 11-2 also shows the relationship between the spacecraft Cruise Frame and the orientation of the aerodynamic coefficients from Figure 11-1, and illustrates that the z-axis aerodynamic frame is aligned with the Y_C axis in the Cruise Frame. Figure 11-3 from Johnson, et al [33] depicts the same Cruise Frame relative to the InSight lander (which is housed inside the aeroshell at entry interface), with the Cruise Frame defined as X_C, Y_C, and Z_C in black font. The figure shows the lander from the bottom of the vehicle, with two red propellant tanks oriented at about 45° in the figure. Note the Cruise Frame y-axis is perpendicular to the lander propellant tanks, and at entry interface the attitude of the entry vehicle is oriented with the propellant tanks parallel to the surface of Mars, with any center of gravity offset in the Cruise Frame y-axis resulting in a pitch moment.

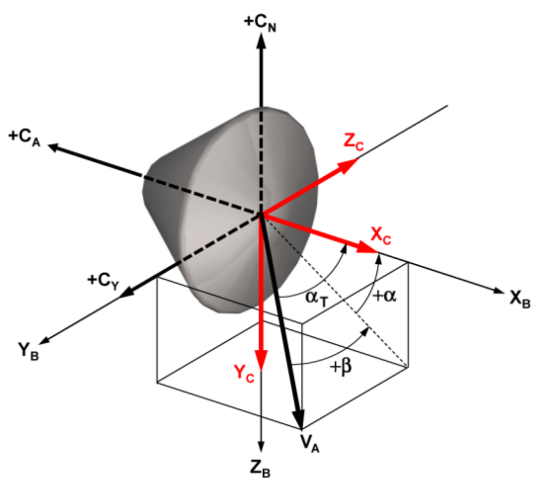


Figure 11-2 InSight entry vehicle coordinate frames. [32]

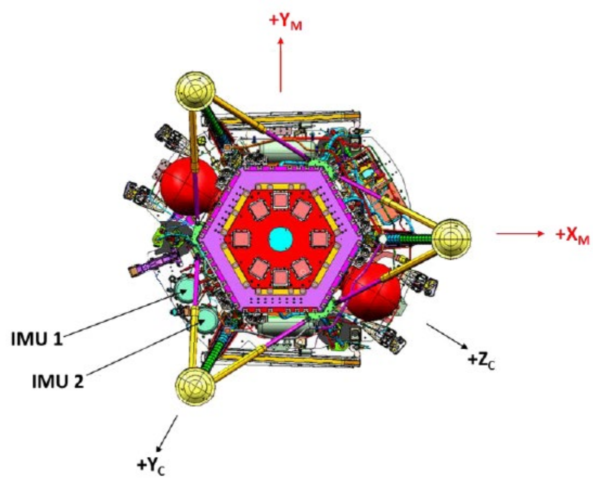


Figure 11-3 InSight lander coordinate frames. [33]

11.2.3 Peak Deceleration

The Monte Carlo Filtering results for the Critical Metric Peak Deceleration for the three InSight epochs are shown in Tables 11-4 – 11-6. Each table lists the mutual information and correlation coefficient result for the top six highest correlated input parameters for the Critical Metric Peak Deceleration. In identifying the most influential parameters, in general parameters were included as long as the magnitude of the correlation coefficient was near or above 0.1. As can be seen in the results, the pitch moment adder in the hypersonic regime shows the strongest correlation with Peak Deceleration. This is not surprising because the pitch moment will determine at what angle of attack the vehicle will trim, which determines the amount of lift the vehicle has as it flies a trajectory. A lift-down or lift-up trajectory strongly influences the Peak Deceleration experienced by the vehicle. The second parameter of strong influence is the atmosphere between 40 km and 20 km, which is the altitude band where the InSight entry vehicle experiences peak deceleration. The third input of strong influence is entry flight path angle, which again is not surprising because a steep entry angle will, in general, result in a higher peak deceleration while a shallower entry flight path angle will lower the peak deceleration. The six most influential input parameters across the three different analysis epochs are the same except for the sixth parameter, which is zcg_ei for the first two epochs ATLO 2.0 and As-Built 2.0, but changes to cn_hc_add for the final epoch Flight Reference Delta. The switch from zcg_ei to cn_hc_add occurs between epochs As-Built 2.0 and Flight Reference Delta, which may be tied to a narrowing of entry vehicle mass properties dispersion after launch of the spacecraft, where zcg_ei input dispersion standard deviation drops from 0.34 mm for ATLO 2.0 and As-Built 2.0, to 0.13 mm for Flight Reference Delta. Interestingly, the magnitude of the mutual information and correlation coefficients do not vary significantly across the three analysis epochs. It should be noted that entry flight path angle (EFPA) shows a range of correlation coefficient and mutual information because EFPA is used as a proxy for components of entry position and entry velocity in the Monte Carlo output.

Table 11-4 Peak Deceleration top influence model input for ATLO 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	cm_hc_add	0.40	-0.68
2	density_sum_40km_20km	0.30	-0.58
3	EFPA	0.19-0.12	0.42-0.15
4	density_sum_60km_40km	0.14	-0.29
5	density_sum_above_100km	0.12	-0.18
6	zcg_ei	0.13	0.15

Table 11-5 Peak Deceleration top influence model inputs for As-Built 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	cm_hc_add	0.40	-0.69
2	density_sum_40km_20km	0.32	-0.59
3	EFPA	0.19-0.12	0.42-0.15
4	density_sum_60km_40km	0.14	-0.29
5	density_sum_above_100km	0.11	-0.18
6	zcg_ei	0.13	0.14

Table 11-6 Peak Deceleration top influence model inputs for Flight Reference Delta.

Var Count	Input	Mutual Info	Correlation Coeff
1	cm_hc_add	0.40	-0.68
2	density_sum_40km_20km	0.30	-0.57
3	EFPA	0.21-0.12	0.40-0.16
4	density_sum_60km_40km	0.15	-0.27
5	density_sum_above_100km	0.13	-0.17
6	cn_hc_add	0.13	0.09

11.2.4 Peak Heat Rate

The Monte Carlo Filtering results for the Critical Metric Peak Heat Rate for the three InSight epochs are shown in Tables 11-7 – 11-9. The top three highest correlated inputs are atmosphere density parameters density_sum_80km_60km and density_sum_above_100km and EFPA. These parameters have the highest correlation because the integrated drag force in the upper levels of the atmosphere affect the rate of energy dissipation prior to peak heating and thus how much energy the entry vehicle carries into the denser altitudes of the atmosphere where peak heating occurs. Both hypersonic axial force coefficient multiplier (ca_hc_mult) and hypersonic pitch moment coefficient adder (cm_hc_add) also are significant influencers. The filtering of the three InSight epoch Monte Carlos returned the same ordered set of inputs of highest influence, and the values for mutual information and correlation coefficient show only minor variations across the three epochs.

Table 11-7 Peak Heat Rate top influence model inputs for ATLO 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	density_sum_80km_60km	0.15	-0.28
2	EFPA	0.15-0.09	0.28-0.17
3	density_sum_above_100km	0.13	-0.22
4	cm_hc_add	0.13	-0.21
5	ca_hc_mult	0.12	-0.19
6	density_sum_60km_40km	0.10	-0.10

Table 11-8 Peak Heat Rate top influence model inputs for As-Built 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	density_sum_80km_60km	0.14	-0.28
2	EFPA	0.15-0.09	0.28-0.09
3	density_sum_above_100km	0.13	-0.22
4	cm_hc_add	0.13	-0.22
5	ca_hc_mult	0.12	-0.19
6	density_sum_60km_40km	0.10	-0.10

Table 11-9 Peak Heat Rate top influence model inputs for Flight Reference Delta.

Var Count	Input	Mutual Info	Correlation Coeff
1	density_sum_80km_60km	0.15	-0.28
2	EFPA	0.17-0.12	0.28-0.10
3	density_sum_above_100km	0.14	-0.21
4	cm_hc_add	0.14	-0.21
5	ca_hc_mult	0.13	-0.19
6	density_sum_60km_40km	0.12	-0.09

11.2.5 Total Heating

The Monte Carlo Filtering results for the Critical Metric Total Heating for the three InSight epochs are shown in Tables 11-10 – 11-12. Interestingly, while both Peak Heat Rate and Total Heating are aerothermal heating metrics and share many of the same inputs of top influence, the ordering of the inputs is somewhat different. This may be because while Peak Heat Rate is an instantaneous metric, Total Heating is an integrated metric. For Total Heating, the input of strongest influence is the hypersonic pitch moment coefficient adder (cm_hc_add) with a correlation coefficient of 0.54 for epoch ATLO 2.0. In the Peak Heat Rate Monte Carlo Filtering results, the hypersonic pitch moment coefficient adder was the fourth most influential input parameter with a negative correlation coefficient of -0.21 for ATLO 2.0. Comparing all inputs of highest influence between Peak Heat Rate and Total Heating, note that the sign of the correlation coefficient shows there is an opposite correlation for each input between the two metrics. This is consistent with the previously mentioned well known opposing relationship between Peak Heat Rate and Total Heating: a rise in one metric is accompanied with a drop in the other metric. Again, the Monte Carlo Filtering results across the three InSight epochs are nearly the same with only small differences in correlation coefficient values.

In assessing the Monte Carlo Filtering results for Peak Heat Rate and Total Heating, it was observed there is a missing simulation model that is central to these two aerothermal Critical Metrics. The well-known Sutton-Graves aerothermal heating model is used in the Monte Carlos to compute stagnation point heating. Because there are no varied model parameters for the Monte Carlo Sutton-Graves model, the model does not appear in the Monte Carlo Filtering results, and therefore is not included in DOE regression modeling or the Model Confidence Assessment. This observation shows an area of future exploration for methodology development. Because the Sutton-Graves model has no dispersed parameters, it implies there is no modeling uncertainty. But given its central role in the modeling of aerothermal heating, should there be a process to include it in the sizing of Target Margin?

Table 11-10 Total Heating top influence model inputs for ATLO 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	cm_hc_add	0.26	0.54
2	density_sum_40km_20km	0.25	0.52
3	EFPA	0.16-0.11	-0.34-0.17
4	density_sum_60km_40km	0.15	0.34
5	density_sum_80km_60km	0.15	0.30
6	ca_hc_mult	0.14	-0.29

Table 11-11 Total Heating top influence model inputs for As-Built 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	cm_hc_add	0.26	0.53
2	density_sum_40km_20km	0.26	0.52
3	EFPA	0.16-0.11	-0.34-0.17
4	density_sum_60km_40km	0.15	0.33
5	density_sum_80km_60km	0.15	0.30
6	ca_hc_mult	0.14	-0.30

Table 11-12 Total Heating top influence model inputs for Flight Reference Delta.

Var Count	Input	Mutual Info	Correlation Coeff
1	cm_hc_add	0.27	0.53
2	density_sum_40km_20km	0.25	0.51
3	EFPA	0.17-0.12	-0.34-0.13
4	density_sum_60km_40km	0.16	0.33
5	ca_hc_mult	0.15	-0.31
6	density_sum_80km_60km	0.16	0.30

11.2.6 Parachute Inflation Load

The Monte Carlo Filtering results for the Critical Metric Parachute Inflation Load for the three InSight epochs are shown in Tables 11-13 – 11-15. For all three epochs there is a strong correlation between the magnitude of dispersed parachute aerodynamic drag (chute_aero_disp) and Parachute Inflation Load. This is because the opening load, or force, is directly proportional to the parachute drag coefficient by the familiar drag force equation [34]

$$D = q_{\infty} S C_d \qquad (11\text{-}4)$$

where D is drag force, q_{∞} is freestream dynamic pressure, S is drag area and C_d is the drag coefficient. The InSight parachute deployment software trigger is a hybrid trigger that deploys on either sensed deceleration (as a proxy for dynamic pressure) or propagated velocity. It is tuned in such a way to deploy nearly always on deceleration, with the velocity trigger used to protect against deploying at a Mach number that exceeds the parachute specification. Sensed entry vehicle deceleration is a proxy for dynamic pressure via a proportionality with the entry vehicle drag coefficient. With perfect knowledge of entry vehicle drag coefficient, the parachute would always deploy at the targeted freestream dynamic pressure, q_{∞}. However, because there is always some uncertainty in entry vehicle drag, there is always some variation in the dynamic pressure at which the parachute will be deployed and thus uncertainty in Parachute Inflation Load. This is reflected in the fact that the entry vehicle supersonic axial force coefficient multiplier is the third input parameter of highest influence. The pitch moment coefficient adders and density parameters all affect the trajectory flown before the parachute is deployed, and thus the deployment conditions.

Table 11-13 Parachute Inflation Load top influence model inputs for ATLO 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	chute_aero_disp	0.69	0.84
2	density_sum_40km_20km	0.17	-0.33
3	ca_mc_mult	0.16	-0.33
4	cm_hc_add	0.15	-0.28
5	cm_mc_add	0.11	-0.15
6	density_sum_60km_40km	0.09	-0.11

Table 11-14 Parachute Inflation Load top influence model inputs for As-Built 2.0.

Var Count	Input	Mutual Info	Correlation Coeff
1	chute_aero_disp	0.72	0.85
2	ca_mc_mult	0.17	-0.33
3	density_sum_40km_20km	0.17	-0.31
4	cm_hc_add	0.15	-0.26
5	cm_mc_add	0.17	-0.15
6	density_sum_60km_40km	0.13	-0.10

Table 11-15 Parachute Inflation Load top influence model inputs for Flight Reference Delta.

Var Count	Input	Mutual Info	Correlation Coeff
1	chute_aero_disp	0.70	0.83
2	density_sum_40km_20km	0.20	-0.35
3	ca_mc_mult	0.18	-0.33
4	cm_hc_add	0.18	-0.31
5	cm_mc_add	0.14	-0.19
6	density_sum_60km_40km	0.13	-0.13

The next section will use the results of Monte Carlo Filtering to generate second-order regression models via the DOE process.

11.3 Design of Experiments (DOE) Regression Model Results

Recall from Section 6 that DOE regression modeling is used to better quantify the level of influence each input has on Critical Metric performance, and to develop a regression model to use in quantifying Target Margin. A 3-level, full-factorial DOE is performed resulting in a regression model of the form:

$$\begin{gathered}\hat{Y} = b_0 + b_1x_1 + b_2x_2 + \cdots + b_px_p \\ +b_{12}x_1x_2 + b_{13}x_1x_3 + \cdots + b_{p-1}x_{p-1}x_p \\ +b_{11}x_1^2 + b_{22}x_2^2 + \cdots + b_{pp}x_p^2\end{gathered} \qquad (11\text{-}5)$$

A t-test and p-value are then employed to determine the statistical significance of each term in the regression model. This section provides the implementation details and results as applied to the three InSight epochs.

11.3.1 Implementation of 3-Level DOE

For each of three InSight epochs, a 3-level, full-factorial DOE analysis was run for each Critical Metric resulting in twelve total DOE analyses. While the data sets used for Monte Carlo Filtering were generated during the InSight project, the DOE analysis data sets were generated as part of this dissertation work. The same simulation tool, POST2, used by the InSight project during the Monte Carlo Simulation Campaigns was used to perform the DOE analysis. Thanks to simulation archiving, the DOE analysis for each InSight epoch was performed with the exact simulation configuration used to perform the Monte Carlo analysis during that InSight epoch. It is an absolute requirement that the simulation environment that generated the Monte Carlo Filtering results be identical to the simulation environment used to perform the DOE analysis.

As discussed in Section 6, for a 3-level, full-factorial DOE analysis where there are *p* input variables, an analysis will require 3^p experiments or simulation runs. Table 11-16 shows an example of a two-variable experimental design, adapted from Rekab and Shaikh [24], having 3^2 experiments or runs to provide all possible combinations of levels for two variables. For six inputs of highest influence serving as DOE variables, a total of 3^6, or 729 simulation runs are required for each of the twelve DOE analyses, as was stated earlier in Section 6. A 3^6 run table was constructed for each DOE analysis to guide simulation set-up. The specification for the first 27 of the 729 runs for Critical Metric Peak Deceleration and epoch As-Built 2.0 is shown in Table 11-18, as an example. Note the ordering of the inputs of highest influence in Table 11-11 is not the same as the ordering in the Monte Carlo Filtering results (Table 11-8). Ordering of variables in the construction of Table 11-18 does not affect the DOE analysis outcome.

Table 11-16 A 3^2 experimental design plan. [24]

Run	Variable 1	Variable 2
1	-1	-1
2	0	-1
3	1	-1
4	-1	0
5	0	0
6	1	0
7	-1	1
8	0	1
9	1	1

11.3.1.1 DOE Levels Represented in Atmosphere Modeling

The process of parametrizing a continuous atmosphere by using the arithmetic sum of density across altitude bands was described in Section 5.3 and utilized in the Monte Carlo Filtering process. Another challenge is creating levels for atmosphere input in DOE analysis that correspond to the same altitude bands used in the Monte Carlo Filtering. To address this challenge, the dispersed Martian atmosphere model used by InSight was used to generate density increase scaling factors based on the 3-sigma density bounds of the ensemble of atmosphere profiles that make up the dispersed InSight atmosphere model. This is depicted in Figure 11-4 via an atmosphere density tornado plot. The tornado plot represents the dispersed atmosphere model used in the Monte Carlo simulation where each dispersed density profile is plotted as a percent difference in density from the model's nominal profile. In Figure 11-4, the red boundary lines show the 3-sigma dispersion in density by altitude, as a percentage of the nominal density at that altitude. Using the 3-sigma boundaries in Figure 11-4, a percent density scaling factor for each altitude band was derived and is shown in Table 11-17. Using names similar to the Monte Carlo Filtering atmosphere variables, the scaled atmosphere bands are referred to as density_40km_20km, density 60km_40km, density_80km_60km, density_100km_80km, and density_above_100km. Using the scale factors, atmosphere profiles were generated by scaling density across altitude bands with the nominal InSight atmosphere profile serving as the basis for all profiles. Given there are six designated altitude bands from the original atmosphere parameterization used in Monte Carlo Filtering, and there are three possible DOE levels per band, there are 3^6, or 729 possible atmosphere profiles needed to capture all combinations of levels across all altitude bands. To produce the 729 scaled atmosphere density profiles needed for the DOE analysis, a MATLAB script was created that receives as input the nominal InSight atmosphere profile and outputs the 729 profiles containing the appropriate level scaling by altitude.

(This space intentionally left blank)

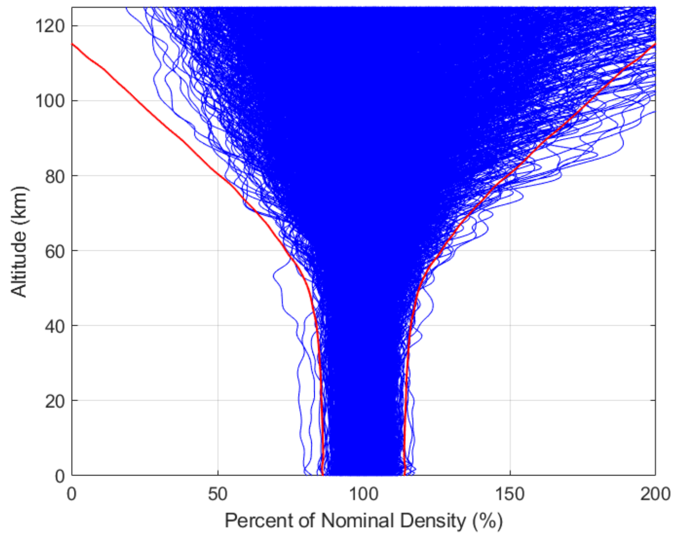


Figure 11-4 Dispersed atmosphere model tornado plot with 3-sigma boundaries.

Table 11-17 Atmosphere altitude band density scaling for 3-level DOE.

Var Count	Altitude Band	Density Scaling			DOE Variable Name
		Low Level (-1)	Mid Level (0)	High Level (+1)	
1	0 km – 20 km	-14.0%	0.0%	14.0%	density_20km_0km
2	20 km – 40 km	-15.0%	0.0%	15.0%	density_40km_20km
3	40 km – 60 km	-19.0%	0.0%	19.0%	density_60km_40km
4	60 km – 80 km	-36.0%	0.0%	36.0%	density_80km_60km
5	80 km – 100 km	-63.0%	0.0%	63.0%	density_100km_80km
6	Above 100 Km	-92.0%	0.0%	90.0%	density_above_100km

(This space intentionally left blank)

Table 11-18 Partial DOE run table for Peak Deceleration and epoch As-Built 2.0 (3^6 experimental design plan).

Case No.	Single Run Caases - All Combinations											
	cm_hc_add		density_40km_20km		EFPA		density_60km_40km		zcg_ei		density_above_100km	
	Level	Input	Level	Input	Level	Input	Level	Input	Level	Input	Level	Input
1	Low	-1	Low	atm_0-1-100-1_profile	Low	-12.21°	Low	atm_0-1-100-1_profile	Low	-0.00085971	Low	atm_0-1-100-1_profile
2	Low	-1	Low	atm_0-1-1000_profile	Low	-12.21°	Low	atm_0-1-1000_profile	Low	-0.00085971	Nom	atm_0-1-1000_profile
3	Low	-1	Low	atm_0-1-1001_profile	Low	-12.21°	Low	atm_0-1-1001_profile	Low	-0.00085971	High	atm_0-1-1001_profile
4	Low	-1	Low	atm_0-1-100-1_profile	Low	-12.21°	Low	atm_0-1-100-1_profile	Nom	0.00016393	Low	atm_0-1-100-1_profile
5	Low	-1	Low	atm_0-1-1000_profile	Low	-12.21°	Low	atm_0-1-1000_profile	Nom	0.00016393	Nom	atm_0-1-1000_profile
6	Low	-1	Low	atm_0-1-1001_profile	Low	-12.21°	Low	atm_0-1-1001_profile	Nom	0.00016393	High	atm_0-1-1001_profile
7	Low	-1	Low	atm_0-1-100-1_profile	Low	-12.21°	Low	atm_0-1-100-1_profile	High	0.0012	Low	atm_0-1-100-1_profile
8	Low	-1	Low	atm_0-1-1000_profile	Low	-12.21°	Low	atm_0-1-1000_profile	High	0.0012	Nom	atm_0-1-1000_profile
9	Low	-1	Low	atm_0-1-1001_profile	Low	-12.21°	Low	atm_0-1-1001_profile	High	0.0012	High	atm_0-1-1001_profile
10	Low	-1	Low	atm_0-1000-1_profile	Low	-12.21°	Nom	atm_0-1000-1_profile	Low	-0.00085971	Low	atm_0-1000-1_profile
11	Low	-1	Low	atm_0-10000_profile	Low	-12.21°	Nom	atm_0-10000_profile	Low	-0.00085971	Nom	atm_0-10000_profile
12	Low	-1	Low	atm_0-10001_profile	Low	-12.21°	Nom	atm_0-10001_profile	Low	-0.00085971	High	atm_0-10001_profile
13	Low	-1	Low	atm_0-1000-1_profile	Low	-12.21°	Nom	atm_0-1000-1_profile	Nom	0.00016393	Low	atm_0-1000-1_profile
14	Low	-1	Low	atm_0-10000_profile	Low	-12.21°	Nom	atm_0-10000_profile	Nom	0.00016393	Nom	atm_0-10000_profile
15	Low	-1	Low	atm_0-10001_profile	Low	-12.21°	Nom	atm_0-10001_profile	Nom	0.00016393	High	atm_0-10001_profile
16	Low	-1	Low	atm_0-1000-1_profile	Low	-12.21°	Nom	atm_0-1000-1_profile	High	0.0012	Low	atm_0-1000-1_profile
17	Low	-1	Low	atm_0-10000_profile	Low	-12.21°	Nom	atm_0-10000_profile	High	0.0012	Nom	atm_0-10000_profile
18	Low	-1	Low	atm_0-10001_profile	Low	-12.21°	Nom	atm_0-10001_profile	High	0.0012	High	atm_0-10001_profile
19	Low	-1	Low	atm_0-1100-1_profile	Low	-12.21°	High	atm_0-1100-1_profile	Low	-0.00085971	Low	atm_0-1100-1_profile
20	Low	-1	Low	atm_0-11000_profile	Low	-12.21°	High	atm_0-11000_profile	Low	-0.00085971	Nom	atm_0-11000_profile
21	Low	-1	Low	atm_0-11001_profile	Low	-12.21°	High	atm_0-11001_profile	Low	-0.00085971	High	atm_0-11001_profile
22	Low	-1	Low	atm_0-1100-1_profile	Low	-12.21°	High	atm_0-1100-1_profile	Nom	0.00016393	Low	atm_0-1100-1_profile
23	Low	-1	Low	atm_0-11000_profile	Low	-12.21°	High	atm_0-11000_profile	Nom	0.00016393	Nom	atm_0-11000_profile
24	Low	-1	Low	atm_0-11001_profile	Low	-12.21°	High	atm_0-11001_profile	Nom	0.00016393	High	atm_0-11001_profile
25	Low	-1	Low	atm_0-1100-1_profile	Low	-12.21°	High	atm_0-1100-1_profile	High	0.0012	Low	atm_0-1100-1_profile
26	Low	-1	Low	atm_0-11000_profile	Low	-12.21°	High	atm_0-11000_profile	High	0.0012	Nom	atm_0-11000_profile
27	Low	-1	Low	atm_0-11001_profile	Low	-12.21°	High	atm_0-11001_profile	High	0.0012	High	atm_0-11001_profile

A plot showing the scaling factors for a representative atmosphere profile from the 729 atmosphere profiles generated for the DOE analysis is shown in Figure 11-5. Overlaid the dispersed atmosphere tornado plot are scaling factors by altitude represented by the black stepwise curve. The scaling factors represent DOE levels for the same atmosphere bands used in Monte Carlo Filtering parameterization, and are scaled by the factors from Table 11-17. Note the scaling variation in the atmosphere profiles is not representative of a possible Mars atmosphere, and the scaling was applied to just density with no adjustments to temperature or pressure, which results in a hydrostatically inconsistent atmosphere. Because density is the primary driver of EDL performance in simulation, scaling density only is deemed acceptable. While the step functions synthetically inserted into the density profiles make for a non-realizable Mars atmosphere, for the purposes of DOE and the intention to providing a system response at the edges of possible atmosphere dispersion, the scaled atmosphere profiles provide an altitude dependent impulse to achieve a system response.

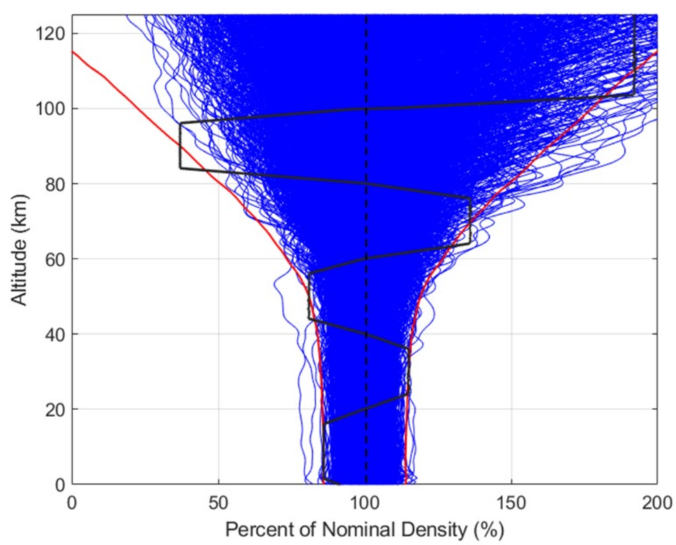


Figure 11-5 Dispersed atmosphere model with representative altitude band scaling factors.

11.3.2 DOE Results for InSight

Over the course of one month, all simulation cases were run for the 12 DOE analyses (a total of 12 x 729, or 8748 simulation runs). The data sets were imported into MATLAB and processed using the linear algebra formulation outlined in Section 6 to generate regression models. Each six-variable second-order regression model has 28 total terms. Additionally, the t-test and p-value were applied to determine the statistical significance of each model term. The results from the DOE analyses for InSight Critical Metrics are provided in the subsections that follow.

11.3.2.1 DOE Results for Peak Deceleration

The regression models generated for Peak Deceleration for the three InSight epochs are shown in Tables 11-19 – 11-21. The first two columns in the tables are the indices of the terms corresponding to the index convention in the regression model formulation in Equation 11-5. An Index 1 value of zero indicates a linear term. When Index 1 and Index 2 are the same value, the term is second-order. When Index 1 and Index 2 are different values and Index 1 is nonzero, the term is an interactive term. The last column, labeled Input, provides the input variable name corresponding to the index number. The Coefficient column provides the numeric value of each regression model term coefficient, and the t-test and p-value columns provide the results of the significance tests. The Significant? column lists a "yes" or "no" depending on the results of the significance tests. As described in Section 6, in implementing a two-sided t-test, the degrees of freedom (DOF) for the regression model are given by

$$DOF = n - L - 1 \tag{11-6}$$

where L is the number of non-constant coefficients in the regression model coefficient vector b, and n is the number of experiments in Y, the DOE response vector. For all InSight Critical Metric regression models derived from DOE Analysis, the number of experiments is 729 and the number of non-constant regression terms is 27. Therefore, the degrees of freedom are 701. A two-sided t-test with 701 degrees of freedom and a significance level of 0.05 has a critical value T^* of 1.9634. Therefore, any t-test in Table 11-19 – Table 11-33 resulting in a value below T^* is deemed not significant and is not included in the regression model. Correspondingly, any p-value above 0.05 is also deemed not significant (again, see Section 6).

The regression models are used by the Target Margin Sizing Tool (TMST) to provide a visual reference during a Target Margin assessment, as was described in Section 6.1.2. Figures 11-6 – 11-8 show the regression model TMST displays for Peak Deceleration for the three InSight epochs, for the first 15 non-constant terms of the regression model. When TMST loads a regression model, it initially orders the linear term coefficients from largest magnitude to smallest magnitude. This is followed by all nonlinear term coefficients ordered from largest magnitude to smallest magnitude. All terms deemed not significant have coefficients set to zero. Linear term coefficients are listed in order first in the TMST Assessment Main Window so those rows can be used for the Input Model Confidence Assessment, the subject of Section 11.5. Also included in the TMST regression model display are the regression model tornado plots, which are used during the assessment to visualize and understand the importance of a particular regression model term on the Critical Metric response.

As stated in Section 6, the magnitude of the regression model term coefficients is a measure of the influence the corresponding input has on Critical Metric performance. As such, the ordering of the linear term coefficients can be compared to the results of the Monte Carlo Filtering where inputs variables are ordered based on correlation coefficient. Comparing the order of the linear terms in Figure 11-6 with Table 11-4, it can be seen the ordering of the inputs is not the same, with density_40km_20km and density_60km_40km each being order one slot higher in the ordered list in Figure 11-6. This is because the DOE coefficients are a measure of influence on Critical Metric performance, while the results of the Monte Carlo Filtering are a measure of correlation. A variable could have perfect correlation to the output, but not significant influence on performance. This is why using correlation analysis to identify Model Inputs of Highest Influence has limitations, the subject of which is discussed further in Section 11.3.2.5

Table 11-22 provides a comparison of the linear terms of the Peak Deceleration regression model across InSight epochs. The input names and coefficient values are listed for each regression model linear term. For ATLO 2.0 and As-Built 2.0 epochs, the percent difference from the Flight Reference Delta linear term coefficients (in the light blue shaded column) is provided. The Flight Reference Delta is chosen as the reference because it represents the most mature version of the EDL simulation having been created just prior to the landing event. For the first four terms, the variation is of the order 1.5% or less, so there is very good consistency in results among the terms with the largest influence on the Critical Metric response. As was already discussed in Section 11.2.3, the fifth term for ATLO 2.0 and As-Built 2.0 is zcg_ei, which became cn_hc_add in the Flight Reference Delta epoch. Also, of note is the final term, density_above_100km, which tested not significant in ATLO 2.0 and As-Built 2.0, but then tested as significant in the Flight Reference Epoch. Exploration of the difference is future work.

In all epochs the linear terms have significantly greater influence on the regression model response than the nonlinear terms. Of the 27 non-constant regression model terms for Peak Deceleration, 12 tested not significant for ATLO 2.0, 12 tested not significant for As-Built 2.0 and 12 tested not significant for Flight Reference Delta. As was stated earlier, the not significant terms are set to zero in applying the regression model.

Table 11-19 DOE results for Peak Deceleration, epoch ATLO 2.0.

Term Indices		Peak Deceleration - ALTO 2.0				
Index 1	Index 2	Coefficient (Earth g)	T-Test	P Value	Significant?	Input Name
NA	NA	7.46735	NA	NA	NA	NA
0	1	-0.46414	2637.00	0.000	Yes	cm_hc_add
0	2	0.73767	482.53	0.000	Yes	density_40km_20km
0	3	-0.33384	766.89	0.000	Yes	EFPA
0	4	-0.35112	347.06	0.000	Yes	density_60km_40km
0	5	-0.00008	365.02	0.000	Yes	zcg_ei
0	6	0.00037	0.08	0.934	No	density_above_100km
1	2	0.04981	0.38	0.701	No	
1	3	0.00604	42.28	0.000	Yes	
1	4	-0.03956	5.12	0.000	Yes	
1	5	0.00001	33.58	0.000	Yes	
1	6	0.00020	0.01	0.994	No	
2	3	-0.00483	0.17	0.862	No	
2	4	-0.04445	4.10	0.000	Yes	
2	5	0.00001	37.73	0.000	Yes	
2	6	0.00001	0.01	0.990	No	
3	4	-0.00004	0.01	0.993	No	
3	5	0.00002	0.03	0.974	No	
3	6	0.00084	0.01	0.989	No	
4	5	0.00001	0.71	0.478	No	
4	6	-0.00024	0.01	0.993	No	
5	6	-0.00005	0.20	0.842	No	
1	1	-0.00638	0.05	0.964	No	
2	2	0.03352	3.83	0.000	Yes	
3	3	0.00918	20.12	0.000	Yes	
4	4	0.00843	5.51	0.000	Yes	
5	5	-0.00001	5.06	0.000	Yes	
6	6	-0.00061	0.01	0.996	No	

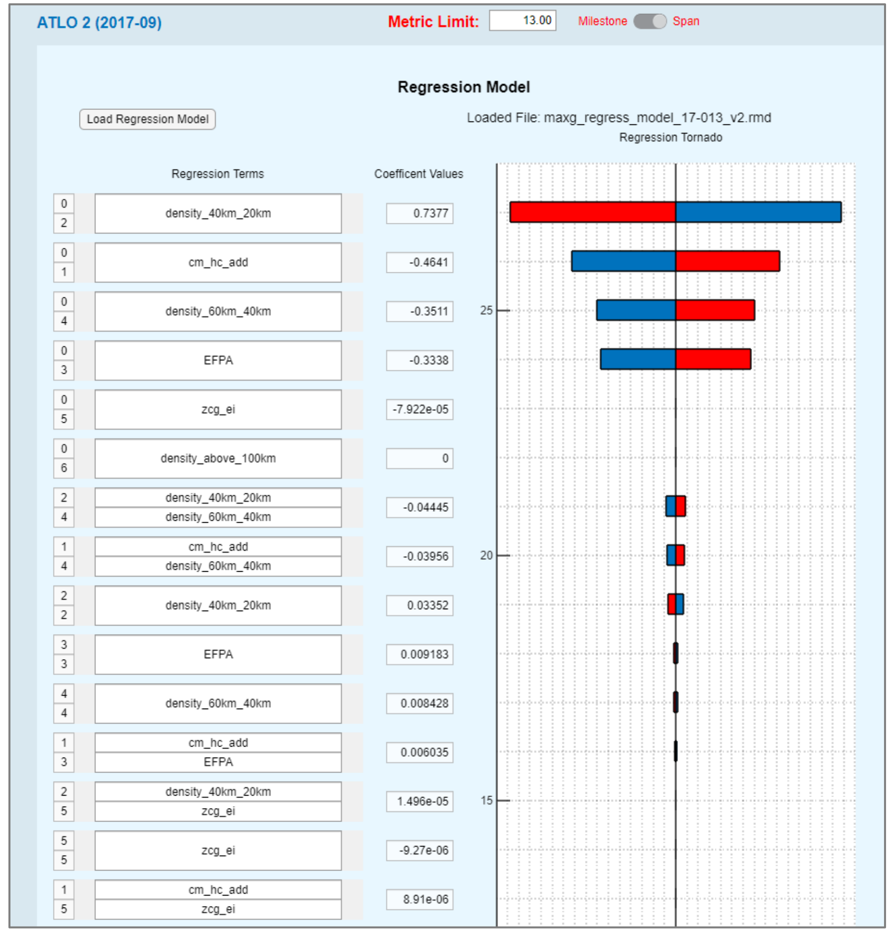


Figure 11-6 Peak Deceleration ordered regression terms and regression tornado, epoch ATLO 2.0.

Table 11-20 DOE results for Peak Deceleration, epoch As-Built 2.0.

Term Indices		Peak Deceleration - As-Built 2.0				
Index 1	Index 2	Coefficient (Earth g)	T-Test	P Value	Significant?	Input Name
NA	NA	7.47659	NA	NA	NA	NA
0	1	-0.46446	2357.98	0.000	Yes	cm_hc_add
0	2	0.74254	431.23	0.000	Yes	density_40km_20km
0	3	-0.33397	689.41	0.000	Yes	EFPA
0	4	-0.35432	310.08	0.000	Yes	density_60km_40km
0	5	0.00003	328.97	0.000	Yes	zcg_ei
0	6	-0.00115	0.03	0.980	No	density_above_100km
1	2	0.04840	1.06	0.288	No	
1	3	0.00695	36.69	0.000	Yes	
1	4	-0.03956	5.27	0.000	Yes	
1	5	-0.00001	29.99	0.000	Yes	
1	6	-0.00003	0.01	0.993	No	
2	3	-0.00727	0.02	0.981	No	
2	4	-0.04404	5.51	0.000	Yes	
2	5	-0.00001	33.38	0.000	Yes	
2	6	-0.00003	0.01	0.994	No	
3	4	-0.00030	0.02	0.981	No	
3	5	0.00001	0.23	0.821	No	
3	6	-0.00002	0.00	0.997	No	
4	5	0.00000	0.02	0.988	No	
4	6	0.00004	0.00	0.998	No	
5	6	-0.00002	0.03	0.974	No	
1	1	-0.00633	0.01	0.989	No	
2	2	0.03258	3.39	0.001	Yes	
3	3	0.01089	17.46	0.000	Yes	
4	4	0.00838	5.84	0.000	Yes	
5	5	0.00000	4.49	0.000	Yes	
6	6	0.00041	0.00	0.999	No	

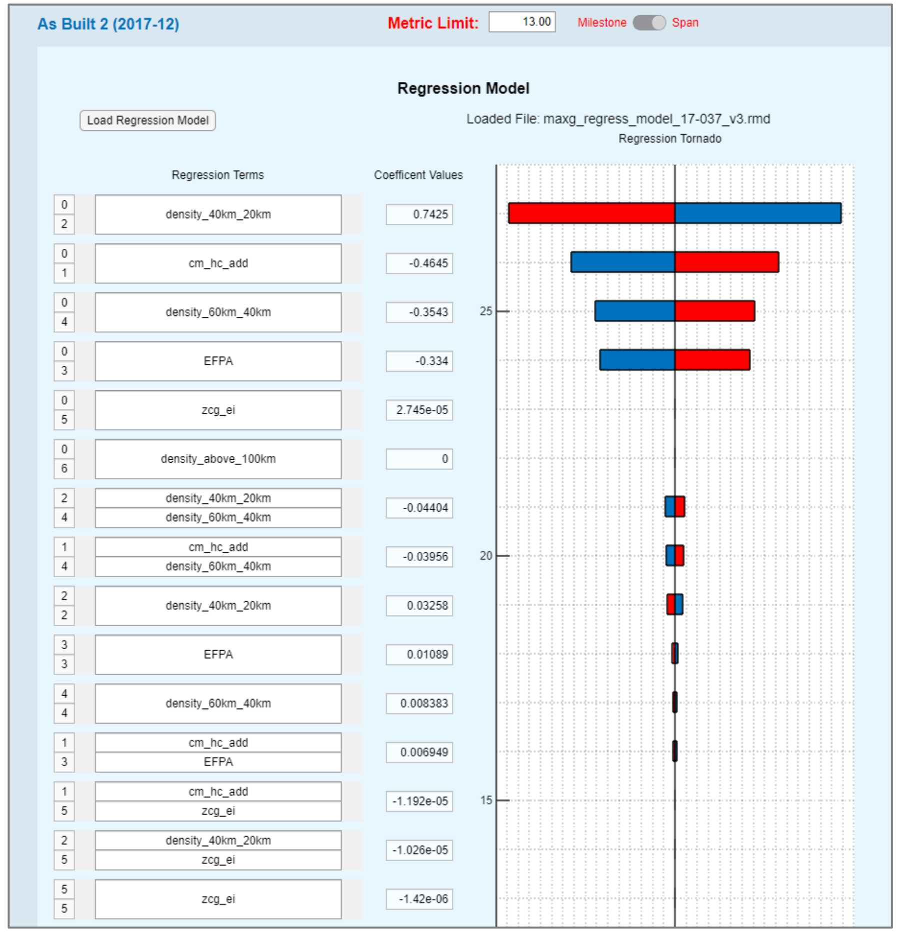


Figure 11-7 Peak Deceleration ordered regression terms and regression tornado, epoch As-Built 2.0.

Table 11-21 DOE results for Peak Deceleration, epoch Flight Reference Delta.

Term Indices		Peak Deceleration - Flight Reference Delta				
Index 1	Index 2	Coefficient (Earth g)	T-Test	P Value	Significant?	Input Name
NA	NA	7.49928	NA	NA	NA	NA
0	1	-0.46757	2284.19	0.000	Yes	cm_hc_add
0	2	0.74418	419.27	0.000	Yes	density_40km_20km
0	3	-0.33192	667.29	0.000	Yes	EFPA
0	4	-0.35605	297.63	0.000	Yes	density_60km_40km
0	5	0.06655	319.26	0.000	Yes	cn_hc_add
0	6	-0.00137	59.68	0.000	Yes	density_above_100km
1	2	0.04714	1.23	0.218	No	
1	3	0.00785	34.52	0.000	Yes	
1	4	-0.03988	5.75	0.000	Yes	
1	5	-0.01079	29.20	0.000	Yes	
1	6	0.00033	7.90	0.000	Yes	
2	3	-0.00837	0.24	0.808	No	
2	4	-0.04610	6.13	0.000	Yes	
2	5	-0.00420	33.75	0.000	Yes	
2	6	-0.00015	3.08	0.002	Yes	
3	4	0.00012	0.11	0.910	No	
3	5	-0.00204	0.09	0.930	No	
3	6	-0.00036	1.49	0.137	No	
4	5	0.00497	0.26	0.792	No	
4	6	0.00001	3.64	0.000	Yes	
5	6	0.00014	0.01	0.995	No	
1	1	-0.01963	0.10	0.920	No	
2	2	0.03429	10.16	0.000	Yes	
3	3	0.01613	17.75	0.000	Yes	
4	4	0.00853	8.35	0.000	Yes	
5	5	0.00037	4.42	0.000	Yes	
6	6	0.00036	0.19	0.848	No	

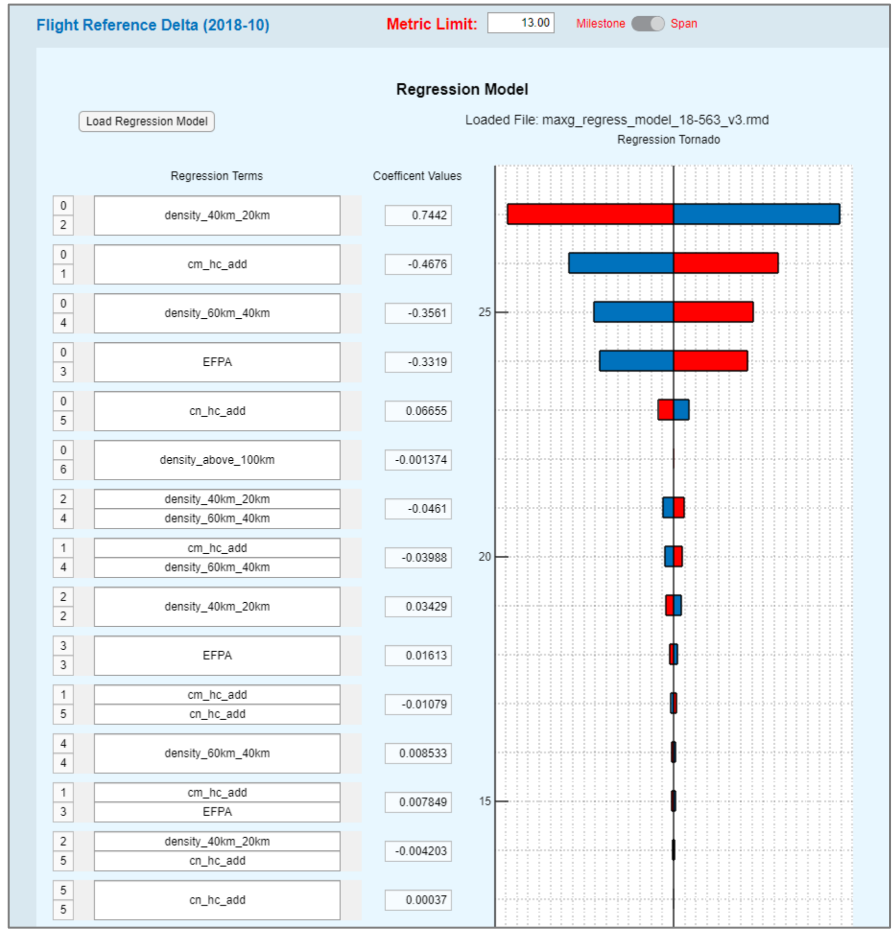


Figure 11-8 Peak Deceleration ordered regression terms and regression tornado, epoch Flight Reference Delta.

Table 11-22 Comparison of Peak Deceleration regression model linear terms across epochs.

Input	Peak Deceleration Linear Coefficient Values & Percent Difference from Flight Reference Delta				
	ATLO 2.0		As-Built 2.0		Flight Ref. Delta
	Coefficient (Earth g)	Percent Difference	Coefficient (Earth g)	Percent Difference	Coefficient (Earth g)
cm_hc_add	-0.46414	-0.73%	-0.46446	-0.67%	-0.46757
density_40km_20km	0.73767	-0.87%	0.74254	-0.22%	0.74418
EFPA	-0.33384	0.58%	-0.33397	0.62%	-0.33192
density_60km_40km	-0.35112	-1.39%	-0.35432	-0.49%	-0.35605
zcg_ei / cn_hc_add	-0.00008	NA	0.00003	NA	0.06655
density_above_100km	0.00036922	NA	-0.00114575	NA	-0.00137

11.3.2.2 DOE Results for Peak Heat Rate

The regression models generated for Peak Heat Rate for the three InSight epochs are shown in Tables 11-23 – 11-25. Figures 11-6 – 11-8 show the regression model TMST displays for Peak Heat Rate for the three InSight epochs, again for the first 15 non-constant terms of the regression model. Comparing the DOE results with the Monte Carlo Filtering results, the order of inputs of highest influence has changed with EFPA being the input of highest influence. Again, this is a result of the difference between correlation and direct influence. Across all epochs, the DOE regression modeling returned the same influence ordering of linear terms and all linear terms were tested as significant. Across the epochs, as seen in Table 11-26, the linear term coefficients show variation at or below 4.72. Easily seen in the regression tornados in Figures 11-9 – 11-11 is the second-order term for density_60km_40km has greater influence on Peak Heat rate performance than any linear term. The strong nonlinear influence for density_60km_40km is likely due to the fact that InSight peak heat rate occurs in this altitude band, and that aerothermal heat rate is directly proportional to the square root of density, and therefore the second-order density DOE input in the region of peak heat rate has strong influence. In all epochs, the second-order term for

Table 11-23 DOE results for Peak Heat Rate, epoch ATLO 2.0.

Term Indices		Peak Heat Rate - ALTO 2.0				
Index 1	Index 2	Coefficient (W/cm^2)	T-Test	P Value	Significant?	Input Name
NA	NA	45.72906	NA	NA	NA	NA
0	1	-0.54775	8303.07	0.000	Yes	density_80km_60km
0	2	-0.01129	292.79	0.000	Yes	density_above_100km
0	3	-0.98094	6.03	0.000	Yes	EFPA
0	4	-0.61370	524.34	0.000	Yes	ca_hc_mult
0	5	-0.57294	328.04	0.000	Yes	cm_hc_add
0	6	-0.78174	306.26	0.000	Yes	density_60km_40km
1	2	-0.00022	417.86	0.000	Yes	
1	3	-0.00499	0.10	0.922	No	
1	4	-0.00937	2.18	0.030	Yes	
1	5	-0.02654	4.09	0.000	Yes	
1	6	0.00501	11.58	0.000	Yes	
2	3	0.00013	2.19	0.029	Yes	
2	4	-0.00008	0.06	0.955	No	
2	5	-0.00006	0.03	0.973	No	
2	6	0.00000	0.03	0.978	No	
3	4	-0.00371	0.00	0.999	No	
3	5	-0.05125	1.62	0.106	No	
3	6	0.16239	22.37	0.000	Yes	
4	5	-0.01319	70.87	0.000	Yes	
4	6	0.07655	5.76	0.000	Yes	
5	6	0.13613	33.41	0.000	Yes	
1	1	0.00383	59.41	0.000	Yes	
2	2	0.00029	1.18	0.238	No	
3	3	-0.01217	0.09	0.928	No	
4	4	0.00831	3.76	0.000	Yes	
5	5	0.21915	2.57	0.011	Yes	
6	6	1.66267	67.63	0.000	Yes	

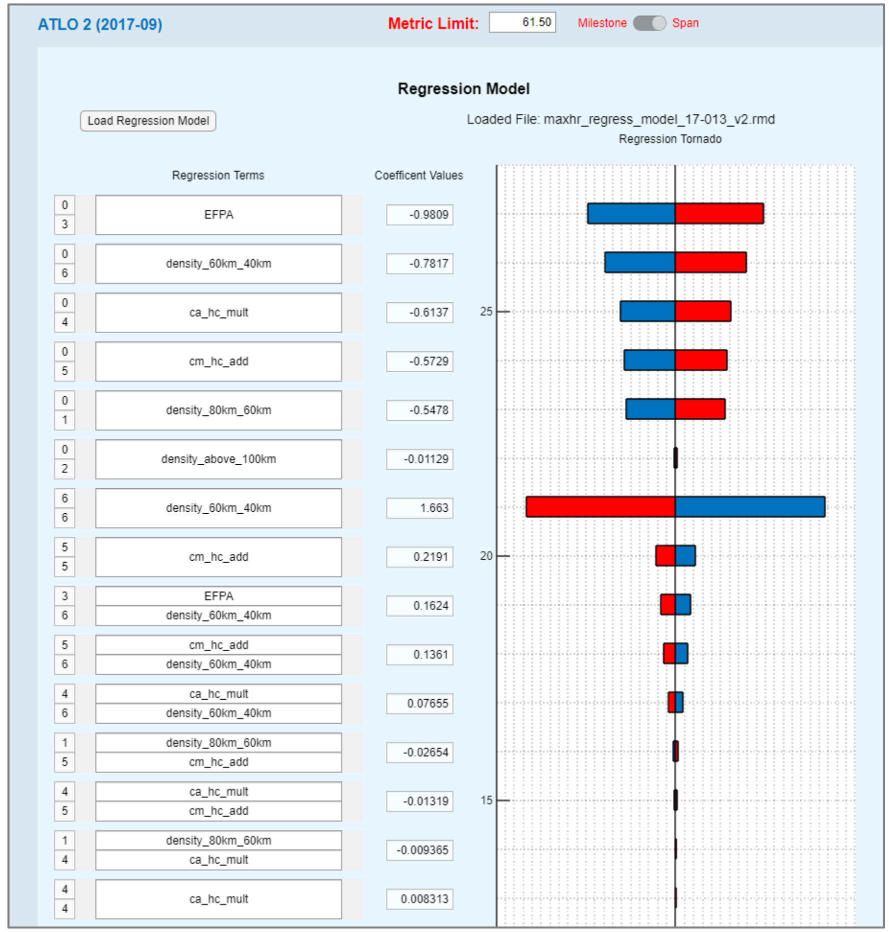


Figure 11-9 Peak Heat Rate ordered regression terms and regression tornado, epoch ATLO 2.0.

Table 11-24 DOE results for Peak Heat Rate, epoch As-Built 2.0.

Term Indices		Peak Heat Rate - As-Built 2.0				
Index 1	Index 2	Coefficient (W/cm²)	T-Test	P Value	Significant?	Input Name
NA	NA	45.77923	NA	NA	NA	NA
0	1	-0.54134	8309.79	0.000	Yes	density_80km_60km
0	2	-0.01091	289.28	0.000	Yes	density_above_100km
0	3	-0.97339	5.83	0.000	Yes	EFPA
0	4	-0.61057	520.16	0.000	Yes	ca_hc_mult
0	5	-0.59887	326.27	0.000	Yes	cm_hc_add
0	6	-0.78458	320.02	0.000	Yes	density_60km_40km
1	2	0.00026	419.26	0.000	Yes	
1	3	-0.00451	0.11	0.910	No	
1	4	-0.00905	1.97	0.050	Yes	
1	5	-0.02671	3.95	0.000	Yes	
1	6	0.00366	11.65	0.000	Yes	
2	3	0.00008	1.60	0.111	No	
2	4	-0.00008	0.04	0.971	No	
2	5	-0.00003	0.03	0.974	No	
2	6	-0.00005	0.01	0.988	No	
3	4	-0.00328	0.02	0.981	No	
3	5	-0.05087	1.43	0.153	No	
3	6	0.15985	22.20	0.000	Yes	
4	5	-0.01364	69.75	0.000	Yes	
4	6	0.07562	5.95	0.000	Yes	
5	6	0.13808	32.99	0.000	Yes	
1	1	-0.00302	60.24	0.000	Yes	
2	2	0.00050	0.93	0.352	No	
3	3	-0.01123	0.16	0.877	No	
4	4	0.00860	3.47	0.001	Yes	
5	5	0.07541	2.65	0.008	Yes	
6	6	1.66282	23.27	0.000	Yes	

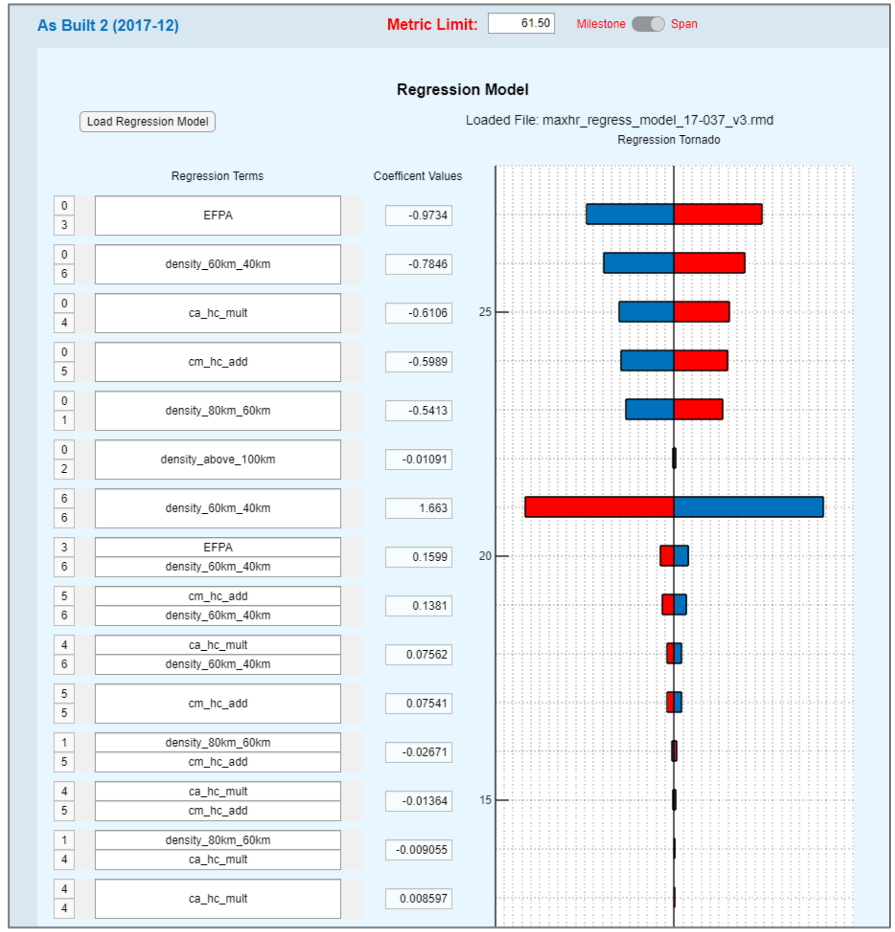


Figure 11-10 Peak Heat Rate ordered regression terms and regression tornado, epoch As-Built 2.0

Table 11-25 DOE results for Peak Heat Rate, epoch Flight Reference Delta.

Term Indices		Peak Heat Rate - Flight Reference Delta				
Index 1	Index 2	Coefficient (W/cm^2)	T-Test	P Value	Significant?	Input Name
NA	NA	45.58880	NA	NA	NA	NA
0	1	-0.54828	7849.45	0.000	Yes	density_80km_60km
0	2	-0.01088	277.91	0.000	Yes	density_above_100km
0	3	-0.97680	5.52	0.000	Yes	EFPA
0	4	-0.61203	495.12	0.000	Yes	ca_hc_mult
0	5	-0.60130	310.23	0.000	Yes	cm_hc_add
0	6	-0.76231	304.79	0.000	Yes	density_60km_40km
1	2	0.00047	386.40	0.000	Yes	
1	3	-0.00430	0.20	0.845	No	
1	4	-0.00934	1.78	0.076	No	
1	5	-0.02709	3.86	0.000	Yes	
1	6	0.00408	11.21	0.000	Yes	
2	3	-0.00014	1.69	0.092	No	
2	4	-0.00011	0.06	0.953	No	
2	5	0.00014	0.05	0.964	No	
2	6	-0.00007	0.06	0.952	No	
3	4	-0.00325	0.03	0.976	No	
3	5	-0.05159	1.35	0.179	No	
3	6	0.16047	21.35	0.000	Yes	
4	5	-0.01361	66.41	0.000	Yes	
4	6	0.07564	5.63	0.000	Yes	
5	6	0.13937	31.30	0.000	Yes	
1	1	0.00211	57.68	0.000	Yes	
2	2	0.00009	0.62	0.538	No	
3	3	-0.00262	0.03	0.979	No	
4	4	0.00800	0.77	0.443	No	
5	5	0.10238	2.34	0.020	Yes	
6	6	1.65478	29.96	0.000	Yes	

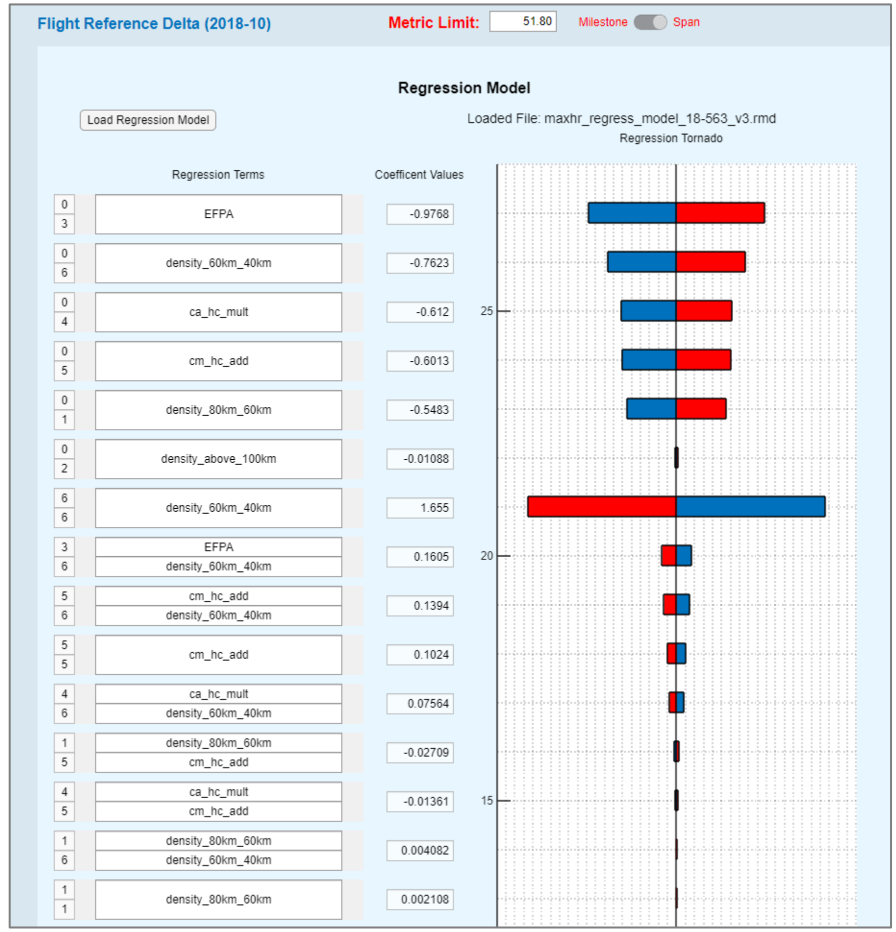


Figure 11-11 Peak Heat Rate ordered regression terms and regression tornado, epoch Flight Reference Delta.

Table 11-26 Comparison of Peak Heat Rate regression model linear terms across epochs.

Input	Peak Heat Rate Linear Coefficient Values & Percent Difference from Flight Reference Delta				
	ATLO 2.0		As-Built 2.0		Flight Ref. Delta
	Coefficient (W/cm^2)	Percent Difference	Coefficient (W/cm^2)	Percent Difference	Coefficient (W/cm^2)
density_80km_60km	-0.54775	-0.10%	-0.54134	-1.27%	-0.54828
density_above_100km	-0.01129	3.70%	-0.01091	0.26%	-0.01088
EFPA	-0.98094	0.42%	-0.97339	-0.35%	-0.97680
ca_hc_mult	-0.61370	0.27%	-0.61057	-0.24%	-0.61203
cm_hc_add	-0.57294	-4.72%	-0.59887	-0.40%	-0.60130
density_60km_40km	-0.78174	2.55%	-0.78458	2.92%	-0.76231

density_60km_40km has the strongest influence, followed by the linear terms. Of the 27 non-constant regression model terms for Peak Heat Rate, eight tested not significant for ATLO 2.0, nine tested not significant for As-Built 2.0 and 11 tested not significant for Flight Reference Delta.

11.3.2.3 DOE Results for Total Heating

The regression models generated for Total Heating for the three InSight epochs are shown in Tables 11-27 – 11-29. Figures 11-12 – 11-14 show the regression model TMST displays for Total Heating for the three InSight epochs, also for the first 15 non-constant terms of the regression model. As with Peak Heat Rate, the influence ordering of the linear term coefficients is the same across all epochs, and all linear terms test as significant. The variations in linear coefficient values across InSight epochs shown in Table 11-30 are below 7.57%, save density_60km_40km, which shows a 24.13% difference from the Flight Reference Delta value for ATLO 2.0, which is the smallest regression model linear term. The regression tornados for Total Heating shown in Figures 11-12 – 11-14 reveal, like Peak Deceleration, across the InSight epochs the linear terms have the strongest influence on the regression model response, with all nonlinear terms having

Table 11-27 DOE results for Total Heating, epoch ATLO 2.0.

Term Indices		Total Heating - ALTO 2.0				
Index 1	Index 2	Coefficient (J/cm^2)	T-Test	P Value	Significant?	Input Name
NA	NA	2716.69958	NA	NA	NA	NA
0	1	68.28279	41665.50	0.000	Yes	cm_hc_add
0	2	-38.05331	3082.99	0.000	Yes	density_40km_20km
0	3	54.94837	1718.12	0.000	Yes	EFPA
0	4	-42.51581	2480.94	0.000	Yes	ca_hc_mult
0	5	2.01036	1919.60	0.000	Yes	density_60km_40km
0	6	29.28842	90.77	0.000	Yes	density_80km_60km
1	2	-4.53907	1322.38	0.000	Yes	
1	3	5.33584	167.33	0.000	Yes	
1	4	-0.80124	196.71	0.000	Yes	
1	5	0.83420	29.54	0.000	Yes	
1	6	1.26892	30.75	0.000	Yes	
2	3	-0.60953	46.78	0.000	Yes	
2	4	0.69237	22.47	0.000	Yes	
2	5	2.14903	25.52	0.000	Yes	
2	6	0.36542	79.22	0.000	Yes	
3	4	-1.11044	13.47	0.000	Yes	
3	5	-1.48811	40.94	0.000	Yes	
3	6	0.46323	54.86	0.000	Yes	
4	5	-0.62072	17.08	0.000	Yes	
4	6	-0.23542	22.88	0.000	Yes	
5	6	0.02187	8.68	0.000	Yes	
1	1	8.81574	0.81	0.420	No	
2	2	2.13029	229.80	0.000	Yes	
3	3	0.72379	55.53	0.000	Yes	
4	4	0.99345	18.87	0.000	Yes	
5	5	-3.48588	25.90	0.000	Yes	
6	6	-4.47618	90.87	0.000	Yes	

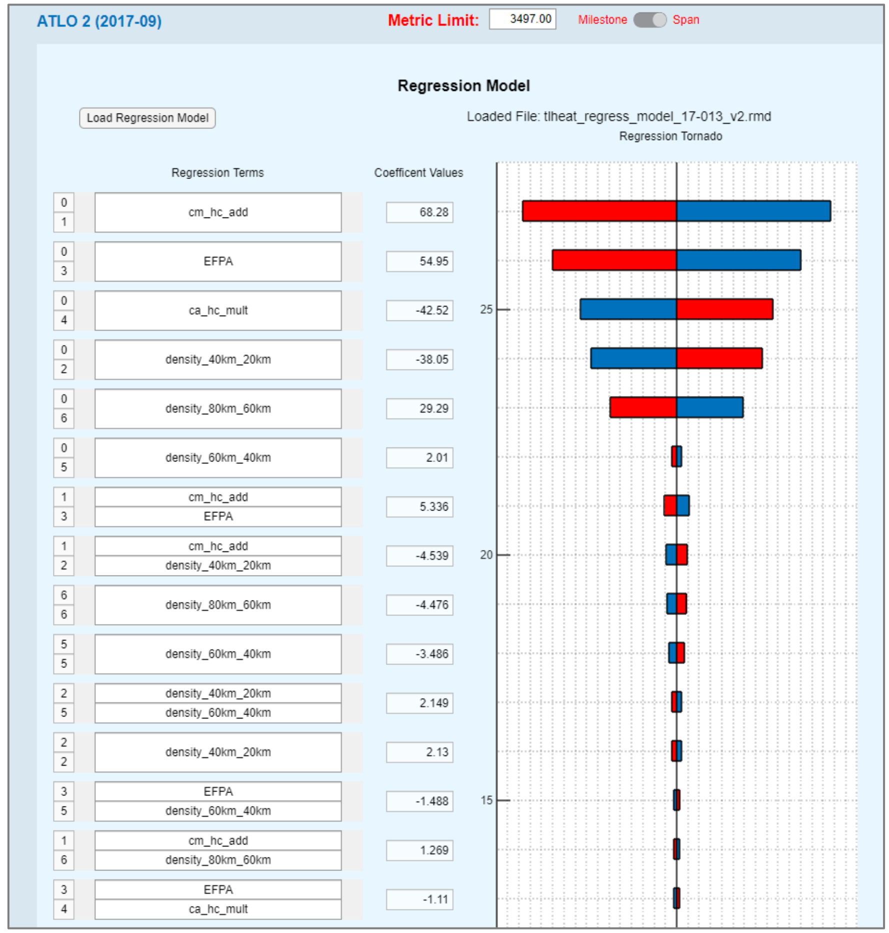


Figure 11-12 Total Heating ordered regression terms and regression tornado, epoch ATLO 2.0.

Table 11-28 DOE results for Total Heating, epoch As-Built 2.0.

Term Indices		Total Heating - As-Built 2.0				
Index 1	Index 2	Coefficient (J/cm^2)	T-Test	P Value	Significant?	Input Name
NA	NA	2682.24806	NA	NA	NA	NA
0	1	64.53960	19111.63	0.000	Yes	cm_hc_add
0	2	-36.95736	1353.79	0.000	Yes	density_40km_20km
0	3	53.55118	775.22	0.000	Yes	EFPA
0	4	-42.40443	1123.29	0.000	Yes	ca_hc_mult
0	5	1.74213	889.48	0.000	Yes	density_60km_40km
0	6	28.75617	36.54	0.000	Yes	density_80km_60km
1	2	-4.39353	603.19	0.000	Yes	
1	3	4.94376	75.25	0.000	Yes	
1	4	-0.82531	84.67	0.000	Yes	
1	5	0.75741	14.14	0.000	Yes	
1	6	1.23595	12.97	0.000	Yes	
2	3	-0.56213	21.17	0.000	Yes	
2	4	0.69307	9.63	0.000	Yes	
2	5	2.11737	11.87	0.000	Yes	
2	6	0.36784	36.26	0.000	Yes	
3	4	-1.12575	6.30	0.000	Yes	
3	5	-1.49517	19.28	0.000	Yes	
3	6	0.45770	25.61	0.000	Yes	
4	5	-0.62020	7.84	0.000	Yes	
4	6	-0.23662	10.62	0.000	Yes	
5	6	0.04433	4.05	0.000	Yes	
1	1	26.73882	0.76	0.448	No	
2	2	1.97961	323.82	0.000	Yes	
3	3	0.52894	23.97	0.000	Yes	
4	4	0.99449	6.41	0.000	Yes	
5	5	-3.46525	12.04	0.000	Yes	
6	6	-4.15863	41.97	0.000	Yes	

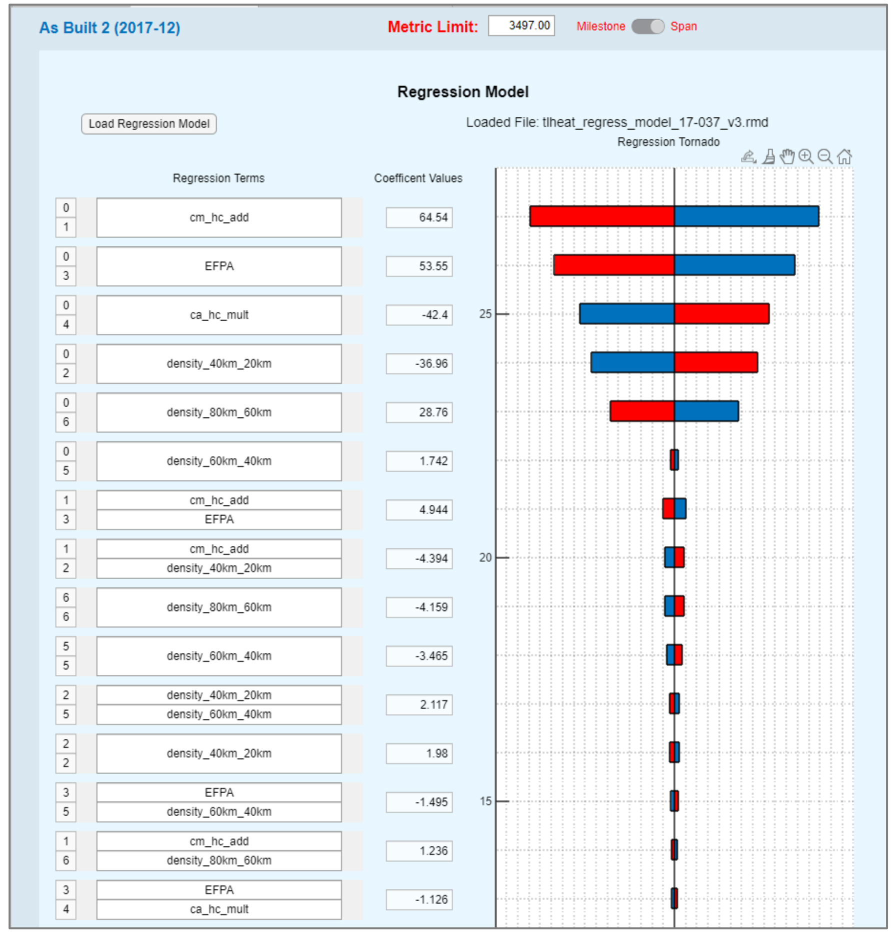


Figure 11-13 Total Heating ordered regression terms and regression tornado, epoch As-Built 2.0.

Table 11-29 DOE results for Total Heating, epoch Flight Reference Delta.

Term Indices		Total Heating - Flight Reference Delta				
Index 1	Index 2	Coefficient (J/cm^2)	T-Test	P Value	Significant?	Input Name
NA	NA	2676.94760	NA	NA	NA	NA
0	1	65.60784	15029.94	0.000	Yes	cm_hc_add
0	2	-36.90149	1084.42	0.000	Yes	density_40km_20km
0	3	53.71845	609.94	0.000	Yes	EFPA
0	4	-42.18069	887.91	0.000	Yes	ca_hc_mult
0	5	1.61953	697.20	0.000	Yes	density_60km_40km
0	6	28.67155	26.77	0.000	Yes	density_80km_60km
1	2	-4.39090	473.91	0.000	Yes	
1	3	5.18176	59.26	0.000	Yes	
1	4	-0.82232	69.93	0.000	Yes	
1	5	0.77891	11.10	0.000	Yes	
1	6	1.25489	10.51	0.000	Yes	
2	3	-0.59862	16.94	0.000	Yes	
2	4	0.62921	8.08	0.000	Yes	
2	5	2.12919	8.49	0.000	Yes	
2	6	0.38210	28.74	0.000	Yes	
3	4	-1.08308	5.16	0.000	Yes	
3	5	-1.49635	14.62	0.000	Yes	
3	6	0.46248	20.19	0.000	Yes	
4	5	-0.60502	6.24	0.000	Yes	
4	6	-0.27543	8.17	0.000	Yes	
5	6	0.03814	3.72	0.000	Yes	
1	1	23.80400	0.51	0.607	No	
2	2	2.04341	227.16	0.000	Yes	
3	3	0.22123	19.50	0.000	Yes	
4	4	0.91975	2.11	0.035	Yes	
5	5	-3.47541	8.78	0.000	Yes	
6	6	-4.69387	33.17	0.000	Yes	

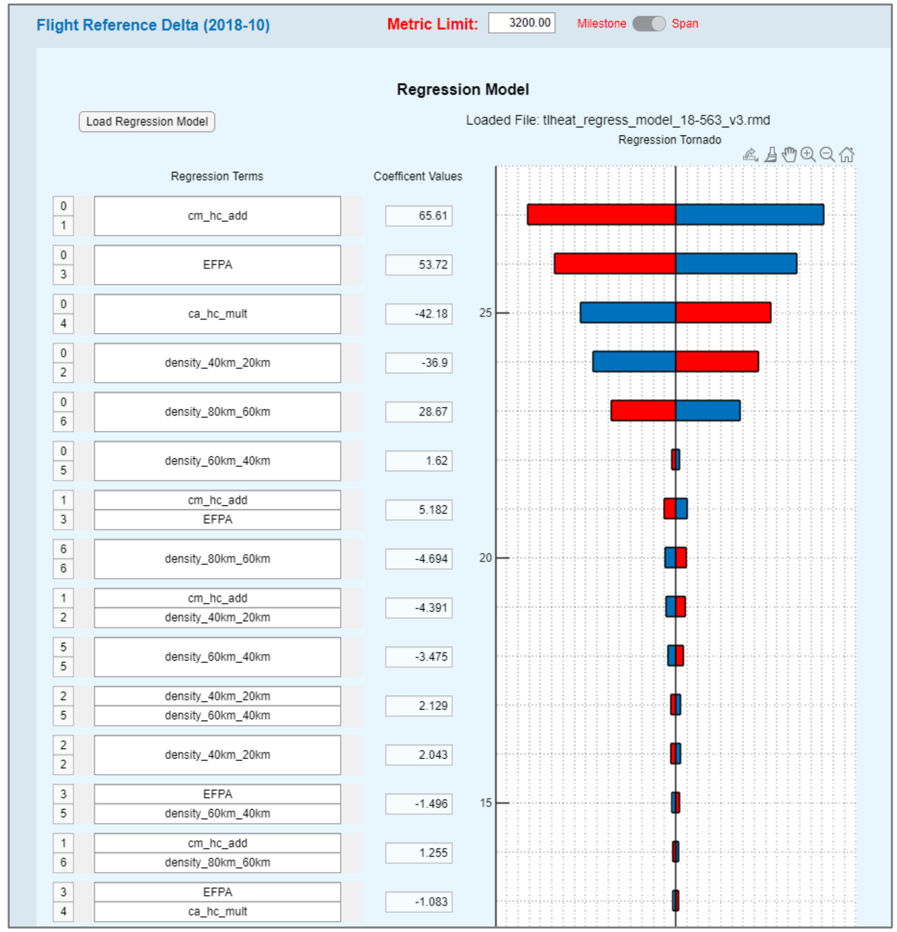


Figure 11-14 Total Heating ordered regression terms and regression tornado, epoch Flight Reference Delta.

Table 11-30 Comparison of Total Heating regression model linear terms across epochs.

Input	Total Heating Linear Coefficient Values & Percent Difference from Flight Reference Delta				
	ATLO 2.0		As-Built 2.0		Flight Ref. Delta
	Coefficient (J/cm^2)	Percent Difference	Coefficient (J/cm^2)	Percent Difference	Coefficient (J/cm^2)
cm_hc_add	68.283	4.08%	64.540	-1.63%	65.608
density_40km_20km	-38.053	3.12%	-36.957	0.15%	-36.901
EFPA	54.948	2.29%	53.551	-0.31%	53.718
ca_hc_mult	-42.516	0.79%	-42.404	0.53%	-42.181
density_60km_40km	2.01036	24.13%	1.74213	7.57%	1.61953
density_80km_60km	29.288	2.15%	28.756	0.30%	28.672

small to insignificant influence on regression model response. Of the 27 non-constant regression model terms for Total Heating, one tested not significant for ATLO 2.0, one tested not significant for As-Built 2.0 and one tested not significant for Flight Reference Delta.

11.3.2.4 DOE Results for Parachute Inflation Load

The regression models generated for Parachute Inflation Load for the three InSight epochs are shown in Tables 11-31 – 11-33. Figures 11-15 – 11-17 show the regression model TMST displays for Parachute Inflation Load for the three InSight epochs, also for the first 15 non-constant terms of the regression model. As with two previous Critical Metrics, the influencing ordering of the linear regression model terms is the same across InSight epochs. For Parachute Inflation Load, the highest variation in linear coefficient values across the epochs, as shown by the percent difference values in Table 11-34, is -18.55% for density_40km_20km for epoch ATLO 2.0.

The regression tornados for Peak Inflation Load shown in Figure 11-15 – 11-17 show a moderate level of influence for some nonlinear terms across the InSight epochs. Of the 27 non-constant regression model terms for Parachute Inflation Load, seven tested not significant for

Table 11-31 DOE results for Parachute Inflation Load, epoch ATLO 2.0.

Term Indices		Parachute Inflation Load - ALTO 2.0				
Index 1	Index 2	Coefficient (lbf)	T-Test	P Value	Significant?	Input Name
NA	NA	12041.25785	NA	NA	NA	NA
0	1	1777.71955	751.89	0.000	Yes	chute_aero_disp
0	2	-818.97436	326.79	0.000	Yes	cm_mc_add
0	3	-78.35085	150.55	0.000	Yes	density_40km_20km
0	4	-1202.24353	14.40	0.000	Yes	ca_mc_mult
0	5	-839.99551	221.00	0.000	Yes	cm_hc_add
0	6	19.28865	154.41	0.000	Yes	density_60km_40km
1	2	-109.12428	3.55	0.000	Yes	
1	3	-9.66141	16.38	0.000	Yes	
1	4	-175.90366	1.45	0.148	No	
1	5	-107.13230	26.40	0.000	Yes	
1	6	2.48692	16.08	0.000	Yes	
2	3	-4.87965	0.37	0.709	No	
2	4	76.80176	0.73	0.464	No	
2	5	-248.31008	11.53	0.000	Yes	
2	6	10.07479	37.27	0.000	Yes	
3	4	-27.49707	1.51	0.131	No	
3	5	-117.71200	4.13	0.000	Yes	
3	6	-2.09537	17.67	0.000	Yes	
4	5	56.46372	0.31	0.753	No	
4	6	0.87890	8.47	0.000	Yes	
5	6	21.41019	0.13	0.895	No	
1	1	-6.34271	3.21	0.001	Yes	
2	2	-88.13760	0.67	0.501	No	
3	3	-25.20447	9.35	0.000	Yes	
4	4	103.21641	2.68	0.008	Yes	
5	5	71.22961	10.95	0.000	Yes	
6	6	4.80238	7.56	0.000	Yes	

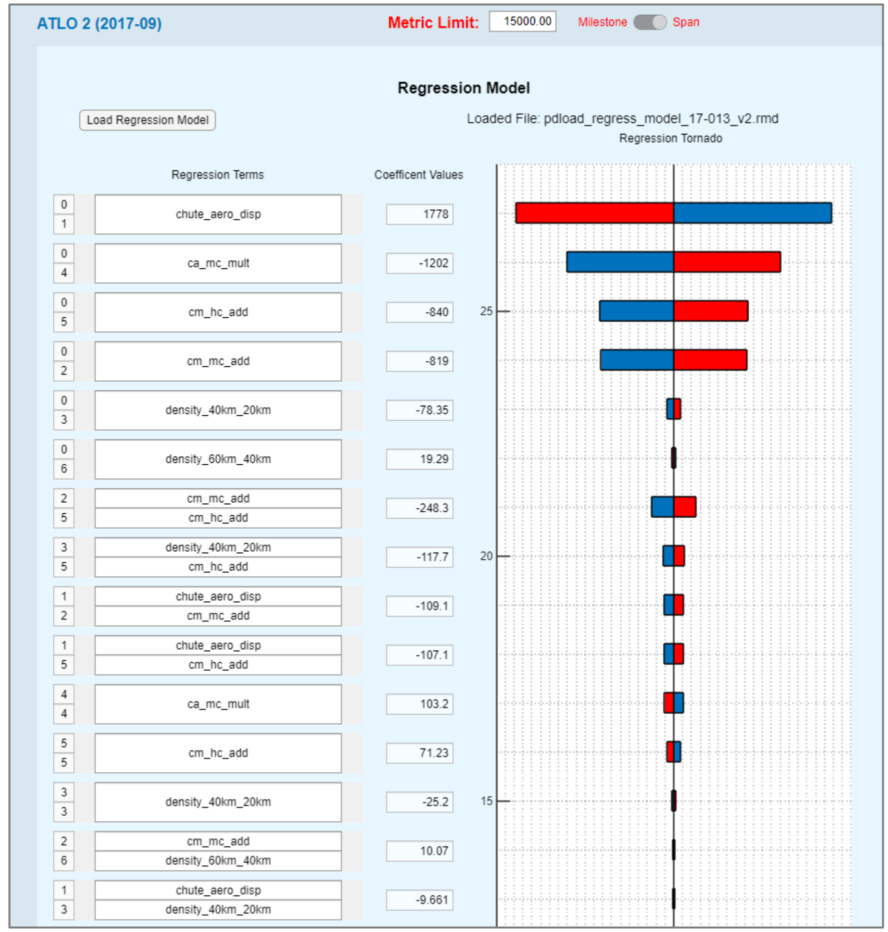


Figure 11-15 Parachute Inflation Load ordered regression terms and regression tornado, epoch ATLO 2.0.

Table 11-32 DOE results for Parachute Inflation Load, epoch As-Built 2.0.

Term Indices		Parachute Inflation Load - As-Built 2.0				
Index 1	Index 2	Coefficient (lbf)	T-Test	P Value	Significant?	Input Name
NA	NA	12223.09620	NA	NA	NA	NA
0	1	1764.58352	382.09	0.000	Yes	chute_aero_disp
0	2	-794.01566	162.39	0.000	Yes	cm_mc_add
0	3	-110.31280	73.07	0.000	Yes	density_40km_20km
0	4	-1175.64332	10.15	0.000	Yes	ca_mc_mult
0	5	-893.31919	108.19	0.000	Yes	cm_hc_add
0	6	19.79487	82.21	0.000	Yes	density_60km_40km
1	2	-101.20439	1.82	0.069	No	
1	3	-13.69234	7.60	0.000	Yes	
1	4	-172.56234	1.03	0.304	No	
1	5	-110.07304	12.97	0.000	Yes	
1	6	2.63945	8.27	0.000	Yes	
2	3	0.48904	0.20	0.843	No	
2	4	92.15099	0.04	0.971	No	
2	5	-403.39859	6.92	0.000	Yes	
2	6	0.95742	30.31	0.000	Yes	
3	4	3.22052	0.07	0.943	No	
3	5	-145.70028	0.24	0.809	No	
3	6	0.45372	10.95	0.000	Yes	
4	5	92.54147	0.03	0.973	No	
4	6	-4.14020	6.95	0.000	Yes	
5	6	16.49740	0.31	0.756	No	
1	1	-5.54702	1.24	0.216	No	
2	2	23.93539	0.29	0.768	No	
3	3	-24.48643	1.27	0.204	No	
4	4	77.19742	1.30	0.194	No	
5	5	-373.25135	4.10	0.000	Yes	
6	6	-9.53309	19.83	0.000	Yes	

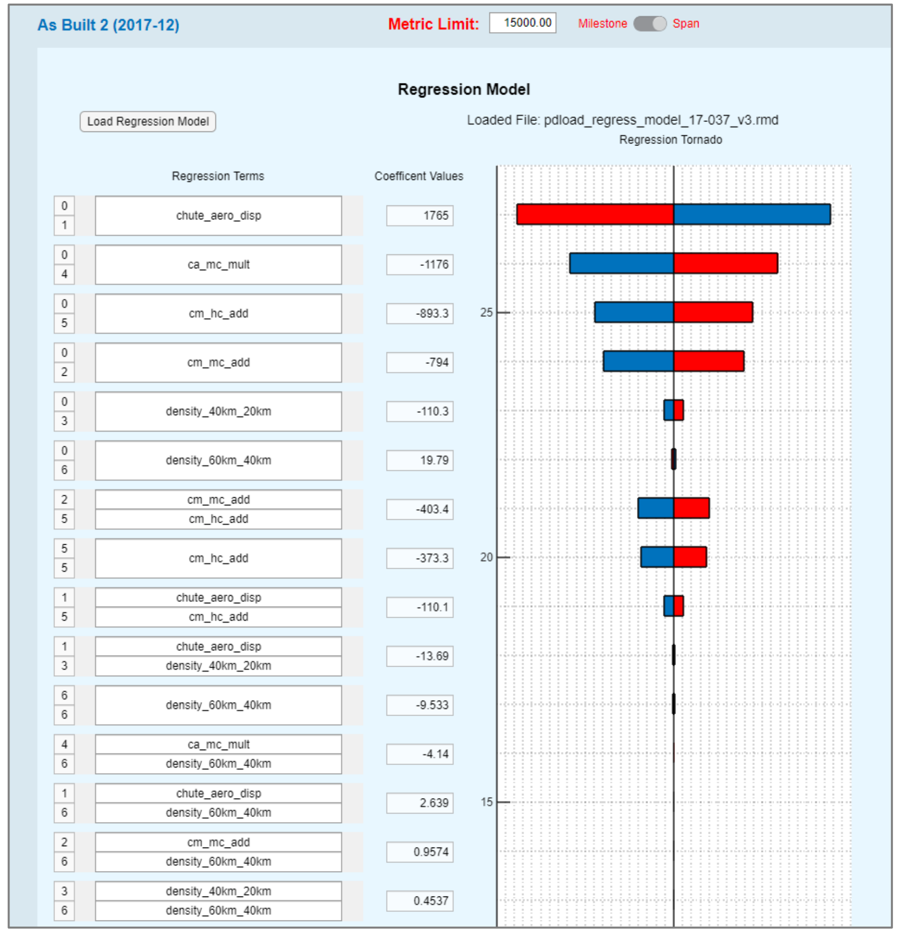


Figure 11-16 Parachute Inflation Load ordered regression terms and regression tornado, epoch As-Built 2.0.

Table 11-33 DOE results for Parachute Inflation Load, epoch Flight Reference Delta.

Term Indices		Parachute Inflation Load - Flight Reference Delta				
Index 1	Index 2	Coefficient (lbf)	T-Test	P Value	Significant?	Input Name
NA	NA	12063.16899	NA	NA	NA	NA
0	1	1756.13288	434.02	0.000	Yes	chute_aero_disp
0	2	-820.98951	186.01	0.000	Yes	cm_mc_add
0	3	-96.19413	86.96	0.000	Yes	density_40km_20km
0	4	-1182.81762	10.19	0.000	Yes	ca_mc_mult
0	5	-920.77241	125.28	0.000	Yes	cm_hc_add
0	6	18.30447	97.53	0.000	Yes	density_60km_40km
1	2	-106.98383	1.94	0.053	No	
1	3	-12.00143	9.25	0.000	Yes	
1	4	-173.42518	1.04	0.300	No	
1	5	-115.80228	15.00	0.000	Yes	
1	6	2.43528	10.01	0.000	Yes	
2	3	6.94057	0.21	0.833	No	
2	4	94.57291	0.60	0.549	No	
2	5	-351.69522	8.18	0.000	Yes	
2	6	2.54315	30.42	0.000	Yes	
3	4	-18.28725	0.22	0.826	No	
3	5	-164.93534	1.58	0.114	No	
3	6	10.11345	14.26	0.000	Yes	
4	5	74.21496	0.87	0.382	No	
4	6	1.01349	6.42	0.000	Yes	
5	6	18.87842	0.09	0.930	No	
1	1	-4.57474	1.63	0.103	No	
2	2	-0.17526	0.28	0.780	No	
3	3	-29.89334	0.01	0.992	No	
4	4	99.25006	1.83	0.068	No	
5	5	-249.37596	6.07	0.000	Yes	
6	6	6.47430	15.25	0.000	Yes	

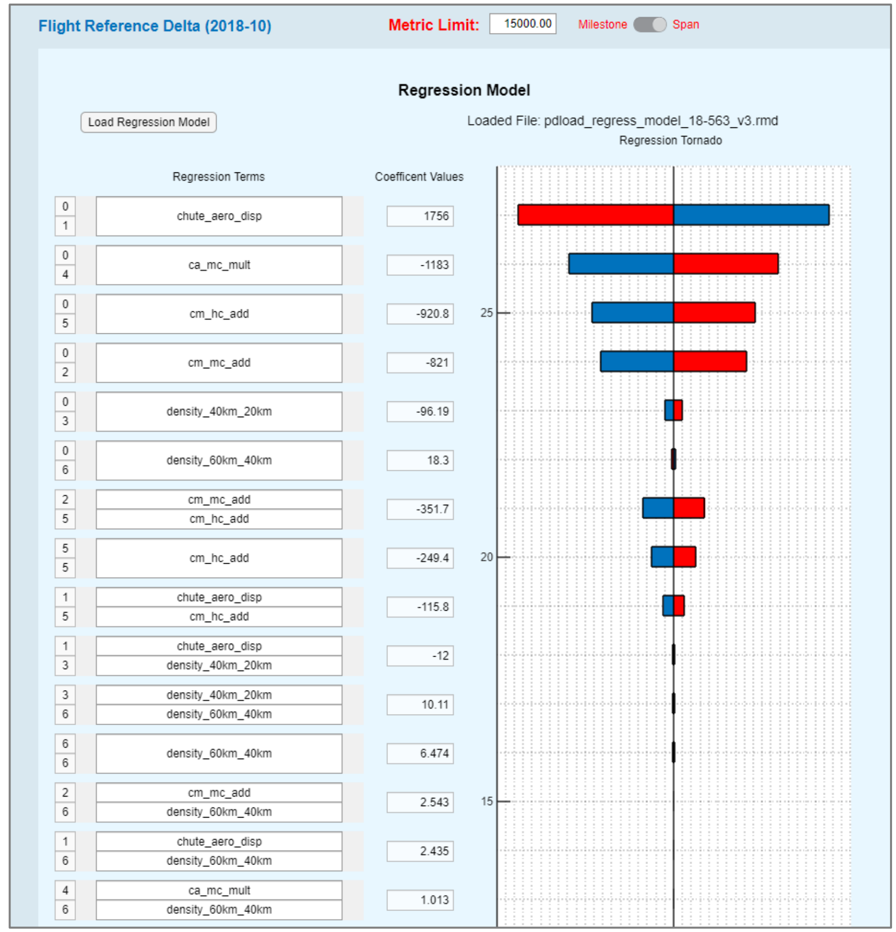


Figure 11-17 Parachute Inflation Load ordered regression terms and regression tornado, epoch Flight Reference Delta.

Table 11-34 Comparison of Parachute Inflation Load regression model linear terms across epochs.

Input	Parachute Inflation Load Linear Coefficient Values & Percent Difference from Flight Reference Delta				
	ATLO 2.0		As-Built 2.0		Flight Ref. Delta
	Coefficient (lbf)	Percent Difference	Coefficient (lbf)	Percent Difference	Coefficient (lbf)
chute_aero_disp	1777.7	1.23%	1764.6	0.48%	1756.1
cm_mc_add	-818.97	-0.25%	-794.02	-3.29%	-820.99
density_40km_20km	-78.351	-18.55%	-110.313	14.68%	-96.194
ca_mc_mult	-1202.2	1.64%	-1175.6	-0.61%	-1182.8
cm_hc_add	-840.00	-8.77%	-893.32	-2.98%	-920.77
density_60km_40km	19.289	5.38%	19.7949	8.14%	18.3045

ATLO 2.0, 12 tested not significant for As-Built 2.0 and 12 tested not significant for Flight Reference Delta.

11.3.2.5 DOE and Model Inputs of Highest Influence

As the results of the DOE analyses show, and was discussed in the previous sections, the ordered strength of influence for the input parameters is somewhat different than the order magnitude of correlation as determined by the Monte Carlo Filtering. This is because an input may have strong correlation by not necessarily strong influence. For the purposes of the research, using a combination of initial Monte Carlo Filtering and DOE analysis to identify and quantify influence serves as a good first method, however, future work should include investigation of a more comprehensive and exhaustive process to identify the inputs of strongest influence, to ensure a best set is identified. Although perhaps computationally intensive, DOE techniques for comprehensively searching for inputs of highest influence should be explored.

11.4 Process Configuration for InSight

This section applies the Target Margin assessment configuration process, as discussed in Section 7, to InSight. As already discussed, in applying the process to InSight, there are four Critical Metrics that are assessed for quantification of Target Margin: Peak Deceleration, Peak Heat Rate, Total Heating and Parachute Inflation Load. The TMST Critical Metric configuration interface, a tab on the Startup Window, is shown in Figure 3-4. Also needed for configuring the InSight process is the epochs over which the assessment will be performed. As introduced in Section 11.1, these epochs are ATLO 2.0, As-Built 2.0, and Flight Reference Delta. Figure 11-18 shows the TMST Startup Window configuration panel for Assessment Epochs.

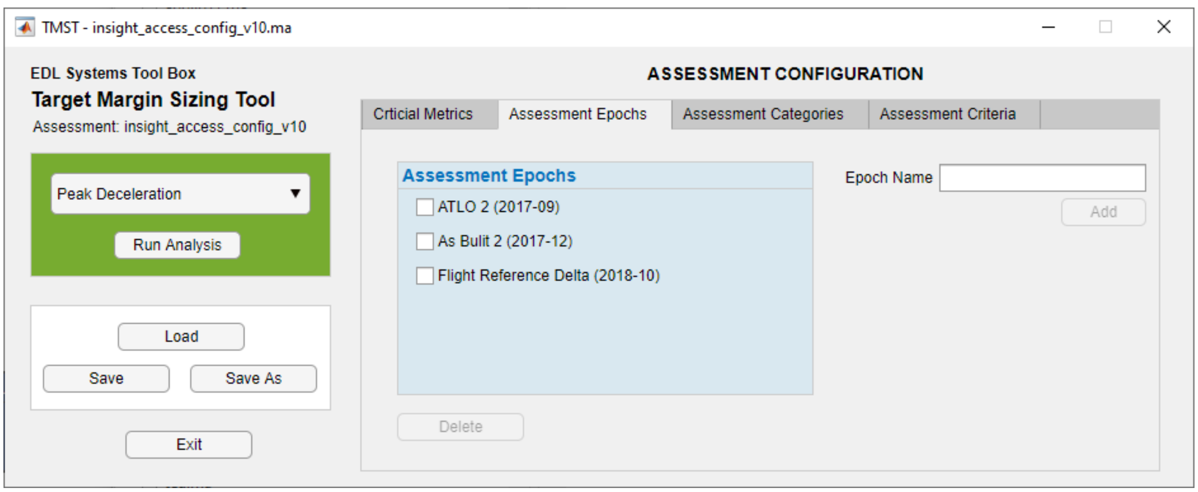


Figure 11-18 TMST Assessment Epochs configuration panel.

11.4.1 Configuring Assessment Categories for InSight

As defined in Section 8.1.1, the Assessment Categories to be deployed in the InSight assessment are Mathematical Model, Code, Validation, Model Parameter. As outlined in Section 8.2, each Assessment Category has an assigned weighting used in the quantification of Target Margin. As part of the assessment configuration process, the assigned weightings can be configured/customized for each assessment epoch. For this initial assessment for InSight, an equal weighting is chosen for Assessment Categories for all three assessment epochs, as is shown in Table 11-35. This provides a baseline with which to compare to the original InSight Seed Methodology introduced in Section 1.3.2. The Assessment Categories TMST panel is shown in Figure 11-19, configured for the InSight baseline Target Margin assessment. An exploration of other weighting configurations is presented in Section 12.

Table 11-35 Initial Assessment Category weightings for InSight, for all assessment epochs.

All Epochs		
Assessment Category	**Weighting**	**Rationale**
Mathematical Model	1.00	Even weighting for initial InSight baseline assessment.
Code	1.00	Even weighting for initial InSight baseline assessment.
Validation	1.00	Even weighting for initial InSight baseline assessment.
Model Parameter	1.00	Even weighting for initial InSight baseline assessment.

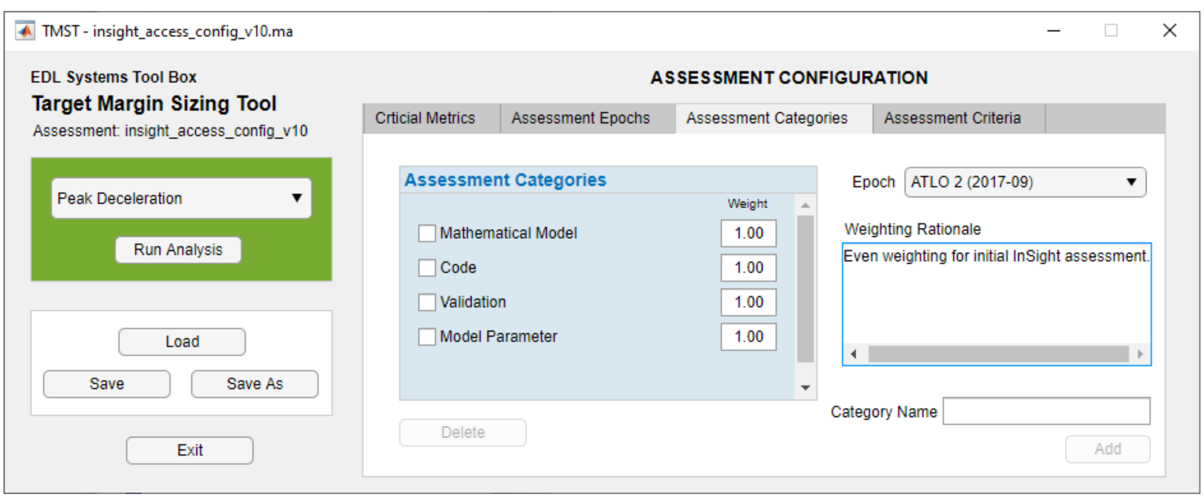


Figure 11-19 TMST Assessment Category initial configuration for InSight.

11.4.2 Configuring Assessment Criteria for InSight

The final configuration step required for performing an InSight Target Margin assessment is configuration of the Assessment Criteria, first introduced in Section 7.2. As outlined, the Evolved Methodology provides for assessments using inputs with normal and uniform distributions. Each distribution type is configured separately, and the methodology provides for configuring Assessment Criteria uniquely for each Assessment Epoch. The following subsections discuss Assessment Criteria configurations for the initial InSight Target Margin baseline assessment.

11.4.2.1 Configuring Normal Distribution Assessment Criteria for InSight

From Section 7.2, Assessment Criteria for normal distributions are specified in a fraction of a standard deviation. The Assessment Criteria chosen for an initial InSight assessment are based on the Method 2 criteria from the original Seed Methodology for InSight, which are discussed in Section 1.3.2.3.2. For performance distributions with modeling of Higher confidence, the Target Margin is based on half a standard deviation, or 0.5σ. For performance distributions with modeling of Lower confidence, the Target Margin is based on a full standard deviation, or 1.0σ. Although the criteria for the original Seed Methodology are applied to performance output rather than inputs, both are based on modeling confidence, and so as an initial set of criteria, the same criteria will be used for the Evolved Methodology for InSight. The initial InSight Assessment Criteria for a normal distribution are summarized in Table 11-36, where the High and Low values are based on the original InSight Seed Methodology criteria, and the midpoint between High and Low is chosen as the criterion for Medium. For the initial assessment, all epochs are configured to use the same Assessment Criteria.

Table 11-36 Normal Assessment Criteria for initial InSight assessment, for all epochs.

Normal Distribution - All Epochs		
Assessment Criteria	Value (Fraction of Sigma)	Rationale
High	0.50	Same sigma fraction used for a Higher rating in the original InSight Target Margin methodology.
Medium	0.75	Chosen as midpoint value between a High rating and Low rating.
Low	1.00	Same sigma fraction used for a Lower rating in the original InSight Target Margin methodology.

11.4.2.2 Configuring Uniform Distribution Assessment Criteria for InSight

The original InSight Target Margin Seed Methodology did not explicitly accommodate uniform distributions, so there is no such basis for Assessment Criteria configuration for this type of distribution. As discussed in Section 7.2, a uniform distribution represents a uniform level of uncertainty across the span of the distribution, while a normal distribution has more probability density concentrated around the mean. Therefore, a uniform distribution can be thought of as representing both greater uncertainty over an equivalent span, and high probability density at the bounds of the distribution, relative to a normal distribution. With uniform input distribution having higher probability of being sampled at the bounds, this in general will have the effect of increasing the 99% performance of a metric in a Monte Carlo simulation. Because of this, for an equivalent Model Confidence Rating there is potentially more contribution to Target Margin needed for a uniform distribution relative to a normal distribution. As a starting place for an initial InSight assessment, a doubling of the normal Assessment Criteria is chosen for uniform criteria. While the sizing is somewhat arbitrary, it is chosen to reflect the more conservative probability distribution represented by a uniform distribution. A method to further refine the sizing of uniform distribution Assessment Criteria is the use of Monte Carlo simulation parametric and sensitivity techniques to establish the mapping of changing uniform input distribution to changing performance metric output. This technique could also be used to tune the Assessment Criteria for any distribution, and is future work.

Table 11-37 summarizes the values used to configure Assessment Criteria for a uniform distribution. Note the magnitude of the values listed in the Value column of Table 11-37 are smaller than those summarized for a normal distribution in Table 11-36. This is because the criteria for normal distributions are expressed as a fraction of standard deviation whereas the uniform distribution criteria are expressed as a fraction of the span of the distribution.

The TMST Startup Window Assessment Criteria configuration interface is shown in Figure 11-20, configured for the initial InSight Target Margin assessment.

Table 11-37 Uniform Assessment Criteria for initial InSight assessment, for all Assessment Epochs.

Uniform Distribution - All Epochs		
Assessment Criteria	Value (Fraction of Span)	Rationale
High	0.17	Treating uniform as representing more uncertainty, therefore doubling Normal fraction (as fraction of span).
Medium	0.25	Treating uniform as representing more uncertainty, therefore doubling Normal fraction (as fraction of span).
Low	0.33	Treating uniform as representing more uncertainty, therefore doubling Normal fraction (as fraction of span).

11.5 Input Model Confidence Assessment

Input Model Confidence Assessment is a key step in the Evolved Methodology for quantifying Target Margin. The process for assessing model confidence discussed in Section 8 is applied to the InSight epochs in this section. There are two assessments needed to complete the

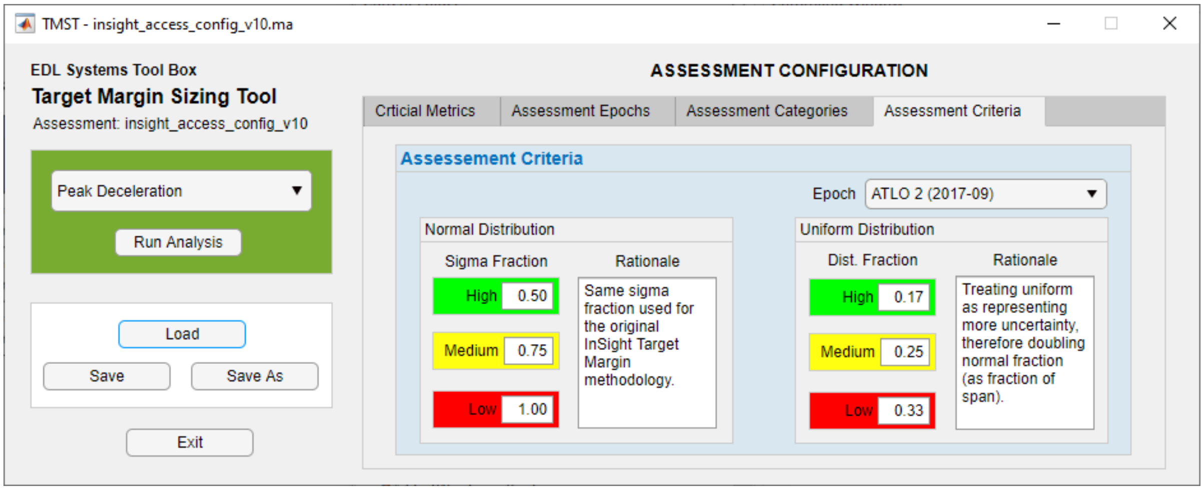


Figure 11-20 TMST Assessment Criteria initial configuration for InSight.

assessment of model confidence: assessing type of uncertainty for each input variable, and providing a model Confidence Rating for each Assessment Category for all input variables.

11.5.1 Assessing Type of Uncertainty

An assessment of uncertainty type was conducted for all input variables across InSight epochs. The assessment includes the uncertainty distribution type (normal or uniform) for each input, as well as an assessment of the percent makeup of epistemic and aleatory uncertainty present in the uncertainty for each input. Documented in Table 11-38 and Table 11-39 are the assessment results and rationales that are discussed in detail in subsequent subsections. The assessment of the percent makeup of each uncertainty type benefitted somewhat by hindsight, which would not necessarily be the case when applying the methodology mid-project lifecycle. Further development of methods for assessing uncertainty makeup without the benefit of hindsight is future work.

11.5.1.1 Aerodynamic Parameters

As implemented by InSight, all entry vehicle aerodynamic parameters (ca_hc_mult, ca_mc_mult, cm_hc_add, cm_mc_add, cn_hc_add) have uncertainty represented by a normal distribution in the InSight EDL Monte Carlo simulation. For all three InSight epochs (and on landing day), the entry vehicle modeling uncertainties remained the same with no opportunity for reduction. Thus, for entry vehicle aerodynamic uncertainties, the uncertainty is classified as 100% aleatory for all epochs. The parachute aerodynamic drag coefficient (chute_aero_disp) uncertainty is represented by a uniform distribution in InSight Monte Carlo simulation, based on a parachute model developed at NASA Langley Research Center. By the same reason used for entry vehicle aerodynamics, parachute drag uncertainty is also classified as irreducible and 100% aleatory for all epochs.

11.5.1.2 Atmospheric Density Parameters

The atmospheric modeling utilized in InSight Monte Carlo simulation is dispersed by altitude by a normal distribution. Thus, each atmosphere density input (density_40km_20km, density_60km_40km, density_80km_60km, density_above_100km) has uncertainty characterized as normal. Because of inherent uncertainty in atmosphere modeling for Mars, the atmosphere modeling carried the same atmosphere model uncertainty into flight operations as was used for the Monte Carlos for the three InSight epochs under assessment. Therefore, with no reducible uncertainty available, all atmosphere density input parameters are classified as 100% aleatory for all epochs.

11.5.1.3 Entry Flight Path Angle

The orbit determination modeling that provides a dispersed entry flight path angle delivery at Mars results in a normally distributed entry flight path angle uncertainty. The entry flight path angle 3-sigma uncertainty used for all three InSight epochs is 0.21°, the InSight entry flight path angle delivery requirement. During flight, as the spacecraft approached Mars, the last estimated entry flight path uncertainty about an hour before landing was only 0.0005° 3-sigma. The 3-sigma entry flight path uncertainty therefore reduced from 0.21° to 0.0005°. This reduction was used to compute an epistemic (reducible) uncertainty percentage of 99.8% and an aleatory uncertainty percentage of 0.2%.

11.5.1.4 Entry Vehicle Z-Axis Center of Gravity Location at Entry Interface

The entry vehicle z-axis center of gravity location at entry interface is a combination of uncertainty types. The contribution to center of gravity location from the structural dry mass of the entry vehicle is characterized by a normal distribution. At entry interface there is also a contribution from tank-to-tank propellant imbalance, which has uniform uncertainty. Because the center of gravity location is dominated by the structural dry mass, the input zcg_ei is treated as a normal distribution. Like entry flight path angle, the entry vehicle z-axis location uncertainty reduces over time. For the epochs ATLO 2.0 and As-Built 2.0, the standard deviation for zcg_ei is 0.34 mm. Following the measurement of spacecraft mass properties prior to launch, the standard deviation for zcg_ei reduced to 0.13 mm for the Flight Reference Delta epoch, which is also the uncertainty carried through to landing. Using these two standard deviation values, for the epochs ATLO 2.0 and As-Built 2.0, the epistemic percentage is calculated to be 61.8% and the aleatory percentage to be 38.2%. The uncertainty standard deviation of 0.13mm is taken as irreducible and thus zcg_ei is 100% aleatory for Flight Reference Delta.

Table 11-38 Input uncertainty type classifications for InSight epochs – part 1.

Input Name	Distribution Type	Epoch	Estimated Percentage		Rationale
			Epistemic	Aleatory	
ca_hc_mult	Normal	ATLO 2.0	0.0%	100.0%	All entry vehicle inputs have no opportunity for uncertainty reduction for the remainder of the InSight project and thus are classified as 100% aleatory for all three epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
ca_mc_mult	Normal	ATLO 2.0	0.0%	100.0%	All entry vehicle inputs have no opportunity for uncertainty reduction for the remainder of the InSight project and thus are classified as 100% aleatory for all three epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
chute_aero_disp	Uniform	ATLO 2.0	0.0%	100.0%	All entry vehicle inputs have no opportunity for uncertainty reduction for the remainder of the InSight project and thus are classified as 100% aleatory for all three epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
cm_hc_add	Normal	ATLO 2.0	0.0%	100.0%	All entry vehicle inputs have no opportunity for uncertainty reduction for the remainder of the InSight project and thus are classified as 100% aleatory for all three epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
cm_mc_add	Normal	ATLO 2.0	0.0%	100.0%	All entry vehicle inputs have no opportunity for uncertainty reduction for the remainder of the InSight project and thus are classified as 100% aleatory for all three epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
cn_hc_add	Normal	ATLO 2.0	0.0%	100.0%	All entry vehicle inputs have no opportunity for uncertainty reduction for the remainder of the InSight project and thus are classified as 100% aleatory for all three epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	

Table 11-39 Input uncertainty type classifications for InSight epochs – part 2.

Input Name	Distribution Type	Epoch	Estimated Percentage		Rationale
			Epistemic	Aleatory	
density_40km_20km	Normal	ATLO 2.0	0.0%	100.0%	The level of atmosphere uncertainty dispersion remained constant and unreduced for all three InSight epochs and flight operations so atmosphere uncertainty is classified 100% aleatory for all epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
density_60km_40km	Normal	ATLO 2.0	0.0%	100.0%	The level of atmosphere uncertainty dispersion remained constant and unreduced for all three InSight epochs and flight operations so atmosphere uncertainty is classified 100% aleatory for all epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
density_80km_60km	Normal	ATLO 2.0	0.0%	100.0%	The level of atmosphere uncertainty dispersion remained constant and unreduced for all three InSight epochs and flight operations so atmosphere uncertainty is classified 100% aleatory for all epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
density_above_100km	Normal	ATLO 2.0	0.0%	100.0%	The level of atmosphere uncertainty dispersion remained constant and unreduced for all three InSight epochs and flight operations so atmosphere uncertainty is classified 100% aleatory for all epochs.
		As-Built 2.0	0.0%	100.0%	
		Flight Reference Delta	0.0%	100.0%	
EFPA	Normal	ATLO 2.0	99.8%	0.2%	Based on a pre-landing delivery uncertainty requirement of 0.21° and a final delivery uncertainty one hour before landing of 0.0005°
		As-Built 2.0	99.8%	0.2%	
		Flight Reference Delta	99.8%	0.2%	
zcg_ei	Normal & Uniform	ATLO 2.0	61.8%	38.2%	Based on a z-axis center of gravity location standard deviation of 0.34mm for ATLO 2.0 and As-Built 2.0, and a reduction to 0.13mm for Flight Reference Delta and on landing day.
		As-Built 2.0	61.8%	38.2%	
		Flight Reference Delta	0.0%	100.0%	

11.5.2 Assessing Model Confidence

The final and important step in configuring a Target Margin assessment is the Input Model Confidence Assessment. In this section, the assessment will be applied to InSight in the four Assessment Categories (Mathematical Modeling, Code, Validation, and Model Parameter) configured for the assessment. In executing the assessment, a rationale is provided for each model input in each category, an important step in that it brings critical thinking to the process and provides documentation of the decisions made in configuring the Evolved Methodology. In formulating a confidence rating for each Assessment Category, it is important to capture only factors contributing to confidence level that are not already captured or represented by uncertainty modeled in the Monte Carlo simulation. The following subsections provide an assessment of modeling confidence for InSight, with the results summarized in Table 11-40 and Table 11-41.

11.5.2.1 Entry Vehicle Aerodynamic Parameters

The entry vehicle aerodynamic parameters ca_hc_mult, ca_mc_mult, cm_hc_add, cm_mc_add and cn_hc_add are static aerodynamic parameters derived using computational fluid dynamic (CFD) modeling. While CFD modeling is a powerful mathematical tool, it more accurately captures hypersonic entry vehicle aerodynamic behavior than supersonic behavior [31], which is why ca_hc_mult, cm_hc_add, cn_hc_add are rated High, and ca_mc_mult and cm_mc_add are Medium in the Assessment Category Mathematical Modeling. The parameters are linearized and coded into an aerodynamic database that is used in Monte Carlo simulations. The database goes through a thorough checkout before delivery to the simulations giving it a High rating in the Assessment Category Code. The validation of the modeling is limited to validation via reconstruction of Mars flight opportunities and through derived means with limited direct measurement. Thus, the Assessment Category Validation is rated Medium because of the limited validation through Mars flight. Finally, for the Assessment Category Model Parameter, all entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function so they are all rated High.

11.5.2.2 Parachute Drag

A parachute as a soft-good has significant uncertainty in aerodynamic drag. The drag model used in EDL simulation is based on drag measurements from the Viking-era Balloon Launched Decelerator Test (BLDT) at high altitude [35], wind tunnel testing in NASA/Langley Research Center's Transonic Dynamics Tunnel (TDT) [36], and flight reconstructions of Mars landed missions. The mathematical model for parachute drag is a parameterized, physics-based model that disperses aerodynamic drag to a set scaling for all Mach numbers. Because is it moderately parameterized, in the Assessment Category Mathematical Model, the parachute drag model is assessed as Medium. For the category Code, the parachute model is assessed as High because as implemented for InSight, the model was independently implemented in multiple simulations and behavior compared. Direct validation of the parachute drag model also suffers from infrequency of flights to Mars and is performed through flight reconstruction that is based on directly measured and derived quantities, and therefore receives a Medium rating for category Validation. For the category Model Parameter, parameters for parachute drag have high pedigree, are complete, relevant and appropriate for their function and therefore are rated High.

11.5.2.3 Atmospheric Density Parameters

The InSight atmosphere model is anchored on atmosphere temperature profiles measured by the Mars Climate Sounder (MCS) instrument on the Mars Reconnaissance Orbiter. Seasonal and diurnal variability is then guided by mesoscale CFD modeling whose boundary conditions are provided by the Mars General Circulation Model (GCM). The MCS measurements and CFD modeling are then combined into a parametrized atmosphere profile generator that provides dispersed density, temperature and pressure profiles for use in Monte Carlo simulations. The InSight atmosphere model creation process was very similar to that employed by Phoenix, which is summarized in *Expected Atmospheric Environment for the Phoenix Landing Season and Location*, by Tamppari, et al [37].

Because the atmosphere model is based on high pedigree, complete physics-based modeling, the atmosphere profile input variables are rated High in the Mathematical Model Assessment Category. While the review of code used to process MCS temperature data was not determined for this application of the methodology, the mesoscale CFD code is widely used and assumed to be well reviewed, and the parametrized profile generator had limited review, which results in a Medium score for the Assessment Category Code. Like the entry vehicle aerodynamic model, atmosphere model validation is limited by infrequent flights to Mars and is based on derived quantities that cannot be separated completely from aerodynamic behavior, so the category Validation is rated as Medium. In the last category, Model Parameters, because MCS cannot measure temperature data below 5 km [37] and must be extrapolated based on surface temperature assumptions, the atmosphere input variables receive a Medium rating except for density_sum_above_100km, where the atmosphere model is very approximated and of low relevance, so its rating is Low.

11.5.2.4 Entry Flight Path Angle

Entry flight path angle modeling is a well-established, physics-based process that has remarkable accuracy. The orbit determination process resulting in knowledge of entry flight path angle utilized by InSight is outlined in *2018 Mars InSight Mission Design and Navigation*

Overview by Abilleira, et al [38]. The mathematical modeling utilizes high fidelity orbital mechanics and statistical state estimation while tracking and incorporating a comprehensive set of factors influencing the flight path of a spacecraft on its transfer orbit to Mars. Because of this, in the Assessment Category Mathematical Model, the modeling is rated as High. The orbit determination tools utilized by InSight to perform the orbit determination is shared across time by multiple missions with results frequently peer reviewed, so in the category Code, modeling of entry flight path angle is rated High. Likewise, the orbit determination techniques utilized in modeling entry flight path angle are utilized by all planetary missions and thus each provides a validation opportunity for the modeling, resulting in a High rating for the category Validation. Finally, for the category Model Parameter, the parameters have high pedigree, completeness and relevance for the application so the rating is High.

11.5.2.5 Entry Vehicle Z-Axis Center of Gravity Location at Entry Interface

Knowledge of center of gravity location of a spacecraft is typically achieved by direct measurement through spin balancing prior to launch. For the InSight entry vehicle, this was not possible because there did not exist ground support equipment that could grapple the entry vehicle alone. Thus, the center of gravity and inertias of the entry vehicle were derived from direct measurements of component mass properties and then the center of gravity location of the entry vehicle was derived from the mathematical combination of component mass properties. Additionally, for the InSight entry vehicle the center of gravity location at entry interface is a function of the amount of propellant used on the journey to Mars, which is not directly measured, but derived from thruster total on-time and expected mass flow rate. The center of gravity location used in Monte Carlo simulation is derived from the pre-launch measurements and statistical representation of fuel use prior to entry interface. The mathematical modeling to derive center of gravity location is rooted in simple high-fidelity physics principles and thus the Mathematical Model category is rate High for entry vehicle center of gravity offset at entry interface. The commercial tools used to derive entry vehicle center of gravity from component measurements are well established and thus have high confidence in code quality. However, in the InSight process, the tool used to generate center of gravity location based on statistical fuel use on the transfer to Mars was a custom tool receiving undocumented heritage code review and thus for the Assessment Category Code, the rating is set as Medium. The tool underwent a spot check validation effort comparing it against other tools, so overall the Validation category for entry vehicle center of gravity location at entry interface is rated Medium. The modeling parameters for center of gravity location are complete and relevant for their application so category Model Parameter is rated High.

Table 11-40 and Table 11-41 contain a summary of the Input Model Confidence Assessment for the initial InSight baseline. Because there were no model changes across the three InSight epochs, these assessments are used for Target Margin quantification for ATLO 2.0, As-Built 2.0 and Flight Reference Delta.

11.6 Quantification of InSight Target Margin

The previous subsections discussed the configuration and Input Model Confidence Assessment for an initial InSight Target Margin baseline assessment. They were used to configure an assessment using LMST to quantify Target Margin for the following Critical Metrics: Peak Deceleration, Peak Heat Rate, Total Heating and Parachute Inflation Load, the results of which are reported in this subsection. With four Critical Metrics and three epochs, a total of twelve assessments are performed, with the TMST Assessment Main Window for the four Critical

Table 11-40 Model confidence assessment for InSight regression model inputs – part 1.

Input Name	Assessment Categories - InSight Initial Assessment							
	Mathematical Model		Code		Validation		Model Parameter	
	Rating	Rationale	Rating	Rationale	Rating	Rationale	Rating	Rationale
ca_hc_mult	High	Linearized aero database derived from CFD analysis.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	High	All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
ca_mc_mult	Medium	Linearized aero database derived from CFD analysis and Viking base pressure correction. Greater uncertainty in supersonic CFD modeling.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	High	All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
chute_aero_disp	Medium	Physics-based model with moderate parameterization.	High	Model is independently implemented across multiple simulations with behavior compared.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	High	Model parameters for parachute drag have high pedigree, are complete, relevant and appropriate for their function.
cm_hc_add	High	Linearized aero database derived from CFD analysis.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	High	All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
cm_mc_add	Medium	Linearized aero database derived from CFD analysis and Viking base pressure correction. Greater uncertainty in supersonic CFD modeling.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	High	All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
cn_hc_add	High	Linearized aero database derived from CFD analysis.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	High	All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.

Table 11-41 Model confidence assessment for InSight regression model inputs – part 2.

Input Name	Assessment Categories - InSight Initial Assessment							
	Mathematical Model		Code		Validation		Model Parameter	
	Rating	Rationale	Rating	Rationale	Rating	Rationale	Rating	Rationale
density_40km_20km	High	Based on high pedigree, complete physics-based CFD modeling.	Medium	Parametrized atmosphere profile generator code not reviewed.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Medium	Temperature is extrapolated at altitudes below 5 km based on surface temperature assumptions.
density_60km_40km	High	Based on high pedigree, complete physics-based CFD modeling.	Medium	Parametrized atmosphere profile generator code not reviewed.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Medium	Temperature is extrapolated at altitudes below 5 km based on surface temperature assumptions.
density_80km_60km	High	Based on high pedigree, complete physics-based CFD modeling.	Medium	Parametrized atmosphere profile generator code not reviewed.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Medium	Temperature is extrapolated at altitudes below 5 km based on surface temperature assumptions.
density_above_100km	High	Based on high pedigree, complete physics-based CFD modeling.	Medium	Parametrized atmosphere profile generator code not reviewed.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Low	At altitudes above 100 km the appropriateness and relevance of the density model parameters becomes low.
EFPA	High	Mathematical modeling utilizes high fidelity orbital mechanics and statistical state estimation while tracking and incorporating a comprehensive set of factors.	High	Orbit determination tools shared by multiple missions with results frequently peer reviewed.	High	Orbit determination techniques utilized in modeling entry flight path angle are utilized by all planetary missions and thus providing numerous validation opportunities.	High	Parameters have high pedigree, completeness and relevance for the application.
zcg_ei	High	Mathematical modeling rooted in simple high-fidelity physics principles.	Medium	Tool used to generate the statistical fuel use on the transfer to Mars is a custom tool and received undocumented heritage code review.	Medium	Tool used to generate the statistical fuel use on the transfer to Mars has some validation against other tool results.	High	Parameters have high pedigree, completeness and relevance for the application.

Metrics for the ATLO 2.0 epoch shown in Figures 11-21 – 11-24 at the end of this subsection as representative of the twelve assessments. The Model Confidence Ratings shown in the figures are those previously documented in Table 11-40 – Table 11-41, with the rationales captured in the rating rationale input feature in TMST.

The Target Margin results for all twelve assessments are summarized in Tables 11-42 – 11-53. The tables provide a summary of the contributions to Target Margin of the six linear regression model terms, and the six highest contributors from the nonlinear regression model terms. For the linear terms (the modeling inputs), the Confidence Score is shown for each in the third column. The higher the Confidence Score, the lower the model confidence and thus greater contribution to Target Margin. The fourth column, titled Target Margin contribution, shows the contribution each regression model term makes to total Target Margin. The final three columns summarize for each model term the estimated uncertainty type split, in percentage, between epistemic, aleatory and combined uncertainty. The bottom row of each table lists the total Target Margin, and the estimated uncertainty type split for total Target Margin.

For the Critical Metric Peak Deceleration, the resulting total Target Margin for the three epochs ranges from 0.46 Earth *g* to 0.49 Earth *g*, as shown in Table 11-42 – Table 11-44. In all three epochs, the first regression model terms density_40km_20km and cm_hc_add contribute to more than half the total Target Margin primarily because of their high level of influence on the Critical Metric. It should be noted that because all three epochs are configured the same and utilize the same model confidence ratings, the regression model unique to each epoch is the sole source of the differences in the resulting total Target Margin (true for all four Critical Metric assessments). Epoch Flight Reference Delta has the largest Target Margin at 0.49 Earth *g*, which seems counter intuitive given it should represent the most mature EDL system. The higher Target Margin is driven by the greater influence of cn_hc_add relative to zcg_ei in the earlier two epoch regression models, and density_above_100km being included in the model by positive significance test result. The estimated separable epistemic uncertainty (mathematically separable from aleatory uncertainty) contributing to the Target Margin ranges from 12.5% to 12.8% across the epochs. The primary contribution to the epistemic uncertainty for Peak Deceleration is EFPA.

As shown in Tables 11-45 – 11-47, for Peak Heat Rate, the most significant contributor to total Target Margin for all epochs is not a linear term, but rather the second-order term for density_60km_40km, which is the density band where most Monte Carlo cases experience peak heat rate. The two biggest linear contributors to Peak Heat Rate Target Margin are EFPA and density_60km_40km. Together with the second-order density_60km_40km term, these three regression model terms contribute more than half of the Target Margin, with a total Target Margin range across all three epochs of 1.8 W/cm^2. For Peak Heat Rate, the estimated separable epistemic uncertainty contribution range across the epochs is 8.9% to 9.2%, again representing potential reduction in uncertainty, and therefore Target Margin, before landing, and primarily attributed to EFPA.

Results for the quantification of Target Margin for the Critical Metric Total Heating range from 55 J/cm^2 to 57 J/cm^2, as shown in Tables 11-48 – 11-50. The most significant contributors to Target Margin are the linear terms cm_hc_add, EFPA and ca_hc_mult, contributing to more than half of the Target Margin primarily because of their strong influence on the metric response via the regression model. The estimated contribution to Target Margin from separable epistemic uncertainty ranges from 16.4% to 16.6% across the epochs, again primarily from EFPA.

Results for Critical Metric Parachute Inflation Load are shown in Tables 11-51 – 11-53. The quantified Target Margin range is 1,607 lbf to 1,756 lbf across the three epochs. This variation is tied to the variation in the regression modeling discussed in Section 11.3.2.4, along with how that variation interacts with the Model Confidence Ratings. Nearly half of the Target Margin for all three epochs is contributed by the linear term chute_aero_disp, which is the parachute drag term. The separable epistemic uncertainty contribution to Target Margin for Parachute Inflation Load is 0% for all epochs because all six input models for the Parachute Inflation Load regression model are assessed to contain 100% aleatory uncertainty. Therefore, no reduction in model uncertainty and Target Margin are expected before landing.

The results of Target Margin quantification for the three InSight epochs ATLO 2.0, As-Built 2.0, and Flight Reference Delta show varying trends across epochs. Generally, as EDL design and simulation capability mature across project lifecycle, it would be expected that Target Margin would reduce over time from early to later epochs. This would be driven by both improving Confidence Ratings and the tightening of input distributions. As stated earlier, because the three epochs are relatively close in time and later in the InSight Project lifecycle, they are configured identically in Confidence Ratings and input distributions, the only difference across the epochs being the regression model is unique to each. Therefore, any differences in the Target Margin results can be attributed to regression model differences. The regression model is derived from the DOE data generated by the EDL simulations via the linear algebra formulation given by Equation 6-3 in Section 6. Because the formulation is deterministic and not responsible for any regression model variation, the only source of variation in the regression models is from the DOE simulation data.

For Critical Metric Peak Deceleration, the Target Margin trend across the epochs, in chronological order, is 0.46 Earth *g*, 0.46 Earth *g*, and 0.49 Earth *g*, displaying the opposite trend

Table 11-42 Target Margin results for Peak Deceleration ATLO 2.0.

Peak Deceleration Regression Model Term (Input Variables)			Confidence Score	ATLO 2.0 Target Margin Contribution (Earth g)	Estimated Uncertainty Contribution Split: Epistemic	Aleatory	Combined
0	2	density_40km_20km	0.2292	0.16900	0.0%	100.0%	
0	1	cm_hc_add	0.1875	0.08703	0.0%	100.0%	
0	4	density_60km_40km	0.2292	0.08046	0.0%	100.0%	
0	3	EFPA	0.1667	0.05564	99.8%	0.2%	
0	5	zcg_ei	0.2083	1.6500E-05	61.8%	38.2%	
0	6	density_above_100km	0.2500	0	-	-	
2	4	density_40km_20km/density_60km_40km		0.02271	0.0%	100.0%	0.0%
1	4	cm_hc_add/density_60km_40km		0.01818	0.0%	100.0%	0.0%
2	2	density_40km_20km		0.01713	0.0%	100.0%	0.0%
3	3	EFPA		0.00332	99.6%	0.0%	0.4%
4	4	density_60km_40km		0.00431	0.0%	100.0%	0.0%
1	3	cm_hc_add/EFPA		0.00233	0.0%	0.2%	99.8%
Aggregate Totals for All Model Terms				0.46	12.8%	86.7%	0.5%

Table 11-43 Target Margin results for Peak Deceleration As-Built 2.0.

Peak Deceleration Regression Model Term (Input Variables)			Confidence Score	As-Built 2.0 Target Margin Contribution (Earth g)	Estimated Uncertainty Contribution Split: Epistemic	Aleatory	Combined
0	2	density_40km_20km	0.2292	0.17020	0.0%	100.0%	
0	1	cm_hc_add	0.1875	0.08709	0.0%	100.0%	
0	4	density_60km_40km	0.2292	0.08120	0.0%	100.0%	
0	3	EFPA	0.1667	0.05566	99.8%	0.2%	
0	5	zcg_ei	0.2083	5.7190E-06	61.8%	38.2%	
0	6	density_above_100km	0.2500	0	-	-	
2	4	density_40km_20km/density_60km_40km		0.02250	0.0%	100.0%	0.0%
1	4	cm_hc_add/density_60km_40km		0.01818	0.0%	100.0%	0.0%
2	2	density_40km_20km		0.01747	0.0%	100.0%	0.0%
3	3	EFPA		0.01664	99.6%	0.0%	0.4%
4	4	density_60km_40km		0.00428	0.0%	100.0%	0.0%
1	3	cm_hc_add/EFPA		0.00268	0.0%	0.2%	99.8%
Aggregate Totals for All Model Terms				0.46	12.8%	86.6%	0.6%

Table 11-44 Target Margin results for Peak Deceleration Flight Reference Delta.

Peak Deceleration Regression Model Term (Input Variables)			Flight Reference Delta Confidence Score	Target Margin Contribution (Earth g)	Estimated Uncertainty Contribution Split Epistemic	Aleatory	Combined
0	2	density_40km_20km	0.2292	0.17050	0.0%	100.0%	
0	1	cm_hc_add	0.1875	0.08767	0.0%	100.0%	
0	4	density_60km_40km	0.2292	0.08159	0.0%	100.0%	
0	3	EFPA	0.1667	0.05532	99.8%	0.2%	
0	5	cn_hc_add	0.1875	0.01248	0.0%	100.0%	
0	6	density_above_100km	0.2500	0.00034	0.0%	100.0%	
2	4	density_40km_20km/density_60km_40km		0.02355	0.0%	100.0%	0.0%
1	4	cm_hc_add/density_60km_40km		0.01833	0.0%	100.0%	0.0%
2	2	density_40km_20km		0.01752	0.0%	100.0%	0.4%
3	3	EFPA		0.00582	99.6%	0.0%	0.4%
1	5	cm_hc_add/cn_hc_add		0.00443	0.0%	100.0%	0.0%
4	4	density_60km_40km		0.00436	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				0.49	12.5%	89.9%	0.6%

Table 11-45 Target Margin results for Peak Heat Rate ATLO 2.0.

Peak Heat Rate Regression Model Term (Input Variables)			ATLO 2.0 Confidence Score	Target Margin Contribution (W/cm^2)	Estimated Uncertainty Contribution Split Epistemic	Aleatory	Combined
0	3	EFPA	0.1667	0.16350	99.8%	0.2%	
0	6	density_60km_40km	0.2292	0.17910	0.0%	100.0%	
0	4	ca_hc_mult	0.1875	0.11510	0.0%	100.0%	
0	5	cm_hc_add	0.1875	0.10740	0.0%	100.0%	
0	1	density_80km_60km	0.2292	0.12550	0.0%	100.0%	
0	2	density_above_100km	0.2500	0.00282	0.0%	100.0%	
6	6	density_60km_40km		0.84940	0.0%	100.0%	0.0%
5	5	cm_hc_add		0.08988	0.0%	100.0%	0.0%
3	6	EFPA/density_60km_40km		0.07048	0.0%	0.2%	99.8%
5	6	cm_hc_add/density_60km_40km		0.06257	0.0%	100.0%	0.0%
4	6	ca_hc_mult/density_60km_40km		0.03519	0.0%	100.0%	0.0%
1	5	density_80km_60km/cm_hc_add		0.01220	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				1.8	8.9%	87.2%	3.8%

Table 11-46 Target Margin results for Peak Heat Rate As-Built 2.0.

Peak Heat Rate Regression Model Term (Input Variables)			As-Built 2.0				
					Estimated Uncertainty Contribution Split		
			Confidence Score	Target Margin Contribution (W/cm^2)	Epistemic	Aleatory	Combined
0	3	EFPA	0.1667	0.16220	99.8%	0.2%	
0	6	density_60km_40km	0.2292	0.17980	0.0%	100.0%	
0	4	ca_hc_mult	0.1875	0.11450	0.0%	100.0%	
0	5	cm_hc_add	0.1875	0.11230	0.0%	100.0%	
0	1	density_80km_60km	0.2292	0.12410	0.0%	100.0%	
0	2	density_above_100km	0.2500	0.00273	0.0%	100.0%	
6	6	density_60km_40km		0.84950	0.0%	100.0%	0.0%
3	6	EFPA/density_60km_40km		0.06938	0.0%	0.2%	99.8%
5	6	cm_hc_add/density_60km_40km		0.06346	0.0%	100.0%	0.0%
4	6	ca_hc_mult/density_60km_40km		0.03476	0.0%	100.0%	0.0%
5	5	cm_hc_add		0.03093	0.0%	100.0%	0.0%
1	5	density_80km_60km/cm_hc_add		0.01228	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				1.8	9.1%	87.0%	0.0%

Table 11-47 Target Margin results for Peak Heat Rate Flight Reference Delta.

Peak Heat Rate Regression Model Term (Input Variables)			Flight Reference Delta				
					Estimated Uncertainty Contribution Split		
			Confidence Score	Target Margin Contribution (W/cm^2)	Epistemic	Aleatory	Combined
0	3	EFPA	0.1626	0.16280	99.8%	0.2%	
0	6	density_60km_40km	0.2292	0.17470	0.0%	100.0%	
0	4	ca_hc_mult	0.1875	0.11480	0.0%	100.0%	
0	5	cm_hc_add	0.1875	0.11270	0.0%	100.0%	
0	1	density_80km_60km	0.2292	0.12560	0.0%	100.0%	
0	2	density_above_100km	0.2500	0.00272	0.0%	100.0%	
6	6	density_60km_40km		0.84530	0.0%	100.0%	0.0%
3	6	EFPA/density_60km_40km		0.06965	0.0%	0.2%	99.8%
5	6	cm_hc_add/density_60km_40km		0.06406	0.0%	100.0%	0.0%
5	5	cm_hc_add		0.04199	0.0%	100.0%	0.0%
4	6	ca_hc_mult/density_60km_40km		0.03477	0.0%	100.0%	0.0%
1	5	density_80km_60km/cm_hc_add		0.01245	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				1.8	9.2%	86.9%	3.9%

Table 11-48 Target Margin results for Total Heating ATLO 2.0.

Total Heating Regression Model Term (Input Variables)			Confidence Score	ATLO 2.0 Target Margin Contribution (J/cm^2)	Estimated Uncertainty Contribution Split Epistemic	Aleatory	Combined
0	1	cm_hc_add	0.1875	12.800	0.0%	100.0%	
0	3	EFPA	0.1667	9.158	99.8%	0.2%	
0	4	ca_hc_mult	0.1875	7.972	0.0%	100.0%	
0	2	density_40km_20km	0.2292	8.721	0.0%	100.0%	
0	6	density_80km_60km	0.2292	6.712	0.0%	100.0%	
0	5	density_60km_40km	0.2292	0.461	0.0%	100.0%	
1	3	cm_hc_add/EFPA		2.057	0.0%	0.2%	99.8%
1	2	cm_hc_add/density_40km_20km		2.086	0.0%	100.0%	0.0%
6	6	density_80km_60km		0.000	0.0%	100.0%	0.0%
5	5	density_60km_40km		0.000	0.0%	100.0%	0.0%
2	5	density_40km_20km/density_60km_40km		1.098	0.0%	100.0%	0.0%
2	2	density_40km_20km		1.088	0.0%	100.0%	0.0%
		Aggregate Totals for All Model Terms		57	16.6%	77.0%	6.3%

Table 11-49 Target Margin results for Total Heating As-Built 2.0.

Total Heating Regression Model Term (Input Variables)			Confidence Score	As-Built 2.0 Target Margin Contribution (J/cm^2)	Estimated Uncertainty Contribution Split Epistemic	Aleatory	Combined
0	1	cm_hc_add	0.1875	12.100	0.0%	100.0%	
0	3	EFPA	0.1667	8.925	99.8%	0.2%	
0	4	ca_hc_mult	0.1875	7.951	0.0%	100.0%	
0	2	density_40km_20km	0.2292	8.469	0.0%	100.0%	
0	6	density_80km_60km	0.2292	6.590	0.0%	100.0%	
0	5	density_60km_40km	0.2292	0.399	0.0%	100.0%	
1	3	cm_hc_add/EFPA		1.905	0.0%	0.2%	99.8%
1	2	cm_hc_add/density_40km_20km		2.019	0.0%	100.0%	0.0%
6	6	density_80km_60km		0.000	0.0%	100.0%	0.0%
5	5	density_60km_40km		0.000	0.0%	100.0%	0.0%
2	5	density_40km_20km/density_60km_40km		1.011	0.0%	100.0%	0.0%
2	2	density_40km_20km		1.011	0.0%	100.0%	0.0%
		Aggregate Totals for All Model Terms		55	16.6%	77.1%	6.3%

Table 11-50 Target Margin results for Total Heating Flight Reference Delta.

Total Heating Regression Model Term (Input Variables)			Flight Reference Delta Confidence Score	Target Margin Contribution (J/cm^2)	Estimated Uncertainty Contribution Split Epistemic	Aleatory	Combined
0	1	cm_hc_add	0.1875	12.300	0.0%	100.0%	
0	3	EFPA	0.1667	8.953	99.8%	0.2%	
0	4	ca_hc_mult	0.1875	7.909	0.0%	100.0%	
0	2	density_40km_20km	0.2292	8.457	0.0%	100.0%	
0	6	density_80km_60km	0.2292	6.571	0.0%	100.0%	
0	5	density_60km_40km	0.2292	0.371	0.0%	100.0%	
1	3	cm_hc_add/EFPA		1.997	0.0%	0.2%	99.8%
6	6	density_80km_60km		0.000	0.0%	100.0%	0.0%
1	2	cm_hc_add/density_40km_20km		2.018	0.0%	100.0%	0.0%
5	5	density_60km_40km		0.000	0.0%	100.0%	0.0%
2	5	density_40km_20km/density_60km_40km		1.088	0.0%	100.0%	0.0%
2	2	density_40km_20km		1.044	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				55	16.4%	77.2%	6.4%

Table 11-51 Target Margin results for Parachute Inflation Load ATLO 2.0.

Parachute Inflation Load Regression Model Term (Input Variables)			ATLO 2.0 Confidence Score	Target Margin Contribution (lbf)	Estimated Uncertainty Contribution Split Epistemic	Aleatory	Combined
0	1	chute_aero_disp	0.4200	746.60	0.0%	100.0%	
0	4	ca_mc_mult	0.2083	250.50	0.0%	100.0%	
0	5	cm_hc_add	0.1875	157.50	0.0%	100.0%	
0	2	cm_mc_add	0.2083	170.60	0.0%	100.0%	
0	3	density_sum_40km_20km	0.2292	17.96	0.0%	100.0%	
0	6	density_sum_60km_40km	0.2292	4.42	0.0%	100.0%	
2	5	cm_mc_add/cm_hc_add		108.00	0.0%	100.0%	0.0%
3	5	density_sum_40km_20km/cm_hc_add		54.10	0.0%	100.0%	0.0%
1	2	chute_aero_disp/cm_mc_add		78.11	0.0%	100.0%	0.0%
1	5	chute_aero_disp/cm_hc_add		73.52	0.0%	100.0%	0.0%
4	4	ca_mc_mult		47.49	0.0%	100.0%	0.0%
5	5	cm_hc_add		29.22	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				1756	0.0%	100.0%	0.0%

Table 11-52 Target Margin results for Parachute Inflation Load As-Built 2.0.

Parachute Inflation Load Regression Model Term (Input Variables)			As-Built 2.0				
					Estimated Uncertainty Contribution Split		
			Confidence Score	Target Margin Contribution (lbf)	Epistemic	Aleatory	Combined
0	1	chute_aero_disp	0.4200	741.10	0.0%	100.0%	
0	4	ca_mc_mult	0.2083	244.90	0.0%	100.0%	
0	5	cm_hc_add	0.1875	167.50	0.0%	100.0%	
0	2	cm_mc_add	0.2083	165.40	0.0%	100.0%	
0	3	density_40km_20km	0.2292	25.28	0.0%	100.0%	
0	6	density_60km_40km	0.2292	4.54	0.0%	100.0%	
2	5	cm_mc_add/cm_hc_add		175.40	0.0%	100.0%	0.0%
5	5	cm_hc_add		0.00	0.0%	100.0%	0.0%
1	5	chute_aero_disp/cm_hc_add		75.54	0.0%	100.0%	0.0%
1	3	chute_aero_disp/density_40km_20km		10.21	0.0%	100.0%	0.0%
6	6	density_60km_40km		0.00	0.0%	100.0%	0.0%
4	6	ca_mc_mult/density_60km_40km		2.01	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				1615	0.0%	100.0%	0.0%

Table 11-53 Target Margin results for Parachute Inflation Load Flight Reference Delta.

Parachute Inflation Load Regression Model Term (Input Variables)			Flight Reference Delta				
					Estimated Uncertainty Contribution Split		
			Confidence Score	Target Margin Contribution (lbf)	Epistemic	Aleatory	Combined
0	1	chute_aero_disp	0.4200	737.60	0.0%	100.0%	
0	4	ca_mc_mult	0.2083	246.40	0.0%	100.0%	
0	5	cm_hc_add	0.1875	172.60	0.0%	100.0%	
0	2	cm_mc_add	0.2083	171.00	0.0%	100.0%	
0	3	density_40km_20km	0.2292	22.04	0.0%	100.0%	
0	6	density_60km_40km	0.2292	4.20	0.0%	100.0%	
2	5	cm_mc_add/cm_hc_add		153.00	0.0%	100.0%	0.0%
5	5	cm_hc_add		0.00	0.0%	100.0%	0.0%
1	3	chute_aero_disp/density_40km_20km		79.47	0.0%	100.0%	0.0%
3	6	density_40km_20km/density_60km_40km		8.95	0.0%	100.0%	0.0%
6	6	density_60km_40km		5.17	0.0%	100.0%	0.0%
2	6	cm_mc_add/density_60km_40km		3.31	0.0%	100.0%	0.0%
Aggregate Totals for All Model Terms				1607	0.0%	100.0%	0.0%

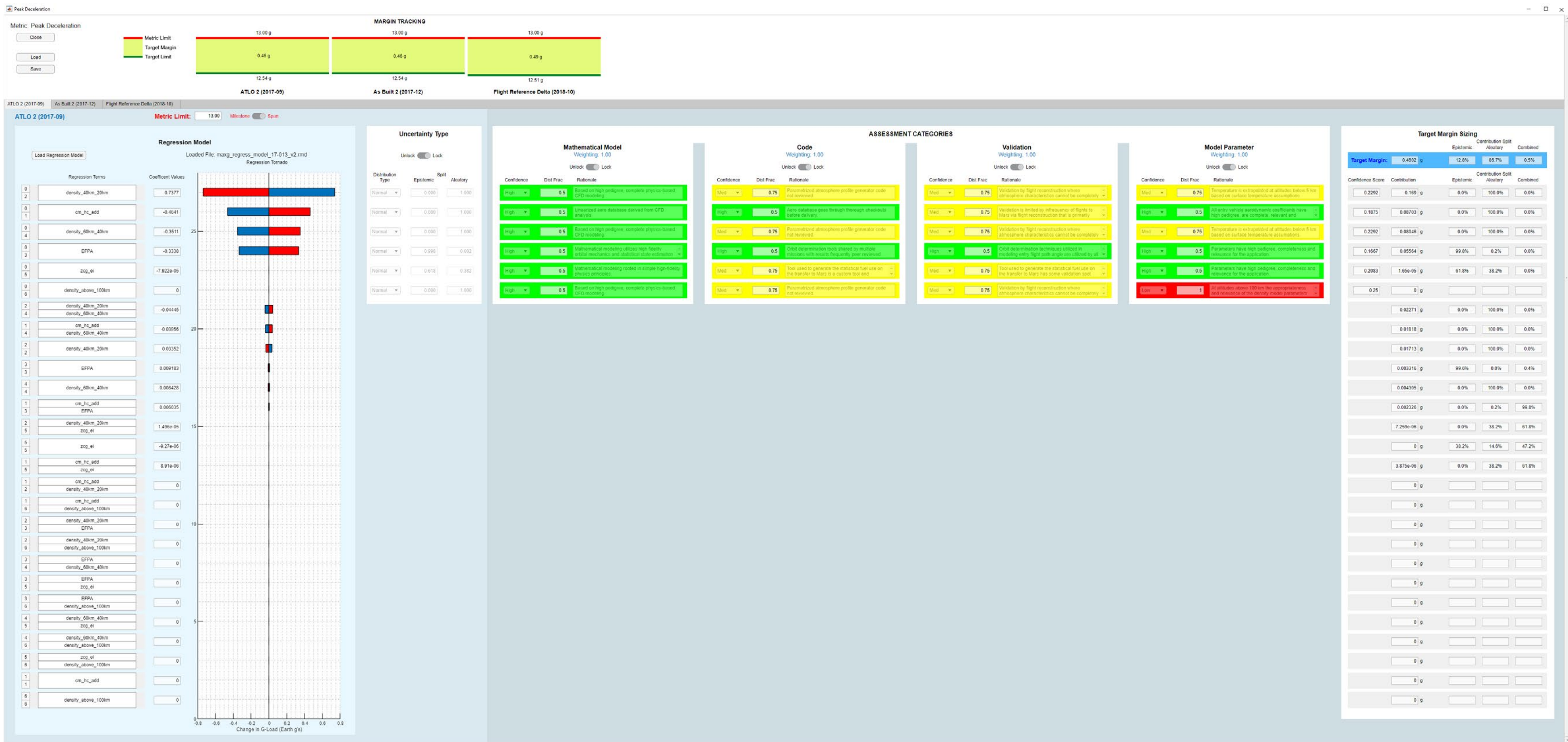


Figure 11-21 TMST configured for InSight Peak Deceleration Target Margin assessment for ATLO 2.0 epoch.

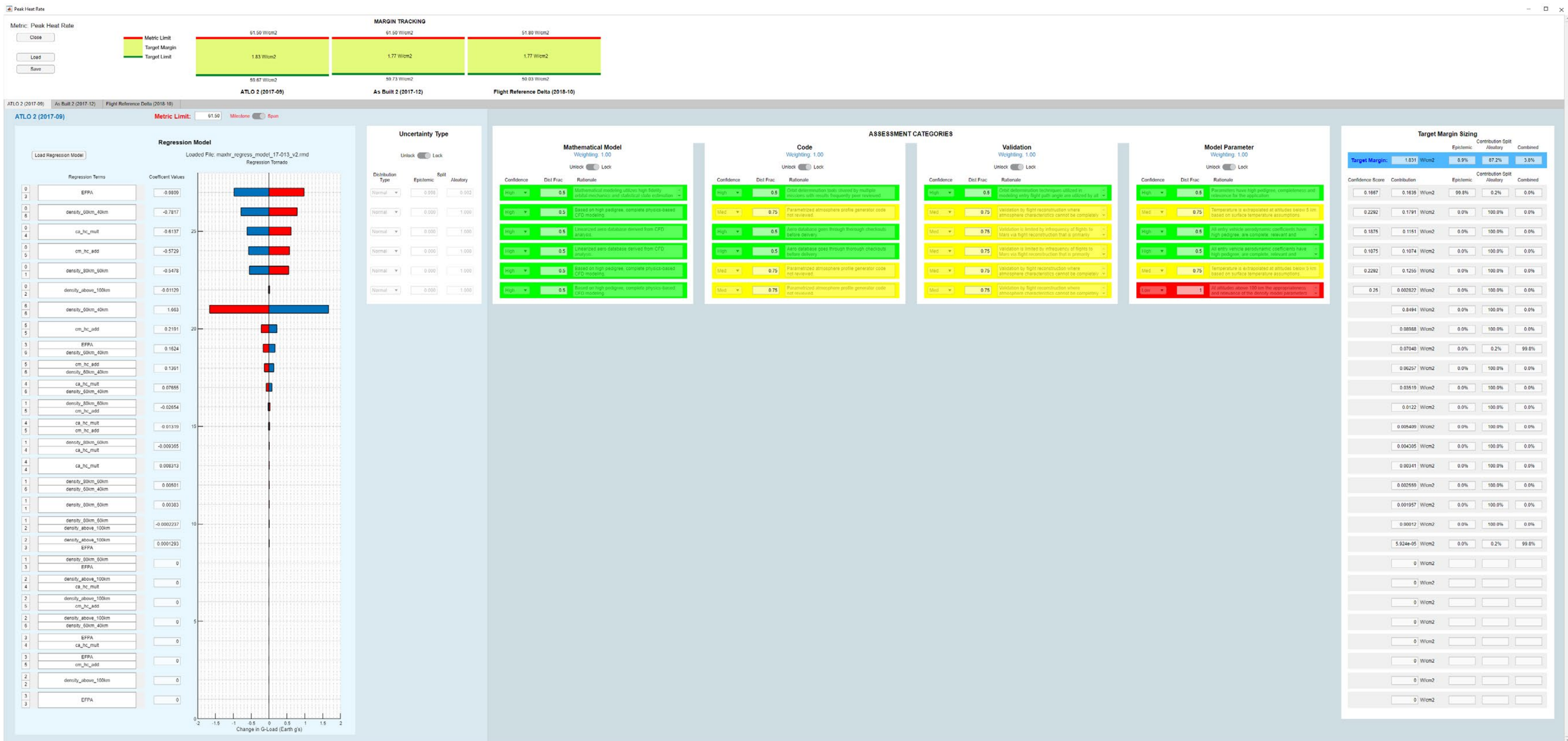


Figure 11-22..TMST configured for InSight Peak Heat Rate Target Margin assessment for ATLO 2.0 epoch.

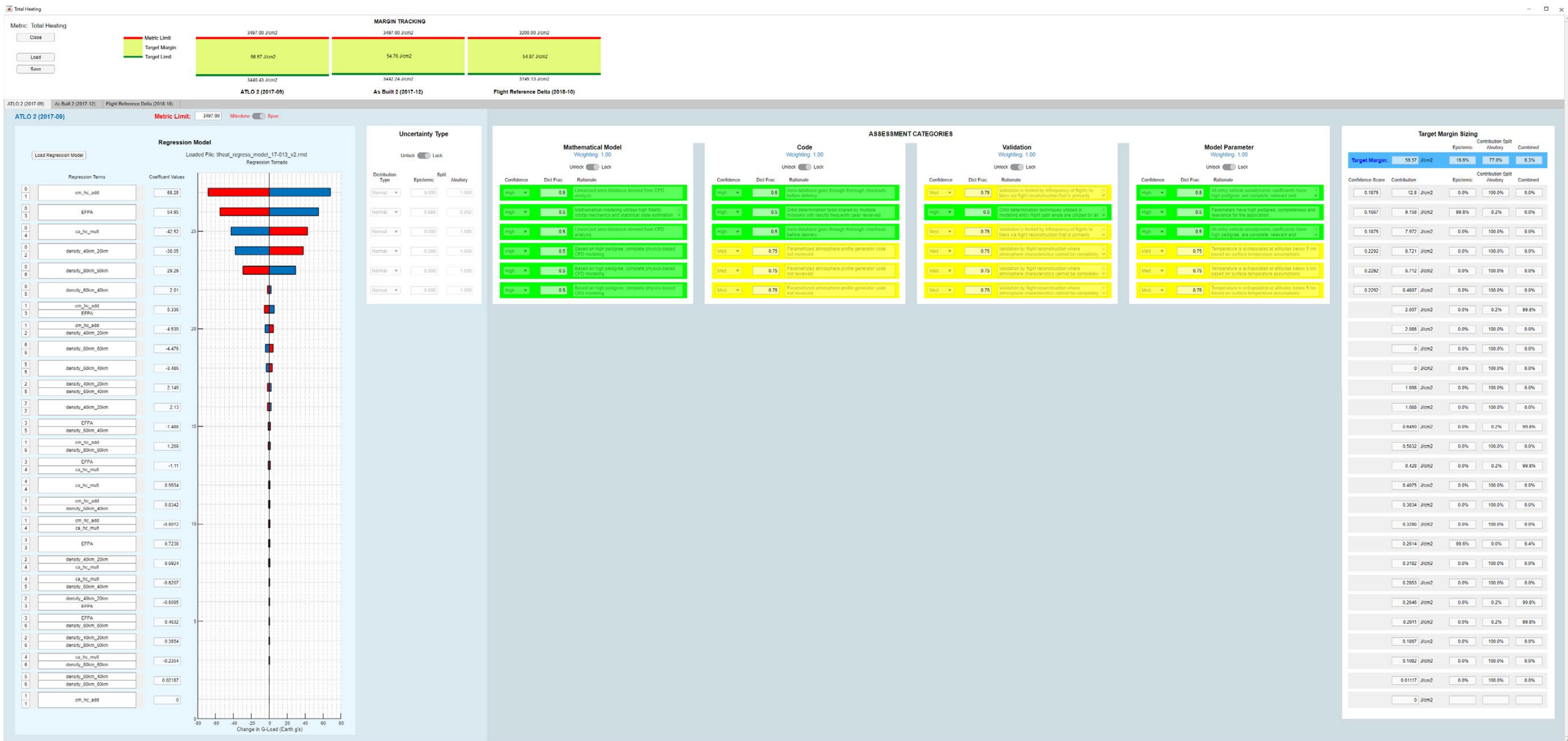


Figure 11-23 TMST configured for InSight Total Heat Target Margin assessment for ATLO 2.0 epoch.

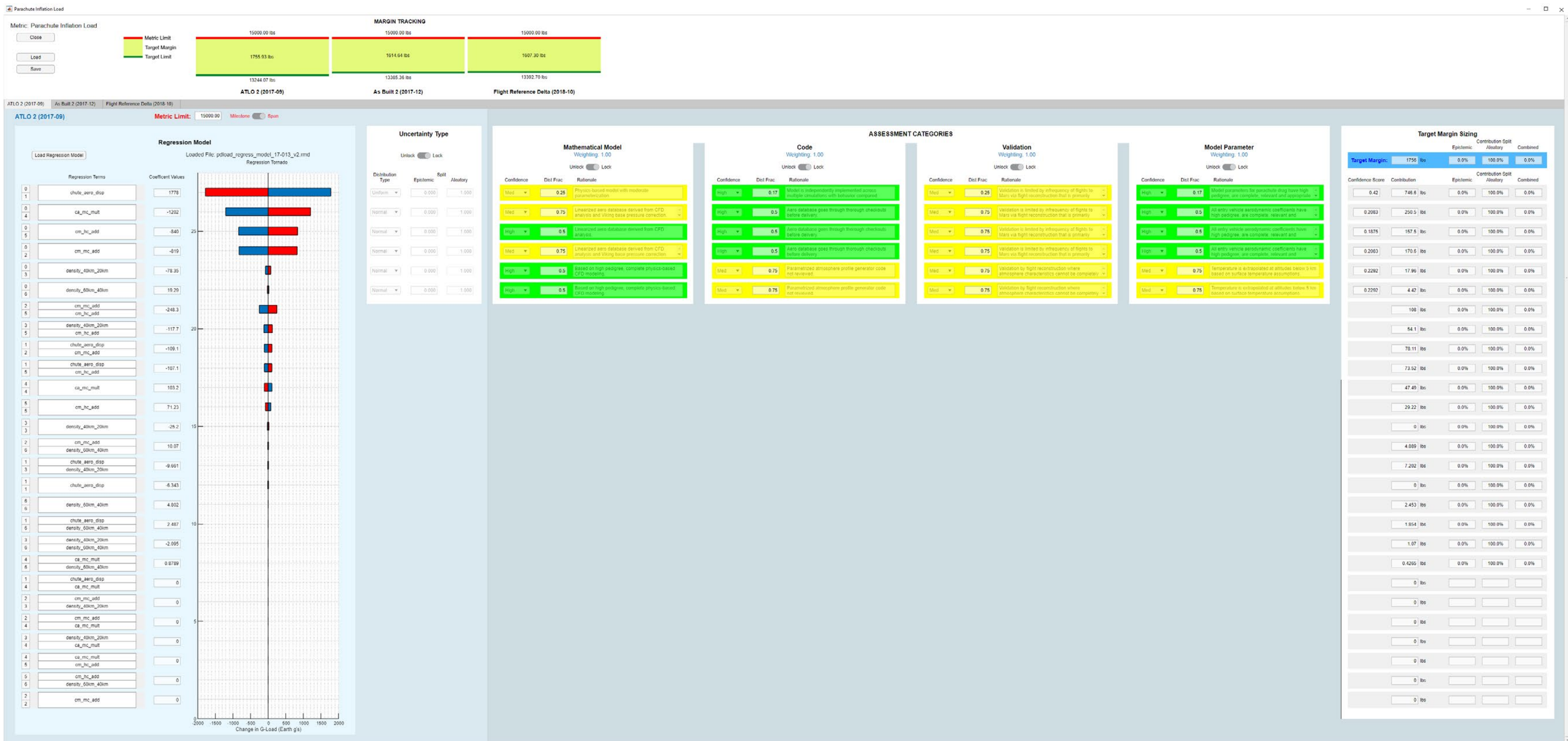


Figure 11-24 TMST configured for InSight Parachute Inflation Load Target Margin assessment for ATLO 2.0 epoch.

one might expect from earlier to later epochs. The primary reason for this is the change in Model Inputs of Highest Influence set from As-Built 2.0 to Flight Reference Delta, where cn_hc_add replaced zcg_ei, with cn_hc_add showing higher influence, and thus greater contribution to Target Margin.

The Target Margin trend for Critical Metric Peak Heat Rate is consistent among all the Critical Metrics. The values across all epochs is 1.8 W/cm^2, with regression model tornado plots provided by TMST showing very consistent regression model behavior.

Critical Metric Total Heating shows more significant Target Margin variation across the epochs, with values 57 J/cm^2, 55 J/cm^2, and 55 J/cm^2. Again, the variation is reflected in the regression model terms, where ATLO 2.0 linear terms of highest influence have slightly stronger influence than the same terms for As-Built 2.0 and Flight Reference Delta.

As was discussed previously, Critical Metric Peak Inflation Load also has significant variation across the epochs with values 1,775 lbf, 1,615 lbf, and 1,607 lbf, with ATLO 2.0 showing the most significant difference in regression model terms.

The main simulation configuration difference across all epochs is vehicle mass properties, although the difference is relatively minor and in some cases trends opposite the effect expected on the Critical Metrics. Investigation into how much variation in Target Margin might be expect from the regression model data is future work. Further discussion of Target Margin quantification results in the context of comparison to the original Seed Methodology is provided in the next subsection.

11.7 Comparison to Seed Methodology

This section continues the discussion of the Target Margin quantification for an initial InSight assessment and extends it to include a comparison to the sizing of Target Margin using the

Seed Methodology. Tables 11-53 – 11-57 contain a summary of the Target Margin quantification for InSight from the previous section for the four Critical Metrics, with each table containing results across the three epochs. Also shown in the tables is the Critical Metric 99% performance value from the Monte Carlo simulation for each epoch, the margin on the 99% performance against the Metric Limit and Target Limit, the Monte Carlo standard deviation for the Critical Metric for each epoch, the Target Margin for each epoch expressed as a fraction/multiple of standard deviation, and the percent contribution from epistemic uncertainty. The bottom two rows of the tables contain Target Margin resulting from applying the Seed Methodology to the Monte Carlo results for each epoch. Above each table, shown in Figure 11-25 – 11-28, is the Margin Tracking display from TMST for the particular Critical Metric.

Figure 11-25 Peak Deceleration Target Margin trends across epochs.

The summary for Peak Deceleration Target Margin quantification is shown in Table 11-54 on the previous page. For each epoch, the Metric Limit is set to 13 Earth *g*, based on the structural capabilities of the spacecraft. This Metric Limit did not change during the lifecycle of InSight. From the previous section, the Target Margin was quantified as 0.46 Earth *g* for ATLO 2.0, 0.46 Earth *g* for As-Built 2.0, and 0.49 Earth *g* for Flight Reference Delta. Thus, the Target Limit for each becomes 12.54 Earth *g*, 12.54 Earth *g*, and 12.51 Earth *g*, respectively. As discussed in

Table 11-54 Summary of Peak Deceleration Target Margin results for InSight.

Peak Deceleration	InSight Epoch		
	ATLO 2.0	As-Built 2.0	Flight Reference Delta
Metric Limit (Earth g)	13.00	13.00	13.00
Target Margin (Earth g)	**0.46**	**0.46**	**0.49**
Target Limit (Earth g)	12.54	12.54	12.51
Monte Carlo 99% High (Earth g)	8.11	8.18	8.15
Monte Carlo Margin Against Metric Limit (Earth g)	4.89	4.82	4.85
Monte Carlo Margin Against Target Limit (Earth g)	4.43	4.36	4.36
Monte Carlo Standard Deviation (Earth g)	0.26	0.26	0.26
Target Margin STD Fraction	1.77	1.77	1.88
Percent Epistemic Contribution (%)	**12.8**	**12.8**	**12.5**
Seed Medthodology Target Margin (Earth g)	0.26	0.26	0.26
Seed Medthodology Target Margin STD Fraction	1.00	1.00	1.00

the previous section, the difference between the resulting Target Margin is due to the difference in regression models for each epoch.

InSight measured Critical Metric margin against 99% Monte Carlo performance, which is shown in row four. Peak Deceleration 99% performance was 8.11 Earth *g*, 8.18 Earth *g*, and 8.15 Earth *g* for the three epochs. Note that this is well below both the Metric Limit and Target Limit and thus the Monte Carlo performance margin for Peak Deceleration against the Metric Limit is 4.89 Earth *g*, 4.82 Earth *g*, and 4.85 Earth *g* across the epochs. The Monte Carlo performance margin against Target Limit is somewhat lower at 4.43 Earth *g*, 4.36 Earth *g*, and 4.36 Earth *g* across the epochs. The quantified Target Margin, which is the desired performance margin between 99% performance and the Metric Limit, is more than an order of magnitude smaller than the Monte Carlo simulation performance margin, so there is very healthy margin for Peak Deceleration. The standard deviation for Peak Deceleration performance across the three epochs is 0.26 Earth *g* for each, and thus the quantified Target Margin for each epoch expressed in terms of standard deviation is 1.77σ, 1.77σ and 1.88σ, as shown in eighth row of Table 11-54.

The last two rows of Table 11-54 show the results of Target Margin quantification using the original Seed Methodology. For Peak Deceleration, the Seed Methodology used a full standard deviation of the performance distribution to size Target Margin, as indicated in the last row of the table. For each epoch, because each had a standard deviation of 0.26 Earth *g*, the Seed Methodology sizes Target Margin for each epoch at 0.26 Earth *g*. The Evolved Methodology applied in this dissertation results in a significantly larger Target Margin of 1.77σ, 1.77σ and 1.88σ, rather than the 1σ specified by the Seed Methodology. A discussion of the reason for this difference follows in Section 11.7.1.

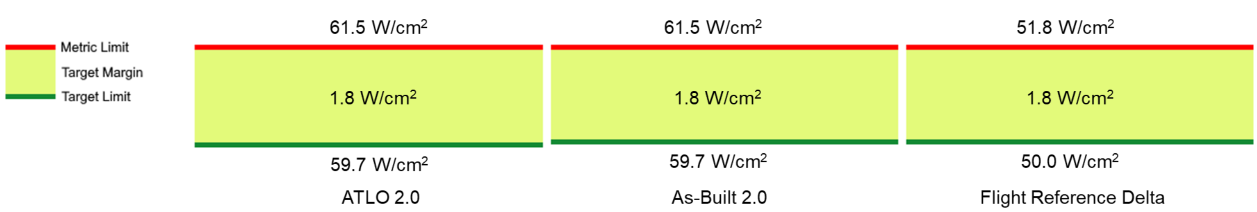


Figure 11-26 Peak Heat Rate Target Margin trends across epochs.

The summary for Peak Heat Rate Target Margin quantification is shown a in Table 11-55. The TMST Margin Tracking display is provided above the table in Figure 11-26. As shown in the table, Peak Heat Rate computed Target Margin is 1.8 W/cm^2 across all three epochs. The Metric Limit for ATLO 2.0 and As-Built 2.0 is 61.5 W/cm^2, but is reduced to 51.8 W/cm^2 for Flight Reference Campaign. This was the result of an updated aerothermal analysis that was conducted prior to launch using updated trajectory information. Subtracting the Target Margin from the

Table 11-55 Summary of Peak Heat Rate Target Margin results for InSight.

Peak Heat Rate	InSight Epoch		
	ATLO 2.0	As-Built 2.0	Flight Reference Delta
Metric Limit (W/cm^2)	61.5	61.5	51.8
Target Margin (W/cm^2)	**1.8**	**1.8**	**1.8**
Target Limit (W/cm^2)	59.7	59.7	50.0
Monte Carlo 99% High (W/cm^2)	48.4	48.3	48.2
Monte Carlo Margin Against Metric Limit (W/cm^2)	13.1	13.2	3.6
Monte Carlo Margin Against Target Limit (W/cm^2)	11.3	11.4	1.8
Monte Carlo Standard Deviation (W/cm^2)	1.06	1.06	1.06
Target Margin STD Fraction	1.73	1.67	1.67
Percent Epistemic Contribution (%)	**8.9**	**9.1**	**9.2**
Seed Medthodology Target Margin (W/cm^2)	0.00	0.00	0.00
Seed Medthodology Target Margin STD Fraction	N/A	N/A	N/A

Metric Limit gives Target Limit, which for the three epochs is 59.7 W/cm^2, 59.7 W/cm^2, and 50.0 W/cm^2. The 99% performance from the Monte Carlos for each epoch for Peak Heat Rate is 48.4 W/cm^2, 48.3 W/cm^2 and 48.2 W/cm^2. This gives a Monte Carlo performance margin with respect to the Metric Limit for each epoch of 13.1 W/cm^2, 13.2 W/cm^2 and 3.6 W/cm^2. With respect to the Target Limit, the Peak Heat Rate Monte Carlo simulation performance margin for each epoch is 11.3 W/cm^2, 11.4 W/cm^2, and 1.8 W/cm^2. The lower margin for the Flight Reference Delta is a result of the lowering of the Metric Limit. The Monte Carlo standard deviation for Peak Heat Rate for each epoch is 1.06 W/cm^2, so that the Target Margin expressed as a fraction of standard deviation for each epoch is 1.73σ, 1.67σ and 1.67σ. The performance margin from the Monte Carlos is nearly an order of magnitude greater than quantified Target Margin for ATLO 2.0 and As-Built 2.0, and about two times the quantity for Flight Reference Delta. This shows the Peak Heat Rate Monte Carlo performance has more than sufficient margin as assessed by the Evolved Methodology. Compared to the original Seed Methodology, these

Target Margins are significantly greater than 1σ, which is the largest Target Margin possible for the Seed Methodology corresponding to a Lower model confidence rating for Method 2. As can be seen in the last two rows of the table, the InSight Seed Methodology required zero Target Margin for Peak Heat Rate because the heat rate computed by the EDL simulations was considered a heating indicator only and not the process by which aerothermal margin was assessed. For InSight, aerothermal margin assessment was provided by a high fidelity CFD analysis using points selected from EDL simulation trajectory. Even with that, it is worthwhile to determine the Peak Heat Rate Target Margin based on aerothermal modeling in the simulation to understand the effects of Modeling Confidence on simulated aerothermal performance results, which is not done via the alternate CFD method of margin assessment.

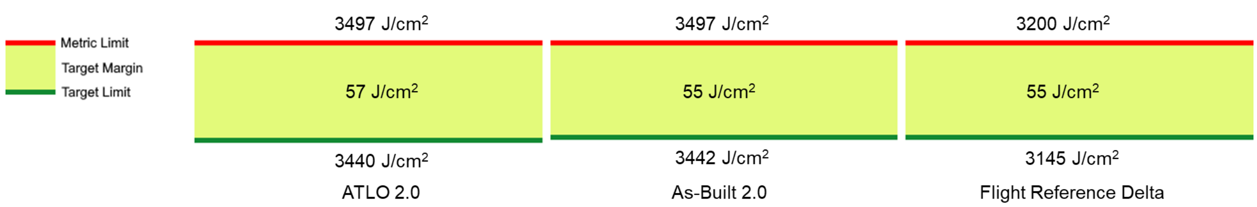


Figure 11-27 Total Heating Target Margin trends across epochs.

The summary for Total Heating Target Margin quantification is shown in Table 11-56, with the TMST Margin Tracking display shown above the table in Figure 11-27. The Target Margin results for Total Heating for the three epochs is 57 J/cm^2, 55 J/cm^2 and 55 J/cm^2 for ATLO 2.0, As-Built 2.0 and Flight Reference Delta, respectively. As with Peak Heat Rate, the Metric Limit for Total Heating was revised down for Flight Reference Delta because of a revised design trajectory, so that the Metric Limit for the three epochs is 3,497 J/cm^2, 3,497 J/cm^2 and 3,200 J/cm^2. Subtracting the Target Margin from the Metric Limit gives the Target Limit,

Table 11-56 Summary of Total Heating Target Margin results for InSight.

Total Heating	InSight Epoch		
	ATLO 2.0	As-Built 2.0	Flight Reference Delta
Metric Limit (J/cm^2)	3497	3497	3200
Target Margin (J/cm^2)	**57**	**55**	**55**
Target Limit (J/cm^2)	3440	3442	3145
Monte Carlo 99% High (J/cm^2)	2832	2814	2804
Monte Carlo Margin Against Metric Limit (J/cm^2)	665	683	396
Monte Carlo Margin Against Target Limit (J/cm^2)	608	628	341
Monte Carlo Standard Deviation (J/cm^2)	50.6	49.9	49.3
Target Margin STD Fraction	1.13	1.10	1.12
Percent Epistemic Contribution (%)	**16.6**	**16.6**	**16.4**
Seed Medthodology Target Margin (J/cm^2)	0.0	0.0	0.0
Seed Medthodology Target Margin STD Fraction	N/A	N/A	N/A

which is 3,440 J/cm^2, 3,442 J/cm^2 and 3,145 J/cm^2 for the three epochs. The 99% Monte Carlo performance for Total Heating for the three epochs is 2,832 J/cm^2, 2,814 J/cm^2 and 2,804 J/cm^2, which results in Monte Carlo simulation performance margin against the Metric Limit for each epoch of 665 J/cm^2, 683 J/cm^2 and 396 J/cm^2. The Monte Carlo performance margin against Target Limit for each epoch is 608 J/cm^2, 628 J/cm^2 and 341 J/cm^2. The Monte Carlo performance margin against both Metric Limit and Target Limit for ATLO 2.0 and As-Built 2.0 is an order of magnitude larger than the computed Target Margin. The Flight Reference Delta has five to six times larger Monte Carlo performance margin relative to both the Metric Limit and the Target Limit than the associated Target Margin. Again, this shows the Monte Carlo performance for Total Heating has significant margin relative to Target Margin sized by the Evolved Methodology. For Total Heating, the Monte Carlo standard deviation for each epoch is 50.6 J/cm^2, 49.9 J/cm^2 and 49.3 J/cm^2, so that the Target Margin for each epoch expressed in terms of standard deviation is 1.13σ, 1.10σ and 1.12σ across the epochs. This says that the Evolved Methodology indicates the Target Margin for Total Heating should be somewhat larger than the 1σ Target Margin specified by the Seed Methodology Method 2 for Lower model confidence. As indicated by the last two rows of Table 11-56, like Peak Heat Rate, the InSight Seed Methodology specified zero Target Margin for Total Heating because, like Peak Heat Rate, total heating computed by the EDL simulation was a heating indicator only and not the process by which aerothermal margin was assessed. As with Peak Heat Rate, even with InSight choice of margin assessment, it is worthwhile to determine the Total Heating Target Margin based on aerothermal modeling in the simulation to understand the effects of Modeling Confidence on simulated aerothermal performance results, which is not done via the alternate method of margin assessment.

The summary for Parachute Inflation Load Target Margin quantification is shown in Table 11-57. As with the other summary tables, the TMST Margin Tracking display is featured above the table in Figure 11-28. The quantified Target Margin for the epochs ATLO 2.0, As-Built 2.0 and Flight Reference Delta are 1,756 lbf, 1,615 lbf and 1,607 lbf, respectively. The Metric Limit for Parachute Inflation Load was set to 15,000 lbf for all three epochs, which, when subtracting Target Margin, results in a Target Limit for each of the epochs of 13,244 lbf, 13,385 lbf and 13,393 lbf. The 99% Monte Carlo performance for Parachute Inflation Load for each epoch is 14,292 lbf, 14,158 lbf and 14,184 lbf. From the 99% performance, actual inflation load performance margin relative to the Metric Limit for each epoch is 708 lbf, 842 lbf and 816 lbf. The performance margin against the Target Limit for Parachute Inflation Load is negative for each epoch, with ATLO 2.0, As-Built 2.0 and Flight Reference Delta being -1,048 lbf, -773 lbf and -791 lbf. While negative margin against the Metric Limit is not acceptable, it was the policy of InSight to accept negative margin against the Target Limit as acceptable but not desirable. The Monte Carlo standard deviation for Parachute Inflation Load for each epoch is 1,183 lbf, 1,170 lbf and 1,194 lbf. In terms of standard deviation, the Parachute Inflation Load Target Margin for each epoch is 1.48σ, 1.38σ and 1.35σ. Shown in the last two rows of Table 11-57, the Seed Methodology originally specified 1σ Target Margin for Parachute Inflation Load, so the assessment using the Evolved Methodology indicates a larger margin is required, as expressed in terms of standard deviation.

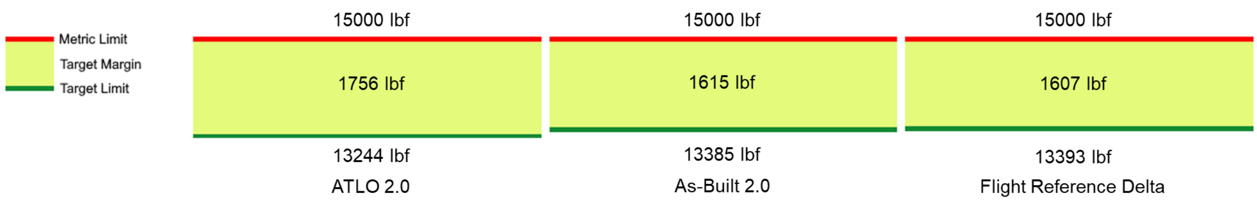


Figure 11-28 Parachute Inflation Target Margin trends across epochs.

Table 11-57 Summary of Parachute Inflation Load Target Margin results for InSight.

Parachute Inflation Load	InSight Epoch		
	ATLO 2.0	As-Built 2.0	Flight Reference Delta
Metric Limit (lbf)	15000	15000	15000
Target Margin (lbf)	**1756**	**1615**	**1607**
Target Limit (lbf)	13244	13385	13393
Monte Carlo 99% High (lbf)	14292	14158	14184
Monte Carlo Margin Against Metric Limit (lbf)	708	842	816
Monte Carlo Margin Against Target Limit (lbf)	-1048	-773	-791
Monte Carlo Standard Deviation (lbf)	1183	1170	1194
Target Margin STD Fraction	1.48	1.38	1.35
Percent Epistemic Contribution (lbf)	**0**	**0**	**0**
Seed Medthodology Target Margin (lbf)	1183	1170	1194
Seed Medthodology Target Margin STD Fraction	1.00	1.00	1.00

For the Critical Metrics Peak Deceleration and Parachute Inflation Load, where the Seed Methodology specified a nonzero Target Margin, the Evolved Methodology results in higher Target Margin. For Peak Deceleration, the Evolved Target Margin is, on average, 1.81σ compared to 1.0σ for the Seed Methodology. Likewise, the Evolved Methodology results in, on average, a Parachute Inflation Load Target Margin of 1.40σ, compared to the Seed Methodology that specified 1.0σ. As discussed earlier, while the Seed Methodology specified zero Target Margin for Peak Heat Rate and Total Heating, the Evolved Methodology results in an average Peak Heat Rate Target Margin of 1.69σ and an average Total Heating Target Margin of 1.12σ. For all Critical Metrics across all InSight epochs, the Evolved Methodology results in specifying larger Target Margin than the Seed Methodology, which has the potential to result in a more robust EDL system.

11.7.1 Key Differences Between Methodologies

An explanation of the result that Evolved Methodology results in larger Target Margin than the Seed Methodology is somewhat challenging to ascertain because the two methodologies are different in how they quantify Target Margin. However, some insights serve to illuminate the reason for the differences.

11.7.1.1 Application of Modeling Confidence

As discussed in Section 3, a key difference between the methodologies is the ability to apply the modeling confidence assessment at a finer resolution for the Evolved Methodology. Because the Evolved Methodology applies the effect of modeling confidence per Model Input of Highest Influence, it provides better resolution than the Seed Methodology, which applies a single combined Confidence Rating to a Critical Metric performance distribution. Additionally, the Seed Methodology Confidence Rating is set by the model of least confidence for a Critical Metric. For example, Table 11-58 lists the Seed Methodology model confidence assessment for the Critical Metric Peak Deceleration. Because Atmosphere Density (Atm Density) and Atmosphere Winds (Atm Winds) are of Lower confidence, the Model Confidence Rating is determined by the Lower rating. Because of this, the resulting Lower Assessment Criteria is applied to the Critical Metric output distribution, the distribution of which is driven by all Model Inputs of Highest Influence. This essentially has the effect of applying a Lower Confidence Rating to all model inputs, independent of their actual Confidence Rating. This leads to the conclusion that the Seed Methodology is more conservative than the Evolved Methodology in the application of modeling confidence to Target Margin quantification.

Table 11-58 Seed Methodology model confidence assessment for Peak Deceleration.

Model	Confidence
Vehicle Mass Props	Higher
Entry Vehicle Aero	Higher
Atm Density	Lower
Atm Winds	Lower

11.7.1.2 Target Margin Arithmetic

Another key difference between the methodologies is the effective way Target Margin contribution from Model Inputs of Highest Influence is summed. For the Seed Methodology, because the Assessment Criteria is applied to the performance output distribution, the individual contributions to Target Margin from the Model Inputs of Highest Influence are combined statistically, which is effectively a root sum square (RSS) process. In the Evolved Methodology, the individual contributions to Target Margin from the Model Inputs of Highest Influence are being added in a stacked fashion through the DOE regression model. It is hypothesized that this effective difference in summing method is a significant contributor to the Evolved Methodology resulting in generally greater Target Margin than the Seed Methodology. To test this hypothesis, the Evolved Methodology was modified to more closely match the Seed Methodology, and applied to Critical Metrics Peak Deceleration and Parachute Inflation Load. The Seed Methodology gave a

Lower Model Confidence Rating to both metrics, so based on the discussion in Section 11.7.1.1, the first modification to the Evolved Methodology is to set the Confidence Rating for all Model Inputs of Highest Influence to Low. The second modification is to RSS the regression model linear terms (the largest contributors to Target Margin), rather than add them. The results are shown in Table 11-59. As can be seen in the table, the modified Evolved Methodology results in a smaller Target Margin than that resulting from the Evolved Methodology, and more closely matches the Target Margin resulting from the Seed Methodology. While this is a necessary result, it is not sufficient to conclusively prove the hypothesis. Further exploration of the drivers of the differences between the methodologies is future work.

Table 11-59 Methodology comparison using modified Evolved Methodology.

Critical Metric	Methodology	Target Margin		
		ATLO 2.0	As-Built 2.0	Flight Reference Delta
Peak Deceleration	Seed Methodology Target Margin (Earth g)	0.26	0.26	0.26
	Modified Evolved Methodology (Earth g)	0.33	0.33	0.34
	Evolved Methodology (Earth g)	0.46	0.46	0.49
Parachute Inflation Load	Seed Methodology Target Margin (lbf)	1183	1170	1194
	Modified Evolved Methodology (lbf)	1300	1293	1292
	Evolved Methodology (lbf)	1756	1615	1607

11.8 Assessment Against InSight Flight Reconstruction

The InSight mission successfully landed on Mars on November 26th, 2018. Upon landing, the spacecraft returned a significant data set recorded during EDL, including 200 Hz inertial measurement unit (IMU) data with rate gyro and accelerometer measurements [39]. These data were used to reconstruct the landing to provide understanding of spacecraft performance during

EDL. The InSight reconstruction effort provides some insight into the Evolved Methodology based on InSight spacecraft performance. Because the methodology is focused on adequate margin at the boundaries of EDL performance space, the mostly nominal InSight landing has, for the most part, performance in the middle of the performance space with large margin against performance limits. Therefore, the conclusions that can be drawn from the InSight flight data relative to the Evolved Methodology is limited. Nonetheless, there are useful observations to be gained.

The following subsections provide an assessment of the Evolved Methodology against the InSight EDL flight reconstruction. The assessment will be made against the Flight Reference Delta epoch, as it represents essentially the flight configuration.

11.8.1 Peak Deceleration

The InSight spacecraft flew a somewhat unexpected EDL trajectory relative to the nominal pre-landing prediction. The first indication of this was seen before touchdown when the spacecraft reached a higher than expected peak deceleration. As measured by the onboard IMU, peak deceleration reached 8.1 Earth *g* while the expected value was 7.6 Earth *g* [40]. As was discovered through post-flight reconstruction, while the vehicle was designed to fly a ballistic trajectory, the vehicle instead trimmed with small angle of attack with significant component in the lift-down direction [31]. This caused the vehicle to fly a steeper trajectory than predicted and carry more velocity into the altitude region of peak deceleration causing a higher-than-expected value. The measured peak deceleration equaled the 99% pre-landing performance prediction. As discussed in the previous subsection, the Metric Limit for the InSight vehicle is 13 Earth *g*, so the margin against the Metric Limit for the actual InSight landing is a very healthy 4.9 Earth *g*. The Evolved Methodology Target Margin against the Metric Limit for the Flight Reference Delta is

0.49 Earth *g*, so during the actual landing the vehicle flew well away from the Metric Limit making it difficult to draw a conclusion about the adequacy of Target Margin quantified using the Evolved Methodology.

The InSight trim angle of attack behavior at 99% pre-landing performance prediction should statistically be a very low probability event, which leads to the conclusion there is something fundamentally misunderstood about the entry vehicle aerodynamic trim behavior. With respect to the Peak Deceleration, cm_hc_add is the input directly tied to trim angle of attack. This would suggest the Input Model Confidence Assessment made for cm_hc_add in quantifying Target Margin should have a lower Confidence Rating than was used in the assessment, particularly in the Assessment Categories for Mathematical Modeling and Validation. It may be that an appropriate uncertainty distribution for cm_hc_add is uniform rather than normal, or a unique Monte Carlo dispersion strategy for cm_hc_add may be required. The issue of predicting trim angle of attack behavior appears to be unique to ballistic entry vehicles like InSight, with vehicles with designed nonzero angle of attack having better trim angle predictability [31].

11.8.2 Peak Heat Rate

The reconstructed flight heatshield stagnation point heat rate for InSight is shown in Figure 11-29, where time is measured from entry interface. The figure was generated by the author using a spline fit of reconstructed heat rate data from Beck, et al [41]. Because InSight had no ability to directly measure aerothermal heating, the heat rate shown in Figure 11-29 is a CFD-predicted flight heat rate based on the reconstructed trajectory and reconstructed atmosphere. As can be seen in the figure, the predicted heat rate is 49 W/cm^2, however because the reconstruction was conducted using CFD analysis, whereas the Metric Limit and Target Limit were derived from simulation data using Sutton-Graves [42] aerothermal modeling, the reconstructed peak heat rate must be adjusted for margin methodology purposes. Based on a mean difference of 3 W/cm^2 between CFD results and Sutton-Graves modeling, the reconstructed peak heat rate is adjusted down to 46 W/cm^2. For the Flight Reference Delta, the Metric Limit for Peak Heat Rate is 51.8 W/cm^2 and the Target Limit is 50.0 W/cm^2, so the InSight as-flown Peak Heat Rate has 5.8 W/cm^2 margin against the Metric Limit and 4.0 W/cm^2 margin against the Target Limit. The Target Margin quantification for Flight Reference Delta is 1.8 W/cm^2, so when comparing the actual flight margin against the Metric Limit, the flight has 3.2 times the minimum margin as determine by Target Margin quantification. The Metric Limit was set by an outlier simulation case landing on a steeper enter flight path angle during a regional dust storm, and it is against this case that the adequacy of the Target Margin quantification should be assessed, which is not the nominal landing condition of the actual InSight landing.

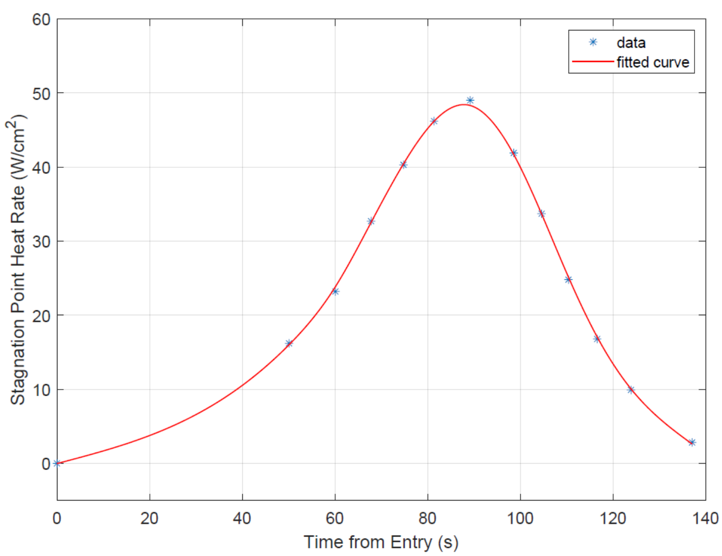


Figure 11-29 InSight reconstructed heatshield stagnation point heat rate.

The 99% performance for Peak Heat Rate from the Flight Reference Delta Monte Carlo is 48.2 W/cm^2 (see Table 11-55) so that the flight reconstruction Peak Heat Rate at 46 W/cm^2 is below the 99% performance and is approximately equivalent to the Monte Carlo 62nd percentile value, showing near nominal performance.

11.8.3 Total Heating

The flight total heating for the InSight landing is computed as the area under the heat rate curve in Figure 1-29. The integrated total is 2,765 J/cm^2. This is the predicted total heating seen at the stagnation point of the heatshield. It should be noted this value is calculated for the purposes of this dissertation and is not an official result of the InSight Project, which has not yet published an official result. Again, this value is CFD-based and therefore a conversion to an equivalent Sutton-Graves-based value is required. For this conversion, the value 2,765 J/cm^2 is the 2nd percentile value from the Monte Carlo simulation using a curve-fit CFD heating model. The corresponding Sutton-Grave 2nd percentile value is 2,590 J/cm^2, which is the value to be used for comparison to the Evolved Methodology. The Metric Limit for Total Heating for Flight Reference Delta is 3,200 J/cm^2, and therefore the flight margin against the Metric Limit is 610 J/cm^2. The Target Limit for Total Heating is 3,145 J/cm^2 so that actual flight margin against the Target Limit is 555 J/cm^2. For the Flight Reference Delta, the Target Margin is 55 J/cm^2, so that the actual flight margin against the Metric Limit is 11.1 times larger than the minimum margin determined by the Evolved Methodology. As stated previously, the flight total heating is equivalent to the 2nd percentile from the Monte Carlo, so flight represents a very non-stressing total heating case and does not lend itself to informing the adequacy of the quantified Target Margin for Total Heating. Given that a 2nd percentile event is also unlikely, like Peak Deceleration, a similar conclusion can be drawn that the Input Model Confidence Assessment for Total Heating may need to be lowered. For Total Heating, cm_hc_add is the most influential input, which is tied to the modeled vehicle angle of attack trim behavior, and so the same conclusion applies: cm_hc_add Confidence Rating should be reassessed. Note that for Peak Heat Rate, cm_hc_add is the fourth most influential input rather than the first, which may be why flight peak heat rate performance is closer to nominal, or the mean of Monte Carlo performance.

11.8.4 Parachute Inflation Load

An official reconstruction of the parachute inflation load has not been published in the public domain, but a preliminary reconstruction calculated the flight parachute inflation load to be 11,200 lbf. Against the Flight Reference Delta Metric Limit of 15,000 lbf, the actual flight Parachute Inflation Load had a margin of 3,800 lbf. Against the Target Limit of 13,393 lbf, the flight margin is 2,193 lbf. The Flight Reference Delta Target Margin is 1,607 lbf, so that the actual flight margin against the Metric Limit is 2.4 times the minimum margin resulting from the Evolved Methodology. The flight Parachute Inflation Load of 11,200 lbf is representative of the 40th percentile from the Flight Reference Delta Monte Carlo simulation, so again it is difficult to draw conclusions on Parachute Inflation Load Target Margin quantification based on the InSight flight reconstruction given the near-nominal performance.

As discussed earlier, there are limits to the insights to be gained about the Evolved Methodology based on the InSight flight data. Because the InSight landing was largely nominal with large margins against the Metric Limit, the comparison of the large as-flown margins to the Evolved Methodology Target Margin is limited to showing that the Target Margins are well under the InSight flight margins based on the InSight EDL flight reconstruction. To compare the sizing of Target Margin to real-world flight, a large number of sampled actual Mars landings would be required to validate the statistics-based quantified Target Margin. The InSight landing did reveal that the confidence in the modeling of cm_hc_add was likely rated too high in the Input Model Confidence Assessment process, given the flight result for Critical Metric Peak Deceleration and

Total Heating were both observed to be unlikely events, 99% event and 2% event respectively. Overall, the InSight landing did prove useful in revealing new insights into entry vehicle trim angle of attack behavior and how associated modeling confidence appears over estimated.

12 Sensitivity to Tool Configuration

The previous section provided the results of a baseline Target Margin assessment for the Critical Metrics for InSight for the three epochs ATLO 2.0, As-Built 2.0 and Flight Reference Delta. In this section the results of sensitivity assessments are presented, also for the InSight landing. The first sensitivity is a look at bounds of possible values of Target Margin given the configuration of the InSight baseline assessment. The second sensitivity is a simulated early lifecycle assessment of Target Margin to assess quantification of Target Margin during the early phase of a project. The final sensitivity is a look at how Assessment Category weighting might be applied in early and late phases of a project.

12.1 Assessing High and Low Target Margin Bounds

To understand the bounds of the Target Margin quantification, i.e., the largest and smallest Target Margin that can result for the InSight baseline assessment configuration, two additional Target Margin assessments were run for all four Critical Metrics. In the first, all Confidence Ratings for all Assessment Categories were set to High, to establish the lower bound on Target Margin quantification. In the second, all Confidence Ratings for all Assessment Categories were set to Low to establish the upper bound of Target Margin quantification. This was done for all Critical Metrics, the results of which are shown in Tables 12-1 – 12-4 and Figures 12-1 – 12-4.

The results for Peak Deceleration are shown in Table 12-1 and Figure 12-1. As can be seen in Table 12-1, the rows show Metric Limit, Target Margin, Target Limit and Percent Epistemic Contribution for each InSight epoch. The first column shows the Target Margin results for the InSight baseline assessment with all Confidence Ratings set to Low, to establish the largest possible Target Margin for the InSight baseline configuration. The middle data column has the

Target Margin result for the InSight Baseline (presented in the previous section), and the last data column has the Target Margin results with all Confidence Ratings set to High, to establish the smallest Target Margin quantification for the InSight baseline configuration. As can be seen in the table, the InSight Baseline falls between the All Low and All High Target Margins and is biased toward the All High Target Margin value for all epochs. This can be seen graphically in Figure 12-1, a plot of Target Margin for all epochs, where the orange data marks are the InSight Baseline case. The bias toward the All High Target Margin quantification can be explained by the fact all three epochs are drawn from the later phases of the InSight project where maturity of simulation modeling results in higher Confidence Ratings for Assessment Categories. The All Low and All High assessments for the Evolved Methodology for Peak Deceleration range from a minimum of 0.37 Earth *g* for All High, to a maximum of 0.79 Earth *g* for All Low. In terms of Monte Carlo performance standard deviation, this is a span of 1.42σ to 3.04σ.

Table 12-1 InSight baseline Target Margin bounding cases for Peak Deceleration.

Epoch	Peak Deceleration	Confidence Rating Configuration		
		InSight All Low	InSight Baseline	InSight All High
ATLO 2.0	Metric Limit (Earth g)	13.00	13.00	13.00
	Target Margin (Earth g)	**0.74**	**0.46**	**0.37**
	Target Limit (Earth g)	12.26	12.54	12.63
	Percent Epistemic Contribution (%)	**16.7**	**12.8**	**16.8**
As-Built 2.0	Metric Limit (Earth g)	13.00	13.00	13.00
	Target Margin (Earth g)	**0.74**	**0.46**	**0.37**
	Target Limit (Earth g)	12.26	12.54	12.63
	Percent Epistemic Contribution (%)	**16.7**	**12.8**	**16.9**
Flight Reference Delta	Metric Limit (Earth g)	13.00	13.00	13.00
	Target Margin (Earth g)	**0.79**	**0.49**	**0.39**
	Target Limit (Earth g)	12.21	12.51	12.61
	Percent Epistemic Contribution (%)	**17.6**	**12.5**	**16.0**

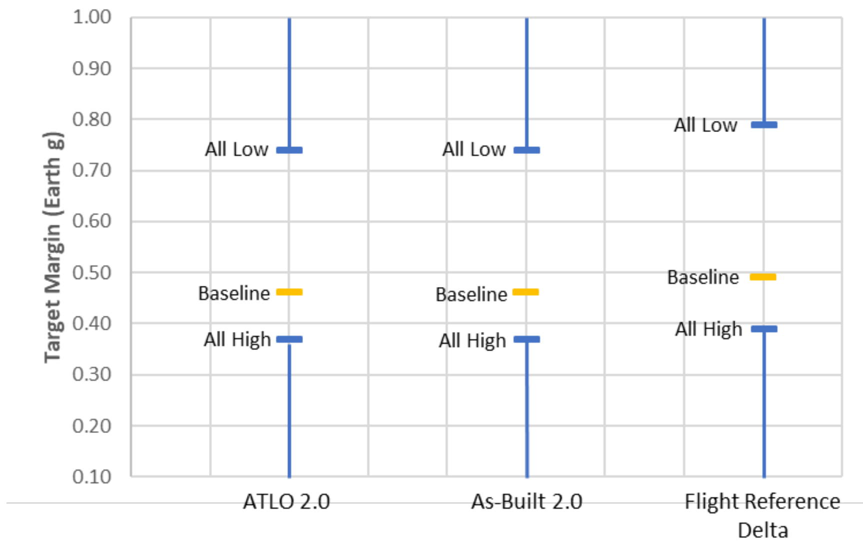


Figure 12-1 InSight Baseline Target Margin bounds - Peak Deceleration.

Interestingly, the InSight All High and InSight All Low configurations have the effect of increasing the separable epistemic uncertainty contribution to Target Margin relative to the InSight baseline, and this contribution for epistemic uncertainty for the two cases is nearly the same for all epochs. This is because when all inputs have the same Confidence Score (All High or All Low) , as with Peak Deceleration, and the nonlinear terms of the regression model make very small contribution to Target Margin, the separable epistemic uncertain contribution to Target Margin tends toward a value given by

$$(|b_1|EU_1 + |b_2|EU_2 + \cdots + |b_6|EU_6)/(|b_1| + |b_2| + \cdots + |b_6|) \qquad (12\text{-}1)$$

where EU_n is the fraction of epistemic uncertainty contained in the *nth* input variable (or *nth* linear term of the regression model), and $b_1 - b_6$ are the regression model linear term coefficients.

The results for Peak Heat Rate are shown in Table 12-2 and Figure 12-2. As with Peak Deceleration, the Target Margin value for Peak Heat Rate for the InSight baseline is biased toward the All High confidence Target Margin bound for all three InSight epochs. Again, this is to be expected given the maturity of the InSight project at the time of the three epochs. This is graphically represented in Figure 12-2, where the InSight Baseline Target Margin is indicated by the orange data marks. The Evolved Methodology analysis indicates that the possible Target Margin for the InSight baseline assessment configuration reaches a minimum of 1.4 W/cm^2 for All High, and a maximum of 3.0 W/cm^2 for All Low. In terms of Monte Carlo standard deviation, this is a span of 1.32σ to 2.83σ.

Table 12-2 InSight baseline Target Margin bounding cases for Peak Heat Rate.

Epoch	Peak Heat Rate	Confidence Rating Configuration		
		InSight All Low	InSight Baseline	InSight All High
ATLO 2.0	Metric Limit (W/cm^2)	61.5	61.5	61.5
	Target Margin (W/cm^2)	**3.0**	**1.8**	**1.4**
	Target Limit (W/cm^2)	58.5	59.7	60.1
	Percent Epistemic Contribution (%)	**11.0**	**8.9**	**11.5**
As-Built 2.0	Metric Limit (W/cm^2)	61.5	61.5	61.5
	Target Margin (W/cm^2)	**2.9**	**1.8**	**1.4**
	Target Limit (W/cm^2)	58.6	59.7	60.1
	Percent Epistemic Contribution (%)	**11.3**	**9.1**	**11.8**
Flight Reference Delta	Metric Limit (W/cm^2)	51.8	51.8	51.8
	Target Margin (W/cm^2)	**2.9**	**1.8**	**1.4**
	Target Limit (W/cm^2)	48.9	50.0	50.4
	Percent Epistemic Contribution (%)	**11.3**	**9.2**	**11.8**

As with Peak Deceleration, the percent contribution from separable epistemic uncertainty is higher for both the All Low and All High assessments than InSight Baseline. Because nonlinear terms of the Peak Heat Rate regression model make a bigger contribution to Target Margin, the separable epistemic uncertainty contribution for All High and All Low assessments differ more in value across the All Low, baseline and All High relative to the result for Peak Deceleration.

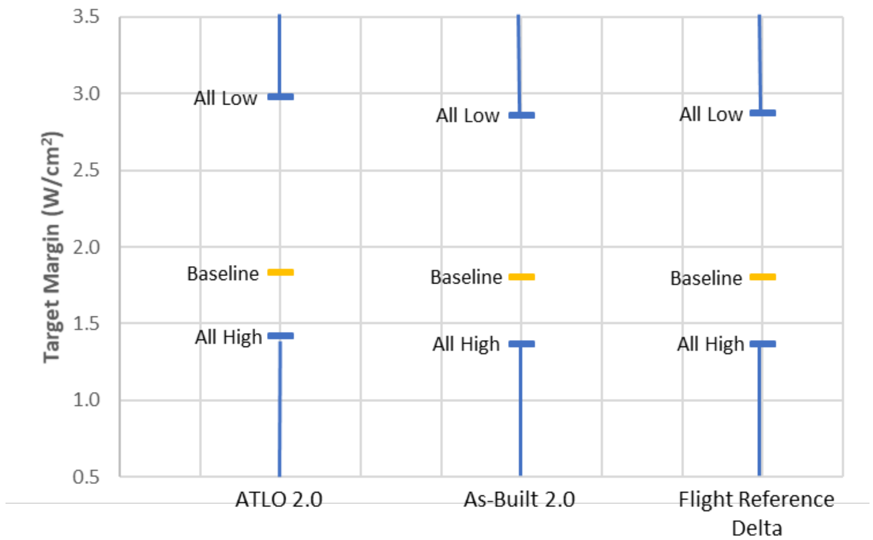


Figure 12-2 InSight Baseline Target Margin bounds - Peak Heat Rate.

The results for Total Heating are shown in Table 12-3 and Figure 12-3. Again, the Target Margin for the InSight baseline is biased toward the All High Target Margin boundary because of the maturity of the project at the time of the InSight epochs. This is graphically illustrated in

Table 12-3 InSight baseline Target Margin bounding cases for Total Heating.

Epoch	Total Heating	Confidence Rating Configuration		
		InSight All Low	InSight Baseline	InSight All High
ATLO 2.0	Metric Limit (J/cm^2)	3497	3497	3497
	Target Margin (J/cm^2)	**97**	**57**	**48**
	Target Limit (J/cm^2)	3400	3440	3449
	Percent Epistemic Contribution (%)	**19.4**	**16.6**	**19.6**
As-Built 2.0	Metric Limit (J/cm^2)	3497	3497	3497
	Target Margin (J/cm^2)	**94**	**55**	**46**
	Target Limit (J/cm^2)	3403	3442	3451
	Percent Epistemic Contribution (%)	**19.4**	**16.6**	**19.6**
Flight Reference Delta	Metric Limit (J/cm^2)	3200	3200	3200
	Target Margin (J/cm^2)	**94**	**55**	**47**
	Target Limit (J/cm^2)	3106	3145	3153
	Percent Epistemic Contribution (%)	**19.1**	**16.4**	**19.4**

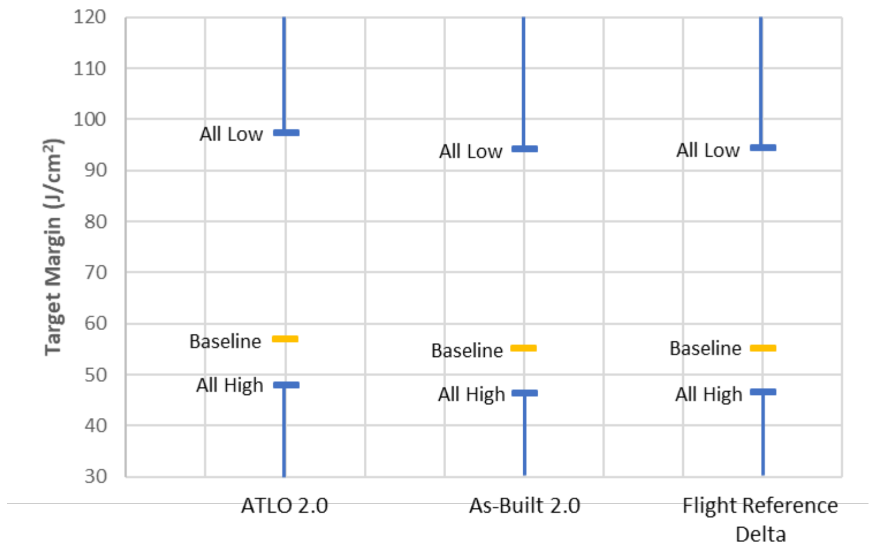


Figure 12-3 InSight Baseline Target Margin bounds – Total Heating.

Figure 12-3. The Total Heating Target Margin from the Evolved Methodology ranges from a minimum of 46 J/cm^2 for All High, to a maximum of 97 J/cm^2 for All Low, or in terms of Monte Carlo standard deviation, a span of about 0.92σ to 1.92σ. Because the regression model for Total Heating has small contribution from nonlinear terms, the separable epistemic uncertainty for All Low and All High is near the same and about 20% higher than InSight Baseline for each epoch.

Finally, the results for Parachute Inflation Load are shown in Table 12-4 and Figure 12-4. Based on the All High and All Low assessments, the minimum Target Margin for InSight as configured is 1,311 lbf, and the maximum is 2,871 lbf, across the three epochs. In terms of Monte Carlo standard deviation, the span is 1.10σ to 2.43σ The InSight Baseline is again biased toward the All High, or minimum Target Margin boundary for all epochs. As with the InSight Baseline, the All Low and All High assessments result in the largest Target Margin quantification variation across the epoch for Parachute Inflation Load, compared to the other Critical Metrics. Because all six input variables for Parachute Inflation Load are assessed as having no epistemic uncertainty, like the InSight Baseline, the All Low and All High assessments show no epistemic uncertainty contribution to Target Margin.

The All-High and All-Low sensitivity assessment provides a context within which to understand the results of the InSight baseline Target Margin assessment. By setting the Confidence Ratings for all model inputs in all Assessment Categories to High and to Low, the maximum and minimum possible Target Margin quantification bounds for the configured InSight baseline assessment are revealed. Because the assessed InSight epochs are all from the later stages of the InSight Project, for each the Evolved Methodology results in Target Margin quantification biased toward the minimum Target Margin boundary.

Table 12-4 InSight baseline Target Margin bounding cases for Parachute Inflation Load.

Epoch	Parachute Inflation Load	Confidence Rating Configuration		
		InSight All Low	InSight Baseline	InSight All High
ATLO 2.0	Metric Limit (lbf)	15000	15000	15000
	Target Margin (lbf)	**2871**	**1756**	**1428**
	Target Limit (lbf)	12129	13244	13572
	Percent Epistemic Contribution (%)	**0.0**	**0.0**	**0.0**
As-Built 2.0	Metric Limit (lbf)	15000	15000	15000
	Target Margin (lbf)	**2634**	**1615**	**1318**
	Target Limit (lbf)	12366	13385	13682
	Percent Epistemic Contribution (%)	**0.0**	**0.0**	**0.0**
Flight Reference Delta	Metric Limit (lbf)	15000	15000	15000
	Target Margin (lbf)	**2619**	**1607**	**1311**
	Target Limit (lbf)	12381	13393	13689
	Percent Epistemic Contribution (%)	**0.0**	**0.0**	**0.0**

(This space intentionally left blank)

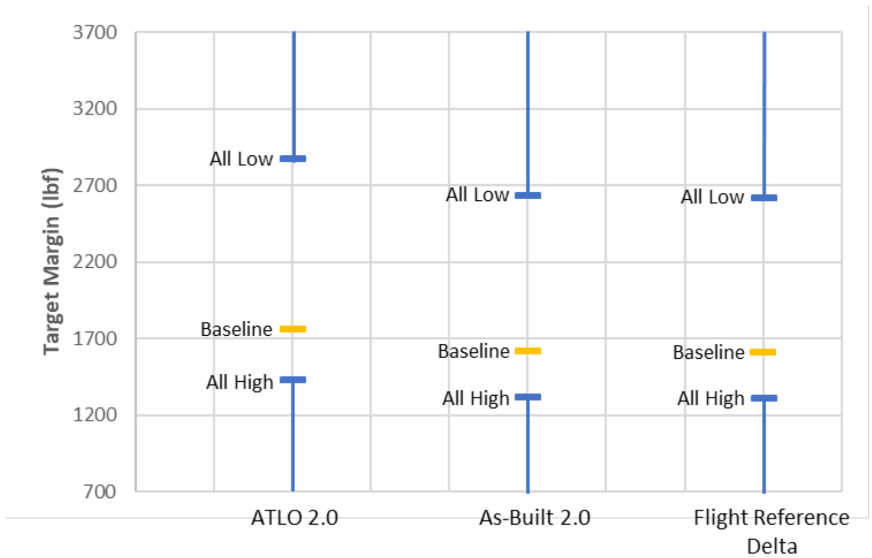


Figure 12-4 InSight Baseline Target Margin bounds – Parachute Inflation Load.

12.2 Simulated Early InSight Assessment

As discussed in Section 11.1, only complete data sets for the InSight epochs ATLO 2.0, As-Built 2.0 and Flight Reference Delta were available for application of the Evolved Methodology to InSight. However, in this section an assessment for an early project phase is simulated to understand or approximate Target Margin quantification that would result from applying the Evolved Methodology early in the lifecycle of a project.

Because the InSight mission was a near copy of NASA's Phoenix mission, the early phase of InSight is not a good representation of a typical project because much of the design of the spacecraft, as well as developed simulation tools, existed at the beginning of the project. Therefore, for the simulation of early InSight, early Phoenix is used to provide an assessment of early lifecycle Target Margin quantification. The early lifecycle simulated epoch is called Phoenix Phase B, where Phase B refers to the period in an early project lifecycle leading up to the Preliminary Design Review (PDR).

For the quantification of Target Margin for Phoenix Phase B, the assessment is configured with the same Assessment Category weightings used for the InSight baseline assessment (equal or even weighting, see Table 11-35), and the same Assessment Criteria are retained (see Table 11-36 and Table 11-37). Because regression models do not exist for Phoenix Phase B, the InSight ATLO 2.0 regression models are used because they are the earliest epoch regression models available. To provide Confidence Ratings for the inputs to the regression models, an Input Model Confidence Assessment is conducted based on the state of the entry, descent and landing (EDL) simulation during Phoenix Phase B. The Confidence Ratings with rationales are shown in Table 12-5 and Table 12-6, and the ratings are discussed in the following subsections.

12.2.1 Entry Vehicle Aerodynamic Parameters Confidence

For Phoenix Phase B, the entry vehicle aerodynamic parameters were developed in the same way as were aerodynamic parameters for InSight, via computational fluid dynamics (CFD) modeling. Therefore, the Assessment Categories Mathematical Model, Code and Validation receive the same ratings as those for InSight. However, during early Phase B, the Phoenix EDL simulations used the aerodynamic database from the Mars Exploration Rovers (MER) mission because of similarities in entry vehicle geometry and entry velocity. A Phoenix-specific aerodynamic database followed after new CFD analyses were conducted based on the Phoenix geometry and trajectory. Because the initial aerodynamic database was not specific to Phoenix, the Model Parameters category is given a Low rating.

12.2.2 Parachute Drag Confidence

During the early stages of Phoenix, the parachute drag was modeled via a drag-only model

Table 12-5 Model confidence assessment for Phoenix Phase B – part 1.

Input Name	Assessment Categories - InSight Initial Assessment							
	Mathematical Model		Code		Validation		Model Parameter	
	Rating	Rationale	Rating	Rationale	Rating	Rationale	Rating	Rationale
ca_hc_mult	High	Linearized aero database derived from CFD analysis.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	Medium	Initial version of aerodynamics database from Mars Exploration Rovers aerodynamics.
ca_mc_mult	Medium	Linearized aero database derived from CFD analysis and Viking base pressure correction. Greater uncertainty in supersonic CFD modeling.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	Medium	Initial version of aerodynamics database from Mars Exploration Rovers aerodynamics.
chute_aero_disp	Low	Drag-only parachute model.	High	Model is independently implemented across multiple simulations with behavior compared.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction, as well as based on Viking era high altitude tests.	Medium	Model parameters have low pedigree as start of project.
cm_hc_add	High	Linearized aero database derived from CFD analysis.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	Medium	Initial version of aerodynamics database from Mars Exploration Rovers aerodynamics.
cm_mc_add	Medium	Linearized aero database derived from CFD analysis and Viking base pressure correction. Greater uncertainty in supersonic CFD modeling.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	Medium	Initial version of aerodynamics database from Mars Exploration Rovers aerodynamics.
cn_hc_add	High	Linearized aero database derived from CFD analysis.	High	Aero database goes through thorough checkouts before delivery.	Medium	Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.	Medium	Initial version of aerodynamics database from Mars Exploration Rovers aerodynamics.

Table 12-6 Model confidence assessment for Phoenix Phase B – part 2.

Input Name	Assessment Categories - InSight Initial Assessment							
	Mathematical Model		Code		Validation		Model Parameter	
	Rating	Rationale	Rating	Rationale	Rating	Rationale	Rating	Rationale
density_sum_40km_20km	Medium	Parametrized atmosphere based on look-up tables generated by NASA Ames Mars General Circulation Model.	Medium	Unknown code review history of a NASA distributed code.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Low	Parameterized atmosphere may not result incompletely appropriate representation.
density_sum_60km_40km	Medium	Parametrized atmosphere based on look-up tables generated by NASA Ames Mars General Circulation Model.	Medium	Unknown code review history of a NASA distributed code.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Low	Parameterized atmosphere may not result incompletely appropriate representation.
density_sum_80km_60km	Medium	Parametrized atmosphere based on look-up tables generated by NASA Ames Mars General Circulation Model.	Medium	Unknown code review history of a NASA distributed code.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Low	Parameterized atmosphere may not result incompletely appropriate representation.
density_sum_above_100km	Medium	Parametrized atmosphere based on look-up tables generated by NASA Ames Mars General Circulation Model.	Medium	Unknown code review history of a NASA distributed code.	Medium	Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.	Low	Parameterized atmosphere may not result incompletely appropriate representation.
EFPA	High	Mathematical modeling utilizes high fidelity orbital mechanics and statistical state estimation while tracking and incorporating a comprehensive set of factors.	High	Orbit determination tools shared by multiple missions with results frequently peer reviewed.	High	Orbit determination techniques utilized in modeling entry flight path angle are utilized by all planetary missions and thus providing numerous validation opportunities.	High	Parameters have high pedigree, completeness and relevance for the application.
zcg_ei	High	Mathematical modeling rooted in simple high-fidelity physics principles.	Low	Unknown code review history.	Low	Unknown validation history.	Low	The pedigree of parameters in early modeling of center of gravity location is low given the state of the vehicle design.

based on Viking-era high altitude tests. Because the model was low fidelity, it is given a Low rating for Mathematical Model. The drag-only model was implemented in multiple simulations and compared, therefore the Code category is rated High. Because of limited opportunity for validation via Mars flights, and because Phoenix Phase B was validated by Viking-era high altitude tests, the Assessment Category receives a Medium rating. Finally, the category Model Parameters receives a Medium rating because of lower pedigree at the start of the project.

12.2.3 Atmosphere Density Parameters Confidence

As with many projects, Phoenix utilized the engineering atmosphere modeling code called Mars Global Reference Atmosphere Model (Mars-GRAM) [43] early in the project lifecycle. At the time of Phoenix Phase B, Mars-GRAM was a parametrized atmosphere model based on look-up tables generated by the NASA Ames Mars General Circulation Model. Because the model is look-up table based, it is given a Medium rating for the category Mathematical Model. The Mars-GRAM code itself has an unknown level of review, but because it is a NASA distributed program, it is assumed it is reviewed. For the category Code, it therefore receives a Medium rating. Like the InSight atmosphere model, validation of the model is by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics and therefore the category Validation receives a Medium rating. For the category Model Parameters, Mars-GRAM receives a low rating because it was a generalized model that did not appropriately capture the structure of the atmosphere as validation efforts revealed a 30% to 45% difference in density near 40 km altitude when compared to the European Mars Climate Database (MCD) atmosphere model at the Phoenix landing zone and arrival time [43].

12.2.4 Entry Flight Path Angle Confidence

Of the twelve model inputs being assessed for Phoenix Phase B, entry flight path angle (EFPA) receives the same confidence ratings as InSight, for the same reasons as summarized in Section 11.5.2.4.

12.2.5 Entry Vehicle Z-Axis Center of Gravity Location at Entry Interface Confidence

As was the case for the InSight epochs, for Phoenix Phase B, the Mathematical Model rating is High for the entry vehicle center of mass model (zcg_ei) because the mathematics is simple and well understood. Because the state of review history of the model code and the validation history are not easily recovered for this assessment, it is assumed tools have a low level of maturity at the beginning of Phoenix, so both Code and Validation receive a Low rating. Also, the category Model Parameter is given a Low rating because the pedigree of the parameters is low at the beginning of a project when there is not significant measurement history or analysis history behind the parameters.

12.2.6 Results for Simulated Early InSight Assessment

Using the ATLO 2.0 regression model, even Assessment Category weightings, the InSight baseline Assessment Criteria, and the Model Confidence Ratings covered in the previous subsections, a Phoenix Phase B Target Margin quantification assessment was conducted. The results for the four InSight Critical Metrics are shown in Tables 12-7 – 12-10 and Figures 12-5 – 12-8.

Table 12-7 has a summary of the Peak Deceleration Target Margin for epoch Phoenix Phase B. Also shown for comparison are InSight All Low, Flight Reference Delta, and InSight All High. As the table shows, the simulated Target Margin assessment resulted in a Target Margin

Table 12-7 Phoenix Phase B Target Margin result for Peak Deceleration.

Peak Deceleration	Confidence Rating Configuration			
	InSight All Low	Phoenix Phase B	Flight Ref Delta	InSight All High
Metric Limit (Earth g)	13.00	13.00	13.00	13.00
Target Margin (Earth g)	**0.79**	**0.53**	**0.49**	**0.39**
Target Limit (Earth g)	12.21	12.47	12.51	12.61
Percent Epistemic Contribution (%)	**17.6**	**19.0**	**12.5**	**16.0**

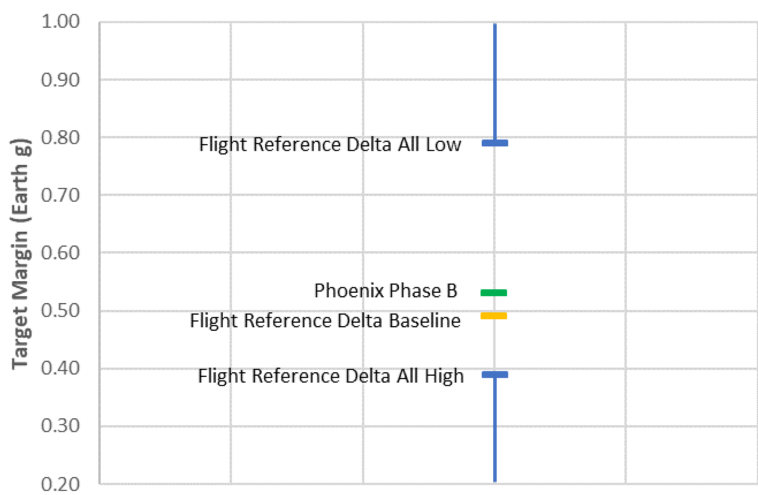


Figure 12-5 Phoenix Phase B Target Margin relative to Flight Reference Delta – Peak Deceleration.

quantification of 0.53 Earth *g*, which is 8% larger than the 0.49 Earth *g* result for the Flight Reference Delta. This suggests the Target Margin during Phase B of a project should be about 8% higher than the Target Margin at the time of the Flight Reference Delta after launch, specific to the InSight spacecraft. As can be seen in Figure 12-5, Phoenix Phase B Target Margin, indicated by the green data mark, falls near the midpoint between the All Low and All High Target Margin boundaries. This also shows for epoch Phoenix Phase B, the EDL system requires 67% of the maximum Target Margin of 0.79 Earth *g* the Evolved Methodology indicates for All Low.

Table 12-8 contains the results for Peak Heat Rate. For epoch Phoenix Phase B, the quantified Target Margin for Peak Heat Rate is 2.1 W/cm^2, 0.3 W/cm^2 larger than Target Margin for Flight Reference Delta at 1.8 W/cm^2. These results suggest, for the InSight spacecraft, that the

Table 12-8 Phoenix Phase B Target Margin result for Peak Heat Rate.

Peak Heat Rate	Confidence Rating Configuration			
	InSight All Low	Phoenix Phase B	Flight Ref Delta	InSight All High
Metric Limit (W/cm^2)	51.8	51.8	51.8	51.8
Target Margin (W/cm^2)	**2.9**	**2.1**	**1.8**	**1.4**
Target Limit (W/cm^2)	48.9	49.7	50.0	50.4
Percent Epistemic Contribution (%)	**11.3**	**7.7**	**9.2**	**11.8**

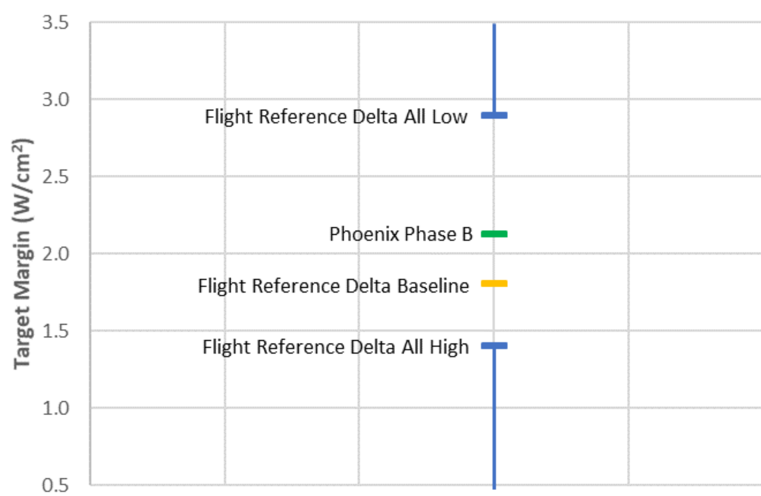


Figure 12-6 Phoenix Phase B Target Margin relative to Flight Reference Delta – Peak Heat Rate.

Peak Heat Rate Target Margin should be 17% higher during Phase B than at the time of the Flight Reference Delta. As can be seen in Figure 12-6, Phoenix Phase B Target Margin, indicated by the green data mark, falls approximately midway between the All Low and All High Flight Reference

Delta Target Margin boundaries, which shows for epoch Phoenix Phase B, the EDL system requires 72% of the maximum Target Margin of 2.9 W/cm^2 the Evolved Methodology indicates for All Low.

Table 12-9 Phoenix Phase B Target Margin result for Total Heating.

Total Heating	Confidence Rating Configuration			
	InSight All Low	**Phoenix Phase B**	**Flight Ref Delta**	**InSight All High**
Metric Limit (J/cm^2)	3200	3200	3200	3200
Target Margin (J/cm^2)	**94**	**63**	**55**	**47**
Target Limit (J/cm^2)	3106	3137	3145	3153
Percent Epistemic Contribution (%)	**19.1**	**14.9**	**16.4**	**19.4**

Table 12-9 contains the results for Total Heating. The Phoenix Phase B Target Margin quantification is 63 J/cm^2, which is 8 J/cm^2 higher than Flight Reference Delta at 55 J/cm^2. This result suggests Target Margin for Total Heating at Phase B should be 15% higher than at the Flight Reference Delta. This can be graphically seen in Figure 12-7, where Phoenix Phase B is indicated by the green data mark. Target Margin for epoch Phoenix Phase B Total Heating is biased

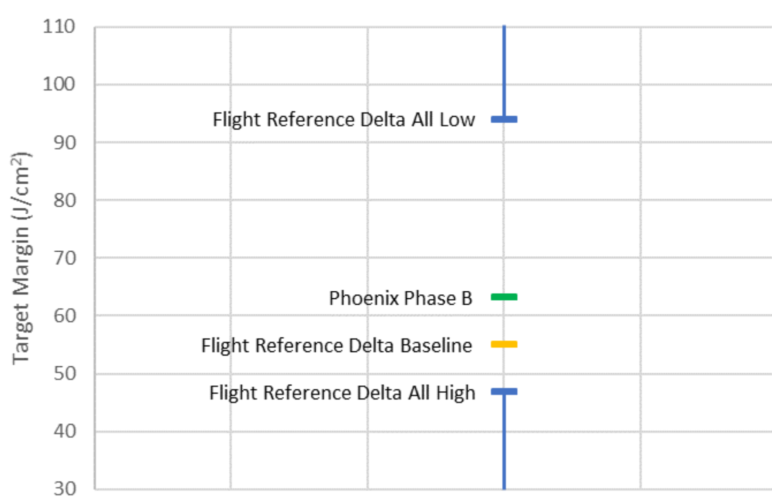


Figure 12-7 Phoenix Phase B Target Margin relative to Flight Reference Delta – Total Heating.

somewhat toward the All High boundary, and short of the midpoint between All Low and All High. The result shows for epoch Phoenix Phase B, the EDL system requires 67% of the maximum Target Margin of 94 J/cm^2 the Evolved Methodology indicates for All Low.

Table 12-10 contains the results for Parachute Inflation Load. The quantified Target Margin for epoch Phoenix Phase B Parachute Inflation Load is 2,023 lbf, or 416 lbf greater than the Target Margin value for Flight Reference Delta of 1,607 lbf. The results suggest Parachute Inflation Load Target Margin for the InSight spacecraft for Phase B should be 26% greater than at the time of the Flight Reference Delta, and is 77% of the maximum Target Margin of 2,619 lbf the Evolved Methodology indicates for All Low. Figure 12-8 provides of graphical representation of the Target Margin data in Table 12-10.

Table 12-10 Phoenix Phase B Target Margin result for Parachute Inflation Load.

Parachute Inflation Load	Confidence Rating Configuration			
	InSight All Low	Phoenix Phase B	Flight Ref Delta	InSight All High
Metric Limit (lbf)	15000	15000	15000	15000
Target Margin (lbf)	**2619**	**2023**	**1607**	**1311**
Target Limit (lbf)	12381	12977	13393	13689
Percent Epistemic Contribution (%)	**0.0**	**0.0**	**0.0**	**0.0**

Utilizing Phoenix Phase B as a simulated early lifecycle epoch provides insight into quantification of Target Margin early in a project when model maturity, vehicle design and level of model validation may be at a low level. By calling upon historical knowledge of the state of Phoenix EDL simulation during Phase B of the project lifecycle, the Evolved Methodology specifies Peak Deceleration Target Margin 8% larger in Phase B than at the time of the Flight Reference Delta; Peak Heat Rate Target Margin 17% higher in Phase B than at the time of Flight

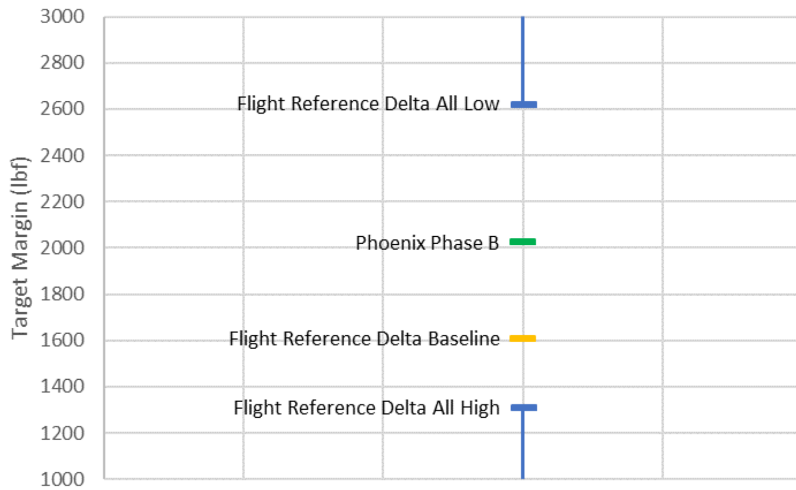


Figure 12-8 Phoenix Phase B Target Margin relative to Flight Reference Delta – Parachute Inf. Load.

Reference Delta; Total Heating Target Margin 15% higher in Phase B than at the time of Flight Reference Delta; and Parachute Inflation Target Margin 26% larger in Phase B than in at the time of Flight Reference Delta. As part of future work, to the extent possible, comparing Phoenix Phase B Target Margin to actual performance margin held by the Phoenix Project during Phase B might provide additional insight and validation of Evolved Methodology Target Margin quantification.

12.3 Application of Assessment Category Weighting

The final sensitivity analysis is an assessment of sensitivity to Assessment Category weighting. The assessment is crafted to simulate how Assessment Category weighting might be applied early in a project lifecycle and late in a project lifecycle. Building off the results of the previous subsection, Phoenix Phase B is chosen to represent an early epoch, and Flight Reference Delta is chosen as a late epoch.

The challenge to crafting an Assessment Category weighting scheme is in choosing what categories should be weighted of higher importance and which of lower importance. In applying a weighting scheme other than even, it becomes clear that attempting to rate, for example, Mathematical Model higher than Validation, is fraught with analytical danger. It is difficult if not impossible to judge how one Assessment Category might be more important than another. An error in model coding, for example, may have an equally detrimental impact to accuracy of simulated performance as a weakness in model validation. As a result, it is difficult to decide or judge which needs more Target Margin, and thus higher weighting, as a result. Understanding that designing a weighting scheme other than even should be treated with great care and perhaps avoided, the following is one approach to applying a weighting scheme.

To perform the weighting sensitivity, a simple, early weighting scheme was designed based on a rationale that emphasizes the immaturity of simulation modeling early in a project. Because simulation models are often of lower fidelity early in a project, Mathematical Model is given a weighting of 2.00 to emphasize that Target Margin size may need to particularly reflect model immaturity. The Assessment Category Validation is also weighted 2.00 because early in a project model validation may be future work, so emphasis on validation protects against expected validation efforts possibly falling short. Assessment Categories Code and Model Parameter retain their baseline weighting of 1.00. This early weighting scheme is summarized in Table 12-11.

A weighting scheme for a late project lifecycle Target Margin assessment is summarized in Table 12-12. The rationale for the weighting choices for a late weighting scheme emphasizes simulation tool maturity which is important for operations once the spacecraft is in flight. For this reason, Code is given higher weighting at 2.00. Likewise, Model Parameter is given a weighting

Table 12-11 A weighting scheme for early project assessment.

Early Weighting - Phoenix Phase B		
Assessment Category	Weighting	Rationale
Mathematical Model	2.00	Additional weighting to emphasize margin needed because of modeling immaturity
Code	1.00	Baseline weighting
Validation	2.00	Additional weighting to emphasize margin needed to protect against expected validation efforts falling short
Model Parameter	1.00	Baseline weighting

Table 12-12 A weighting scheme for late project assessment.

Late weighting - Flight Reference Delta		
Assessment Category	Weighting	Rationale
Mathematical Model	1.00	Baseline weighting
Code	2.00	Additional weighting to emphasize tool accuracy and maturity in preparation for operations.
Validation	1.00	Baseline weighting
Model Parameter	2.00	Additional weighting to emphasize tool accuracy and maturity in preparation for operations.

of 2.00 for the same reason that tool accuracy and reliability are important in operations. The categories Mathematical Model and Validation are given a baseline rating of 1.00 because late in a project it is assumed model development and validation efforts are complete.

Using these Assessment Category weighting schemes, Phoenix Phase B and Flight Reference Baseline were run again using the Target Margin Sizing Tool (TMST), with the results of the weighted assessment summarized in Tables 12-13 – 12-16. Each table provides the Target Margin results of early weighting and late weighting assessments, along with the results for Phoenix Phase B and Flight Reference Delta with the even (equal) weighting that was used for the InSight baseline assessment in Section 11.4.1. As can be seen in Table 12-13, for Critical Metric Peak Deceleration, both Phoenix Phase B and Flight Reference Delta show little to no sensitivity to the early and late weighting schemes, as the results are nearly identical or identical to even weighting Target Margin results. Critical Metrics Peak Heat Rate and Total Heating show essentially no sensitivity to the designed early and late weighting schemes, as shown in Table 12-14 and Table 12-15. Finally, Critical Metric Parachute Inflation Load does show some sensitivity to the early and late weighting schemes, as summarized in Table 12-16. The early weighting scheme results in an increase in Phoenix Phase B Target Margin from 2,023 lbf for even weighting, to 2,095 lbf for early weighting, or a about a 4% increase in Target Margin when emphasizing Mathematical Model and Validation. This is because categories Mathematical Model and Validation have on average lower confidence ratings for the inputs of highest influence relative to the lower weighted categories Code and Model Parameter. For Parachute Inflation Load, the late weighting scheme results in a lowering of Flight Reference Delta Target Margin from 1,607 lbf for even weighting to 1,513 lbf for late weighting, or about a 6% decrease. This is because the categories Code and Model Parameter have, on average, higher confidence ratings for inputs of strongest influence relative to lower weighted categories Mathematical Model and Validation. Therefore, the late weighting scheme suggests less Parachute Inflation Load Target Margin is needed relative to an even weighting scheme for InSight late in the project lifecycle. Likewise, the early weighting scheme suggests for InSight more Target Margin is needed for Parachute Inflation Load early in the lifecycle of the project.

Table 12-13 Peak Deceleration Target Margin sensitivity to early and late weighting.

Peak Deceleration	Confidence Rating Configuration			
	Phoenix Phase B		Flight Refernece Delta	
	Even Weighting	Early Weighting	Even Weighting	Late Weighting
Metric Limit (Earth g)	13.00	13.00	13.00	13.00
Target Margin (Earth g)	**0.53**	**0.52**	**0.49**	**0.49**
Target Limit (Earth g)	12.47	12.48	12.51	12.51
Percent Epistemic Contribution (%)	**19.0**	**18.8**	**12.5**	**19.5**

Table 12-14 Peak Heat Rate Target Margin sensitivity to early and late weighting.

Peak Heat Rate	Confidence Rating Configuration			
	Phoenix Phase B		Flight Refernece Delta	
	Even Weighting	Early Weighting	Even Weighting	Late Weighting
Metric Limit (W/cm^2)	51.8	51.8	51.8	51.8
Target Margin (W/cm^2)	**2.1**	**2.1**	**1.8**	**1.8**
Target Limit (W/cm^2)	49.7	49.7	50.0	50.0
Percent Epistemic Contribution (%)	**7.7**	**7.8**	**9.2**	**9.0**

Table 12-15 Total Heating Target Margin sensitivity to early and late weighting.

Total Heating	Confidence Rating Configuration			
	Phoenix Phase B		Flight Refernece Delta	
	Even Weighting	Early Weighting	Even Weighting	Late Weighting
Metric Limit (J/cm^2)	3200	3200	3200	3200
Target Margin (J/cm^2)	**63**	**63**	**55**	**55**
Target Limit (J/cm^2)	3137	3137	3145	3145
Percent Epistemic Contribution (%)	**14.9**	**15.0**	**16.4**	**16.5**

Table 12-16 Parachute Inflation Load Target Margin sensitivity to early and late weighting.

Parachute Inflation Load	Confidence Rating Configuration			
	Phoenix Phase B		Flight Refernece Delta	
	Even Weighting	Early Weighting	Even Weighting	Late Weighting
Metric Limit (lbf)	15000	15000	15000	15000
Target Margin (lbf)	**2023**	**2095**	**1607**	**1513**
Target Limit (lbf)	12977	12905	13393	13487
Percent Epistemic Contribution (%)	**0.0**	**0.0**	**0.0**	**0.0**

The level of sensitivity to Assessment Category weighting is in part related to the Confidence Rating levels for the model inputs in the category, as was illustrated by the effect that weighting has on Parachute Inflation Load Target Margin. Parametrically varying weighting levels is another study approach that would be valuable in understanding the full sensitivity of Target Margin quantification to the design of Assessment Category weighting.

13 Conclusions

The research in this dissertation develops and implements a methodology for the determination of entry, descent and landing design performance margins for spacecraft landing on planetary bodies. The methodology, termed the Evolved Methodology, is seeded by a Seed Methodology employed by NASA's InSight Project. The methodology utilizes Monte Carlo simulation, Monte Carlo Filtering, and Design of Experiments methods to develop second-order regression models that are utilized in the quantification of desired performance margin, called Target Margin. The research resulted in the successful development of the methodology and in this conclusion section the level to which the original research objectives have been met is assessed, a summary of research finding is presented and areas for future work are summarized.

13.1 Assessment of Research Objectives

Early in the research presented in this dissertation, four research objectives were identified that guided the direction of the research. These objectives were derived from identified areas needing improvement in the Seed Methodology. The following subsections assess the degree to which Evolved Methodology addresses the research objectives.

13.1.1 Objective 1: Input Model Influence

Objective: Develop a more rigorous method for identifying simulation models that most influence performance distributions of Critical Metrics and quantify their level of influence.

The Seed Methodology uses experienced-based engineering judgement to identify models with highest influence on Critical Metric response, and does not quantify their level of influence. The Evolved Methodology uses Monte Carlo Filtering as a more rigorous process of identifying models of highest influence, and then uses a three-level full factorial design of experiments (DOE) analysis, resulting in a second-order regression model, the coefficients of which provide a quantification of input influence on Critical Metric response. The Evolved Methodology completely addresses Objective 1.

13.1.2 Objective 2: Mapping Input Model Confidence to Target Margin

Objective: Develop a more rigorous, higher resolution technique to map input model confidence to Target Margin sizing via metric performance distribution.

The Seed Methodology sizes Target Margin as a fraction of Critical Metric Monte Carlo performance distribution, with the size of fraction based on model confidence assessed using engineering judgement. The Evolved Methodology uses instead fractions of input distributions that become inputs into the DOE regression model to size Target Margin. By applying the process of using distribution fraction to size Target Margin, higher resolution is achieved because the distribution fraction process is applied to each input variable based on a methodical assessment of modeling confidence under multiple assessment categories. The Seed Methodology applies a single confidence assessment to size fraction for a single Critical Metric performance distribution. By evolving the methodology to applying distribution fraction to modeling inputs, the Evolved Methodology provides a higher resolution technique to sizing Target Margin. The Evolved Methodology meets Objective 2.

13.1.3 Objective 3: Technique for Epistemic and Aleatory Uncertainty

Objective: Develop a method to handle epistemic and aleatory uncertainty appropriately within the methodology.

The Seed Methodology had no process for addressing epistemic and aleatory uncertainty, which is uncertainty that is reducible and irreducible, respectively. The importance of identifying and understanding epistemic and aleatory uncertainty is found in understanding where uncertainty may be reduced during the lifecycle of a project. The Evolved Methodology provides a means for estimating epistemic and aleatory uncertainty, and quantifying how epistemic and aleatory uncertainty contribute to Target Margin. The Evolved Methodology meets Objective 3.

13.1.4 Objective 4: Evolvable Methodology

Objective: Develop an evolvable methodology that can be applied throughout the lifecycle of a project.

The Evolved Methodology addresses Objective 4 in a number of ways. First, the methodology provides the ability to apply the methodology at selected epochs across a project lifecycle. It also provides a number of ways to tailor the configuration of a Target Margin assessment depending on system engineering needs. Tailoring of Target Margin assessment configuration is provided by the ability to set Assessment Criteria, choose Assessment Categories and category weightings, and change Model Confidence ratings as modeling fidelity and validation evolves. Through these abilities, the Evolved Methodology allows the Target Margin assessments to evolve across project lifecycle, and meets Objective 4.

13.2 Findings & Areas for Future Work

The following is a list of findings and future work areas that were identified in the development and application of the Evolved Methodology.

- In the Monte Carlo Filtering process, a stepwise sum of density represents an initial technique for parametrizing an atmosphere density profile. Future development of the methodology should explore other methods of parameterizing a density profile.

- The process of using Monte Carlo Filtering to identify Model Inputs of Highest Influence did not capture the Sutton-Graves aerothermal heating model for Critical Metrics Peak Heat Rate and Total Heating. It was not identified because Sutton-Graves model parameters are not dispersed. Future development of the Evolved Methodology should address this observation.
- The process of doing an initial identification of Model Inputs of Highest Influence via Monte Carlo Filtering to feed the DOE analysis may not provide a fully comprehensive search. Future development should include exploration of DOE search methods to potentially provide a more exhaustive search of input parameters for parameters of highest influence.
- The Evolved Methodology supports normal and uniform input distribution types. Further development of the methodology needs to expand to accommodate a wider range of input distributions.
- The estimation of epistemic and aleatory uncertainty in the Evolved Methodology represents an initial treatment, and benefited from hindsight. Future work needs to evolve the treatment of epistemic and aleatory uncertainty and further develop formal methods of estimating epistemic and aleatory uncertainty content in Model Inputs of Highest Influence.
- When executing the Input Model Confidence Assessment, it is important to focus the Confidence Rating only on factors that are not already captured in modeled uncertainty in simulation, thus avoiding double accounting of sources of uncertainty. This was adhered to while assessing Input Modeling Confidence.

- Compared to the Seed Methodology, the Evolved Methodology provides the ability to quantify how much individual model inputs are contributing to total Target Margin, and results in a more conservative total Target Margin than the Seed Methodology. This has the potential to result in a more robust EDL system.
- Because this dissertation implemented a single set of Assessment Criteria for normal and uniform distributions, a more extensive exploration of sensitivity to Assessment Criteria sizing should be part of future work.
- As future work, to the extent possible, comparing actual margins held by Phoenix in Phase B to results of Evolved Methodology assessment for Phoenix Phase B might provide useful insight into Evolved Methodology Target Margin quantification.
- A broader parametric weighting assessment would provide a clearer understanding of Evolved Methodology sensitivity to weightings, and is an area of future work.
- The research and development performed in this dissertation was focused on applying the Evolved Methodology to Critical Metrics whose Metric Limit is on the high side of its performance distribution. Future work includes application to Critical Metrics with low side and two-sided Metric Limits.
- Applying the Evolved Methodology to more complex EDL systems is future work.
- A future area of extended research is the potential to use the results of the Evolved Methodology to develop a weighted process to convolve or combine Target Margin per Critical Metric into an overall Integrated Total Margin, for use in guiding conceptual design and balancing risk.

13.3 Summary

A methodology for determination of entry, descent, and landing performance Target Margin was developed, evolved from a Seed Methodology employed by NASA's InSight Project. Termed the Evolved Methodology, it successfully addresses shortcomings of the Seed Methodology, identified as objectives of the presented research. The methodology was applied to four Critical Metrics across three analysis epochs for the NASA InSight Mars landing and was successfully compared to the Seed Methodology, showing the Evolved Methodology calls for larger, potentially more robust Target Margin, as compared to the Seed Methodology, for all four Critical Metrics. Sensitivity analyses using the Evolved Methodology established the minimum and maximum bounding values for InSight EDL system Target Margin, and examined Target Margin for a simulated early InSight epoch to compare application of the Evolved Methodology to early and late lifecycle epochs. In addition, sensitivity analysis examined hypothetical Assessment Category weightings for early lifecycle and late lifecycle assessments, and found low sensitivity to the chosen hypothetical weightings. Through successful application to the InSight Project, the Evolved Methodology is assessed as a viable method for quantifying critical design performance margins for the InSight EDL system, and future work includes addressing the findings of this research and expanding application of the methodology to landings on other planetary bodies.

References

[1] Spencer, D. A., et al, "Mars Pathfinder Entry, Descent, and Landing Reconstruction," *Journal of Spacecraft and Rockets,* vol. 36, no. 3, 1999, pp. 357-366.

[2] Smrekar, S., et al, "Deep Space 2: The Mars Microprobe Mission," *Journal of Geophysical Research*, vol. 104, no. E11, 1999, pp. 27,013-27,030.

[3] Grover, M. R., et al, "Overview of the Phoenix Entry, Descent and Landing System Architecture," AIAA 2008-7218, Honolulu, Hawaii, 2008.

[4] Steltzner, A., D., et al, "Mars Science Laboratory Entry, Descent, and Landing Systems Development Challenges," *Journal of Spacecraft and Rockets*, vol. 51, no. 4, 2014, pp. 994-1003.

[5] Grover, M. R., et al, "InSight EDL Overview and As-Flown Performance,", 16th International Planetary Probe Workshop, Mars Exploration Session, Oxford, UK, 2019.

[6] Metropolis, N., "The Beginning of the Monte Carlo Method," Los Alamos Science Special Issue Number 15,1987, pp. 125-130.

[7] Maddock, R. W., Dwyer-Cianciolo, A. M., Litton, D. K., and Zumwalt, C. H. "InSight Entry, Descent and Landing Pre-Flight Performance Predictions," AIAA 2020-1273, Orlando, FL, 2019.

[8] Desai, P., N. and Knocke, P., C., "Mars Exploration Rovers Entry, Descent, and Landing Trajectory Analysis," AIAA 2004-5092, Providence, Rhode Island, 2004.

[9] Prince, J. L., et al, "Mars Phoenix Entry, Descent, and Landing Simulation Design and Modeling Analysis," AIAA 2008-7507, Honolulu, Hawaii, 2008.

[10] Desai, P. N., et al, "Entry, Descent, and Landing Operations Analysis for the Stardust Entry Capsule," *Journal of Spacecraft and Rockets,* vol. 45, no. 6, 2008, pp. 1262-1268.

[11] Striepe, S. A., et al, "Mars Science Laboratory Simulations for Entry, Descent, and Landing," *Journal of Spacecraft and Rockets,* vol. 43, no. 2, 2006, pp. 311-323.

[12] Mehta, U. B., et al, "Simualtion Credibility: Advances in Verification, Validation, and Uncertainty Quantification," NASA/TP-2016-219422, Moffett Field, California, 2016.

[13] Noyes, C., et al, "Sensitivity Analysis and Uncertainty Quantification of a Mars Ascent Vehicle Concept," 2017 IEEE Aerospace Conference, Big Sky, Montana, 2017.

[14] Dutta, S., et al, "Uncertainty Quantification for Mars Entry, Descent and Landing Reconstruction Using Adaptive Filtering," AIAA 2013-0026, Grapevine, Texas, 2013.

[15] Olds, J. R. and Way, D., "Uncertainty Optimization Applied to the Monte Carlo Analysis of Planetary Entry Trajectories," NGT-1-52163, NASA Langley Research Center, Atlanta, GA, 2001.

[16] Helton, J. C., "Conceptual and Computational Basis for the Quantification of Margins and Uncertainty," Sandia Report SAND2009-3055, Sandia National Laboratories, Albuquerque, NM, 2009.

[17] Pilch, M., Trucano, T. G. and Helton, J. C., "Ideas Underlying Quantification of Margins and Uncertainties (QMU): A White Paper," Sandia Report SAND2006-5001, Sandia National Laboratories, Albuquerque, NM, 2006.

[18] "Evaluation of Quantification of Margins and Uncertainties Methodology for Assessing and Certifying the Reliability of the Nuclear Stockpile," National Research Council, The National Academies Press, Washington, DC, 2009. https://doi.org/10.17226/12531.

[19] Eardley, D., et al, "Quantification of Margins and Uncertainties (QMU)," Report Number JSR-04-330, JASON Program Office, The MITRE Corporation, McLean, Virginia, 2005.

[20] Grover, M. R., "InSight Entry, Descent and Landing Margins Policy – Initial Release," Jet Propulsion Laboratory Doc. No. D-81371, Pasadena, CA, 2013.

[21] The SciPy Community, "numpy.corrcoef," The SciPy Community, 31 January 2021, [https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html. Accessed 10 May 2021].

[22] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, NY, 1991, pg. 152.

[23] Kraskov, A., Stogbaur, H. and Gassberger, P., "Estimating Mutual Information," *Physical Review*, Vol. 69, Issue 6 2004, pg. 066138-1.

[24] Rekab, K., and Shaikh, M., *Statistical Design of Experiments with Engineering Applications*, CRC Press, Boca Raton, FL: 2005, pp. 193-202.

[25] Frost, J., "How to Interpret Regression Analysis Results: P-values and Coefficients," The Minitab Blog, 01 July 2013. [https://blog.minitab.com/blog/adventures-in-statistics-2/how-to-interpret-regression-analysis-results-p-values-and-coefficients, Accessed 08 09 2020].

[26] Sandia National Laboratories, "What is PIRT," Sandia Report SAND2016-6466, Sandia National Laboratories, Albuquerque, NM, 2016.

[27] Sandia National Laboratories, "How to PIRT," Sandia Report SAND2016-6465, Sandia National Laboratories, Albuquerque, NM, 2016.

[28] Vastag, B., "NASA will send robot drill to Mars in 2016," The Washington Post, 20 August 2012, [https://www.washingtonpost.com/national/health-science/nasa-will-send-robot-drill-to-mars-in-2016/2012/08/20/43bf1980-eaef-11e1-9ddc-340d5efb1e9c_story.html, Accessed 8 April 2021].

[29] NASA Langley Research Center, "Program to Optimize Simulated Trajectories II (POST2)," 2 April 2021, [https://www.nasa.gov/post2, Accessed 10 April 2021].

[30] Schoenenberger, M., et al, "Aerodynamic Challenges for the Mars Science Laboratory Entry, Descent and Landing,", AIAA 2009-3914, San Antionio, TX, 2009.

[31] Korzun, A. M., et al, "Aerodynamic Performance of the 2018 InSight Mars Lander," AIAA 2020-1272, Orlando, FL, 2020.

[32] Blanchard, R. C. and Desai, P. N., "Mars Phoenix Entry, Descent, and Landing Trajectory and Atmosphere Reconstruction," *Journal of Spacecraft and Rockets*, Vol. 48, No. 5, 2011, pp. 809-821.

[33] Johnson, M. A., et al, "Performance of the InSight Spacecraft During Entry, Descent and Landing," 16th International Planetary Probe Workshop, Mars Exploration Session, Oxford, UK, 2019.

[34] Anderson Jr, J. D., *Introduction to Flight*, 3rd Edition, McGraw-Hill, Boston, Massachusetts, 1989, pg. 185.

[35] Dickinson, D., et al, "Balloon Launched Decelerator Test Program, Post–Test Test Report, BLDT Vehicle AV-4," NASA CR-1121179, 1972.

[36] Cruz, J. R., et al, "Wind Tunnel Testing of Various Disk-Gap-Band Parachutes," AIAA 2003-2129, Monterey, CA, 2003.

[37] Tamppari, L. K. , et al, "Expected Atmosphere Environment for the Phoenix Season and Location," *J. Geophys. Res*., Vol. 113, No. E3, 2008.

[38] Abilleria, F., et al, "2018 Mars InSight Mission Design and Navigation Overview," AAS 18-251, Snowbird, UT, 2018.

[39] Karlgaard, C., et al, "InSight Trajectory and Atmosphere Reconstruction," 16th International Planetary Probe Workshop, Mars Exploration Session, Oxford, UK, 2019.

[40] Maddock, R. W., and et al, "InSight Entry, Descent and Landing Post-Flight Performance Assessment," AIAA 2020-1270, Orlando, FL, 2020.

[41] Beck, R. A. S., et al, "InSight Aerothermal Environment Assessment," AIAA 2020-1273, Orlando, FL, 2020.

[42] Sutton, K. and Graves, Jr, R. A., "A General Stagnation-Point Convective-Heating Equation for Arbitrary Gas Mixtures," NASA Technical Report R-376, Hampton, VA, 1972.

[43] Justus, C. G., Duvall, A. and Keller, V. W., "Validation of Mars-GRAM and Planned New Features," 35th COSPAR Scientific Assembly Meeting, Paris, France, 2004.

[44] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, NY, 1991, pp. 151-152.

Appendix A TMST Code – Startup Window

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Target Margin Assessment Tool    %
% R. Grover                        %
% 5/4/2021                         %
% StartWindow_v10.mapp             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

classdef StartWindow_v10 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        TMST                          matlab.ui.Figure
        TabGroup                      matlab.ui.container.TabGroup
        CrticialMetricsTab            matlab.ui.container.Tab
        CriticalMetricsPanel          matlab.ui.container.Panel
        UnitsLabel                    matlab.ui.control.Label
        MetricNameEditFieldLabel      matlab.ui.control.Label
        MetricNameEditField           matlab.ui.control.EditField
        MetricAddButton               matlab.ui.control.Button
        MetricsDeleteButton           matlab.ui.control.Button
        AssessmentEpochsTab           matlab.ui.container.Tab
        AssessmentEpochsPanel         matlab.ui.container.Panel
        EpochNameEditFieldLabel       matlab.ui.control.Label
        EpochNameEditField            matlab.ui.control.EditField
        EpochAddButton                matlab.ui.control.Button
        EpochDeleteButton             matlab.ui.control.Button
        AssessmentCategoriesTab       matlab.ui.container.Tab
        AssessmentCategoriesPanel     matlab.ui.container.Panel
        WeightLabel                   matlab.ui.control.Label
        CatDeleteButton               matlab.ui.control.Button
        CategoryNameEditFieldLabel    matlab.ui.control.Label
        CategoryNameEditField         matlab.ui.control.EditField
        CatAddButton                  matlab.ui.control.Button
        EpochDropDownLabel            matlab.ui.control.Label
        CatEpochDropDown              matlab.ui.control.DropDown
        WeightRationaleTextArea       matlab.ui.control.TextArea
        WeightRationaleLabel          matlab.ui.control.Label
        AssessmentCriteriaTab         matlab.ui.container.Tab
        AssessementCriteriaPanel      matlab.ui.container.Panel
        NormalDistributionPanel       matlab.ui.container.Panel
        NormalHighPanel               matlab.ui.container.Panel
        HighEditFieldLabel            matlab.ui.control.Label
        NormalHighEditField           matlab.ui.control.NumericEditField
        NormalMediumPanel             matlab.ui.container.Panel
        MediumEditFieldLabel          matlab.ui.control.Label
        NormalMediumEditField         matlab.ui.control.NumericEditField
        NormalLowPanel                matlab.ui.container.Panel
        LowEditFieldLabel             matlab.ui.control.Label
        NormalLowEditField            matlab.ui.control.NumericEditField
        NormalRationaleTextArea       matlab.ui.control.TextArea
        NormalRationaleLabel          matlab.ui.control.Label
        UniformRationaleLabel_3       matlab.ui.control.Label
        UniformDistributionPanel      matlab.ui.container.Panel
        UniformHighPanel              matlab.ui.container.Panel
        HighEditField_2Label          matlab.ui.control.Label
        UniformHighEditField          matlab.ui.control.NumericEditField
        UniformMediumPanel            matlab.ui.container.Panel
        MediumEditField_2Label        matlab.ui.control.Label
        UniformMediumEditField        matlab.ui.control.NumericEditField
        UniformLowPanel               matlab.ui.container.Panel
        LowEditField_2Label           matlab.ui.control.Label
        UniformLowEditField           matlab.ui.control.NumericEditField
        UniformRationaleTextArea      matlab.ui.control.TextArea
        UniformRationaleLabel         matlab.ui.control.Label
        UniformRationaleLabel_2       matlab.ui.control.Label
        EpochDropDown_2Label          matlab.ui.control.Label
        CriteriaEpochDropDown         matlab.ui.control.DropDown
        ExitButton                    matlab.ui.control.Button
        ASSESSMENTCONFIGURATIONLabel  matlab.ui.control.Label
        Panel                         matlab.ui.container.Panel
        LoadButton                    matlab.ui.control.Button
        SaveButton                    matlab.ui.control.Button
        SaveAsButton                  matlab.ui.control.Button
        TargetMarginSizingToolLabel   matlab.ui.control.Label
        EDLSystemsToolBoxLabel        matlab.ui.control.Label
        Panel_2                       matlab.ui.container.Panel
        Label                         matlab.ui.control.Label
        CriticalMetricDropDown        matlab.ui.control.DropDown
        RunAnalysisButton             matlab.ui.control.Button
        AssessmentLabel               matlab.ui.control.Label
        Label_2                       matlab.ui.control.Label
    end


    properties (Access = public)
        Property % Description
        metric_number_limit = 50;       % Maximum number of metrics allowed
        metric_checkboxes(1,50)         matlab.ui.control.CheckBox
        metric_unit_edit(1,50)          matlab.ui.control.EditField
        epoch_number_limit = 10;        % Maximum number of epochs allowed
        epoch_checkboxes(1,10)          matlab.ui.control.CheckBox
        cat_number_limit = 10;          % Maximum number of categories allowed
        cat_checkboxes(1,10)            matlab.ui.control.CheckBox
        cat_weight_edit(1,10)           matlab.ui.control.NumericEditField
        number_of_metrics = 0;          % Number of critical metrics
        metric_names;                   % Array of cells with metric names
        metric_units;                   % Array of cells with metric units
        number_of_epochs = 0;           % Number of epochs
        epoch_names;                    % Array of cells with epoch names
        number_of_categories = 0;       % Number of model confidence assessement catagories
        category_names;                 % Array of cells with catagory names
        category_weight;                % Array of category weights
        metric_config_file;             % Array of metric config file names
        dist_frac;                      % Dist fraction for normal and uniform (high, med, low)
        change_flag = false;            % Flag to track if a save is required for changes
        current_file;                   % Name of current file
        current_path;                   % Path to current file
        normal_scores_rationale;        % Rationale for normal distribution scoring
        normal_rationale_lines_cnt;     % Number of string lines in normal rationale
        normal_rationale_max_cnt;       % Maximum line count in category weight rationales
        uniform_scores_rationale;       % Rationale for uniform distribution scoring
        uniform_rationale_lines_cnt;    % Number of string lines in uniform rationale
        uniform_rationale_max_cnt;      % Maximum line count in category weight rationales
        cat_weight_rationale;           % Rationale for category weighting
        cat_weight_rationale_lines_cnt; % Number of string lines in category weight rationale
        cat_weight_rationale_max_cnt;   % Maximum line count in category weight rationales
        data_loaded_flag = false;       % Flag to track when data is loaded from file
        AssessmentApp;                  % Handle array for opening multiple Assessment Windows
        assessment_window_cnt = 0;      % Number of open assessment windows
        regression_model_path;          % Array of string cells containing all the regression model paths
        metric_index;                   % Index of metric selected in dropdown menu
        cat_epoch_index;                % Index of epoch selected in category epoch dropdown menu
        criteria_epoch_index;           % Index of epoch selected in criteria epoch dropdown menu
    end

    methods (Access = private)

        % Loads critical metric dropdown control
        function LoadDropDown(app)

            % Populate dropdwn menu
            app.CriticalMetricDropDown.Items = {};
            for i=1:app.number_of_metrics
                app.CriticalMetricDropDown.Items{i} = app.metric_names{i};
            end

            % Enable dropdown box and run analysis button
            app.CriticalMetricDropDown.Enable = true;

        end

        % Loads critical metric checkboxes
        function LoadMetricCheckboxes(app)

            % Delete any exisitng metric checkboxes
            for i=1:app.metric_number_limit
                app.metric_checkboxes(i).delete;
                app.metric_unit_edit(i).delete;
            end

            % Create new metric checkboxes
            for i=1:app.number_of_metrics
                app.metric_checkboxes(i) =
                uicheckbox(app.CriticalMetricsPanel,'Visible','off','Enable','off','ValueChangedFcn',@(metric_checkboxes,ev
                ent) cMetricBoxChanged(app,i));
                app.metric_checkboxes(i).Text = app.metric_names{i};
                if (10+27*app.number_of_metrics > 164)
                    app.metric_checkboxes(i).Position = [15 10+27*app.number_of_metrics-27*i 218 22];
                else
                    app.metric_checkboxes(i).Position = [15 164-27*i 218 22];
                end
                app.metric_checkboxes(i).Enable = true;
                app.metric_unit_edit(i) =
                uieditfield(app.CriticalMetricsPanel,'Visible','off','Enable','off','ValueChangedFcn',@(metric_unit_edit,ev
                ent) cUnitsChanged(app,i));
                app.metric_unit_edit(i).Value = sprintf('%s',app.metric_units{i});
                app.metric_unit_edit(i).HorizontalAlignment = 'Center';

                if (10+27*app.number_of_metrics > 164)
                    app.metric_unit_edit(i).Position = [252 10+27*app.number_of_metrics-27*i 51 22];
                    app.UnitsLabbel.Position(2) = 6+27*app.number_of_metrics;
                else
                    app.metric_unit_edit(i).Position = [252 164-27*i 51 22];
                    app.UnitsLabel.Position(2) = 164;
                end
                app.metric_unit_edit(i).Enable = true;


```

```
        end

        % Make metric checkboxes visible
        for i=1:app.number_of_metrics
            app.metric_checkboxes(i).Visible = true;
            app.metric_unit_edit(i).Visible = true;
        end

    end


    % Loads epoch check boxes
    function LoadEpochCheckboxes(app)

        % Delete any exisitng epoch checkboxes
        for i=1:app.epoch_number_limit
            app.epoch_checkboxes(i).delete;
        end

        % Create new epoch checkboxes
        for i=1:app.number_of_epochs
            app.epoch_checkboxes(i) = uicheckbox(app.AssessmentEpochsPanel,'Visible','off','Enable','off','ValueChangedFcn',@(epoch_checkboxes,event) cEpochBoxChanged(app,i));
            app.epoch_checkboxes(i).Text = app.epoch_names{i};
            if (10+27*app.number_of_epochs > 164)
                app.epoch_checkboxes(i).Position = [15 10+27*app.number_of_epochs-27*i 266 22];
            else
                app.epoch_checkboxes(i).Position = [15 164-27*i 266 22];
            end
            app.epoch_checkboxes(i).Enable = true;
        end


        % Make epoch checkboxes visible
        for i=1:app.number_of_epochs
            app.epoch_checkboxes(i).Visible = true;
        end

    end


    % Loads category checkboxes
    function LoadCategoryCheckboxes(app)

        % Delete any exisitng category checkboxes
        for i=1:app.cat_number_limit
            app.cat_checkboxes(i).delete;
            app.cat_weight_edit(i).delete;
        end

        % Set epoch index
        app.cat_epoch_index = 1;
        app.CatEpochDropDown.Value = app.epoch_names{app.cat_epoch_index};

        % Create new category checkboxes
        for i=1:app.number_of_categories
            app.cat_checkboxes(i) = uicheckbox(app.AssessmentCategoriesPanel,'Visible','off','Enable','off','ValueChangedFcn',@(cat_checkboxes,event) cCatBoxChanged(app,i));
            app.cat_checkboxes(i).Text = app.category_names{i};
            if (10+27*app.number_of_categories > 164)
                app.cat_checkboxes(i).Position = [15 10+27*app.number_of_categories-27*i 218 22];
            else
                app.cat_checkboxes(i).Position = [15 164-27*i 218 22];
            end
            app.cat_checkboxes(i).Enable = true;
            app.cat_weight_edit(i) = uieditfield(app.AssessmentCategoriesPanel,'numeric','Visible','off','Enable','off','ValueChangedFcn',@(cat_weight_edit,event) cWeightChanged(app,i));
            app.cat_weight_edit(i).Value = app.category_weight(app.cat_epoch_index,i);
            app.cat_weight_edit(i).ValueDisplayFormat = '%.2f';
            app.cat_weight_edit(i).HorizontalAlignment = 'Center';
            if (10+27*app.number_of_categories > 164)
                app.cat_weight_edit(i).Position = [252 10+27*app.number_of_categories-27*i 44 22];
                app.WeightLabel.Position(2) = 6+27*app.number_of_categories;
            else
                app.cat_weight_edit(i).Position = [252 164-27*i 44 22];
                app.WeightLabel.Position(2) = 164;
            end
            app.cat_weight_edit(i).Enable = true;
        end


        % Make category checkboxes visible
        for i=1:app.number_of_categories
            app.cat_checkboxes(i).Visible = true;
            app.cat_weight_edit(i).Visible = true;
        end



    end


    % Metric checkbox changed callback function
    function cMetricBoxChanged(app,i)

        % If checkbox has been checked, enable delete button
        if (app.metric_checkboxes(i).Value)
            app.MetricsDeleteButton.Enable = true;

       % Disable delete button if all are unchecked
        else
            checked_sum = 0;
            for (j=1:app.number_of_metrics)
                checked_sum = checked_sum + app.metric_checkboxes(j).Value;
            end
            if (checked_sum == 0)
                app.MetricsDeleteButton.Enable = false;
            end
        end
    end


    % Epoch checkbox changed callback function
    function cEpochBoxChanged(app,i)

        % If checkbox has been checked, enable delete button
        if (app.epoch_checkboxes(i).Value)
            app.EpochDeleteButton.Enable = true;

       % Disable delete button if all are unchecked
        else
            checked_sum = 0;
            for (j=1:app.number_of_epochs)
                checked_sum = checked_sum + app.epoch_checkboxes(j).Value;
            end
            if (checked_sum == 0)
                app.EpochDeleteButton.Enable = false;
            end
        end
    end


    % Category checkbox changed callback function
    function cCatBoxChanged(app,i)

        % If checkbox has been checked, enable delete button
        if (app.cat_checkboxes(i).Value)
            app.CatDeleteButton.Enable = true;

       % Disable delete button if all are unchecked
        else
            checked_sum = 0;
            for j=1:app.number_of_categories
                checked_sum = checked_sum + app.cat_checkboxes(j).Value;
            end
            if (checked_sum == 0)
                app.CatDeleteButton.Enable = false;
            end
        end
    end


    % Category weight edit field change callback function
    function cWeightChanged(app,i)

        % Set change to weight
        app.category_weight(app.cat_epoch_index,i) = app.cat_weight_edit(i).Value;
        % Set change flag
        app.change_flag = true;

    end


    % Metric unit edit field change callback function
    function cUnitsChanged(app,i)

        % Set change to metric units
        app.metric_units{i} = app.metric_unit_edit(i).Value;

        % Set change flag
        app.change_flag = true;

    end

    % Function to save off all setup window control states
```

```
        function save_success = SaveAll(app)

            save_success = false;

            % Open file for saving
            fid = fopen(sprintf('%s%s',app.current_path,app.current_file),'w');

            % Write out header
            fprintf(fid,'%% .MA file containing configuration data for Margin Assessment\n');

            % Write out header
            fprintf(fid,'=== Number of Metrics ===\n');
            % Write out number of critical metric
            fprintf(fid,'%d\n',app.number_of_metrics);

            % Write out metric names
            for i=1:app.number_of_metrics
                 % Write out header
                fprintf(fid,'=== Metric %d ===\n',i);
                fprintf(fid,'%s\n',app.metric_names{i});
                fprintf(fid,'%s\n',app.metric_units{i});
                fprintf(fid,'%s\n',app.metric_config_file{i});
            end

            % Write out header
            fprintf(fid,'=== Number of Epochs ===\n');
            % Write out number of epochs
            fprintf(fid,'%d\n',app.number_of_epochs);

            % Write out header
            fprintf(fid,'=== Epoch Names ===\n');
            % Write out epoch names
            for i=1:app.number_of_epochs
                fprintf(fid,'%s\n',app.epoch_names{i});
            end

            % Write out header
            fprintf(fid,'=== Number of Categories ===\n');
            % Write out number of categories
            fprintf(fid,'%d\n',app.number_of_categories);

            % Write out category names and category weightings
            for i=1:app.number_of_categories
                % Write out header
                fprintf(fid,'=== Category %d ===\n',i);
                fprintf(fid,'%s\n',app.category_names{i});
                for j=1:app.number_of_epochs
                    fprintf(fid,'%4.2f\n',app.category_weight(j,i));
                end
            end

            % Write out header
            fprintf(fid,'=== Category Weight Rationales (per epoch) ===\n');
            % Write out category weighting rationales max line count
            fprintf(fid,'%d\n',app.cat_weight_rationale_max_cnt);
            %Write out line counts and lines
            for i=1:app.number_of_epochs
                fprintf(fid,'%d\n',app.cat_weight_rationale_lines_cnt(i));
                for j=1:app.cat_weight_rationale_lines_cnt(i)
                    fprintf(fid,'%s\n',app.cat_weight_rationale{i,j});
                end
            end

            % Write out header
            fprintf(fid,'=== Normal Fractions ===\n');
            % Write out normal fractions
            for i=1:app.number_of_epochs
                for j=1:3
                    fprintf(fid,'%4.2f\n',app.dist_frac(i,1,j));
                end
            end

            % Write out header
            fprintf(fid,'=== Uniform Fractions ===\n');
            % Write out uniform fractions
            for i=1:app.number_of_epochs
                for j=1:3
                    fprintf(fid,'%4.2f\n',app.dist_frac(i,2,j));
                end
            end

            % Write out header
            fprintf(fid,'=== Normal Fractions Rationale ===\n');

            % Write out normal rationale max line count
            fprintf(fid,'%d\n',app.normal_rationale_max_cnt);

            for i=1:app.number_of_epochs

                % Write out normal rationale line count
                fprintf(fid,'%d\n',app.normal_rationale_lines_cnt(i));

                % Write out normal rationale lines
                for j=1:app.normal_rationale_lines_cnt(i)
                    fprintf(fid,'%s\n',app.normal_scores_rationale{i,j});
                end
            end

            % Write out header
            fprintf(fid,'=== Uniform Fractions Rationale ===\n');

            % Write out normal rationale max line count
            fprintf(fid,'%d\n',app.uniform_rationale_max_cnt);

            for i=1:app.number_of_epochs

                % Write out uniform rationale line count
                fprintf(fid,'%d\n',app.uniform_rationale_lines_cnt(i));

                % Write out uniform rationale lines
                for j=1:app.uniform_rationale_lines_cnt(i)
                    fprintf(fid,'%s\n',app.uniform_scores_rationale{i,j});
                end

            end

            fclose(fid);

            % Reset change flag
            app.change_flag = false;

            save_success = true;

        end


        % Function to provide Save As functionality
        function SaveAs(app)

            % Call save as dialog box
            [app.current_file,app.current_path] = uiputfile('*.ma','Save As');

            % Save off data if user clicked save
            if(app.current_file ~= 0)
                % Update figure title to include file name
                SaveAll(app);
                app.TMST.Name = append('TMST - ',app.current_file);
                app.Label_2.Text = app.current_file(1:end-3);
            end

        end

    end



    methods (Access = public)

        % Is called upon return to Start Window
        function ReturnToSartWindow(app)

            app.RunAnalysisButton.Enable = true;

        end
    end


    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: LoadButton
        function LoadButtonPushed(app, event)

            % Check to see if a save is required
            if (app.change_flag)
               save_before_load = uiconfirm(app.TMST,'Save the current assessment before
               loading?','Save','Options',{'Yes','No'});
               if (strcmp(save_before_load,'Yes'))
                   SaveAll(app);
                   app.change_flag = false;
               end
            end


            % Get file open dialog box and open, if not cancelled
            temp_current_file = app.current_file;
            temp_current_path = app.current_path;
            [app.current_file,app.current_path] = uigetfile(...
```

```
                        {'*.ma', 'Margin Analysis Files (*.ma)';...
                         '*.*',  'All Files (*.*)'}, ...
                         'Load Margin Analysis');


        if (app.current_file ~= 0)
            fid = fopen(sprintf('%s%s',app.current_path,app.current_file));

                % Reader in header
                fgetl(fid);

                % Reader in header
                fgetl(fid);
                % Read in number of critical metrics
                app.number_of_metrics = str2num(fgetl(fid));

                % Set size of metric name
                app.metric_names = cell(app.number_of_metrics);

                % Set size of metric units
                app.metric_units = cell(app.number_of_metrics);

                % Set size of metric config file names array
                app.metric_config_file = cell(app.number_of_metrics);

                % Read in metric names
                for i=1:app.number_of_metrics
                    % Reader in header
                    fgetl(fid);
                    app.metric_names{i} = fgetl(fid);
                    app.metric_units{i} = fgetl(fid);
                    app.metric_config_file{i} = fgetl(fid);
                end

                % Reader in header
                fgetl(fid);
                % Read in number of assessment epochs
                app.number_of_epochs = str2num(fgetl(fid));

                % Set size of epoch name
                app.epoch_names = cell(app.number_of_metrics);

                % Reader in header
                fgetl(fid);
                % Read in epoch names
                for i=1:app.number_of_epochs
                    app.epoch_names{i} = fgetl(fid);
                end

                % Reader in header
                fgetl(fid);
                % Read in number of assessment categories
                app.number_of_categories = str2num(fgetl(fid));

                % Set size of category names
                app.category_names = cell(app.number_of_metrics);
                app.category_weight = zeros(app.number_of_epochs,app.number_of_categories);

                % Read in category names and weights
                for i=1:app.number_of_categories
                    % Reader in header
                    fgetl(fid);
                    app.category_names{i} = fgetl(fid);
                    for j=1:app.number_of_epochs
                        app.category_weight(j,i) = str2double(fgetl(fid));
                    end
                end

                % Reader in header
                fgetl(fid);
                % Read in maximum line count from rationales
                app.cat_weight_rationale_max_cnt = str2num(fgetl(fid));

                % Create cell array for rationale and line count array
                app.cat_weight_rationale = cell(app.number_of_epochs,app.cat_weight_rationale_max_cnt);
                app.cat_weight_rationale_lines_cnt = zeros(app.number_of_epochs,1);

                % Read in category weight rationale lines
                for i=1:app.number_of_epochs
                    app.cat_weight_rationale_lines_cnt(i) = str2num(fgetl(fid));
                    for j=1:app.cat_weight_rationale_lines_cnt(i)
                        app.cat_weight_rationale{i,j} = fgetl(fid);
                    end
                end

                % Read sigma/uniform distribution fractions
                % Create distribution fraction array
                app.dist_frac = zeros(app.number_of_epochs,2,3);
                % Reader in header
                fgetl(fid);
                % Read in normal fractions
                for i=1:app.number_of_epochs
                    app.dist_frac(i,1,1) = str2double(fgetl(fid));
                    app.dist_frac(i,1,2) = str2double(fgetl(fid));
                    app.dist_frac(i,1,3) = str2double(fgetl(fid));
                end
                % Reader in header
                fgetl(fid);
                % Read in uniform fractions
                for i=1:app.number_of_epochs
                    app.dist_frac(i,2,1) = str2double(fgetl(fid));
                    app.dist_frac(i,2,2) = str2double(fgetl(fid));
                    app.dist_frac(i,2,3) = str2double(fgetl(fid));
                end

                % Reader in header
                fgetl(fid);

                % Read in max line count for normal rationales
                app.normal_rationale_max_cnt = str2num(fgetl(fid));

                % Make normal rationale cell array
                app.normal_scores_rationale = cell(0);
                app.normal_scores_rationale = cell(app.number_of_epochs,app.normal_rationale_max_cnt);
                app.normal_rationale_lines_cnt = zeros(app.number_of_epochs,1);

                % Read in rationales
                for i=1:app.number_of_epochs
                    % Line count
                    app.normal_rationale_lines_cnt(i) = str2num(fgetl(fid));

                    % Read in the rationale lines
                    for j=1:app.normal_rationale_lines_cnt(i)
                        app.normal_scores_rationale{i,j} = fgetl(fid);
                    end
                end

                % Reader in header
                fgetl(fid);

                % Read in max line count for normal rationales
                app.uniform_rationale_max_cnt = str2num(fgetl(fid));

                % Make normal rationale cell array
                app.uniform_scores_rationale = cell(0);
                app.uniform_scores_rationale = cell(app.number_of_epochs,app.uniform_rationale_max_cnt);
                app.uniform_rationale_lines_cnt = zeros(app.number_of_epochs,1);

                % Read in rationales
                for i=1:app.number_of_epochs
                    % Line count
                    app.uniform_rationale_lines_cnt(i) = str2num(fgetl(fid));

                    % Read in the rationale lines
                    for j=1:app.uniform_rationale_lines_cnt(i)
                        app.uniform_scores_rationale{i,j} = fgetl(fid);
                    end
                end

                % Close file
                fclose(fid);

            % Set up start window based on loaded metric data
            if (app.number_of_metrics > 0)

                % Load critical metrics dropdown box
                LoadDropDown(app);

                % Load critical metric checkboxes
                LoadMetricCheckboxes(app);

            end


            % Set up start window based on loaded epoch data
            if (app.number_of_epochs > 0)

                % Load critical epoch checkboxes
                LoadEpochCheckboxes(app);

                % Load in epoch names in category epoch dropdown menu
                for i=1:app.number_of_epochs
                    app.CatEpochDropDown.Items{i} = app.epoch_names{i};
                end

                % Load in epoch names in criteria epoch dropdown menu
                app.CriteriaEpochDropDown.Enable = true;
                for i=1:app.number_of_epochs
```

```
                    app.CriteriaEpochDropDown.Items{i} = app.epoch_names{i};
                end

            end


            % Set up start window based on loaded category data
            if (app.number_of_categories > 0)

                % Enable epoch dropdown menu
                app.CatEpochDropDown.Enable = true;

                % Load category epoch checkboxes
                LoadCategoryCheckboxes(app);

                % Load in category weight rationale
                app.WeightRationaleTextArea.Value =
                app.cat_weight_rationale(1,1:app.cat_weight_rationale_lines_cnt(1));

            end

            if (app.number_of_epochs > 0)

                % Load in sigma and uniform fractions
                app.criteria_epoch_index = 1;
                app.NormalHighEditField.Value = app.dist_frac(app.criteria_epoch_index,1,1);
                app.NormalMediumEditField.Value = app.dist_frac(app.criteria_epoch_index,1,2);
                app.NormalLowEditField.Value = app.dist_frac(app.criteria_epoch_index,1,3);
                app.UniformHighEditField.Value = app.dist_frac(app.criteria_epoch_index,2,1);
                app.UniformMediumEditField.Value = app.dist_frac(app.criteria_epoch_index,2,2);
                app.UniformLowEditField.Value = app.dist_frac(app.criteria_epoch_index,2,3);

                % Load in rationale strings
                app.criteria_epoch_index = 1;
                app.NormalRationaleTextArea.Value = app.normal_scores_rationale{app.criteria_epoch_index,...
                                                   1:app.normal_rationale_lines_cnt(app.criteria_epoch_index)};
                app.UniformRationaleTextArea.Value = app.uniform_scores_rationale{app.criteria_epoch_index,...
                                                   1:app.uniform_rationale_lines_cnt(app.criteria_epoch_index)};
            end

            % If all conditions for analysis are met, enable run analysis buttoon
            if (app.number_of_metrics > 0 && app.number_of_epochs > 0 && app.number_of_categories > 0)
                app.RunAnalysisButton.Enable = true;
            end

            % Enable Save As function
            app.SaveAsButton.Enable = true;
            app.SaveButton.Enable = true;

            % Update figure title to include file name
            app.TMST.Name = append('TMST - ',app.current_file);
            app.Label_2.Text = app.current_file(1:end-3);

            % Set data loaded flag
            app.data_loaded_flag = true;

        else

            % Copy back current file and path if no selection
            app.current_file = temp_current_file;
            app.current_path = temp_current_path;

        end


    end

    % Button pushed function: ExitButton
    function ExitButtonPushed(app, event)

        % Check to see if a save is required
        if (app.change_flag)
           save_before_load = uiconfirm(app.TMST,'Save the current assessment before
           closing?','Save','Options',{'Yes','No'});
           if (strcmp(save_before_load,'Yes'))
               if (app.data_loaded_flag)
                   SaveAll(app);
               else
                   SaveAs(app);
               end
               app.change_flag = false;
           end
        end



        % Close application
        close(app.TMST);


    end

    % Value changing function: MetricNameEditField
    function MetricNameEditFieldValueChanging(app, event)
        changingValue = event.Value;

        % Enable/disable add button based on edit field
        if (isempty(changingValue))
            app.MetricAddButton.Enable = false;
        elseif (length(changingValue) == 1)
            app.MetricAddButton.Enable = true;
        end

    end

    % Value changing function: EpochNameEditField
    function EpochNameEditFieldValueChanging(app, event)
        changingValue = event.Value;

        % Enable/disable add button based on edit field
        if (isempty(changingValue))
            app.EpochAddButton.Enable = false;
        elseif (length(changingValue) == 1)
            app.EpochAddButton.Enable = true;
        end

    end

    % Value changing function: CategoryNameEditField
    function CategoryNameEditFieldValueChanging(app, event)
        changingValue = event.Value;

        % Enable/disable add button based on edit field
        if (isempty(changingValue))
            app.CatAddButton.Enable = false;
        elseif (length(changingValue) == 1)
            app.CatAddButton.Enable = true;
        end

    end

    % Button pushed function: MetricAddButton
    function MetricAddButtonPushed(app, event)

         % Check that metric does not already exist
         metric_exists = false;
         for i=1:app.number_of_metrics
             if (strcmp(app.metric_names{i},app.MetricNameEditField.Value))
                 uialert(app.TMST,'Metric name already exists','Invalid Name');
                 metric_exists = true;
             end
         end

         % Add new metric if name is not duplicate
         if (~metric_exists)

            % Temporarily save current metric names
            temp_metric_names = app.metric_names;
            temp_metric_units = app.metric_units;
            temp_metric_config_files = app.metric_config_file;

            % Clear size of metric name
            app.metric_names = cell(0);
            app.metric_units = cell(0);
            app.metric_config_file = cell(0);

            % Increment metric count
            app.number_of_metrics = app.number_of_metrics + 1;

            % Set size of metric name
            app.metric_names = cell(app.number_of_metrics);
            app.metric_units = cell(app.number_of_metrics);
            app.metric_config_file = cell(app.number_of_metrics);

            % Copy back current names
            if (app.number_of_metrics > 1)
                for i=1:app.number_of_metrics-1
                    app.metric_names{i} = temp_metric_names{i};
                    app.metric_units{i} = temp_metric_units{i};
                    app.metric_config_file{i} = temp_metric_config_files{i};
                end
            end
            app.metric_names{app.number_of_metrics} = app.MetricNameEditField.Value;
            app.metric_units{app.number_of_metrics} = '';
            app.metric_config_file{app.number_of_metrics} =
            sprintf('%s_config.mcf',strrep(app.metric_names{app.number_of_metrics},' ','_'));

            % Add metric folder, if it does not exist
```

```
                metric_folder = sprintf('%s/%s',app.current_path,strrep(app.metric_names{app.number_of_metrics},' ',' '));
                if ~exist(metric_folder, 'dir')
                    % Create directory
                    mkdir(metric_folder);
                end

                % Create file path
                config_file_path = sprintf('%s/%s', metric_folder,app.metric_config_file{app.number_of_metrics} );

                % If default config file doesn't exist, create one
                if ~exist(config_file_path)
                    % Open file for writing
                    fid = fopen(config_file_path,'w');

                    % Write out default contents
                    fprintf(fid,'%%.MCF file containing configuration data metric assessment\n');
                    for i=1:app.number_of_epochs
                        fprintf(fid,'NULL\n');
                    end

                    % Close file
                    fclose(fid);
                end



                % Load critical metric checkboxes
                LoadMetricCheckboxes(app);

                % Load critical metrics dropdown box
                LoadDropDown(app);

                % Clear add edit box
                app.MetricNameEditField.Value = "";

                % Disbale add button if maximum metrics reached
                if (app.number_of_metrics == app.metric_number_limit)
                    app.MetricAddButton.Enable = false;
                end

                % Enable delete button, run analysis and dropdown menu if first item added
                if (app.number_of_metrics == 1)
                    app.MetricsDeleteButton.Enable = true;
                    app.MetricsDeleteButton.Enable = true;
                    app.RunAnalysisButton.Enable = true;
                end

                % Update path and file name matrix
                temp_path = app.regression_model_path;
                temp_file = app.regression_model_file;
                app.regression_model_path = cell(0);
                app.regression_model_file = cell(0);
                app.regression_model_path = cell(app.number_of_metrics,app.number_of_epochs);
                app.regression_model_file = cell(app.number_of_metrics,app.number_of_epochs);
                for i=1:(app.number_of_metrics-1)
                   for j=1:app.number_of_epochs
                        app.regression_model_path{i,j} = temp_path{i,j};
                        app.regression_model_file{i,j} = temp_file{i,j};
                   end
                end
                for j=1:app.number_of_epochs
                    app.regression_model_path{app.number_of_metrics,j} = 'NULL';
                    app.regression_model_file{app.number_of_metrics,j} = 'NULL';
                end

                % Set change flag
                app.change_flag = true;

            end

        end

        % Button pushed function: EpochAddButton
        function EpochAddButtonPushed(app, event)

            % Check that epoch does not already exist
            epoch_exists = false;
            for i=1:app.number_of_epochs
                if (strcmp(app.epoch_names{i},app.EpochNameEditField.Value))
                    uialert(app.TMST,'Epoch name already exists','Invalid Name');
                    epoch_exists = true;
                end
            end

            % If epoch name is new
            if (~epoch_exists)

                % Temporarily save current epoch names
                temp_epoch_names = app.epoch_names;
```

```

                % Clear size of epoch name
                app.epoch_names = cell(0);

                % Increment epoch count
                app.number_of_epochs = app.number_of_epochs + 1;

                % Set size of epoch name
                app.epoch_names = cell(app.number_of_epochs);

                % Copy back current names
                if (app.number_of_epochs > 1)
                    for i=1:app.number_of_epochs-1
                        app.epoch_names(i) = temp_epoch_names(i);
                    end
                end
                app.epoch_names{app.number_of_epochs} = app.EpochNameEditField.Value;

                % Load epoch checkboxes
                LoadEpochCheckboxes(app);

                % Clear add edit box
                app.EpochNameEditField.Value = "";

                % Disbale add button if maximum metrics reached
                if (app.number_of_epochs == app.epoch_number_limit)
                    app.EpochAddButton.Enable = false;
                end

                % Enable delete button if first item added
                if (app.number_of_epochs == 1)
                    app.EpochDeleteButton.Enable = true;
                end

                % Update path and file name matrix
                temp_path = app.regression_model_path;
                temp_file = app.regression_model_file;
                app.regression_model_path = cell(0);
                app.regression_model_file = cell(0);
                app.regression_model_path = cell(app.number_of_metrics,app.number_of_epochs);
                app.regression_model_file = cell(app.number_of_metrics,app.number_of_epochs);
                for i=1:(app.number_of_metrics)
                   for j=1:(app.number_of_epochs-1)
                        app.regression_model_path{i,j} = temp_path{i,j};
                        app.regression_model_file{i,j} = temp_file{i,j};
                   end
                end
                for i=1:app.number_of_metrics
                    app.regression_model_path{i,app.number_of_epochs} = 'NULL';
                    app.regression_model_file{i,app.number_of_epochs} = 'NULL';
                end

                % Set change flag
                app.change_flag = true;

            end

        end

        % Button pushed function: CatAddButton
        function CatAddButtonPushed(app, event)

            % Check that category does not already exist
            cat_exists = false;
            for i=1:app.number_of_categories
                if (strcmp(app.category_names{i},app.CategoryNameEditField.Value))
                    uialert(app.TMST,'Category name already exists','Invalid Name');
                    cat_exists = true;
                end
            end

            % If categroy name is new
            if (~cat_exists)

                % Temporarily save current category names
                temp_category_names = app.category_names;
                temp_category_weight = app.category_weight;

                % Clear size of categroy name
                app.category_names = cell(0);

                % Increment category count
                app.number_of_categories = app.number_of_categories + 1;

                % Set size of category name
                app.category_names = cell(app.number_of_categories);
                app.category_weight = zeros(app.number_of_epochs,app.number_of_categories);

                % Copy back current names
```

```
                if (app.number_of_categories > 1)
                    for i=1:app.number_of_categories-1
                        app.category_names(i) = temp_category_names(i);
                        for j=1:app.number_of_epochs
                          app.category_weight(j,i) = temp_category_weight(j,i);
                        end
                    end
                end
                app.category_names{app.number_of_categories} = app.CategoryNameEditField.Value;
                for i=1:app.number_of_epochs
                    app.category_weight(i,app.number_of_categories) = 1.0;
                end

                % Load epoch checkboxes
                LoadCategoryCheckboxes(app);

                % Clear add edit box
                app.CategoryNameEditField.Value = "";

                % Disbale add button if maximum categories reached
                if (app.number_of_categories == app.cat_number_limit)
                    app.CatAddButton.Enable = false;
                end

                % Enable delete button if first item added
                if (app.number_of_categories == 1)
                    app.CatDeleteButton.Enable = true;
                end

                % Set change flag
                app.change_flag = true;

            end


        end

        % Button pushed function: MetricsDeleteButton
        function MetricsDeleteButtonPushed(app, event)

            % Copy over the temp path and file matrix
            temp_regression_model_path = app.regression_model_path;
            temp_regression_model_file = app.regression_model_file;
            temp_metric_config_file = app.metric_config_file;


             % Count number of metrics to keep
             temp_number_of_metrics = 0;
             for i=1:app.number_of_metrics
                 if (app.metric_checkboxes(i).Value == false)
                     temp_number_of_metrics = temp_number_of_metrics + 1;
                 end
             end


             % Reset metric names and path/file matrices
             app.metric_names = cell(0);
             app.metric_units = cell(0);
             app.metric_config_file = cell(0);
             app.regression_model_file = cell(0);
             app.regression_model_path = cell(0);

             % Set size of metric name and path/file matrices
             app.metric_names = cell(temp_number_of_metrics);
             app.metric_units = cell(temp_number_of_metrics);
             app.metric_config_file = cell(temp_number_of_metrics);
             app.regression_model_file = cell(temp_number_of_metrics,app.number_of_epochs);
             app.regression_model_path = cell(temp_number_of_metrics,app.number_of_epochs);

             % Copy over metric names
             j = 0;
             for i=1:app.number_of_metrics
                 if (app.metric_checkboxes(i).Value == false)
                     j = j + 1;
                     app.metric_names{j} = app.metric_checkboxes(i).Text;
                     app.metric_units{j} = app.metric_unit_edit(i).Value;
                     app.metric_config_file{j} = temp_metric_config_file{i};
                     for k=1:app.number_of_epochs
                        app.regression_model_file{j,k} = temp_regression_model_file{i,k};
                        app.regression_model_path{j,k} = temp_regression_model_path{i,k};
                     end
                 end
             end

             % Update number of metrics
             app.number_of_metrics = temp_number_of_metrics;

             % Load critical metric checkboxes
             LoadMetricCheckboxes(app);

             % Load critical metrics dropdown box
             LoadDropDown(app);

             % Take care of control states
             if (app.number_of_metrics == 0)
                 % Diable delete button, run analysis button and dropdown menu
                 app.MetricsDeleteButton.Enable = false;
                 app.RunAnalysisButton.Enable = false;
                 app.CriticalMetricDropDown.Enable = false;
             elseif (app.number_of_metrics == app.metric_number_limit-1)
                 % Enable Add button
                 app.MetricAddButton.Enable = true;
             end

             % Set change flag
             app.change_flag = true;

             % Disable delete button
             app.MetricsDeleteButton.Enable = false;


        end

        % Button pushed function: EpochDeleteButton
        function EpochDeleteButtonPushed(app, event)

            % Copy over the temp path and file matrix
            temp_regression_model_path = app.regression_model_path;
            temp_regression_model_file = app.regression_model_file;

             % Count number of epochs to keep
             temp_number_of_epochs = 0;
             for i=1:app.number_of_epochs
                 if (app.epoch_checkboxes(i).Value == false)
                     temp_number_of_epochs = temp_number_of_epochs + 1;
                 end
             end

             % Reset epoch names and path/file matrices
             app.epoch_names = cell(0);
             app.regression_model_file = cell(0);
             app.regression_model_path = cell(0);

             % Set size of epoch name
             app.epoch_names = cell(temp_number_of_epochs);
             app.regression_model_file = cell(app.number_of_metrics,temp_number_of_epochs);
             app.regression_model_path = cell(app.number_of_metrics,temp_number_of_epochs);

             % Copy over epoch names
             j = 0;
             for i=1:app.number_of_epochs
                 if (app.epoch_checkboxes(i).Value == false)
                     j = j + 1;
                     app.epoch_names{j} = app.epoch_checkboxes(i).Text;
                     for k=1:app.number_of_metrics
                        app.regression_model_file{k,j} = temp_regression_model_file{k,i};
                        app.regression_model_path{k,j} = temp_regression_model_path{k,i};
                     end
                 end
             end
             app.regression_model_file{app.number_of_metrics,j}

             % Update number of epochs
             app.number_of_epochs = temp_number_of_epochs;

             % Load critical metric checkboxes
             LoadEpochCheckboxes(app);

             % Take care of control states
             if (app.number_of_epochs == 0)
                 % Disable delete button
                 app.EpochDeleteButton.Enable = false;
             elseif (app.number_of_epochs == app.epoch_number_limit-1)
                 % Enable Add button
                 app.EpochAddButton.Enable = true;
             end

             % Set change flag
             app.change_flag = true;

             % Disable delete button
             app.EpochDeleteButton.Enable = false;


        end

        % Button pushed function: CatDeleteButton
        function CatDeleteButtonPushed(app, event)
```

```
            % Count number of categories to keep
            temp_number_of_categories = 0;
            for i=1:app.number_of_categories
                if (app.cat_checkboxes(i).Value == false)
                    temp_number_of_categories = temp_number_of_categories + 1;
                end
            end

            % Temporarily save category weights
            temp_category_weight = app.category_weight;

            % Reset category names
            app.category_names = cell(0);

            % Set size of category name
            app.category_names = cell(temp_number_of_categories);
            app.category_weight = zeros(app.number_of_epochs,temp_number_of_categories);


            % Copy over category names and weights
            j = 0;
            for i=1:app.number_of_categories
                if (app.cat_checkboxes(i).Value == false)
                    j = j + 1;
                    app.category_names{j} = app.cat_checkboxes(i).Text;
                    for k=1:app.number_of_epochs
                       app.category_weight(k,j) = temp_category_weight(k,i);
                    end
                end
            end

            % Update number of categories
            app.number_of_categories = temp_number_of_categories;

            % Load critical metric checkboxes
            LoadCategoryCheckboxes(app);

            % Take care of control states
            if (app.number_of_categories == 0)
                % Disable delete button
                app.CatDeleteButton.Enable = false;
            elseif (app.number_of_categories == app.cat_number_limit-1)
                % Enable Add button
                app.CatAddButton.Enable = true;
            end

            % Set change flag
            app.change_flag = true;

            % Disable delete button
            app.CatDeleteButton.Enable = false;

        end

        % Button pushed function: SaveButton
        function SaveButtonPushed(app, event)

            % Save configuration
            SaveAll(app);

        end

        % Value changed function: NormalHighEditField
        function NormalHighEditFieldValueChanged(app, event)
            value = app.NormalHighEditField.Value;

            % Set new Normal High Value
            app.dist_frac(app.criteria_epoch_index,1,1) = value;

            % Set change flag
            app.change_flag = true;

        end

        % Value changed function: NormalMediumEditField
        function NormalMediumEditFieldValueChanged(app, event)
            value = app.NormalMediumEditField.Value;

            % Set new Normal Medium Value
            app.dist_frac(app.criteria_epoch_index,1,2) = value;

            % Set change flag
            app.change_flag = true;

        end
```

```
        % Value changed function: NormalLowEditField
        function NormalLowEditFieldValueChanged(app, event)
            value = app.NormalLowEditField.Value;

            % Set new Normal Low Value
            app.dist_frac(app.criteria_epoch_index,1,3) = value;

            % Set change flag
            app.change_flag = true;

        end

        % Value changed function: UniformHighEditField
        function UniformHighEditFieldValueChanged(app, event)
            value = app.UniformHighEditField.Value;

            % Set new Uniform High Value
            app.dist_frac(app.criteria_epoch_index,2,1) = value;

            % Set change flag
            app.change_flag = true;

        end

        % Value changed function: UniformMediumEditField
        function UniformMediumEditFieldValueChanged(app, event)
            value = app.UniformMediumEditField.Value;

            % Set new Uniform Medium Value
            app.dist_frac(app.criteria_epoch_index,2,2) = value;

            % Set change flag
            app.change_flag = true;

        end

        % Value changed function: UniformLowEditField
        function UniformLowEditFieldValueChanged(app, event)
            value = app.UniformLowEditField.Value;

            % Set new Uniform Low Value
            app.dist_frac(app.criteria_epoch_index,2,3) = value;

            % Set change flag
            app.change_flag = true;

        end

        % Button pushed function: SaveAsButton
        function SaveAsButtonPushed(app, event)

         % Save off all config data
         SaveAs(app);

        end

        % Value changed function: NormalRationaleTextArea
        function NormalRationaleTextAreaValueChanged(app, event)
            value = app.NormalRationaleTextArea.Value;

            % Get number of line
            N = size(value);

            app.normal_rationale_lines_cnt(app.criteria_epoch_index) = N(1,1);

            % Reset max rationale lines
            app.normal_rationale_max_cnt = max(app.normal_rationale_lines_cnt);

            % Copy rationales to temp variable
            temp_normal_rationales = app.normal_scores_rationale;

            % Reset uniform rationale string cells
            app.normal_scores_rationale = cell(0);
            app.normal_scores_rationale = cell(app.number_of_epochs,app.normal_rationale_max_cnt);
            for i=1:app.number_of_epochs
               if (i == app.criteria_epoch_index)
                    app.normal_scores_rationale(i,1:app.normal_rationale_lines_cnt(i)) = value;
               else
                  app.normal_scores_rationale(i,1:app.normal_rationale_lines_cnt(i)) =...
                      temp_normal_rationales(i,1:app.normal_rationale_lines_cnt(i));
               end
            end


            % Set change flag
            app.change_flag = true;

        end
```

```
        % Value changed function: UniformRationaleTextArea
        function UniformRationaleTextAreaValueChanged(app, event)
            value = app.UniformRationaleTextArea.Value;

            % Get number of line
            N = size(value);
            app.uniform_rationale_lines_cnt(app.criteria_epoch_index) = N(1,1);

            % Reset max rationale lines
            app.uniform_rationale_max_cnt = max(app.uniform_rationale_lines_cnt);

            % Copy rationales to temp variable
            temp_uniform_rationales = app.uniform_scores_rationale;

            % Reset uniform rationale string cells
            app.uniform_scores_rationale = cell(0);
            app.uniform_scores_rationale = cell(app.number_of_epochs,app.uniform_rationale_max_cnt);
            app.uniform_rationale_lines_cnt(:,1)
            for i=1:app.number_of_epochs
               if (i == app.criteria_epoch_index)
                    app.uniform_scores_rationale(i,1:app.uniform_rationale_lines_cnt(i)) = value;
               else
                  app.uniform_scores_rationale(i,1:app.uniform_rationale_lines_cnt(i)) =...
                      temp_uniform_rationales(i,1:app.uniform_rationale_lines_cnt(i));
               end
            end

            % Set change flag
            app.change_flag = true;

        end

        % Button pushed function: RunAnalysisButton
        function RunAnalysisButtonPushed(app, event)

            % disable run analysis buttion
            app.RunAnalysisButton.Enable = false;

            % Get selected metric index
            for i=1:app.number_of_metrics
                if (strcmp(app.CriticalMetricDropDown.Value,app.metric_names{i}))
                    app.metric_index = i;
                end
            end

            % Call assessment window app with input values
            app.AssessmentApp = AssessmentWindow_v8(app,app.CriticalMetricDropDown.Value,app.metric_index);

        end

        % Value changed function: CatEpochDropDown
        function CatEpochDropDownValueChanged(app, event)
            value = app.CatEpochDropDown.Value;

            % Get selected metric index
            for i=1:app.number_of_epochs
                if (strcmp(value,app.epoch_names{i}))
                    app.cat_epoch_index = i;
                end
            end

            % Set new weight values to edit fields
            for i=1:app.number_of_categories
                app.cat_weight_edit(i).Value = app.category_weight(app.cat_epoch_index,i);
            end

            % Load in cat weigth rationale for epoch
            app.WeightRationaleTextArea.Value = app.cat_weight_rationale(app.cat_epoch_index,...
                                                1:app.cat_weight_rationale_lines_cnt(app.cat_epoch_index));

        end

        % Value changed function: CriteriaEpochDropDown
        function CriteriaEpochDropDownValueChanged(app, event)
            value = app.CriteriaEpochDropDown.Value;

            % Get selected metric index
            for i=1:app.number_of_epochs
                if (strcmp(value,app.epoch_names{i}))
                    app.criteria_epoch_index = i;
                end
            end

            % Set new fraction values to edit fields
            app.NormalHighEditField.Value = app.dist_frac(app.criteria_epoch_index,1,1);
            app.NormalMediumEditField.Value = app.dist_frac(app.criteria_epoch_index,1,2);
            app.NormalLowEditField.Value = app.dist_frac(app.criteria_epoch_index,1,3);
            app.UniformHighEditField.Value = app.dist_frac(app.criteria_epoch_index,2,1);
            app.UniformMediumEditField.Value = app.dist_frac(app.criteria_epoch_index,2,2);
            app.UniformLowEditField.Value = app.dist_frac(app.criteria_epoch_index,2,3);

            % Set rationales
            app.NormalRationaleTextArea.Value = app.normal_scores_rationale(app.criteria_epoch_index,...
                                                    1:app.normal_rationale_lines_cnt(app.criteria_epoch_index,1));
            app.UniformRationaleTextArea.Value = app.uniform_scores_rationale(app.criteria_epoch_index,...
                                                    1:app.uniform_rationale_lines_cnt(app.criteria_epoch_index,1));

        end

        % Value changed function: WeightRationaleTextArea
        function WeightRationaleTextAreaValueChanged(app, event)
            value = app.WeightRationaleTextArea.Value;

            % Get number of line
            N = size(value);
            app.cat_weight_rationale_lines_cnt(app.cat_epoch_index) = N(1,1);

            % Reset max rationale lines
            app.cat_weight_rationale_max_cnt = max(app.cat_weight_rationale_lines_cnt);

            % Caopy rationales to temp variable
            temp_cat_weight_rationales = app.cat_weight_rationale;

            % Reset uniform rationale string cells
            app.cat_weight_rationale = cell(0);
            app.cat_weight_rationale = cell(app.number_of_epochs,app.cat_weight_rationale_max_cnt);
            for i=1:app.number_of_epochs
               if (i == app.cat_epoch_index)
                    app.cat_weight_rationale(i,1:app.cat_weight_rationale_lines_cnt(i)) = value;
               else
                  app.cat_weight_rationale(i,1:app.cat_weight_rationale_lines_cnt(i)) =...
                      temp_cat_weight_rationales(i,1:app.cat_weight_rationale_lines_cnt(i));
               end
            end

            % Set change flag
            app.change_flag = true;


        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create TMST and hide until all components are created
            app.TMST = uifigure('Visible', 'off');
            app.TMST.Position = [100 100 936 346];
            app.TMST.Name = 'TMST';
            app.TMST.Resize = 'off';

            % Create TabGroup
            app.TabGroup = uitabgroup(app.TMST);
            app.TabGroup.Position = [281 13 630 293];

            % Create CrticialMetricsTab
            app.CrticialMetricsTab = uitab(app.TabGroup);
            app.CrticialMetricsTab.Title = 'Crticial Metrics';
            app.CrticialMetricsTab.BackgroundColor = [1 1 1];

            % Create CriticalMetricsPanel
            app.CriticalMetricsPanel = uipanel(app.CrticialMetricsTab);
            app.CriticalMetricsPanel.ForegroundColor = [0 0.4471 0.7412];
            app.CriticalMetricsPanel.Title = 'Critical Metrics';
            app.CriticalMetricsPanel.BackgroundColor = [0.851 0.9098 0.9412];
            app.CriticalMetricsPanel.FontName = 'Arial';
            app.CriticalMetricsPanel.FontWeight = 'bold';
            app.CriticalMetricsPanel.Scrollable = 'on';
            app.CriticalMetricsPanel.FontSize = 14;
            app.CriticalMetricsPanel.Position = [29 60 328 184];

            % Create UnitsLabel
            app.UnitsLabel = uilabel(app.CriticalMetricsPanel);
            app.UnitsLabel.HorizontalAlignment = 'center';
            app.UnitsLabel.FontName = 'Arial';
            app.UnitsLabel.FontSize = 10;
            app.UnitsLabel.Position = [258 145 36 15];
            app.UnitsLabel.Text = 'Units';

            % Create MetricNameEditFieldLabel
            app.MetricNameEditFieldLabel = uilabel(app.CrticialMetricsTab);
            app.MetricNameEditFieldLabel.HorizontalAlignment = 'right';
            app.MetricNameEditFieldLabel.Position = [377 218 74 22];
            app.MetricNameEditFieldLabel.Text = 'Metric Name';
```

```
            % Create MetricNameEditField
            app.MetricNameEditField = uieditfield(app.CrticialMetricsTab, 'text');
            app.MetricNameEditField.ValueChangingFcn = createCallbackFcn(app, @MetricNameEditFieldValueChanging, true);
            app.MetricNameEditField.Position = [456 218 163 22];

            % Create MetricAddButton
            app.MetricAddButton = uibutton(app.CrticialMetricsTab, 'push');
            app.MetricAddButton.ButtonPushedFcn = createCallbackFcn(app, @MetricAddButtonPushed, true);
            app.MetricAddButton.Enable = 'off';
            app.MetricAddButton.Position = [552 191 67 22];
            app.MetricAddButton.Text = 'Add';

            % Create MetricsDeleteButton
            app.MetricsDeleteButton = uibutton(app.CrticialMetricsTab, 'push');
            app.MetricsDeleteButton.ButtonPushedFcn = createCallbackFcn(app, @MetricsDeleteButtonPushed, true);
            app.MetricsDeleteButton.Enable = 'off';
            app.MetricsDeleteButton.Position = [29 24 100 22];
            app.MetricsDeleteButton.Text = 'Delete';

            % Create AssessmentEpochsTab
            app.AssessmentEpochsTab = uitab(app.TabGroup);
            app.AssessmentEpochsTab.Title = 'Assessment Epochs';

            % Create AssessmentEpochsPanel
            app.AssessmentEpochsPanel = uipanel(app.AssessmentEpochsTab);
            app.AssessmentEpochsPanel.ForegroundColor = [0 0.4471 0.7412];
            app.AssessmentEpochsPanel.Title = 'Assessment Epochs';
            app.AssessmentEpochsPanel.BackgroundColor = [0.851 0.9098 0.9412];
            app.AssessmentEpochsPanel.FontName = 'Arial';
            app.AssessmentEpochsPanel.FontWeight = 'bold';
            app.AssessmentEpochsPanel.Scrollable = 'on';
            app.AssessmentEpochsPanel.FontSize = 14;
            app.AssessmentEpochsPanel.Position = [29 60 328 184];

            % Create EpochNameEditFieldLabel
            app.EpochNameEditFieldLabel = uilabel(app.AssessmentEpochsTab);
            app.EpochNameEditFieldLabel.HorizontalAlignment = 'right';
            app.EpochNameEditFieldLabel.Position = [376 220 75 22];
            app.EpochNameEditFieldLabel.Text = 'Epoch Name';

            % Create EpochNameEditField
            app.EpochNameEditField = uieditfield(app.AssessmentEpochsTab, 'text');
            app.EpochNameEditField.ValueChangingFcn = createCallbackFcn(app, @EpochNameEditFieldValueChanging, true);
            app.EpochNameEditField.Position = [456 220 163 22];

            % Create EpochAddButton
            app.EpochAddButton = uibutton(app.AssessmentEpochsTab, 'push');
            app.EpochAddButton.ButtonPushedFcn = createCallbackFcn(app, @EpochAddButtonPushed, true);
            app.EpochAddButton.Enable = 'off';
            app.EpochAddButton.Position = [552 193 67 22];
            app.EpochAddButton.Text = 'Add';

            % Create EpochDeleteButton
            app.EpochDeleteButton = uibutton(app.AssessmentEpochsTab, 'push');
            app.EpochDeleteButton.ButtonPushedFcn = createCallbackFcn(app, @EpochDeleteButtonPushed, true);
            app.EpochDeleteButton.Enable = 'off';
            app.EpochDeleteButton.Position = [29 24 100 22];
            app.EpochDeleteButton.Text = 'Delete';

            % Create AssessmentCategoriesTab
            app.AssessmentCategoriesTab = uitab(app.TabGroup);
            app.AssessmentCategoriesTab.Title = 'Assessment Categories';
            app.AssessmentCategoriesTab.BackgroundColor = [1 1 1];

            % Create AssessmentCategoriesPanel
            app.AssessmentCategoriesPanel = uipanel(app.AssessmentCategoriesTab);
            app.AssessmentCategoriesPanel.ForegroundColor = [0 0.4471 0.7412];
            app.AssessmentCategoriesPanel.Title = 'Assessment Categories';
            app.AssessmentCategoriesPanel.BackgroundColor = [0.851 0.9098 0.9412];
            app.AssessmentCategoriesPanel.FontName = 'Arial';
            app.AssessmentCategoriesPanel.FontWeight = 'bold';
            app.AssessmentCategoriesPanel.Scrollable = 'on';
            app.AssessmentCategoriesPanel.FontSize = 14;
            app.AssessmentCategoriesPanel.Position = [29 60 328 184];

            % Create WeightLabel
            app.WeightLabel = uilabel(app.AssessmentCategoriesPanel);
            app.WeightLabel.HorizontalAlignment = 'center';
            app.WeightLabel.FontName = 'Arial';
            app.WeightLabel.FontSize = 10;
            app.WeightLabel.Position = [255 145 36 15];
            app.WeightLabel.Text = 'Weight';

            % Create CatDeleteButton
            app.CatDeleteButton = uibutton(app.AssessmentCategoriesTab, 'push');
            app.CatDeleteButton.ButtonPushedFcn = createCallbackFcn(app, @CatDeleteButtonPushed, true);
            app.CatDeleteButton.Enable = 'off';
            app.CatDeleteButton.Position = [29 24 100 22];
            app.CatDeleteButton.Text = 'Delete';
```

```

            % Create CategoryNameEditFieldLabel
            app.CategoryNameEditFieldLabel = uilabel(app.AssessmentCategoriesTab);
            app.CategoryNameEditFieldLabel.HorizontalAlignment = 'right';
            app.CategoryNameEditFieldLabel.Position = [361 40 90 22];
            app.CategoryNameEditFieldLabel.Text = 'Category Name';

            % Create CategoryNameEditField
            app.CategoryNameEditField = uieditfield(app.AssessmentCategoriesTab, 'text');
            app.CategoryNameEditField.ValueChangingFcn = createCallbackFcn(app, @CategoryNameEditFieldValueChanging, true);
            app.CategoryNameEditField.Position = [455 41 163 22];

            % Create CatAddButton
            app.CatAddButton = uibutton(app.AssessmentCategoriesTab, 'push');
            app.CatAddButton.ButtonPushedFcn = createCallbackFcn(app, @CatAddButtonPushed, true);
            app.CatAddButton.Enable = 'off';
            app.CatAddButton.Position = [551 11 67 22];
            app.CatAddButton.Text = 'Add';

            % Create EpochDropDownLabel
            app.EpochDropDownLabel = uilabel(app.AssessmentCategoriesTab);
            app.EpochDropDownLabel.HorizontalAlignment = 'right';
            app.EpochDropDownLabel.Position = [367 220 51 22];
            app.EpochDropDownLabel.Text = 'Epoch';

            % Create CatEpochDropDown
            app.CatEpochDropDown = uidropdown(app.AssessmentCategoriesTab);
            app.CatEpochDropDown.Items = {};
            app.CatEpochDropDown.ValueChangedFcn = createCallbackFcn(app, @CatEpochDropDownValueChanged, true);
            app.CatEpochDropDown.Enable = 'off';
            app.CatEpochDropDown.Position = [425 219 190 24];
            app.CatEpochDropDown.Value = {};

            % Create WeightRationaleTextArea
            app.WeightRationaleTextArea = uitextarea(app.AssessmentCategoriesTab);
            app.WeightRationaleTextArea.ValueChangedFcn = createCallbackFcn(app, @WeightRationaleTextAreaValueChanged,
            true);
            app.WeightRationaleTextArea.WordWrap = 'off';
            app.WeightRationaleTextArea.Position = [376 87 243 97];

            % Create WeightRationaleLabel
            app.WeightRationaleLabel = uilabel(app.AssessmentCategoriesTab);
            app.WeightRationaleLabel.Position = [378 183 197 22];
            app.WeightRationaleLabel.Text = 'Weighting Rationale';

            % Create AssessmentCriteriaTab
            app.AssessmentCriteriaTab = uitab(app.TabGroup);
            app.AssessmentCriteriaTab.Title = 'Assessment Criteria';

            % Create AssessementCriteriaPanel
            app.AssessementCriteriaPanel = uipanel(app.AssessmentCriteriaTab);
            app.AssessementCriteriaPanel.ForegroundColor = [0 0.4471 0.7412];
            app.AssessementCriteriaPanel.Title = 'Assessement Criteria';
            app.AssessementCriteriaPanel.BackgroundColor = [0.851 0.9098 0.9412];
            app.AssessementCriteriaPanel.FontName = 'Arial';
            app.AssessementCriteriaPanel.FontWeight = 'bold';
            app.AssessementCriteriaPanel.Scrollable = 'on';
            app.AssessementCriteriaPanel.FontSize = 14;
            app.AssessementCriteriaPanel.Position = [26 11 577 244];

            % Create NormalDistributionPanel
            app.NormalDistributionPanel = uipanel(app.AssessementCriteriaPanel);
            app.NormalDistributionPanel.Title = 'Normal Distribution';
            app.NormalDistributionPanel.Position = [19 9 246 178];

            % Create NormalHighPanel
            app.NormalHighPanel = uipanel(app.NormalDistributionPanel);
            app.NormalHighPanel.AutoResizeChildren = 'off';
            app.NormalHighPanel.BackgroundColor = [0 1 0];
            app.NormalHighPanel.Position = [10 101 100 30];

            % Create HighEditFieldLabel
            app.HighEditFieldLabel = uilabel(app.NormalHighPanel);
            app.HighEditFieldLabel.HorizontalAlignment = 'right';
            app.HighEditFieldLabel.Position = [21 4 30 22];
            app.HighEditFieldLabel.Text = 'High';

            % Create NormalHighEditField
            app.NormalHighEditField = uieditfield(app.NormalHighPanel, 'numeric');
            app.NormalHighEditField.Limits = [0 Inf];
            app.NormalHighEditField.ValueDisplayFormat = '%.2f';
            app.NormalHighEditField.ValueChangedFcn = createCallbackFcn(app, @NormalHighEditFieldValueChanged, true);
            app.NormalHighEditField.Position = [54 4 41 22];

            % Create NormalMediumPanel
            app.NormalMediumPanel = uipanel(app.NormalDistributionPanel);
            app.NormalMediumPanel.AutoResizeChildren = 'off';
            app.NormalMediumPanel.BackgroundColor = [1 1 0.0667];
            app.NormalMediumPanel.Position = [10 58 100 30];
```

```
        % Create MediumEditFieldLabel
        app.MediumEditFieldLabel = uilabel(app.NormalMediumPanel);
        app.MediumEditFieldLabel.HorizontalAlignment = 'right';
        app.MediumEditFieldLabel.Position = [4 4 48 22];
        app.MediumEditFieldLabel.Text = 'Medium';

        % Create NormalMediumEditField
        app.NormalMediumEditField = uieditfield(app.NormalMediumPanel, 'numeric');
        app.NormalMediumEditField.Limits = [0 Inf];
        app.NormalMediumEditField.ValueDisplayFormat = '%.2f';
        app.NormalMediumEditField.ValueChangedFcn = createCallbackFcn(app, @NormalMediumEditFieldValueChanged, true);
        app.NormalMediumEditField.Position = [56 4 39 22];

        % Create NormalLowPanel
        app.NormalLowPanel = uipanel(app.NormalDistributionPanel);
        app.NormalLowPanel.AutoResizeChildren = 'off';
        app.NormalLowPanel.BackgroundColor = [1 0 0];
        app.NormalLowPanel.Position = [10 12 100 30];

        % Create LowEditFieldLabel
        app.LowEditFieldLabel = uilabel(app.NormalLowPanel);
        app.LowEditFieldLabel.HorizontalAlignment = 'right';
        app.LowEditFieldLabel.Position = [22 4 28 22];
        app.LowEditFieldLabel.Text = 'Low';

        % Create NormalLowEditField
        app.NormalLowEditField = uieditfield(app.NormalLowPanel, 'numeric');
        app.NormalLowEditField.Limits = [0 Inf];
        app.NormalLowEditField.ValueDisplayFormat = '%.2f';
        app.NormalLowEditField.ValueChangedFcn = createCallbackFcn(app, @NormalLowEditFieldValueChanged, true);
        app.NormalLowEditField.Position = [54 4 41 22];

        % Create NormalRationaleTextArea
        app.NormalRationaleTextArea = uitextarea(app.NormalDistributionPanel);
        app.NormalRationaleTextArea.ValueChangedFcn = createCallbackFcn(app, @NormalRationaleTextAreaValueChanged,
        true);
        app.NormalRationaleTextArea.Position = [124 12 109 119];

        % Create NormalRationaleLabel
        app.NormalRationaleLabel = uilabel(app.NormalDistributionPanel);
        app.NormalRationaleLabel.HorizontalAlignment = 'center';
        app.NormalRationaleLabel.Position = [149 132 59 22];
        app.NormalRationaleLabel.Text = 'Rationale';

        % Create UniformRationaleLabel_3
        app.UniformRationaleLabel_3 = uilabel(app.NormalDistributionPanel);
        app.UniformRationaleLabel_3.HorizontalAlignment = 'center';
        app.UniformRationaleLabel_3.Position = [17 132 86 22];
        app.UniformRationaleLabel_3.Text = 'Sigma Fraction';

        % Create UniformDistributionPanel
        app.UniformDistributionPanel = uipanel(app.AssessementCriteriaPanel);
        app.UniformDistributionPanel.Title = 'Uniform Distribution';
        app.UniformDistributionPanel.Position = [313 9 246 178];

        % Create UniformHighPanel
        app.UniformHighPanel = uipanel(app.UniformDistributionPanel);
        app.UniformHighPanel.AutoResizeChildren = 'off';
        app.UniformHighPanel.BackgroundColor = [0 1 0];
        app.UniformHighPanel.Position = [11 100 100 30];

        % Create HighEditField_2Label
        app.HighEditField_2Label = uilabel(app.UniformHighPanel);
        app.HighEditField_2Label.HorizontalAlignment = 'right';
        app.HighEditField_2Label.Position = [21 4 30 22];
        app.HighEditField_2Label.Text = 'High';

        % Create UniformHighEditField
        app.UniformHighEditField = uieditfield(app.UniformHighPanel, 'numeric');
        app.UniformHighEditField.Limits = [0 1];
        app.UniformHighEditField.ValueDisplayFormat = '%.2f';
        app.UniformHighEditField.ValueChangedFcn = createCallbackFcn(app, @UniformHighEditFieldValueChanged, true);
        app.UniformHighEditField.Position = [54 4 41 22];

        % Create UniformMediumPanel
        app.UniformMediumPanel = uipanel(app.UniformDistributionPanel);
        app.UniformMediumPanel.AutoResizeChildren = 'off';
        app.UniformMediumPanel.BackgroundColor = [1 1 0.0667];
        app.UniformMediumPanel.Position = [11 57 100 30];

        % Create MediumEditField_2Label
        app.MediumEditField_2Label = uilabel(app.UniformMediumPanel);
        app.MediumEditField_2Label.HorizontalAlignment = 'right';
        app.MediumEditField_2Label.Position = [4 4 48 22];
        app.MediumEditField_2Label.Text = 'Medium';

        % Create UniformMediumEditField
        app.UniformMediumEditField = uieditfield(app.UniformMediumPanel, 'numeric');
        app.UniformMediumEditField.Limits = [0 1];
        app.UniformMediumEditField.ValueDisplayFormat = '%.2f';
        app.UniformMediumEditField.ValueChangedFcn = createCallbackFcn(app, @UniformMediumEditFieldValueChanged, true);
        app.UniformMediumEditField.Position = [56 4 39 22];

        % Create UniformLowPanel
        app.UniformLowPanel = uipanel(app.UniformDistributionPanel);
        app.UniformLowPanel.AutoResizeChildren = 'off';
        app.UniformLowPanel.BackgroundColor = [1 0 0];
        app.UniformLowPanel.Position = [11 11 100 30];

        % Create LowEditField_2Label
        app.LowEditField_2Label = uilabel(app.UniformLowPanel);
        app.LowEditField_2Label.HorizontalAlignment = 'right';
        app.LowEditField_2Label.Position = [22 4 28 22];
        app.LowEditField_2Label.Text = 'Low';

        % Create UniformLowEditField
        app.UniformLowEditField = uieditfield(app.UniformLowPanel, 'numeric');
        app.UniformLowEditField.Limits = [0 1];
        app.UniformLowEditField.ValueDisplayFormat = '%.2f';
        app.UniformLowEditField.ValueChangedFcn = createCallbackFcn(app, @UniformLowEditFieldValueChanged, true);
        app.UniformLowEditField.Position = [54 4 41 22];

        % Create UniformRationaleTextArea
        app.UniformRationaleTextArea = uitextarea(app.UniformDistributionPanel);
        app.UniformRationaleTextArea.ValueChangedFcn = createCallbackFcn(app, @UniformRationaleTextAreaValueChanged,
        true);
        app.UniformRationaleTextArea.Position = [126 11 109 121];

        % Create UniformRationaleLabel
        app.UniformRationaleLabel = uilabel(app.UniformDistributionPanel);
        app.UniformRationaleLabel.HorizontalAlignment = 'center';
        app.UniformRationaleLabel.Position = [151 133 59 22];
        app.UniformRationaleLabel.Text = 'Rationale';

        % Create UniformRationaleLabel_2
        app.UniformRationaleLabel_2 = uilabel(app.UniformDistributionPanel);
        app.UniformRationaleLabel_2.HorizontalAlignment = 'center';
        app.UniformRationaleLabel_2.Position = [23 133 76 22];
        app.UniformRationaleLabel_2.Text = 'Dist. Fraction';

        % Create EpochDropDown_2Label
        app.EpochDropDown_2Label = uilabel(app.AssessementCriteriaPanel);
        app.EpochDropDown_2Label.HorizontalAlignment = 'right';
        app.EpochDropDown_2Label.Position = [310 194 51 22];
        app.EpochDropDown_2Label.Text = 'Epoch';

        % Create CriteriaEpochDropDown
        app.CriteriaEpochDropDown = uidropdown(app.AssessementCriteriaPanel);
        app.CriteriaEpochDropDown.Items = {};
        app.CriteriaEpochDropDown.ValueChangedFcn = createCallbackFcn(app, @CriteriaEpochDropDownValueChanged, true);
        app.CriteriaEpochDropDown.Enable = 'off';
        app.CriteriaEpochDropDown.Position = [368 193 190 24];
        app.CriteriaEpochDropDown.Value = {};

        % Create ExitButton
        app.ExitButton = uibutton(app.TMST, 'push');
        app.ExitButton.ButtonPushedFcn = createCallbackFcn(app, @ExitButtonPushed, true);
        app.ExitButton.Position = [96 23 100 22];
        app.ExitButton.Text = 'Exit';

        % Create ASSESSMENTCONFIGURATIONLabel
        app.ASSESSMENTCONFIGURATIONLabel = uilabel(app.TMST);
        app.ASSESSMENTCONFIGURATIONLabel.FontSize = 14;
        app.ASSESSMENTCONFIGURATIONLabel.FontWeight = 'bold';
        app.ASSESSMENTCONFIGURATIONLabel.Position = [507 315 225 22];
        app.ASSESSMENTCONFIGURATIONLabel.Text = 'ASSESSMENT CONFIGURATION';

        % Create Panel
        app.Panel = uipanel(app.TMST);
        app.Panel.BackgroundColor = [1 1 1];
        app.Panel.Position = [21 61 237 83];

        % Create LoadButton
        app.LoadButton = uibutton(app.Panel, 'push');
        app.LoadButton.ButtonPushedFcn = createCallbackFcn(app, @LoadButtonPushed, true);
        app.LoadButton.Position = [69 47 100 22];
        app.LoadButton.Text = 'Load';

        % Create SaveButton
        app.SaveButton = uibutton(app.Panel, 'push');
        app.SaveButton.ButtonPushedFcn = createCallbackFcn(app, @SaveButtonPushed, true);
        app.SaveButton.Enable = 'off';
        app.SaveButton.Position = [10 14 100 22];
        app.SaveButton.Text = 'Save';

        % Create SaveAsButton
        app.SaveAsButton = uibutton(app.Panel, 'push');
```

```
            app.SaveAsButton.ButtonPushedFcn = createCallbackFcn(app, @SaveAsButtonPushed, true);
            app.SaveAsButton.Position = [126 14 100 22];
            app.SaveAsButton.Text = 'Save As';

            % Create TargetMarginSizingToolLabel
            app.TargetMarginSizingToolLabel = uilabel(app.TMST);
            app.TargetMarginSizingToolLabel.FontName = 'Arial';
            app.TargetMarginSizingToolLabel.FontSize = 16;
            app.TargetMarginSizingToolLabel.FontWeight = 'bold';
            app.TargetMarginSizingToolLabel.Position = [21 294 200 22];
            app.TargetMarginSizingToolLabel.Text = 'Target Margin Sizing Tool';

            % Create EDLSystemsToolBoxLabel
            app.EDLSystemsToolBoxLabel = uilabel(app.TMST);
            app.EDLSystemsToolBoxLabel.FontWeight = 'bold';
            app.EDLSystemsToolBoxLabel.Position = [21 315 136 22];
            app.EDLSystemsToolBoxLabel.Text = 'EDL Systems Tool Box';

            % Create Panel_2
            app.Panel_2 = uipanel(app.TMST);
            app.Panel_2.BackgroundColor = [0.4667 0.6745 0.1882];
            app.Panel_2.Position = [21 166 237 98];

            % Create Label
            app.Label = uilabel(app.Panel_2);
            app.Label.HorizontalAlignment = 'right';
            app.Label.Enable = 'off';
            app.Label.Position = [34 56 25 22];
            app.Label.Text = '';

            % Create CriticalMetricDropDown
            app.CriticalMetricDropDown = uidropdown(app.Panel_2);
            app.CriticalMetricDropDown.Items = {};
            app.CriticalMetricDropDown.Enable = 'off';
            app.CriticalMetricDropDown.Position = [16 49 205 33];
            app.CriticalMetricDropDown.Value = {};

            % Create RunAnalysisButton
            app.RunAnalysisButton = uibutton(app.Panel_2, 'push');
            app.RunAnalysisButton.ButtonPushedFcn = createCallbackFcn(app, @RunAnalysisButtonPushed, true);
            app.RunAnalysisButton.Enable = 'off';
            app.RunAnalysisButton.Position = [66 14 100 22];
            app.RunAnalysisButton.Text = 'Run Analysis';

            % Create AssessmentLabel
            app.AssessmentLabel = uilabel(app.TMST);
            app.AssessmentLabel.Position = [23 273 74 22];
            app.AssessmentLabel.Text = 'Assessment:';

            % Create Label_2
            app.Label_2 = uilabel(app.TMST);
            app.Label_2.Position = [95 273 160 22];
            app.Label_2.Text = '  ';

            % Show the figure after all components are created
            app.TMST.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = StartWindow_v10

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.TMST)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.TMST)
        end
    end
end
```

Appendix B TMST Code – Assessment Main Window

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Target Margin Assessment Tool    %
% R. Grover                        %
% 5/4/2021                         %
% AssessmentWindow_v8.mapp         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

classdef AssessmentWindow_v8 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        AssessmentWindow  matlab.ui.Figure
        BasePanel         matlab.ui.container.Panel
        MetricNameLabel   matlab.ui.control.Label
        MetricLabel       matlab.ui.control.Label
        CloseButton       matlab.ui.control.Button
        EpochTabGroup     matlab.ui.container.TabGroup
        Tab1              matlab.ui.container.Tab
        SaveButton        matlab.ui.control.Button
        LoadButton        matlab.ui.control.Button
        MarginImage       matlab.ui.control.Image
    end

    properties (Access = private)

    end

    properties (Access = public)
        RedColor = [1, 0, 0];                       % Define red background color
        YellowColor = [1, 1, 0];                    % Define yellow background color
        GreenColor = [0, 1, 0];                     % Define green background color
        CallingApp;                                 % Handle to calling app
        var_limit = 10                              % Limit of the number of variables in regress model
        epoch_tabs(1,10);                           % Handles for each epoch tab
        load_panel(1,10);                           % Panel per tab for load button
        uncertainty_panel(1,10);                    % Panel per tab for uncertianty classification
        category_panel(10,10);                      % Panels per tab for assessment categories
        category_title_bar(1,10)                    % Panel per tab for assessment categories bar
        load_button(1,10);                          % Load button per tab
        regression_model_files;                     % Array of regression model names
        saved_regression_model_files;               % Array of saved regression model names
        regression_label(1,10);                     % Regression label per tab
        regression_model_file_label(1,10);          % Regression model file label per tab
        epoch_label(1,10);                          % Epoch label per tab
        current_tab = 1;                            % Track which is active tab
        var_cnt(1,10);                              % Number of regression model variables, per tab
        max_var_cnt;                                % Maximum input variable count
        var_name;                                   % Names of input variables
        var_data(10,66,3);                          % Variable indices and coefficients
        var_data_sorted(10,66,3);                   % Variable indices and coefficients sorts
        term_cnt(1,10);                             % Number of terms (without const term) in regression model
        regress_heading_label(1,10);                % Label per tab of regression model section
        uncertainty_heading_label(1,10);            % Label per tab of uncertainty section
        category_heading_label(10,10);              % Labels per tab per category section
        category_weighting_label (10,10);           % Label per tab per category for category weighting
        category_bar_label(1,10);                   % Label for category bar
        input_cat_assess_panel(10,10,10);           % Colored panels for input confidence assessments
        input_cat_assess_dropdown(10,10,10);        % Dropdown menu for high, medium and low
        dist_frac_editfield(10,10,10);              % Edit field for distribution fraction
        rationale_textarea(10,10,10);               % Assessment rationale edit field
        dist_type_dropdown(10,10);                  % Uncertainty distribution type dropdown menu
        epistemic_editfield(10,10);                 % Edit field for epistemic fraction
        aleatory_editfield(10,10);                  % Edit field for aleatory fraction
        uncertainty_type_label(1,10);               % Label for distribution type column
        uncertainty_split_label(1,10);              % Label for uncertainty split
        uncertainty_epistemic_label(1,10);          % Label for epistemic uncertainty column
        uncertainty_aleatory_label(1,10);           % Label for aleatory uncertainty column
        cat_confidence_label(10,10);                % Label for category confidence column
        cat_dist_frac_label(10,10);                 % Label for catetegy distribution fraction
        cat_rationale_label(10,10);                 % Label for category confidence rationale
        scroll_panel(1,10);                         % Panel for scrolling across assessment categories
        results_panel(1,10);                        % Panel for all the results objects
        results_heading_label(1,10);                % Label for results section heading
        results_color_panel(10,66);                 % Color panel for results edit fields
        ave_dist_frac_editfield(10,66);             % Edit field for average distribution fraction
        contribution_editfield(10,66);              % Edit field for term contribtuion to target margin
        contribution_units_label(10,66);            % Array of labels for units
        epistemic_cont_editfield(10,66);            % Edit field for epistemic contribution
        aleatory_cont_editfield(10,66);             % Edit field for aleatory contribution
        combined_editfield(10,66);                  % Edit field for combined epistemic/aleatory contribution
        target_margin_result_panel(1,10);           % Panel that contains target margin results objects
        target_margin_editfield(1,10);              % Edit field for target margin
        epistemic_tar_margin_editfield(1,10);       % Edit field for epistemic target margin
        aleatory_tar_margin_editfield(1,10);        % Edit field for aleatory target margin
        combined_tar_margin_editfield(1,10);        % Edit field for combined epistemic/aleatory target margin
        ave_dist_frac_label(1,10);                  % Average distribution fraction column label
        contribution_label(1,10);                   % Contribution column label
        epistemic_label(2,10);                      % Epistemic column label
        aleatory_label(2,10);                       % Aleatory column label
        combined_label(2,10);                       % Combined column label
        contribution_split_label(2,10);             % Label for uncertainty split
        total_target_margin_label(1,10);            % Label for total target margin
        total_target_margin_units(1,10)             % Units label for target margin
        dist_frac_select(10,10);                    % Distribution fraction selected per input
        cat_lock_switch(10,10);                     % Switch to lock category assessments
        uncertainty_lock_switch(1,10);              % Switch to lock uncertain classification
        regression_terms_editfield(10,66);          % Edit fields for numeric terms of regression model
        regress_term_names_panel(10,66);            % Panels for regression model term names
        regress_term_ind_editfield(10,66,2);        % Edit fields for regression model indices
        regress_term_names_editfield(10,66,2);      % Edit field for term names
        tornado_ax(1,10);                           % Axes object for tornado plot
        regress_names_header_label(1,10);           % Regression term names header labels
        regress_values_header_label(1,10);          % Regression term values header labels
        regress_tornado_header_label(1,10);         % Regression tornado header labels
        epoch_metric_limit_label(1,10);             % Epoch metric limit labels
        epoch_metric_limit_editfield(1,10);         % Epoch metric limit editfield
        milestone_span_switch(1,10);                % Milestone or time span switch
        target_margin_component(10,66);             % Array to store target margin components
        target_margin_component_epi(10,66);         % Array to store target margin components
        target_margin_component_alt(10,66);         % Array to store target margin components
        target_margin_component_cmb(10,66);         % Array to store target margin components
        target_margin_total(1,10);                  % Total target margin per epoch
        weighted_ave_frac(10,10);                   % Weighted average distribution fraction per input
        weighting_sum;                              % Sum of category weight for each epoch
        metric_limit_line_panel(1,10);              % Panel for metric limit line
        metric_limit_line_label(1,10);              % Label for metric limit line
        target_margin_span_panel(1,10);             % Panels for taget margin span
        target_margin_span_label(1,10);             % Label for taget margin span
        target_limit_line_panel(1,10);              % Panel for target limit line
        target_limit_line_label(1,10);              % Label for target limit line
        margin_grapics_epoch_label(1,10);           % Label for epochs in margin graphic
        margin_graphics_header_label;               % Label for heading over margin graphics
        cat_assess_boundaries(2,2);                 % Boundaries for assessing margin score color
        target_limit_values(1,10);                  % Array holding target limit values
        metric_limit_values(1,10);                  % Array holding metric limit values
        saved_metric_limit_values(1,10);            % Array holidng saved metric limit values
        saved_epochs;                               % Epochs saved in metric config file
        saved_epochs_cnt;                           % Number of epochs saved in metrics config file
        saved_epoch_type;                           % Array saved type of epoch: milestone or span
        epoch_type;                                 % Array type of epoch: milestone or span
        saved_var_cnt(1,10);                        % Saved number of regression model input variables per epoch
        saved_uncertainty_lock;                     % Saved uncertainty type lock state
        uncertainty_lock;                           % Uncertainty type lock state
        saved_uncertainty_type;                     % Saved uncertainty type
        uncertainty_type;                           % Uncertainty type
        saved_uncertainty_split(10,10);             % Saved fraction uncertainty split (epistemic)
        uncertainty_split(10,10);                   % Fraction uncertainty split (epistemic)
        saved_categories;                           % Categories saved in metric config file
        saved_categories_cnt;                       % Number of categories saved in metric config file
        saved_category_lock;                        % Saved category lock state
        category_lock;                              % Sategory lock state
        saved_var_confidence;                       % Saved input variable confidence rating
        var_confidence;                             % Input variable confidence rating
        saved_var_confidence_rationale;             % Saved input variable confidence rating rationale
        var_confidence_rationale;                   % Input variable confidence rating rationale
        compare_app;                                % Handle to comparison app
        call_compare_msg_flag = false;              % Flag to call comparison dialog box
        proceed_flag = false;                       % Flag returned from compare dialog box to signal proceed
        current_metric_name;                        % Names of metric being assessed
        current_metric_index;                       % Index of metric being assessed
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% WINODW DIMENSIONS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        width_seperator = 30;                       % Pixel width of separation between panels
        height_border = 45;                         % Pixel height of border
        width_regression_area = 890;                % Pixel width of regression model area
        height_regression_header = 106;             % Pixel height of regression model header area
        width_uncertainty_area = 300;               % Pixel width of uncertainty type area
        width_category_area = 525;                  % Pixel width of assessment category area
        width_tabs;                                 % Pixel width of tabs
        height_tabs;                                % Pixel height of tabs
        height_input_row = 51;                      % Pixel height of input row
        height_category_header = 152;               % Pixel height of header of category assessment area
        height_category_bar = 45;                   % Pixel height of category title bar
        left_uncertainty_panel = 935;               % Left anchor of uncertianty panel
        width_results_area = 557;                   % Pixel width of results area
        height_margin_span = 92;                    % Height of target margin span
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    end


    methods (Access = private)

        % Call file open dialog and load regression model
        function cLoadButtonPushed(app)

            regression_model_path = sprintf('%s/%s',app.CallingApp.current_path,strrep(app.current_metric_name,' ','_'));
            % Get file open dialog box and open, if not cancelled
            [app.regression_model_files{app.current_tab},...
```

```
             regression model path] = uigetfile(...
                             {'*.rmd', 'Margin Analysis Files (*.rmd)';...
                              '*.*',  'All Files (*.*)'}, ...
                              'Load Regression Model');

            if (app.regression_model_files{app.CallingApp.metric_index,app.current_tab} ~= 0)

                 load_regression_model(app);

            end

            set(app.regression_label(app.current_tab),'Text','Fred');
        end


        % Function to open and load assessment configuration file
        function load_metric_config_file(app,metric_name,metric_index)

            % PLEASE NOTE
            % !!! Need to come back and figure out how to handle when epochs
            % !!! and categories have changed since config file was saved.

            % Set compare msg flag
            app.call_compare_msg_flag = false;

            % Open metric config file
            file_path = sprintf('%s%s\\%s',app.CallingApp.current_path,strrep(metric_name,' ','_'),...
                                app.CallingApp.metric_config_file{metric_index});

            fid = fopen(file_path,'r');

            % Read file header
            fgetl(fid);

            % Read section header
            fgetl(fid);
            % Reader in number of epochs
            app.saved_epochs_cnt = str2num(fgetl(fid));

            % Check number of epochs
            if (app.CallingApp.number_of_epochs ~= app.saved_epochs_cnt)
                app.call_compare_msg_flag = true;
            end

            % Initialize string cells
            app.saved_epochs = cell(app.saved_epochs_cnt);
            app.saved_regression_model_files = cell(app.saved_epochs_cnt);
            app.saved_epoch_type = cell(app.saved_epochs_cnt);

            % Read section header
            fgetl(fid);
            % Read in epoch names
            for i=1:app.saved_epochs_cnt
                app.saved_epochs{i} = fgetl(fid);
            end

            % Read section header
            fgetl(fid);
            % Read in epoch type
            for i=1:app.saved_epochs_cnt
                app.saved epoch type{i} = fgetl(fid);
            end

            % Read section header
            fgetl(fid);
            % Read in regression model file names
            for i=1:app.saved epochs cnt
                app.saved_regression_model_files{i} = fgetl(fid);
            end

            % Read section header
            fgetl(fid);
            % Read in regression model variable count
            for i=1:app.saved_epochs_cnt
                app.saved_var_cnt(i) = str2num(fgetl(fid));
            end

            % Read section header
            fgetl(fid);
            % Read in metric limit values per epoch
            for i=1:app.saved_epochs_cnt
                app.saved_metric_limit_values(i) = str2num(fgetl(fid));
            end

            % Check epoch names are the same in saved and config
            temp =
            ismember(app.CallingApp.epoch_names(1:app.CallingApp.number_of_epochs),app.saved_epochs(1:app.saved_epochs_cnt)
            );
            if (sum(temp) ~= app.CallingApp.number_of_epochs)
```

```
               app.call compare msg flag = true;
            end

            % Read section header
            fgetl(fid);
            % Reader in number of categories
            app.saved categories cnt = str2num(fgetl(fid));

            % Initialize category cells
            app.saved_categories = cell(app.saved_categories_cnt);

            % Check number of categories
            if (app.CallingApp.number of categories ~= app.saved categories cnt)
                app.call_compare_msg_flag = true;
            end

            % Read section header
            fgetl(fid);
            % Read in regression category names
            for i=1:app.saved_categories_cnt
                app.saved_categories{i} = fgetl(fid);
            end

            % Check category names are the same in saved and config
            temp =
            ismember(app.CallingApp.category_names(1:app.CallingApp.number_of_categories),app.saved_categories(1:app.saved_
            categories_cnt));
            if (sum(temp) ~= app.CallingApp.number_of_categories)
               app.call_compare_msg_flag = true;
            end

            % Call comparison dialog to check for consistency
            if (app.call_compare_msg_flag)
                app.compare_app  = compare_msg_box_v2(app, app.CallingApp);
                waitfor(app.compare_app);
                if (app.proceed_flag)
                    disp('I am back - proceed');
                end
            end

            % Initialize string variables
            app.saved_uncertainty_lock = cell(app.saved_epochs_cnt);
            app.saved uncertainty type = cell(app.saved epochs cnt,10);
            app.saved_category_lock = cell(app.saved_epochs_cnt, app.saved_categories_cnt);
            app.saved_var_confidence = cell(app.saved_epochs_cnt, app.saved_categories_cnt,10);
            app.saved_var_confidence_rationale = cell(app.saved_epochs_cnt, app.saved_categories_cnt,10);


            % Load assessment configuration for each epoch
            for i=1:app.saved_epochs_cnt

                if (~strcmp(app.saved_regression_model_files{i},'NULL'))

                    % Read section header
                    fgetl(fid);
                    % Read Uncertainty Type Lock State
                    app.saved_uncertainty_lock{i} = fgetl(fid);

                    % Read section header
                    fgetl(fid);
                    % Read uncertainty types
                    for j=1:app.saved_var_cnt(i)
                        app.saved_uncertainty_type{i,j} = fgetl(fid);
                    end

                    % Read section header
                    fgetl(fid);
                    % Read uncertainty splits
                    for j=1:app.saved_var_cnt(i)
                        app.saved_uncertainty_split(i,j) = str2num(fgetl(fid));
                    end

                    for j=1:app.saved categories cnt

                        % Read section header
                        fgetl(fid);
                        % Read catergory lock state
                        app.saved_category_lock{i,j} = fgetl(fid);

                        % Read section header
                        fgetl(fid);
                        % Read variable confidence values
                        for k=1:app.saved_var_cnt(i)
                            app.saved_var_confidence{i,j,k} = fgetl(fid);
                        end

                        % Read section header
                        fgetl(fid);
                        for k=1:app.saved_var_cnt(i)
```

```
                    line_cnt = str2num(fgetl(fid));
                    temp = cell(line_cnt);
                    for n=1:line_cnt
                      temp{n,1} = fgetl(fid);
                    end
                    app.saved_var_confidence_rationale{i,j,k} = temp(:,1);
                end

            end

        end

    end

    % Close file
    fclose(fid);

    % Copy over saved regression model file names
    for i=1:app.CallingApp.number_of_epochs

         % Check to see if saved epoch is part of current configuration copy epoch data if so
         [dumby,saved_index] = ismember(app.CallingApp.epoch_names(i),app.saved_epochs(1:app.saved_epochs_cnt));
         if (saved_index ~= 0)

             % Copy saved regression model file names
             app.regression_model_files{i} = app.saved_regression_model_files{saved_index};

             % Copy over regression model variable count
             app.var_cnt(i) = app.saved_var_cnt(saved_index);

         end

    end

end


% Function for laoding regression model from file
function load_regression_model(app,metric_name,metric_index)

    regression_model_path = sprintf('%s/%s/%s',app.CallingApp.current_path,strrep(metric_name,' ','_'),...
                                    app.regression_model_files{app.current_tab});
    fid = fopen(regression_model_path,'r');

    % Reader in header
    S = fgetl(fid);

    % Read in number of variables
    app.var_cnt(app.current_tab) = str2num(fgetl(fid));

    % initialize counter
    cnt = 0;

    % Read in constant term
    cnt = cnt + 1;
    temp = split(fgetl(fid));
    app.var_data(app.current_tab,cnt,1) = str2num(temp{1});
    app.var_data(app.current_tab,cnt,2) = str2num(temp{2});
    app.var_data(app.current_tab,cnt,3) = str2double(temp{3});

    % Read in first order terms and variable names
    for i=1:app.var_cnt(app.current_tab)
        % Parse variable info
        cnt = cnt + 1;
        temp = split(fgetl(fid));
        app.var_data(app.current_tab,cnt,1) = str2num(temp{1});
        app.var_data(app.current_tab,cnt,2) = str2num(temp{2});
        app.var_data(app.current_tab,cnt,3) = str2double(temp{3});
        app.var_name{i,app.current_tab} = temp{4};
    end

    % Read data for interaction coefficient terms
    for i=1:(app.var_cnt(app.current_tab)-1)
        for j=i+1:app.var_cnt(app.current_tab)
            cnt = cnt + 1;
            temp = split(fgetl(fid));
            app.var_data(app.current_tab,cnt,1) = str2num(temp{1});
            app.var_data(app.current_tab,cnt,2) = str2num(temp{2});
            app.var_data(app.current_tab,cnt,3) = str2double(temp{3});
        end
    end

    % Read in second order terms
    for i=1:app.var_cnt(app.current_tab)
        % Parse variable info
        cnt = cnt + 1;
        temp = split(fgetl(fid));
        app.var_data(app.current_tab,cnt,1) = str2num(temp{1});
        app.var_data(app.current_tab,cnt,2) = str2num(temp{2});
        app.var_data(app.current_tab,cnt,3) = str2double(temp{3});
    end

    % Close regression model file
    fclose(fid);

    % Set row counter
    app.term_cnt(app.current_tab) = cnt - 1;

end

% Sort terms of regression model
function sort_terms(app,tab_index)

        % Sort variables (linear terms)
        temp_data = app.var_data(tab_index,2:app.var_cnt(tab_index)+1,1:3);
        [temp, sort_index] = sort(abs(temp_data(1,:,3)'),'descend');
        app.var_data_sorted(tab_index,1:app.var_cnt(tab_index),1:3) =...
            [temp_data(1,sort_index,1)', temp_data(1,sort_index,2)', temp_data(1,sort_index,3)'];


        % Sort other terms (nonlinear terms)
        temp_data = app.var_data(tab_index,app.var_cnt(tab_index)+2:app.term_cnt(tab_index)+1,1:3);
        [temp, sort_index] = sort(abs(temp_data(1,:,3)'),'descend');
        app.var_data_sorted(tab_index,app.var_cnt(tab_index)+1:app.term_cnt(tab_index),1:3) =...
            [temp_data(1,sort_index,1)', temp_data(1,sort_index,2)', temp_data(1,sort_index,3)'];

end


% Populate names in editfields of regression model display
function populate_term_names(app,tab_index)

    % Populate names and index columns
    for i=1:app.term_cnt(tab_index)
        if (app.var_data_sorted(tab_index,i,1) == 0) ||...
           (app.var_data_sorted(tab_index,i,1) == app.var_data_sorted(tab_index,i,2))

           app.CallingApp.category_names{app.var_data_sorted(tab_index,i,2)};

           set(app.regress_term_names_editfield(tab_index,i,2),'Position',[44,1,257,42],...
               'Value',app.var_name{app.var_data_sorted(tab_index,i,2),tab_index});
           set(app.regress_term_ind_editfield(tab_index,i,1),'Value',app.var_data_sorted(tab_index,i,1));
           set(app.regress_term_ind_editfield(tab_index,i,2),'Value',app.var_data_sorted(tab_index,i,2));

        else

           set(app.regress_term_names_editfield(tab_index,i,1),'Position',[44,21,257,22],...
               'Value',app.var_name{app.var_data_sorted(tab_index,i,1),tab_index});
           set(app.regress_term_names_editfield(tab_index,i,2),'Position',[44,1,257,22],...
               'Value',app.var_name{app.var_data_sorted(tab_index,i,2),tab_index});
           set(app.regress_term_ind_editfield(tab_index,i,1),'Value',app.var_data_sorted(tab_index,i,1));

           set(app.regress_term_ind_editfield(tab_index,i,2),'Value',app.var_data_sorted(tab_index,i,2));


        end

    end

end


% Function to create tornado plot
function make_tornado_plot(app, tab_index)

        % Set bar width
        BarWidth = 0.4;

        % Set high side / low side values
        HighSide = flip(app.var_data_sorted(tab_index,1:app.term_cnt(tab_index),3));
        LowSide = flip(-1*app.var_data_sorted(tab_index,1:app.term_cnt(tab_index),3));

        % Create tornado plot
        app.tornado_ax(tab_index) = uiaxes('Parent', app.load_panel(tab_index),'Interactions',[]);
        barh(app.tornado_ax(tab_index),HighSide,BarWidth);
        hold(app.tornado_ax(tab_index),'on');
        barh(app.tornado_ax(tab_index),LowSide,BarWidth,'r');
        set(app.tornado_ax(tab_index),'Position',[461 8 400 app.height_input_row*app.term_cnt(tab_index)+90]);
        xlabel(app.tornado_ax(tab_index),'Change in G-Load (Earth g''s)')
        grid(app.tornado_ax(tab_index),'minor')

end


% Function to update target margin fields upon change or at startup
function update_target_margin(app,tab_index)

    % Compute weighed average of distrution fractions per inputs
```

```
            for n=1:app.var_cnt(tab_index)
                app.weighted_ave_frac(tab_index,n) = 0;
                for m=1:app.CallingApp.number_of_categories
                    app.weighted_ave_frac(tab_index,n) = app.weighted_ave_frac(tab_index,n) +...

                        app.CallingApp.category_weight(tab_index,m)*get(app.dist_frac_editfield(tab_index,m,n),'Value');

                end

                % Calculate weighted average
                app.weighted_ave_frac(tab_index,n) = app.weighted_ave_frac(tab_index,n)/app.weighting_sum(tab_index,1);

                % Map weighted average to DOE interval depending on distribution type
                S = get(app.dist_type_dropdown(tab_index,n),'Value');
                if strcmp(S,'Normal')
                    app.weighted_ave_frac(tab_index,n) = (1/3)*app.weighted_ave_frac(tab_index,n);
                elseif strcmp(S,'Uniform')
                    app.weighted_ave_frac(tab_index,n) = 2*app.weighted_ave_frac(tab_index,n);
                end

                % Set values to edit fields
                set(app.ave_dist_frac_editfield(tab_index,n),'Value',app.weighted_ave_frac(tab_index,n));

            end

            % Compute indivdual components of target margin
            for n=1:app.term_cnt(tab_index)


                % If linear terms
                if (app.var_data_sorted(tab_index,n,1) == 0)

                    temp1 = abs(app.var_data_sorted(tab_index,n,3))*1;
                    temp2 = abs(app.var_data_sorted(tab_index,n,3))*(1 + app.weighted_ave_frac(tab_index,n));
                    app.target_margin_component(tab_index,n) = temp2 - temp1;

                    % Compute uncertainty type contribution fractions
                    epistemic_frac = get(app.epistemic_editfield(tab_index,n),'Value');
                    aleatory_frac =  get(app.aleatory_editfield(tab_index,n),'Value');
                    combined_frac = 0.0;


                % If second order terms
                elseif (app.var_data_sorted(tab_index,n,1) == app.var_data_sorted(tab_index,n,2))

                    int_index1 = 0;
                    for m=1:app.var_cnt(tab_index)
                        if (app.var_data_sorted(tab_index,n,1) == app.var_data_sorted(tab_index,m,2))
                            int_index1 = m;
                        end
                    end
                    temp1 = abs(app.var_data_sorted(tab_index,n,3))*1^2;
                    temp2 = abs(app.var_data_sorted(tab_index,n,3))*(1 + app.weighted_ave_frac(tab_index,int_index1))^2;
                    app.target_margin_component(tab_index,n) = temp2 - temp1;

                    % Compute uncertainty type contribution fractions
                    epistemic_frac = get(app.epistemic_editfield(tab_index,int_index1),'Value')^2;
                    aleatory_frac =  get(app.aleatory_editfield(tab_index,int_index1),'Value')^2;
                    combined_frac =
                    2*get(app.epistemic_editfield(tab_index,int_index1),'Value')*get(app.aleatory_editfield(tab_index,int_index1),'Value');

                % If interactive terms
                else

                    % Find correct variable
                    int_index1 = 0;
                    int_index2 = 0;
                    for m=1:app.var_cnt(tab_index)
                        if (app.var_data_sorted(tab_index,n,1) == app.var_data_sorted(tab_index,m,2))
                            int_index1 = m;
                        end
                        if (app.var_data_sorted(tab_index,n,2) == app.var_data_sorted(tab_index,m,2))
                            int_index2 = m;
                        end
                    end
                    temp1 = abs(app.var_data_sorted(tab_index,n,3))*1*1;
                    temp2 = abs(app.var_data_sorted(tab_index,n,3))*(1 + app.weighted_ave_frac(tab_index,int_index1))*(1 + app.weighted_ave_frac(tab_index,int_index2));
                    app.target_margin_component(tab_index,n) = temp2 - temp1;

                    % Compute uncertainty type contribution fractions
                    epistemic_frac =
                    get(app.epistemic_editfield(tab_index,int_index1),'Value')*get(app.epistemic_editfield(tab_index,int_index2),'Value');
                    aleatory_frac =
                    get(app.aleatory_editfield(tab_index,int_index1),'Value')*get(app.aleatory_editfield(tab_index,int_index2),'Value');
                    combined_frac =
                    get(app.epistemic_editfield(tab_index,int_index1),'Value')*get(app.aleatory_editfield(tab_index,int_index2),'Value') +...

                                    get(app.epistemic_editfield(tab_index,int_index2),'Value')*get(app.aleatory_editfield(tab_index,int_index1),'Value');

                end

                % Set value to target margin component edit field
                set(app.contribution_editfield(tab_index,n),'Value', app.target_margin_component(tab_index,n));

                % Set conttribution edit fields
                if (app.var_data_sorted(tab_index,n,3) ~= 0)
                    set(app.epistemic_cont_editfield(tab_index,n),'Value',sprintf('%4.1f%%',epistemic_frac*100));
                    set(app.aleatory_cont_editfield(tab_index,n),'Value',sprintf('%4.1f%%',aleatory_frac*100));
                    set(app.combined_editfield(tab_index,n),'Value',sprintf('%4.1f%%',combined_frac*100));
                end
                app.target_margin_component_epi(tab_index,n) = epistemic_frac*app.target_margin_component(tab_index,n);
                app.target_margin_component_alt(tab_index,n) = aleatory_frac*app.target_margin_component(tab_index,n);
                app.target_margin_component_cmb(tab_index,n) = combined_frac*app.target_margin_component(tab_index,n);

            end

            % Compute total target margin and set to edit field
            app.target_margin_total(tab_index) = sum(app.target_margin_component(tab_index,:));
            set(app.target_margin_editfield(tab_index),'Value',app.target_margin_total(tab_index));
            set(app.epistemic_tar_margin_editfield(tab_index),'Value',...

                sprintf('%4.1f%%',100*(sum(app.target_margin_component_epi(tab_index,:))/get(app.target_margin_editfield(tab_index),'Value'))));
            set(app.aleatory_tar_margin_editfield(tab_index),'Value',...

                sprintf('%4.1f%%',100*(sum(app.target_margin_component_alt(tab_index,:))/get(app.target_margin_editfield(tab_index),'Value'))));
            set(app.combined_tar_margin_editfield(tab_index),'Value',...

                sprintf('%4.1f%%',100*(sum(app.target_margin_component_cmb(tab_index,:))/get(app.target_margin_editfield(tab_index),'Value'))));

            % Update target margin color panels
            for n=1:app.var_cnt(tab_index)
                if (app.weighted_ave_frac(tab_index,n) <=
                app.cat_assess_boundaries(app.dist_frac_select(tab_index,n),1))
                    %set(app.results_color_panel(tab_index,n), 'BackgroundColor','green');
                    set(app.results_color_panel(tab_index,n), 'BackgroundColor','[0.94, 0.94, 0.94]');
                elseif (app.weighted_ave_frac(tab_index,n) <=
                app.cat_assess_boundaries(app.dist_frac_select(tab_index,n),2))
                    %set(app.results_color_panel(tab_index,n), 'BackgroundColor','yellow');
                    set(app.results_color_panel(tab_index,n), 'BackgroundColor','[0.94, 0.94, 0.94]');
                else
                    %set(app.results_color_panel(tab_index,n), 'BackgroundColor','red');
                    set(app.results_color_panel(tab_index,n), 'BackgroundColor','[0.94, 0.94, 0.94]');
                end
            end

        end


        % Write out metric configuration file
        function save_metric_config_file(app)

            % Open metric config file
            file_path = sprintf('%s%s\\%s',app.CallingApp.current_path,strrep(app.current_metric_name,' ','_'),...
                                app.CallingApp.metric_config_file{app.current_metric_index});

            % Open metric config file
            fid = fopen(file_path,'w');

            % Write file header
            fprintf(fid,'%% .MCF file containing configuration data metric assessment\n');

            % Write out section heading
            fprintf(fid,'=== Number of Epochs ===\n');
            % Write out number of epochs
            fprintf(fid,'%d\n',app.CallingApp.number_of_epochs);

            % Write out section heading
            fprintf(fid,'=== Epoch Names ===\n');
            for i=1:app.CallingApp.number_of_epochs
                fprintf(fid,'%s\n',app.CallingApp.epoch_names{i});
            end

            % Write out section heading
            fprintf(fid,'=== Epoch Duration ===\n');
            % Write epoch duration types
            for i=1:app.CallingApp.number_of_epochs
                S = get(app.milestone_span_switch(i),'Value');
                fprintf(fid,'%s\n',S);
            end
```

```

            % Write out section heading
            fprintf(fid,'=== Regression Model File for Each Epoch =====\n');
            % Write out regression model names
            for i=1:app.CallingApp.number_of_epochs
                fprintf(fid,'%s\n',app.regression_model_files{i});
            end

            % Write out section heading
            fprintf(fid,'=== Regression Model Number of Inputs ===\n');
            % Write out regression model number of inputs
            for i=1:app.CallingApp.number_of_epochs
                fprintf(fid,'%d\n',app.var cnt(i));
            end

            % Write out section heading
            fprintf(fid,'=== Metric Limit for Each Epoch ===\n');
            % Write out metric limits for each epoch
            for i=1:app.CallingApp.number of epochs
                S = get(app.epoch_metric_limit_editfield(i),'Value');
                fprintf(fid,'%f\n',S);
            end

            % Write out section heading
            fprintf(fid,'=== Number of Categories ===\n');
            % Write out number of categories
            fprintf(fid,'%d\n',app.CallingApp.number_of_categories);

            % Write out section heading
            fprintf(fid,'=== Category Names ===\n');
            for i=1:app.CallingApp.number of categories
                fprintf(fid,'%s\n',app.CallingApp.category_names{i});
            end

            % Save assessment configuration for each epoch
            for i=1:app.CallingApp.number_of_epochs

                if (~strcmp(app.regression_model_files{i},'NULL'))

                    % Save section header
                    fprintf(fid,'=== %s Uncertainty Type Lock ===\n',app.CallingApp.epoch_names{i});
                    % Save Uncertainty Type Lock State
                    S = get(app.uncertainty lock switch(i),'Value');
                    fprintf(fid,'%s\n',S);

                    % Save section header
                    fprintf(fid,'=== %s Input Variable Distribution ===\n',app.CallingApp.epoch_names{i});
                    % Save uncertainty types
                    for j=1:app.var cnt(i)
                        S = get(app.dist_type_dropdown(i,j),'Value');
                        fprintf(fid,'%s\n',S);
                    end

                    % Save section header
                    fprintf(fid,'=== %s Input Type Split (Epistemic) ===\n',app.CallingApp.epoch names{i});
                    % Save uncertainty splits
                    for j=1:app.var_cnt(i)
                        S = get(app.epistemic_editfield(i,j),'Value');
                        fprintf(fid,'%f\n',S);
                    end

                    for j=1:app.CallingApp.number_of_categories

                        % Save section header
                        fprintf(fid,'=== %s %s Lock ===\n',app.CallingApp.epoch_names{i},app.CallingApp.category_names{j});
                        % Save catergory lock state
                        S = get(app.cat lock switch(i,j),'Value');
                        fprintf(fid,'%s\n',S);

                        % Save section header
                        fprintf(fid,'=== %s %s Confidence
                        ===\n',app.CallingApp.epoch_names{i},app.CallingApp.category_names{j});
                        % Save variable confidence values
                        for k=1:app.var_cnt(i)
                            S = get(app.input_cat_assess_dropdown(i,j,k),'Value');
                            fprintf(fid,'%s\n',S);
                        end

                        % Save section header
                        fprintf(fid,'=== %s %s Rationale
                        ===\n',app.CallingApp.epoch_names{i},app.CallingApp.category_names{j});
                        % Save category confidence rationale
                        for k=1:app.var_cnt(i)
                            line_cnt = length(get(app.rationale_textarea(i,j,k),'Value'));
                            fprintf(fid,'%d\n',line cnt);
                            S = get(app.rationale_textarea(i,j,k),'Value');
                            for n=1:line_cnt
                                fprintf(fid,'%s\n',S{n});
                            end
                        end

                    end

                end

            end

            % Close metric config file
            fclose(fid)

        end


        % Function to initiaze all variable
        function initialize_variables(app)

            app.epoch_type = cell(app.CallingApp.number_of_epochs);
            app.regression model files = cell(app.CallingApp.number of epochs);
            app.uncertainty_lock = cell(app.CallingApp.number_of_epochs);
            app.uncertainty_type = cell(app.CallingApp.number_of_epochs,10);
            app.var_confidence = cell(app.CallingApp.number_of_epochs,app.CallingApp.number_of_categories,10);
            app.var_confidence_rationale = cell(app.CallingApp.number_of_epochs,app.CallingApp.number_of_categories,10);
            app.category_lock = cell(app.CallingApp.number_of_epochs,app.CallingApp.number_of_categories);
            for i=1:app.CallingApp.number of epochs
                app.epoch_type{i} = 'Span';
                app.regression_model_files{i} = 'NULL';
                app.var_cnt = 0;
                app.metric_limit_values(i) = 0;
                app.uncertainty_lock{i} = 'Unlock';
                for j=1:app.CallingApp.number of categories
                    app.category_lock{i,j} = 'Unlock';
                    for k=1:10
                        app.var_confidence{i,j,k} = 'High';
                        app.var_confidence_rationale{i,j,k} = {'<Enter Rationale>'};
                    end
                end
            end
        end

    end

    methods (Access = public)

    end


    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app, startwindowapp, MetricName, MetricIndex)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DO BASIC TAB SETUP  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            % Store main app object
            app.CallingApp = startwindowapp;

            % Set metric info
            app.current_metric_name = MetricName;
            app.current metric index = MetricIndex;

            % Initialize contraol state variable
            initialize_variables(app);

            % Load metric configuration file
            load metric config file(app,MetricName,MetricIndex)

            % Set metric name to Assessment Window
            app.AssessmentWindow.Name = MetricName;
            app.MetricNameLabel.Text = MetricName;

           % Compute sums of category weights
           app.weighting_sum = zeros(app.CallingApp.number_of_epochs,1);
           for i=1:app.CallingApp.number_of_epochs
             app.weighting_sum(i,1) = sum(app.CallingApp.category_weight(i,1:app.CallingApp.number_of_categories));
           end

           % Initize target margin componenets
           app.target_margin_component = zeros(10,66);
           app.target_margin_component_epi = zeros(10,66);
           app.target_margin_component_alt = zeros(10,66);
           app.target_margin_component_cmb = zeros(10,66);


           % Initialize distribution fraction selection
           app.dist_frac_select = zeros(10,10) + 1;

           % Compute assessment boundaries for target margin column
```

```
            app.cat_assess_boundaries(1,2) = app.CallingApp.dist_frac(1,1,3) + (app.CallingApp.dist_frac(1,1,2) -
            app.CallingApp.dist_frac(1,1,3))/2;
            app.cat_assess_boundaries(1,1) = app.CallingApp.dist_frac(1,1,2) + (app.CallingApp.dist_frac(1,1,1) -
            app.CallingApp.dist_frac(1,1,2))/2;
            app.cat_assess_boundaries(2,2) = app.CallingApp.dist_frac(1,2,3) + (app.CallingApp.dist_frac(1,2,2) -
            app.CallingApp.dist_frac(1,2,3))/2;
            app.cat_assess_boundaries(2,1) = app.CallingApp.dist_frac(1,2,2) + (app.CallingApp.dist_frac(1,2,1) -
            app.CallingApp.dist_frac(1,2,2))/2;

            % Compute tab sizes
            app.width_tabs = app.width_seperator +...
                             app.width_regression_area +...
                             app.width_seperator +...
                             app.width_uncertainty_area +...
                             app.width_seperator +...
                             app.CallingApp.number_of_categories*app.width_category_area +...
                             app.CallingApp.number_of_categories*app.width_seperator+...
                             app.width_results_area +...
                             app.width_seperator;

            % Size base panel
            app.BasePanel.Position(3) = app.width_tabs;

            % Size tab group
            app.EpochTabGroup.Position(3) = app.width_tabs;

            % Add epoch tabs to the assessment window
            delete(app.Tab1);
            for i=1:startwindowapp.number_of_epochs
                % Create tab
                app.epoch_tabs(i) = uitab(app.EpochTabGroup,'Title',
                startwindowapp.epoch_names{i},'BackgroundColor',[0.851 0.9098 0.9412]);
                % Create Epoch label on tab
                app.epoch_label(i) =
                uilabel('Parent',app.epoch_tabs(i),'FontName','Arial','FontSize',16,'FontWeight','Bold',...

                    'FontColor',[0,0.45,0.74],'Position',[app.width_seperator,app.EpochTabGroup.Position(4)-52,350,22],'Tex
                    t',startwindowapp.epoch_names{i});
                % Place epoch metric limit editfield
                app.epoch_metric_limit_editfield(i) = uieditfield('numeric','Parent',app.epoch_tabs(i),...
                                    'Position',[500,...
                                                app.EpochTabGroup.Position(4) - 54,...
                                                70, 22],'ValueDisplayFormat', '%.2f','ValueChangedFcn', @(app,event)
                                                MetricLimitChanged(i));
                % Place epoch metric limit label
                app.epoch_metric_limit_label(i) = uilabel('Parent',app.epoch_tabs(i),...
                                    'Position',[500 - 210,...
                                                app.EpochTabGroup.Position(4) - 56, 200, 22],...
                                                'HorizontalAlignment','right',...
                                                'FontSize',16,...
                                                'FontColor','red',...
                                                'FontWeight','bold',...
                                                'Text','Metric Limit:');
                % Add milestone lock switch
                app.milestone_span_switch(i) = uiswitch('slider','Parent',app.epoch_tabs(i),'Position',...
                                 [500 + 150,app.EpochTabGroup.Position(4) - 52, 35, 20],'Items',{'Milestone' 'Span'},...
                                 'FontColor','red','ValueChangedFcn',@(app,event)
                                 DurationSwitchChanged(i));
                % Create regression model load and info area
                app.load_panel(i) = uipanel('Parent',app.epoch_tabs(i),'Title','','BackgroundColor',[0.910 0.969
                1],'BorderType','none',...
                    'Position',[app.width_seperator,...
                                app.EpochTabGroup.Position(4) - app.height_regression_header - app.height_border - 24,...
                                app.width_regression_area,...
                                app.height_regression_header]);
                app.regress_heading_label(i) =
                uilabel('Parent',app.load_panel(i),'FontName','Arial','FontSize',16,'FontWeight','Bold','HorizontalAlignmen
                t','Center',...

                    'Position',[(app.width_regression_area-175)/2,app.height_regression_header-32,175,22],'Text','Regressio
                    n Model');
                app.load_button(i) = uibutton('Parent',app.load_panel(i),"Text",'Load Regression
                Model','Position',[45,(app.height_regression_header-22)/2,...
                    142,22],'ButtonPushedFcn',@(load_button,event) cLoadButtonPushed(app));
                app.regression_label(i) =
                uilabel('Parent',app.load_panel(i),'FontName','Arial','FontSize',14,'Position',...

                    [app.width_regression_area/2,app.height_regression_header-62,80,22],'HorizontalAlignment','Right','Text
                    ','Loaded File:');
                app.regression_model_file_label(i) =
                uilabel('Parent',app.load_panel(i),'FontName','Arial','FontSize',14,'Position',...

                    [app.width_regression_area/2+83,app.height_regression_header-62,350,22],'HorizontalAlignment','Left','T
                    ext','File Name');
                % Create model uncertainty panel area
                app.uncertainty_panel(i) =
                uipanel('Parent',app.epoch_tabs(i),'Title','','BackgroundColor','white','BorderType','none',...
                    'Position',[app.width_seperator + app.width_regression_area + app.width_seperator,...
                                app.EpochTabGroup.Position(4) - app.height_regression_header - app.height_border - 24,...
                                app.width_uncertainty_area,...
                                app.height_regression_header]);
                app.uncertainty_heading_label(i) =
                uilabel('Parent',app.uncertainty_panel(i),'FontName','Arial','FontSize',16,'FontWeight','Bold','HorizontalA
                lignment','Center',...

                    'Position',[(app.width_uncertainty_area-150)/2,app.height_regression_header-32,150,22],'Text','Uncertai
                    nty Type');
                % Create scrolling panel
                %scroll_panel_width = app.CallingApp.number_of_categories*app.width_category_area +
                (app.CallingApp.number_of_categories - 1)*app.width_seperator;
                scroll_panel_width = app.EpochTabGroup.Position(3) - (2.5*app.width_seperator + app.width_regression_area
                + app.width_uncertainty_area);
                app.scroll_panel(i) = uipanel('Parent',app.epoch_tabs(i),'Title','','BackgroundColor',[0.808 0.863
                0.886],'BorderType','none','Scrollable','on',...
                    'Position',[app.width_seperator + app.width_regression_area + app.width_seperator +
                    app.width_uncertainty_area + app.width_seperator/2,...
                                1,...
                                scroll_panel_width,...
                                app.EpochTabGroup.Position(4)-24]);
                % Create category panels
                for j=1:app.CallingApp.number_of_categories
                    app.category_panel(i,j) =
                    uipanel('Parent',app.scroll_panel(i),'Title','','BackgroundColor','white','BorderType','none',...
                                'Position',[app.width_seperator/2 + (j - 1)*(app.width_category_area +
                                app.width_seperator),...
                                app.EpochTabGroup.Position(4) - app.height_regression_header - app.height_border - 24,...
                                app.width_category_area,...
                                app.height_regression_header]);
                    app.category_heading_label(i,j) =
                    uilabel('Parent',app.category_panel(i,j),'FontName','Arial','FontSize',16,'FontWeight','Bold','Horizont
                    alAlignment','Center',...

                                'Position',[(app.width_category_area-300)/2,(app.height_regression_header-app.height_catego
                                ry_bar-22),300,22],'Text',app.CallingApp.category_names(j));
                    app.category_weighting_label(i,j) =
                    uilabel('Parent',app.category_panel(i,j),'FontName','Arial','FontSize',14,'HorizontalAlignment','Center
                    ','FontColor',[0,0.45,0.74],...

                                'Position',[(app.width_category_area-300)/2,(app.height_regression_header-app.height_catego
                                ry_bar-40),300,22],'Text',sprintf('Weighting: %3.2f',app.CallingApp.category_weight(i,j)));
                end

                % Create spanning category title bar
                bar_width = app.CallingApp.number_of_categories*app.width_category_area +
                (app.CallingApp.number_of_categories - 1)*app.width_seperator;
                app.category_title_bar(i) =
                uipanel('Parent',app.scroll_panel(i),'Title','','BackgroundColor','white','BorderType','none',...
                                'Position',[app.width_seperator/2,...
                                app.EpochTabGroup.Position(4) - app.height_category_bar - app.height_border - 24,...
                                bar_width,...
                                app.height_category_bar]);
                app.category_bar_label(i) =
                uilabel('Parent',app.category_title_bar(i),'FontName','Arial','FontSize',16,'FontWeight','Bold','Horizontal
                Alignment','center',...
                    'Position',[(bar_width-300)/2,app.height_category_bar-32,300,22],'Text','ASSESSMENT CATEGORIES');

                % Create results panel
                app.results_panel(i) =
                uipanel('Parent',app.scroll_panel(i),'Title','','BackgroundColor','white','BorderType','none',...
                    'Position',[(app.CallingApp.number_of_categories + 0.5)*app.width_seperator +
                    app.CallingApp.number_of_categories*app.width_category_area,...
                                app.EpochTabGroup.Position(4) - app.height_regression_header - app.height_border - 24,...
                                app.width_results_area,...
                                app.height_regression_header]);
                app.results_heading_label(i) =
                uilabel('Parent',app.results_panel(i),'FontName','Arial','FontSize',16,'FontWeight','Bold','HorizontalAlign
                ment','center',...
                    'Position',[(app.width_results_area-200)/2,app.height_regression_header-32,200,22],'Text','Target
                    Margin Sizing');

            end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            % Initialize variable name array
            app.var_name = cell(app.var_limit,app.CallingApp.epoch_number_limit);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DO REGRESSION MODEL SETUP  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            for i=1:app.CallingApp.number_of_epochs
                if (strcmp(app.regression_model_files{i},'NULL'))
                    set(app.regression_model_file_label(i),'Text', '');
                    app.var_cnt(i) = 0;
                else

                    % Assign input count
                    max_inputs(i) = app.var_cnt(i);

```

```
                    % Set model name to regression model label
                    set(app.regression_model_file_label(i),'Text', app.regression_model_files(i));

                    % Load regression model from file
                    app.current_tab = i;

                    load_regression_model(app,MetricName,MetricIndex);


                end
            end

            % Set max input variable count
            [app.max_var_cnt, maxIndex] = max(app.var_cnt);

            % Set up tabs with active regression models
            if (app.max_var_cnt > 0)

                % Compute tab height based on maximum number of inputs
                set(app.BasePanel, 'Position', [1,1,app.EpochTabGroup.Position(3),...
                    192 + 24 + app.height_input_row*app.term_cnt(maxIndex) + 2*app.height_border + 222]);
                app.MetricLabel.Position(2) = app.BasePanel.Position(4) - 40;
                app.MetricNameLabel.Position(2) = app.BasePanel.Position(4) - 40;
                app.CloseButton.Position(2) = app.BasePanel.Position(4) - 70;
                app.LoadButton.Position(2) = app.BasePanel.Position(4) - 122;
                app.SaveButton.Position(2) = app.BasePanel.Position(4) - 153;

                % Move margin image
                app.MarginImage.Position(2) = app.BasePanel.Position(4) - 121;

                set(app.EpochTabGroup, 'Position', ([1,1,app.EpochTabGroup.Position(3),...
                    192 + app.height_input_row*app.term_cnt(maxIndex) + 2*app.height_border + 24]));

                for i=1:app.CallingApp.number_of_epochs


                    set(app.epoch_label(i),'Position',[app.width_seperator,app.EpochTabGroup.Position(4)-57,350,22]);

                    % Place epoch metric limit editfield
                    set(app.epoch_metric_limit_editfield(i),...
                                    'Position',[500,...
                                                app.EpochTabGroup.Position(4) - 54, 70,
                                                22]);
                    % Place epoch metric limit label
                    set(app.epoch_metric_limit_label(i),...
                                    'Position',[500 - 210, app.EpochTabGroup.Position(4) - 56, 200,
                                    22])
                    % Add milestone lock switch
                    set(app.milestone_span_switch(i),'Position',...
                                 [500 + 150,app.EpochTabGroup.Position(4) - 52, 35, 20]);

                    if (app.var_cnt(i) > 0)
                        set(app.load_panel(i),'Position',[app.width_seperator,...
                                (app.EpochTabGroup.Position(4)-24) - (192 + app.height_input_row*app.term_cnt(i) +
                                0.5*app.height_border) - app.height_border,...
                                app.width_regression_area,...
                                192 + app.height_input_row*app.term_cnt(i)+0.5*app.height_border]);
                        set(app.regress_heading_label(i),'Position',[(app.width_regression_area-175)/2,(192 +
                        app.height_input_row*app.term_cnt(i))-32,175,22]);
                        set(app.regression_label(i),'Position',...
                                [app.width_regression_area/2,(192 + app.height_input_row*app.term_cnt(i))-62,80,22]);
                        set(app.regression_model_file_label(i),'Position',...
                                [app.width_regression_area/2+83,(192 + app.height_input_row*app.term_cnt(i))-62,350,22]);
                        set(app.load_button(i),'Position',[45,((192 + app.height_input_row*app.term_cnt(i))-64),142,22]);
                        set(app.uncertainty_panel(i),'Position',[app.width_seperator + app.width_regression_area +
                        app.width_seperator,...
                                app.EpochTabGroup.Position(4) - (160 + app.height_input_row*app.var_cnt(i)) -
                                app.height_border - 24,...
                                app.width_uncertainty_area,...
                                160 + app.height_input_row*app.var_cnt(i)]);
                        set(app.uncertainty_heading_label(i), 'Position',...
                                [(app.width_uncertainty_area-150)/2,(160 +
                                app.height_input_row*app.var_cnt(i))-32,150,22]);
                        app.uncertainty_type_label(i) = uilabel('Parent',app.uncertainty_panel(i),'Position',[13,8 +
                        app.height_input_row*app.var_cnt(i),...

                                70,35],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text',sprintf('Dist
                                ribution\nType'));
                        app.uncertainty_split_label(i) = uilabel('Parent',app.uncertainty_panel(i),'Position',[179,22 +
                        app.height_input_row*app.var_cnt(i),...
                                35,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Split');
                        app.uncertainty_epistemic_label(i) = uilabel('Parent',app.uncertainty_panel(i),'Position',[115,8
                        + app.height_input_row*app.var_cnt(i),...

                                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Epistemic');

                        app.uncertainty_aleatory_label(i) = uilabel('Parent',app.uncertainty_panel(i),'Position',[208,8 +
                        app.height_input_row*app.var_cnt(i),...
                                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Aleatory');
```

```
                        % Create uncertainty panel child controls
                        for j=1:app.var_cnt(i)
                            % Create distribution type dropdown menu
                            app.dist_type_dropdown(i,j) = uidropdown('Parent',app.uncertainty_panel(i),'Position',[10
                            (160 + app.height_input_row*app.var_cnt(i)) - 185 - (j-1)*app.height_input_row 80 22],...
                                'Items',{'Normal','Uniform'},'ValueChangedFcn',@(app,event) DistDropDownChanged(i,j));
                            % Create epistemic edit field
                            app.epistemic_editfield(i,j) = uieditfield('numeric','Parent',app.uncertainty_panel(i),...
                                'Position',[116 (160 + app.height_input_row*app.var_cnt(i)) - 185 -
                                (j-1)*app.height_input_row 70 22],...
                                'Editable','on','ValueDisplayFormat','%.3f','Value',1.00,'Limits',[0
                                1],'ValueChangedFcn',@(app,event) EpistemicChanged(i,j));
                            % Create aleatory edit field
                            app.aleatory_editfield(i,j) = uieditfield('numeric','Parent',app.uncertainty_panel(i),...
                                'Position',[210 (160 + app.height_input_row*app.var_cnt(i)) - 185 -
                                (j-1)*app.height_input_row 70 22],...
                                'Editable','off','ValueDisplayFormat','%.3f','Value',0.00);
                        end

                        % Add uncertainty lock switch
                        app.uncertainty_lock_switch(i) = uiswitch('slider','Parent',app.uncertainty_panel(i),'Position',...
                                 [(app.width_uncertainty_area-35)/2, 75 + app.height_input_row*app.var_cnt(i), 35,
                                 20],'Items',{'Unlock' 'Lock'},...
                                 'ValueChangedFcn',@(app,event) UncertaintyLockChangedChanged(i));

                        % Reset scroll panel size
                        set(app.scroll_panel(i),'Position',...
                            [app.width_seperator + app.width_regression_area + app.width_seperator +
                            app.width_uncertainty_area + app.width_seperator/2,...
                             1,...
                             scroll_panel_width,...
                             app.EpochTabGroup.Position(4)-24]);

                        % Create category panels and child controls
                        for j=1:app.CallingApp.number_of_categories
                            set(app.category_panel(i,j),'Position',[app.width_seperator/2 + (j -
                            1)*(app.width_category_area + app.width_seperator),...
                                    app.EpochTabGroup.Position(4) - (160 + app.height_input_row*app.var_cnt(i)) -
                                    app.height_border - 24,...
                                    app.width_category_area,...
                                    160 + app.height_input_row*app.var_cnt(i)]);
                             set(app.category_heading_label(i,j),'Position',...
                                    [(app.width_category_area-300)/2,((160 +
                                    app.height_input_row*app.var_cnt(i))-app.height_category_bar-22),300,22]);
                             set(app.category_weighting_label(i,j),'Position',...
                                    [(app.width_category_area-300)/2,((160 +
                                    app.height_input_row*app.var_cnt(i))-app.height_category_bar-40),300,22]);
                             % Create column labels
                             app.cat_confidence_label(i,j) = uilabel('Parent',app.category_panel(i,j),'Position',[28,8 +
                             app.height_input_row*app.var_cnt(i),...

                                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Confidence')
                                ;
                             app.cat_dist_frac_label(i,j) = uilabel('Parent',app.category_panel(i,j),'Position',[125,8 +
                             app.height_input_row*app.var_cnt(i),...
                                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Dist
                                Frac');
                             app.cat_rationale_label(i,j) = uilabel('Parent',app.category_panel(i,j),'Position',[212,8 +
                             app.height_input_row*app.var_cnt(i),...

                                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Rationale');


                             % Add category lock switch
                             app.cat_lock_switch(i,j) = uiswitch('slider','Parent',app.category_panel(i,j),'Position',...
                                 [(app.width_category_area-35)/2, 45 + app.height_input_row*app.var_cnt(i), 35,
                                 20],'Items',{'Unlock' 'Lock'},...
                                 'ValueChangedFcn',@(app,event) CatLockChangedChanged(i,j));

                             for k=1:app.var_cnt(i)
                                % Create colored assessment panels
                                app.input_cat_assess_panel(i,j,k) =
                                uipanel('Parent',app.category_panel(i,j),'Title','','BackgroundColor','green','BorderType',
                                'none',...
                                    'Position',[20,(160 + app.height_input_row*app.var_cnt(i)) - 195 -
                                    (k-1)*app.height_input_row,...
                                     490,42]);
                                % Create assessment dropdown menu
                                app.input_cat_assess_dropdown(i,j,k) =
                                uidropdown('Parent',app.input_cat_assess_panel(i,j,k),'Position',[10 11 70 22]);

                                set(app.input_cat_assess_dropdown(i,j,k),'Items',{'High','Med','Low'},'ValueChangedFcn',@(a
                                pp,event) CatDropDownChanged(i,j,k));
                                % Create distribution fraction edit field
                                app.dist_frac_editfield(i,j,k) =
                                uieditfield('numeric','Parent',app.input_cat_assess_panel(i,j,k),'Position',[107 11 70
                                22],'Editable','off','Value',app.CallingApp.dist_frac(i,app.dist_frac_select(i,k),1));
                                % Create rationale edit field
                                app.rationale_textarea(i,j,k) =
```

```
                uitextarea('Parent',app.input cat assess panel(i,j,k),'Position',[192 5 288
                34],'Editable','on');
              end

        end
        set(app.category_title_bar(i), 'Position',[app.width_seperator/2,...
                app.EpochTabGroup.Position(4) - app.height category bar - app.height border - 24,...
                bar_width,...
                app.height_category_bar]);

        % Resize results area panel
        set(app.results_panel(i),'Position',...
            [(app.CallingApp.number of categories + 0.5)*app.width seperator +
            app.CallingApp.number_of_categories*app.width_category_area,...
                (app.EpochTabGroup.Position(4)-24) - (150 + app.height_input_row*app.term_cnt(i) +
                0.5*app.height_border) - app.height_border,...
                app.width_results_area,...
                150 + app.height_input_row*app.term_cnt(i)+0.5*app.height_border]);
        set(app.results heading label(i),'Position',...
            [(app.width_results_area-200)/2,150 + app.height_input_row*app.term_cnt(i) +
            0.5*app.height_border - 32,200,22]);

        % Create results columns
        for j=1:app.term_cnt(i)
            if (j <= app.var cnt(i))
                %app.results_color_panel(i,j) =
                uipanel('Parent',app.results_panel(i),'Title','','BackgroundColor','green','BorderType','no
                ne',...
                app.results_color_panel(i,j) =
                uipanel('Parent',app.results_panel(i),'Title','','BackgroundColor',[0.94, 0.94,
                0.94],'BorderType','none',...
                    'Position',[18,(160 + app.height_input_row*app.term_cnt(i)) + 0.5*app.height_border -
                    205 - (j-1)*app.height_input_row,...
                    521,42]);
            else
                app.results_color_panel(i,j) =
                uipanel('Parent',app.results panel(i),'Title','','BackgroundColor',[0.94, 0.94,
                0.94],'BorderType','none',...
                    'Position',[18,(160 + app.height_input_row*app.term_cnt(i)) + 0.5*app.height_border -
                    205 - (j-1)*app.height_input_row,...
                    521,42]);
            end

            % Create average distribution fraction edit field
            if (j <= app.var_cnt(i))
                app.ave_dist_frac_editfield(i,j) =
                uieditfield('numeric','Parent',app.results_color_panel(i,j),'Position',[15 11 70
                22],'Editable','off');
            end
            % Create contribution edit field
            app.contribution_editfield(i,j) =
            uieditfield('numeric','Parent',app.results_color_panel(i,j),'Position',[115 11 70
            22],'Editable','off');
            % Create units labels
            app.contribution units label(i,j) =
            uilabel('Parent',app.results_color_panel(i,j),'Position',[190 11 70 22],...

                'HorizontalAlignment','left','FontName','Arial','FontSize',12,'Text',sprintf('%s',app.Calli
                ngApp.metric_units{app.CallingApp.metric_index}));
            % Create epistemic contribution edit field
            app.epistemic cont editfield(i,j) =
            uieditfield('Parent',app.results_color_panel(i,j),'Position',[265 11 70
            22],'Editable','off','HorizontalAlignment','center');
            % Create aleatory contribution edit field
            app.aleatory_cont_editfield(i,j) =
            uieditfield('Parent',app.results_color_panel(i,j),'Position',[352 11 70
            22],'Editable','off','HorizontalAlignment','center');
            % Create aleatory contribution edit field
            app.combined_editfield(i,j) =
            uieditfield('Parent',app.results_color_panel(i,j),'Position',[439 11 70
            22],'Editable','off','HorizontalAlignment','center');
        end

        app.ave_dist_frac_label(i) =
        uilabel('Parent',app.results_panel(i),'Position',[16,app.height_input_row*app.term_cnt(i) +
        0.5*app.height_border - 2,...

                100,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Confidence
                Score');
        app.contribution_label(i) =
        uilabel('Parent',app.results_panel(i),'Position',[128,app.height_input_row*app.term_cnt(i) +
        0.5*app.height_border - 2,...

                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Contribution
                ');
        app.epistemic_label(1,i) =
        uilabel('Parent',app.results_panel(i),'Position',[282,app.height_input_row*app.term_cnt(i) +
        0.5*app.height_border - 2,...

                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Epistemic');

        app.aleatory_label(1,i) =
        uilabel('Parent',app.results_panel(i),'Position',[367,app.height_input_row*app.term_cnt(i) +
        0.5*app.height_border - 2,...
                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Aleatory');
        app.combined label(1,i) =
        uilabel('Parent',app.results_panel(i),'Position',[454,app.height_input_row*app.term_cnt(i) +
        0.5*app.height_border - 2,...

                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Combined');

        app.contribution split label(1,i) = uilabel('Parent',app.results panel(i),'Position',[350,14 +
        app.height_input_row*app.term_cnt(i) + 0.5*app.height_border,...

                100,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Contributio
                n Split');

        % Create target margin results box
        app.target_margin_result_panel(i) =
        uipanel('Parent',app.results_panel(i),'Title','','BackgroundColor',[0.310, 0.722,
        1.0],'BorderType','none',...
                    'Position',[18,(150 + app.height_input_row*app.term_cnt(i)) + 0.5*app.height_border -
                    110,...
                    521,42]);
        app.target_margin_editfield(i) =
        uieditfield('numeric','Parent',app.target_margin_result_panel(i),'Position',[115 11 70
        22],'Editable','off');
        % Create units label
        app.total_target_margin_units(i) =
        uilabel('Parent',app.target margin result panel(i),'Position',[190 11 100 22],...

                'HorizontalAlignment','left','FontName','Arial','FontSize',12,'Text',sprintf('%s',app.Calli
                ngApp.metric_units{app.CallingApp.metric_index}));
        % Create epistemic contribution edit field
        app.epistemic_tar_margin_editfield(i) =
        uieditfield('Parent',app.target margin result panel(i),'Position',[265 11 70
        22],'Editable','off','HorizontalAlignment','center');
        % Create aleatory contribution edit field
        app.aleatory_tar_margin_editfield(i) =
        uieditfield('Parent',app.target_margin_result_panel(i),'Position',[352 11 70
        22],'Editable','off','HorizontalAlignment','center');
        % Create combine contribution edit field
        app.combined_tar_margin_editfield(i) =
        uieditfield('Parent',app.target_margin_result_panel(i),'Position',[439 11 70
        22],'Editable','off','HorizontalAlignment','center');
        % Create total target margin label
        app.total_target_margin_label(i) =
        uilabel('Parent',app.target margin result panel(i),'Position',[7 11 100 22],...

                'HorizontalAlignment','center','FontColor','blue','FontName','Arial','FontSize',14,'FontWei
                ght','bold','Text','Target Margin:');
        app.epistemic_label(2,i) =
        uilabel('Parent',app.results_panel(i),'Position',[282,app.height_input_row*app.term_cnt(i) +
        0.5*app.height border + 83,...

                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Epistemic');

        app.aleatory_label(2,i) =
        uilabel('Parent',app.results_panel(i),'Position',[367,app.height_input_row*app.term_cnt(i) +
        0.5*app.height border + 83,...
                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Aleatory');
        app.combined_label(2,i) =
        uilabel('Parent',app.results_panel(i),'Position',[454,app.height_input_row*app.term_cnt(i) +
        0.5*app.height_border + 83,...
                70,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Combined');
        app.contribution split label(1,i) = uilabel('Parent',app.results panel(i),'Position',[350,99 +
        app.height_input_row*app.term_cnt(i) + 0.5*app.height_border,...

                100,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Contributio
                n Split');

        % Create regression model term column
        for j=1:app.term_cnt(i)
            app.regress_term_names_panel(i,j) =
            uipanel('Parent',app.load_panel(i),'Title','','BackgroundColor',[0.941, 0.941,
            0.941],'BorderType','none',...
                    'Position',[18,(202 + app.height_input_row*app.term_cnt(i)) + 0.5*app.height_border -
                    205 - (j-1)*app.height input row,...
                    322,42]);

            app.regress_term_ind_editfield(i,j,1) =
            uieditfield('numeric','Parent',app.regress_term_names_panel(i,j),...
                    'HorizontalAlignment','center','Position',[1,21,22,22],...
                    'Editable','off');

            app.regress_term_ind_editfield(i,j,2) =
            uieditfield('numeric','Parent',app.regress_term_names_panel(i,j),...
                    'HorizontalAlignment','center','Position',[1,1,22,22],...
```

```
                                'Editable','off');

                        app.regress_term_names_editfield(i,j,1) =
                        uieditfield('Parent',app.regress_term_names_panel(i,j),...
                                'FontSize',12,'HorizontalAlignment','center','Position',[44,21,257,22],...
                                'Editable','on','Value',sprintf('Frank'));

                        app.regress_term_names_editfield(i,j,2) =
                        uieditfield('Parent',app.regress_term_names_panel(i,j),...
                                'FontSize',12,'HorizontalAlignment','center','Position',[44,1,257,22],...
                                'Editable','on','Value',sprintf('Harry'));

                        app.regression_terms_editfield(i,j) = uieditfield('numeric','Parent',app.load_panel(i),...
                                'Position',[364,(202 + app.height_input_row*app.term_cnt(i)) + 0.5*app.height_border
                                - 195 - (j-1)*app.height_input_row, 70, 22],...
                                'Editable','off');


                    end

                    % Place regression terms header
                    app.regress_names_header_label(i) = uilabel('Parent',app.load_panel(i),'Position',[89,70 +
                    app.height_input_row*app.term_cnt(i),...

                            200,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Regression
                            Terms');

                    % Place regression terms values header
                    app.regress_values_header_label(i) = uilabel('Parent',app.load_panel(i),'Position',[298,70 +
                    app.height_input_row*app.term_cnt(i),...

                            200,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Coefficent
                            Values');

                    % Place regression tornado header
                    app.regress_tornado_header_label(i) = uilabel('Parent',app.load_panel(i),'Position',[560,108 +
                    app.height_input_row*app.term_cnt(i),...

                            200,22],'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Regression
                            Tornado');


                    % Sort regression terms by magnitude
                    sort_terms(app,i);
                    for j=1:app.term_cnt(i)
                        set(app.regression_terms_editfield(i,j),'Value',app.var_data_sorted(i,j,3));
                    end

                    % Populate terms names
                    populate_term_names(app,i);

                    % Create plot
                    make_tornado_plot(app, i);

                    % Update assessment
                    update_target_margin(app,i);

                else
                    set(app.load_panel(i),'Position',[app.width_seperator,...
                            app.EpochTabGroup.Position(4) - app.height_regression_header - app.height_border - 24,...
                            app.width_regression_area,...
                            app.height_regression_header]);
                    set(app.uncertainty_panel(i),'Position',[app.width_seperator + app.width_regression_area +
                    app.width_seperator,...
                            app.EpochTabGroup.Position(4) - app.height_regression_header - app.height_border - 24,...
                            app.width_uncertainty_area,...
                            app.height_regression_header]);
                    for j=1:app.CallingApp.number_of_categories
                        set(app.category_panel(i,j),'Position',[app.width_seperator + app.width_regression_area +
                        app.width_seperator + app.width_uncertainty_area + app.width_seperator + (j -
                        1)*(app.width_category_area + app.width_seperator),...
                                app.EpochTabGroup.Position(4) - app.height_regression_header - app.height_border -
                                24,...
                                app.width_category_area,...
                                app.height_regression_header]);
                    end
                    set(app.category_title_bar(i), 'Position',[app.width_seperator + app.width_regression_area +
                    app.width_seperator + app.width_uncertainty_area + app.width_seperator,...
                            app.EpochTabGroup.Position(4) - app.height_category_bar - app.height_border - 24,...
                            bar_width,...
                            app.height_category_bar]);
                end

            end

        end

        % Create margin trend objects
        left_indent = 500;
        epoch_panel_width = (app.BasePanel.Position(3) - left_indent -
        app.width_seperator)/app.CallingApp.epoch_number_limit;

        for i=1:app.CallingApp.epoch_number_limit

            app.metric_limit_line_panel(i) =
            uipanel('Parent',app.BasePanel,'Title','','BackgroundColor','red','BorderType','none',...
                'Position',[left_indent + (i-1)*epoch_panel_width + 4,...
                            app.BasePanel.Position(4) - 63,...
                            epoch_panel_width - 8,...
                            5]);
             PP = get(app.metric_limit_line_panel(i),'Position');
             app.target_margin_span_panel(i) = uipanel('Parent',app.BasePanel,'Title','','BackgroundColor',[0.89,
             0.98, 0.48],'BorderType','none',...
                'Position',[left_indent + (i-1)*epoch_panel_width + 4,...
                            PP(2) - app.height_margin_span,...
                            epoch_panel_width - 8,...
                            app.height_margin_span]);
             PP = get(app.target_margin_span_panel(i),'Position');
             app.target_limit_line_panel(i) =
             uipanel('Parent',app.BasePanel,'Title','','BackgroundColor',[0.075,0.54,0.2],'BorderType','none',...
                'Position',[left_indent + (i-1)*epoch_panel_width + 4,...
                            PP(2) - 5,...
                            epoch_panel_width - 8,...
                            5]);
             PP = get(app.metric_limit_line_panel(i),'Position');
             app.metric_limit_line_label(i) = uilabel('Parent',app.BasePanel,'Position',[PP(1),...
                            PP(2) + PP(4),...
                            PP(3),...
                            22],...
                            'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Placehoder g');
             PP = get(app.target_margin_span_panel(i),'Position');
             app.target_margin_span_label(i) = uilabel('Parent',app.target_margin_span_panel(i),'Position',[0,...
                            (PP(4) - 22)/2,...
                            PP(3),...
                            22],...
                            'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Placehoder g');
             PP = get(app.target_limit_line_panel(i),'Position');
             app.target_limit_line_label(i) = uilabel('Parent',app.BasePanel,'Position',[PP(1),...
                            PP(2) - 22,...
                            PP(3),...
                            22],...
                            'HorizontalAlignment','center','FontName','Arial','FontSize',12,'Text','Placehoder
                            g');

               if (i <= app.CallingApp.number_of_epochs)
                    app.margin_grapics_epoch_label = uilabel('Parent',app.BasePanel,'Position',[PP(1),...
                            app.BasePanel.Position(4) - 212,...
                            PP(3),...
                            22],'HorizontalAlignment','center','FontName','Arial','FontSize',14,...
                            'FontWeight','bold','Text',app.CallingApp.epoch_names{i});
               end

              if (i > app.CallingApp.number_of_epochs)

                 set(app.metric_limit_line_panel(i),'Visible','off');
                 set(app.target_margin_span_panel(i),'Visible','off');
                 set(app.target_limit_line_panel(i),'Visible','off');
                 set(app.metric_limit_line_label(i),'Visible','off');
                 set(app.target_margin_span_label(i),'Visible','off');
                 set(app.target_limit_line_label(i),'Visible','off');

             end

        end

        % Place the graphics sections header
        PP = get(app.metric_limit_line_panel(1),'Position');
        app.margin_graphics_header_label = uilabel('Parent',app.BasePanel,'Position',[PP(1),...
                            app.BasePanel.Position(4) - 28,...
                            epoch_panel_width*app.CallingApp.number_of_epochs,...
                            22],'HorizontalAlignment','center','FontName','Arial','FontSize',14,...
                            'FontWeight','bold','Text','MARGIN TRACKING');

        % Reset current tab
        app.current_tab = 1;

        % Update margin tracking graphics
        update_margin_tracker();

        % Set all state variables to their saved values
        set_the_table(app);

%%%%%%%%%%%%%%% LOCAL CALLBACK FUNCTIONS  %%%%%%%%%%%%%%%%%%%%%%%

    % Callback function for category assessment dropdown menus
    function CatDropDownChanged(i,j,k)

        % Change panel color
```

```
                switch get(app.input_cat_assess_dropdown(i,j,k),'Value')
                    case 'Low'
                        set(app.input_cat_assess_panel(i,j,k),'BackgroundColor',app.RedColor);

                        set(app.dist_frac_editfield(i,j,k),'Value',app.CallingApp.dist_frac(i,app.dist_frac_select(i,k),3))
                        ;
                    case 'Med'
                        set(app.input_cat_assess_panel(i,j,k),'BackgroundColor',app.YellowColor);

                        set(app.dist_frac_editfield(i,j,k),'Value',app.CallingApp.dist_frac(i,app.dist_frac_select(i,k),2))
                        ;
                    case 'High'
                        set(app.input_cat_assess_panel(i,j,k),'BackgroundColor',app.GreenColor);

                        set(app.dist_frac_editfield(i,j,k),'Value',app.CallingApp.dist_frac(i,app.dist_frac_select(i,k),1))
                        ;
                    otherwise
                end

                % Update assessment
                update_target_margin(app,i);

                % Update margin tracking graphic
                update_margin_tracker();
        end


        % Callback function for distribution selection dropdown menu
        function DistDropDownChanged(i,j)

                % Change dropdown selection value
                switch get(app.dist_type_dropdown(i,j),'Value')
                    case 'Normal'
                        app.dist_frac_select(i,j) = 1;
                    case 'Uniform'
                        app.dist_frac_select(i,j) = 2;
                    otherwise
                end

                % Reset jth distribution fractions across categories
                for n=1:app.CallingApp.number_of_categories

                    switch get(app.input_cat_assess_dropdown(i,n,j),'Value')
                        case 'Low'

                            set(app.dist_frac_editfield(i,n,j),'Value',app.CallingApp.dist_frac(i,app.dist_frac_select(i,j)
                            ,3));
                        case 'Med'

                            set(app.dist_frac_editfield(i,n,j),'Value',app.CallingApp.dist_frac(i,app.dist_frac_select(i,j)
                            ,2));
                        case 'High'

                            set(app.dist_frac_editfield(i,n,j),'Value',app.CallingApp.dist_frac(i,app.dist_frac_select(i,j)
                            ,1));
                        otherwise
                    end
                end

                % Update assessment
                update_target_margin(app,i);

                % Update margin tracking graphic
                update_margin_tracker();

        end


        % Callback function for category lock switch
        function CatLockChangedChanged(i,j)

                for n=1:app.var_cnt(i)

                    if (strcmp(get(app.cat_lock_switch(i,j),'Value'),'Lock'))
                        % Disable controls
                        set(app.input_cat_assess_dropdown(i,j,n),'Enable',false);
                        set(app.rationale_textarea(i,j,n),'Enable',false);
                    else
                        % Enable controls
                        set(app.input_cat_assess_dropdown(i,j,n),'Enable',true);
                        set(app.rationale_textarea(i,j,n),'Enable',true);
                    end

                end
        end


        % Callback function for uncertainty lock switch
        function UncertaintyLockChangedChanged(i)

                for n=1:app.var_cnt(i)

                    if (strcmp(get(app.uncertainty_lock_switch(i),'Value'),'Lock'))
                        % Disable controls
                        set(app.dist_type_dropdown(i,n),'Enable',false);
                        set(app.epistemic_editfield(i,n),'Enable',false);
                        set(app.aleatory_editfield(i,n),'Enable',false);
                    else
                        % Enable controls
                        set(app.dist_type_dropdown(i,n),'Enable',true);
                        set(app.epistemic_editfield(i,n),'Enable',true);
                        set(app.aleatory_editfield(i,n),'Enable',true);
                    end

                end

        end


        % Callback function for epistemic fraction change
        function EpistemicChanged(i,j)

                % Update aleatory uncertainty
                set(app.aleatory_editfield(i,j),'Value', 1 - get(app.epistemic_editfield(i,j),'Value'));

                % Update target margin
                update_target_margin(app,i);

        end


        % Callback function for metric limit change
        function MetricLimitChanged(i)

            % Update metric limit value
            app.metric_limit_values(i) = get(app.epoch_metric_limit_editfield(i),'Value');

            % Update margin tracker
            update_margin_tracker();

        end


        % Callback function for epoch duration switch
        function DurationSwitchChanged(i)

            % Call for margin tracker update
            update_margin_tracker();

        end


        % Updates margin tracking graphics when something changes
        function update_margin_tracker()

            % Find largest target margin value
            max_target_margin = max(app.target_margin_total);

            for n=1:app.CallingApp.number_of_epochs

                temp_width = epoch_panel_width;
                add_width = 0;
                if (strcmp(get(app.milestone_span_switch(n),'Value'),'Milestone'))
                    add_width = temp_width*(3/8);
                    temp_width = temp_width/4;
                end

                 % Compute target limit
                 app.target_limit_values = app.metric_limit_values(n) - app.target_margin_total(n);

                 new_span_height = app.height_margin_span*(app.target_margin_total(n)/max_target_margin);
                 PP = get(app.metric_limit_line_panel(n),'Position');
                 set(app.target_margin_span_panel(n),'Position',[left_indent + (n-1)*epoch_panel_width + add_width + 4,...
                                PP(2) - new_span_height,...
                                temp_width - 8,...
                                new_span_height]);
                 PP = get(app.target_margin_span_panel(n),'Position');

                 set(app.target_limit_line_panel(n),'Parent',app.BasePanel,'Title','','BackgroundColor',[0.075,0.54,0.2],'B
                 orderType','none',...
                    'Position',[left_indent + (n-1)*epoch_panel_width + add_width + 4,...
                                PP(2) - 5,...
                                temp_width - 8,...
                                5]);
                 PP = get(app.target_margin_span_panel(n),'Position');
                 set(app.target_margin_span_label(n),...
                     'Position',[0,...
                                (PP(4) - 22)/2,...
```

```
                                PP(3),...
                                22],...
                    'Text',sprintf('%7.2f
                    %s',app.target_margin_total(n),app.CallingApp.metric_units{app.CallingApp.metric_index}))
                PP = get(app.target_limit_line_panel(n),'Position');
                set(app.target_limit_line_label(n),'Position',[PP(1),...
                                PP(2) - 22,...
                                PP(3),...
                                22],...
                                'Text',sprintf('%7.2f
                                %s',app.target_limit_values(n),app.CallingApp.metric_units{app.CallingApp.metric_index}));

                set(app.metric limit line label(n),'Text',sprintf('%7.2f
                %s',app.metric_limit_values(n),app.CallingApp.metric_units{app.CallingApp.metric_index}));


                set(app.metric_limit_line_panel(n),...
                    'Position',[left_indent + (n-1)*epoch_panel_width + add_width + 4,...
                                 app.BasePanel.Position(4) - 63,...
                                 temp_width - 8,...
                                 5]);

            end

        end


        % Set up the user interface for the Target Margin assessment
        function set_the_table(app)

            % Set up all assessment variables based on if saved epoch and category are incuded in current config
            for n=1:app.CallingApp.number_of_epochs

                % Check to see if saved epoch is part of current configuration copy epoch data if so
                [dumby,saved_epoch_index] =
                ismember(app.CallingApp.epoch_names(n),app.saved_epochs(1:app.saved_epochs_cnt));
                if (saved epoch index ~= 0)

                    % Copy over epoch type
                    app.epoch_type{n} = app.saved_epoch_type{saved_epoch_index};

                    % Copy saved epoch metric limt
                    app.metric limit values(n)  = app.saved metric limit values(saved epoch index);

                    % Copy over uncertainy lock
                    app.uncertainty_lock{n} = app.saved_uncertainty_lock{saved_epoch_index};

                    % Copy over uncertainty types & split
                    for m=1:app.var cnt(n)
                        app.uncertainty_type{n,m} = app.saved_uncertainty_type{saved_epoch_index,m};
                        app.uncertainty_split(n,m) = app.saved_uncertainty_split(saved_epoch_index,m);
                    end

                    % Check to see if saved category is part of current configuration, and if so copy
                    for m=1:app.CallingApp.number of categories
                       [dumby,saved_cat_index] =
                       ismember(app.CallingApp.category_names(m),app.saved_categories(1:app.saved_categories_cnt));
                       if (saved_cat_index)

                           % Copy over category lock state
                           app.category lock{n,m} = app.saved category lock{saved epoch index,saved cat index};

                           % Copy over variable confidence and rationale
                           for p=1:app.var_cnt(n)
                               app.var_confidence{n,m,p} = app.saved_var_confidence{saved_epoch_index,saved_cat_index,p};
                               app.var_confidence_rationale{n,m,p} =
                               app.saved var confidence rationale{saved epoch index,saved cat index,p};
                           end

                       end

                    end

                end

            end

            % Copy all settings to the GUI controls
            for n=1:app.CallingApp.number of epochs

                % Set epoch duration switch for each span
                set(app.milestone_span_switch(n),'Value',app.epoch_type{n});
                DurationSwitchChanged(n);

                % Set epoch metric limit value
                set(app.epoch_metric_limit_editfield(n),'Value',app.metric_limit_values(n));
                MetricLimitChanged(n);

                % Set regression model is loaded
                if (~strcmp(app.regression model files{n},'NULL'))

                   % Set uncertainty lock switch
                   set(app.uncertainty_lock_switch(n),'Value',app.uncertainty_lock{n});
                   UncertaintyLockChangedChanged(n);

                   % Set category locks
                   for m=1:app.CallingApp.number_of_categories
                       set(app.cat_lock_switch(n,m),'Value',app.category_lock{n,m});
                       CatLockChangedChanged(n,m);
                   end

                   % For each inout variable
                   for m=1:app.var_cnt(n)

                       % Set input var distribution types
                       set(app.dist_type_dropdown(n,m),'Value',app.uncertainty_type{n,m});
                       DistDropDownChanged(n,m);

                       % Set epistemic fraction
                       set(app.epistemic_editfield(n,m),'Value',app.saved_uncertainty_split(n,m));
                       EpistemicChanged(n,m)

                       % Set category input variable confidence
                       for p=1:app.CallingApp.number of categories

                            % Set category input variable confidence
                            set(app.input_cat_assess_dropdown(n,p,m),'Value',app.var_confidence{n,p,m});
                            CatDropDownChanged(n,p,m);

                            % Set confidence rationale
                            set(app.rationale_textarea(n,p,m),'Value',app.var_confidence_rationale{n,p,m});

                       end

                   end

                end

            end

        end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        end

        % Close request function: AssessmentWindow
        function AssessmentWindowCloseRequest(app, event)

            ReturnToSartWindow(app.CallingApp);

            delete(app);

        end

        % Callback function
        function LoadRegressionModelButtonPushed(app, event)


        end

        % Button pushed function: CloseButton
        function CloseButtonPushed(app, event)

            % Close assessment window
            close(app.AssessmentWindow);

        end

        % Selection change function: EpochTabGroup
        function EpochTabGroupSelectionChanged(app, event)

            selectedTab = app.EpochTabGroup.SelectedTab;

            % Select a tab
             app.current_tab = find(selectedTab == app.epoch_tabs);

        end

        % Button pushed function: SaveButton
        function SaveButtonPushed(app, event)

        save metric config file(app);

        end
    end
```

```
    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create AssessmentWindow and hide until all components are created
            app.AssessmentWindow = uifigure('Visible', 'off');
            app.AssessmentWindow.AutoResizeChildren = 'off';
            app.AssessmentWindow.Position = [100 100 1366 763];
            app.AssessmentWindow.Name = 'UI Figure';
            app.AssessmentWindow.CloseRequestFcn = createCallbackFcn(app, @AssessmentWindowCloseRequest, true);
            app.AssessmentWindow.Scrollable = 'on';

            % Create BasePanel
            app.BasePanel = uipanel(app.AssessmentWindow);
            app.BasePanel.AutoResizeChildren = 'off';
            app.BasePanel.BackgroundColor = [1 1 1];
            app.BasePanel.Position = [2 2 1364 762];

            % Create MetricNameLabel
            app.MetricNameLabel = uilabel(app.BasePanel);
            app.MetricNameLabel.FontName = 'Arial';
            app.MetricNameLabel.FontSize = 16;
            app.MetricNameLabel.Position = [72 705 217 22];
            app.MetricNameLabel.Text = 'Metric Name';

            % Create MetricLabel
            app.MetricLabel = uilabel(app.BasePanel);
            app.MetricLabel.HorizontalAlignment = 'right';
            app.MetricLabel.FontName = 'Arial';
            app.MetricLabel.FontSize = 16;
            app.MetricLabel.Position = [14 705 53 22];
            app.MetricLabel.Text = 'Metric:';

            % Create CloseButton
            app.CloseButton = uibutton(app.BasePanel, 'push');
            app.CloseButton.ButtonPushedFcn = createCallbackFcn(app, @CloseButtonPushed, true);
            app.CloseButton.Position = [35 673 100 22];
            app.CloseButton.Text = 'Close';

            % Create EpochTabGroup
            app.EpochTabGroup = uitabgroup(app.BasePanel);
            app.EpochTabGroup.AutoResizeChildren = 'off';
            app.EpochTabGroup.SelectionChangedFcn = createCallbackFcn(app, @EpochTabGroupSelectionChanged, true);
            app.EpochTabGroup.Position = [-1 -1 1365 572];

            % Create Tab1
            app.Tab1 = uitab(app.EpochTabGroup);
            app.Tab1.AutoResizeChildren = 'off';
            app.Tab1.Title = 'Tab1';
            app.Tab1.BackgroundColor = [0.851 0.9098 0.9412];

            % Create SaveButton
            app.SaveButton = uibutton(app.BasePanel, 'push');
            app.SaveButton.ButtonPushedFcn = createCallbackFcn(app, @SaveButtonPushed, true);
            app.SaveButton.Position = [35 588 100 22];
            app.SaveButton.Text = 'Save';

            % Create LoadButton
            app.LoadButton = uibutton(app.BasePanel, 'push');
            app.LoadButton.Position = [35 621 100 22];
            app.LoadButton.Text = 'Load';

            % Create MarginImage
            app.MarginImage = uiimage(app.BasePanel);
            app.MarginImage.Position = [315 627 147 72];
            app.MarginImage.ImageSource = 'margin_graphic.gif';

            % Show the figure after all components are created
            app.AssessmentWindow.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = AssessmentWindow v8(varargin)

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.AssessmentWindow)

            % Execute the startup function
            runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.AssessmentWindow)
        end
    end
end
```

Appendix C TMST Code – Compare Message Box

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Target Margin Assessment Tool    %
% R. Grover                        %
% 5/4/2021                         %
% compare_msg_box_v2.mapp          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

classdef compare_msg_box_v2 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                   matlab.ui.Figure
        OKButton                   matlab.ui.control.Button
        EpochsPanel                matlab.ui.container.Panel
        EpochsConfiguredPanel      matlab.ui.container.Panel
        EpochsLoadingPanel         matlab.ui.container.Panel
        AssessmentCategoriesPanel  matlab.ui.container.Panel
        CatConfiguredPanel         matlab.ui.container.Panel
        CatLoadingPanel            matlab.ui.container.Panel
        MetricAssessmentComparisonLabel  matlab.ui.control.Label
        Epochs1Label               matlab.ui.control.Label
        Cat1Label                  matlab.ui.control.Label
    end


    properties (Access = private)
        Property % Description

        config_epoch_list_panel(1,10);              % Panel for list of configured epochs
        config_cats_list_panel(1,10);               % Panel for list of configured categories
        saved_epoch_list_panel(1,10);               % Panel for list of saved epochs
        saved_cats_list_panel(1,10);                % Panel for list of saved categories

        config_epoch_list_label(1,10);              % Label for list of configured epochs
        config_cats_list_label(1,10);               % Label for list of configured categories
        saved_epoch_list_label(1,10);               % Label for list of saved epochs
        saved_cats_list_label(1,10);                % Label for list of saved categories


        width_list_background_panel = 275;           % Width of background list panel
        height_list_background_panel = 20;           % Height of background list panel
        AssessApp;
    end


    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app, saved_app, calling_app)

        % Assign the assessment app
        app.AssessApp = saved_app;

        % Set proceed flag
        app.AssessApp.proceed_flag = false;

        % Populate configured epochs
        list_top = app.EpochsConfiguredPanel.Position(4) - 20;
        if (list_top < calling_app.number_of_epochs*app.height_list_background_panel + 5)
            list_top = calling_app.number_of_epochs*app.height_list_background_panel + 5;
        end

        for i=1:calling_app.number_of_epochs

            app.config_epoch_list_panel(i) =
            uipanel('Parent',app.EpochsConfiguredPanel,'Title','','BackgroundColor','white','BorderType','none',...
                                  'Position',[1,list_top - app.height_list_background_panel*(i),...
                                   app.width_list_background_panel,app.height_list_background_panel]);

            app.config_epoch_list_label(i) =
            uilabel('Parent',app.config_epoch_list_panel(i),'FontName','Arial','FontSize',12,...

                    'Position',[5,1,app.width_list_background_panel-5,app.height_list_background_panel],'Text',calling_app.
                    epoch_names{i});
        end

        % Populate configured categories
        list_top = app.CatConfiguredPanel.Position(4) - 20;
        if (list_top < calling_app.number_of_categories*app.height_list_background_panel + 5)
            list_top = calling_app.number_of_categories*app.height_list_background_panel + 5;
        end

        for i=1:calling_app.number_of_categories

            app.config_cats_list_panel(i) =
            uipanel('Parent',app.CatConfiguredPanel,'Title','','BackgroundColor','white','BorderType','none',...
                                  'Position',[1,list_top - app.height_list_background_panel*(i),...
                                   app.width_list_background_panel,app.height_list_background_panel]);

            app.config_cats_list_label(i) =
            uilabel('Parent',app.config_cats_list_panel(i),'FontName','Arial','FontSize',12,...

                    'Position',[5,1,app.width_list_background_panel-5,app.height_list_background_panel],'Text',calling_app.
                    category_names{i});
        end

        % Find saved common epochs
        saved_common_epochs = ismember(saved_app.saved_epochs(1:saved_app.saved_epochs_cnt),...
            calling_app.epoch_names(1:calling_app.number_of_epochs));

        % Populate loading epochs
        list_top = app.EpochsLoadingPanel.Position(4) - 20;
        if (list_top < saved_app.saved_epochs_cnt*app.height_list_background_panel + 5)
            list_top = saved_app.saved_epochs_cnt*app.height_list_background_panel + 5;
        end

        for i=1:saved_app.saved_epochs_cnt

            app.saved_epoch_list_panel(i) =
            uipanel('Parent',app.EpochsLoadingPanel,'Title','','BackgroundColor','green','BorderType','none',...
                                  'Position',[1,list_top - app.height_list_background_panel*(i),...
                                   app.width_list_background_panel,app.height_list_background_panel]);

            if (~saved_common_epochs(i))
                set(app.saved_epoch_list_panel(i),'BackgroundColor','white');
            end

            app.saved_epoch_list_label(i) =
            uilabel('Parent',app.saved_epoch_list_panel(i),'FontName','Arial','FontSize',12,...

                    'Position',[5,1,app.width_list_background_panel-5,app.height_list_background_panel],'Text',saved_app.sa
                    ved_epochs{i});
        end

        % Find saved common categories
        saved_common_categories = ismember(saved_app.saved_categories(1:saved_app.saved_categories_cnt),...
            calling_app.category_names(1:calling_app.number_of_categories));

        % Populate loading categories
        list_top = app.CatLoadingPanel.Position(4) - 20;
        if (list_top < saved_app.saved_categories_cnt*app.height_list_background_panel + 5)
            list_top = saved_app.saved_categories_cnt*app.height_list_background_panel + 5;
        end

        for i=1:saved_app.saved_categories_cnt

            app.saved_cats_list_panel(i) =
            uipanel('Parent',app.CatLoadingPanel,'Title','','BackgroundColor','green','BorderType','none',...
                                  'Position',[1,list_top - app.height_list_background_panel*(i),...
                                   app.width_list_background_panel,app.height_list_background_panel]);

            if (~saved_common_categories(i))
                set(app.saved_cats_list_panel(i),'BackgroundColor','white');
            end

            app.saved_cats_list_label(i) =
            uilabel('Parent',app.saved_cats_list_panel(i),'FontName','Arial','FontSize',12,...

                    'Position',[5,1,app.width_list_background_panel-5,app.height_list_background_panel],'Text',saved_app.sa
                    ved_categories{i});
        end


        end

        % Button pushed function: OKButton
        function OKButtonPushed(app, event)

        % Set proceed flag
        app.AssessApp.proceed_flag = true;

        % Delete app
        delete(app);

        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [100 100 681 566];
            app.UIFigure.Name = 'UI Figure';
```

```
            % Create OKButton
            app.OKButton = uibutton(app.UIFigure, 'push');
            app.OKButton.ButtonPushedFcn = createCallbackFcn(app, @OKButtonPushed, true);
            app.OKButton.Position = [293 26 100 22];
            app.OKButton.Text = 'OK';

            % Create EpochsPanel
            app.EpochsPanel = uipanel(app.UIFigure);
            app.EpochsPanel.TitlePosition = 'centertop';
            app.EpochsPanel.Title = 'Epochs';
            app.EpochsPanel.BackgroundColor = [1 1 1];
            app.EpochsPanel.FontName = 'Arial';
            app.EpochsPanel.FontWeight = 'bold';
            app.EpochsPanel.FontSize = 14;
            app.EpochsPanel.Position = [67 304 551 166];

            % Create EpochsConfiguredPanel
            app.EpochsConfiguredPanel = uipanel(app.EpochsPanel);
            app.EpochsConfiguredPanel.TitlePosition = 'centertop';
            app.EpochsConfiguredPanel.Title = 'CONFIGURED';
            app.EpochsConfiguredPanel.BackgroundColor = [1 1 1];
            app.EpochsConfiguredPanel.FontName = 'Arial';
            app.EpochsConfiguredPanel.Scrollable = 'on';
            app.EpochsConfiguredPanel.Position = [0 0 276 147];

            % Create EpochsLoadingPanel
            app.EpochsLoadingPanel = uipanel(app.EpochsPanel);
            app.EpochsLoadingPanel.TitlePosition = 'centertop';
            app.EpochsLoadingPanel.Title = 'LOADING';
            app.EpochsLoadingPanel.BackgroundColor = [1 1 1];
            app.EpochsLoadingPanel.FontName = 'Arial';
            app.EpochsLoadingPanel.Scrollable = 'on';
            app.EpochsLoadingPanel.Position = [275 0 276 147];

            % Create AssessmentCategoriesPanel
            app.AssessmentCategoriesPanel = uipanel(app.UIFigure);
            app.AssessmentCategoriesPanel.TitlePosition = 'centertop';
            app.AssessmentCategoriesPanel.Title = 'Assessment Categories';
            app.AssessmentCategoriesPanel.BackgroundColor = [1 1 1];
            app.AssessmentCategoriesPanel.FontName = 'Arial';
            app.AssessmentCategoriesPanel.FontWeight = 'bold';
            app.AssessmentCategoriesPanel.FontSize = 14;
            app.AssessmentCategoriesPanel.Position = [67 75 551 166];

            % Create CatConfiguredPanel
            app.CatConfiguredPanel = uipanel(app.AssessmentCategoriesPanel);
            app.CatConfiguredPanel.TitlePosition = 'centertop';
            app.CatConfiguredPanel.Title = 'CONFIGURED';
            app.CatConfiguredPanel.BackgroundColor = [1 1 1];
            app.CatConfiguredPanel.FontName = 'Arial';
            app.CatConfiguredPanel.Scrollable = 'on';
            app.CatConfiguredPanel.Position = [0 0 276 147];

            % Create CatLoadingPanel
            app.CatLoadingPanel = uipanel(app.AssessmentCategoriesPanel);
            app.CatLoadingPanel.TitlePosition = 'centertop';
            app.CatLoadingPanel.Title = 'LOADING';
            app.CatLoadingPanel.BackgroundColor = [1 1 1];
            app.CatLoadingPanel.FontName = 'Arial';
            app.CatLoadingPanel.Scrollable = 'on';
            app.CatLoadingPanel.Position = [275 0 276 147];

            % Create MetricAssessmentComparisonLabel
            app.MetricAssessmentComparisonLabel = uilabel(app.UIFigure);
            app.MetricAssessmentComparisonLabel.FontSize = 16;
            app.MetricAssessmentComparisonLabel.FontWeight = 'bold';
            app.MetricAssessmentComparisonLabel.Position = [67 524 364 22];
            app.MetricAssessmentComparisonLabel.Text = 'Metric Assessment Configuration Comparison';

            % Create Epochs1Label
            app.Epochs1Label = uilabel(app.UIFigure);
            app.Epochs1Label.Position = [72 472 524 22];
            app.Epochs1Label.Text = 'The epochs highlighted in green below are included in the saved assessment and will
            be loaded';

            % Create Cat1Label
            app.Cat1Label = uilabel(app.UIFigure);
            app.Cat1Label.Position = [72 244 542 22];
            app.Cat1Label.Text = 'The categories highlighted in green below are included in the saved assessment and will
            be loaded';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = compare_msg_box_v2(varargin)

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            % Execute the startup function
            runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

Appendix D Example DOE Regression Model Output File

```
PEAK HEAT RATE REGRESSION MODEL
Significance Level: 0.0500
Degrees of Freedom: 701
T-Test Critical Value: 1.9634

COEFFICIENTS
          Index   Coefficient  T-Test Value       P-Value  Significant?
              0   45.72905847            NA            NA            NA
              1   -0.54775284     8303.0655        0.0000           Yes
              2   -0.01128674      292.7904        0.0000           Yes
              3   -0.98093972        6.0331        0.0000           Yes
              4   -0.61370087      524.3419        0.0000           Yes
              5   -0.57294460      328.0417        0.0000           Yes
              6   -0.78173708      306.2562        0.0000           Yes
              7   -0.00022373      417.8621        0.0000           Yes
              8   -0.00499218        0.0976        0.9222            No
              9   -0.00936515        2.1788        0.0297           Yes
             10   -0.02653655        4.0873        0.0000           Yes
             11    0.00501016       11.5817        0.0000           Yes
             12    0.00012926        2.1866        0.0291           Yes
             13   -0.00007731        0.0564        0.9550            No
             14   -0.00006213        0.0337        0.9731            No
             15    0.00000378        0.0271        0.9784            No
             16   -0.00371378        0.0016        0.9987            No
             17   -0.05124900        1.6208        0.1055            No
             18    0.16238679       22.3672        0.0000           Yes
             19   -0.01318648       70.8724        0.0000           Yes
             20    0.07655480        5.7551        0.0000           Yes
             21    0.13613414       33.4117        0.0000           Yes
             22    0.00382994       59.4147        0.0000           Yes
             23    0.00029194        1.1820        0.2376            No
             24   -0.01217313        0.0901        0.9282            No
             25    0.00831311        3.7568        0.0002           Yes
             26    0.21914636        2.5655        0.0105           Yes
             27    1.66267264       67.6310        0.0000           Yes
```

Appendix E TMST Regression Model Input File

```
% Regression model for peak heat rate
6
0 0   45.72905847   CONSTANT
0 1   -0.54775284   density_80km_60km
0 2   -0.01128674   density_above_100km
0 3   -0.98093972   EFPA
0 4   -0.61370087   ca_hc_mult
0 5   -0.57294460   cm_hc_add
0 6   -0.78173708   density_60km_40km
1 2   -0.00022373
1 3           0.0
1 4   -0.00936515
1 5   -0.02653655
1 6    0.00501016
2 3    0.00012926
2 4           0.0
2 5           0.0
2 6           0.0
3 4           0.0
3 5           0.0
3 6    0.16238679
4 5   -0.01318648
4 6    0.07655480
5 6    0.13613414
1 1    0.00382994
2 2           0.0
3 3           0.0
4 4    0.00831311
5 5    0.21914636
6 6    1.66267264
```

Appendix F TMST Margin Assessment Configuration File

```
% .ma file containing configuration data for Margin Assessment
=== Number of Metrics ===
4
=== Metric 1 ===
Peak Deceleration
g
peak_decel.mcf
=== Metric 2 ===
Peak Heating
W/cm2
peak_heating.mcf
=== Metric 3 ===
Total Heating
J/cm2
total_heat.mcf
=== Metric 4 ===
Parachute Inflation Load
lbs
parachute_inf_load.mcf
=== Number of Epochs ===
3
=== Epoch Names ===
ATLO 2 (2017-09)
As Built 2 (2017-12)
Flight Reference Delta (2018-10)
=== Number of Categories ===
4
=== Category 1 ===
Mathematical Model
1.00
1.00
1.00
=== Category 2 ===
Code
1.00
1.00
1.00
=== Category 3 ===
Validation
1.00
1.00
1.00
=== Category 4 ===
Model Parameter
1.00
1.00
1.00
=== Category Weight Rationales (per epoch) ===
1
1
Even weighting for initial InSight assessment.
1
Even weighting for initial InSight assessment.
1
Even weighting for initial InSight assessment.
=== Normal Fractions ===
0.50
0.75
1.00
0.50
0.75
1.00
0.50
0.75
1.00
=== Uniform Fractions ===
```

67 0.17
68 0.25
69 0.33
70 0.17
71 0.25
72 0.33
73 0.17
74 0.25
75 0.33
76 === Normal Fractions Rationale ===
77 1
78 1
79 Same sigma fraction used for the original InSight Target Margin methodology.
80 1
81 Same sigma fraction used for the original InSight Target Margin methodology.
82 1
83 Same sigma fraction used for the original InSight Target Margin methodology.
84 === Uniform Fractions Rationale ===
85 1
86 1
87 Treating uniform as representing more uncertainty, therefore doubling normal fraction (as fraction of span).
88 1
89 Treating uniform as representing more uncertainty, therefore doubling normal fraction (as fraction of span).
90 1
91 Treating uniform as representing more uncertainty, therefore doubling normal fraction (as fraction of span).

Appendix G TMST Margin Configuration File

```
% .mcf file containing configuration data metric assessment
=== Number of Epochs ===
3
=== Epoch Names ===
ATLO 2 (2017-09)
As Built 2 (2017-12)
Flight Reference Delta (2018-10)
=== Epoch Duration ===
Span
Span
Span
=== Regression Model File for Each Epoch =====
maxhr_regress_model_17-013_update.rmd
maxhr_regress_model_17-037.rmd
maxhr_regress_model_18-563.rmd
=== Regression Model Number of Inputs ===
6
6
6
=== Metric Limit for Each Epoch ===
61.500000
61.500000
51.800000
=== Number of Categories ===
4
=== Category Names ===
Mathematical Model
Code
Validation
Model Parameter
=== ATLO 2 (2017-09) Uncertainty Type Lock ===
Lock
=== ATLO 2 (2017-09) Input Variable Distribution ===
Normal
Normal
Normal
Normal
Normal
Normal
=== ATLO 2 (2017-09) Input Type Split (Epistemic) ===
0.998000
0.000000
0.000000
0.000000
0.000000
0.000000
=== ATLO 2 (2017-09) Mathematical Model Lock ===
Lock
=== ATLO 2 (2017-09) Mathematical Model Confidence ===
High
High
High
High
High
High
=== ATLO 2 (2017-09) Mathematical Model Rationale ===
1
Mathematical modeling utilizes high fidelity orbital mechanics and statistical state
estimation while tracking and incorporating a comprehensive set of factors.
1
Linearized aero database derived from CFD analysis.
1
Linearized aero database derived from CFD analysis.
1
Based on high pedigree, complete physics-based CFD modeling.
1
```

66 Based on high pedigree, complete physics-based CFD modeling.
67 1
68 Based on high pedigree, complete physics-based CFD modeling.
69 === ATLO 2 (2017-09) Code Lock ===
70 Lock
71 === ATLO 2 (2017-09) Code Confidence ===
72 High
73 High
74 High
75 Med
76 Med
77 Med
78 === ATLO 2 (2017-09) Code Rationale ===
79 1
80 Orbit determination tools shared by multiple missions with results frequently peer reviewed.
81 1
82 Aero database goes through thorough checkouts before delivery.
83 1
84 Aero database goes through thorough checkouts before delivery.
85 1
86 Parametrized atmosphere profile generator code not reviewed.
87 1
88 Parametrized atmosphere profile generator code not reviewed.
89 1
90 Parametrized atmosphere profile generator code not reviewed.
91 === ATLO 2 (2017-09) Validation Lock ===
92 Lock
93 === ATLO 2 (2017-09) Validation Confidence ===
94 High
95 Med
96 Med
97 Med
98 Med
99 Med
100 === ATLO 2 (2017-09) Validation Rationale ===
101 1
102 Orbit determination techniques utilized in modeling entry flight path angle are utilized by all planetary missions and thus providing numerous validation opportunities.
103 1
104 Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.
105 1
106 Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.
107 1
108 Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
109 1
110 Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
111 1
112 Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
113 === ATLO 2 (2017-09) Model Parameter Lock ===
114 Lock
115 === ATLO 2 (2017-09) Model Parameter Confidence ===
116 High
117 High
118 High
119 Med
120 Med
121 Low
122 === ATLO 2 (2017-09) Model Parameter Rationale ===
123 1
124 Parameters have high pedigree, completeness and relevance for the application.

```
125   1
126   All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant
      and appropriate for their function.
127   1
128   All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant
      and appropriate for their function.
129   1
130   Temperature is extrapolated at altitudes below 5 km based on surface temperature
      assumptions.
131   1
132   Temperature is extrapolated at altitudes below 5 km based on surface temperature
      assumptions.
133   1
134   At altitudes above 100 km the appropriateness and relevance of the density model
      parameters becomes low.
135   === As Bulit 2 (2017-12) Uncertainty Type Lock ===
136   Lock
137   === As Bulit 2 (2017-12) Input Variable Distribution ===
138   Normal
139   Normal
140   Normal
141   Normal
142   Normal
143   Normal
144   === As Bulit 2 (2017-12) Input Type Split (Epistemic) ===
145   0.998000
146   0.000000
147   0.000000
148   0.000000
149   0.000000
150   0.000000
151   === As Bulit 2 (2017-12) Mathematical Model Lock ===
152   Lock
153   === As Bulit 2 (2017-12) Mathematical Model Confidence ===
154   High
155   High
156   High
157   High
158   High
159   High
160   === As Bulit 2 (2017-12) Mathematical Model Rationale ===
161   1
162   Mathematical modeling utilizes high fidelity orbital mechanics and statistical state
      estimation while tracking and incorporating a comprehensive set of factors.
163   1
164   Linearized aero database derived from CFD analysis.
165   1
166   Linearized aero database derived from CFD analysis.
167   1
168   Based on high pedigree, complete physics-based CFD modeling.
169   1
170   Based on high pedigree, complete physics-based CFD modeling.
171   1
172   Based on high pedigree, complete physics-based CFD modeling.
173   === As Bulit 2 (2017-12) Code Lock ===
174   Lock
175   === As Bulit 2 (2017-12) Code Confidence ===
176   High
177   High
178   High
179   Med
180   Med
181   Med
182   === As Bulit 2 (2017-12) Code Rationale ===
183   1
184   Orbit determination tools shared by multiple missions with results frequently peer
```

reviewed.
185 1
186 Aero database goes through thorough checkouts before delivery.
187 1
188 Aero database goes through thorough checkouts before delivery.
189 1
190 Parametrized atmosphere profile generator code not reviewed.
191 1
192 Parametrized atmosphere profile generator code not reviewed.
193 1
194 Parametrized atmosphere profile generator code not reviewed.
195 === As Bulit 2 (2017-12) Validation Lock ===
196 Lock
197 === As Bulit 2 (2017-12) Validation Confidence ===
198 High
199 Med
200 Med
201 Med
202 Med
203 Med
204 === As Bulit 2 (2017-12) Validation Rationale ===
205 1
206 Orbit determination techniques utilized in modeling entry flight path angle are utilized by all planetary missions and thus providing numerous validation opportunities.
207 1
208 Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.
209 1
210 Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.
211 1
212 Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
213 1
214 Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
215 1
216 Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
217 === As Bulit 2 (2017-12) Model Parameter Lock ===
218 Lock
219 === As Bulit 2 (2017-12) Model Parameter Confidence ===
220 High
221 High
222 High
223 Med
224 Med
225 Low
226 === As Bulit 2 (2017-12) Model Parameter Rationale ===
227 1
228 Parameters have high pedigree, completeness and relevance for the application.
229 1
230 All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
231 1
232 All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
233 1
234 Temperature is extrapolated at altitudes below 5 km based on surface temperature assumptions.
235 1
236 Temperature is extrapolated at altitudes below 5 km based on surface temperature assumptions.
237 1
238 At altitudes above 100 km the appropriateness and relevance of the density model parameters becomes low.

```
=== Flight Reference Delta (2018-10) Uncertainty Type Lock ===
Lock
=== Flight Reference Delta (2018-10) Input Variable Distribution ===
Normal
Normal
Normal
Normal
Normal
Normal
=== Flight Reference Delta (2018-10) Input Type Split (Epistemic) ===
0.998000
0.000000
0.000000
0.000000
0.000000
0.000000
=== Flight Reference Delta (2018-10) Mathematical Model Lock ===
Lock
=== Flight Reference Delta (2018-10) Mathematical Model Confidence ===
High
High
High
High
High
High
=== Flight Reference Delta (2018-10) Mathematical Model Rationale ===
1
Mathematical modeling utilizes high fidelity orbital mechanics and statistical state estimation while tracking and incorporating a comprehensive set of factors.
1
Linearized aero database derived from CFD analysis.
1
Linearized aero database derived from CFD analysis.
1
Based on high pedigree, complete physics-based CFD modeling.
1
Based on high pedigree, complete physics-based CFD modeling.
1
Based on high pedigree, complete physics-based CFD modeling.
=== Flight Reference Delta (2018-10) Code Lock ===
Lock
=== Flight Reference Delta (2018-10) Code Confidence ===
High
High
High
Med
Med
Med
=== Flight Reference Delta (2018-10) Code Rationale ===
1
Orbit determination tools shared by multiple missions with results frequently peer reviewed.
1
Aero database goes through thorough checkouts before delivery.
1
Aero database goes through thorough checkouts before delivery.
1
Parametrized atmosphere profile generator code not reviewed.
1
Parametrized atmosphere profile generator code not reviewed.
1
Parametrized atmosphere profile generator code not reviewed.
=== Flight Reference Delta (2018-10) Validation Lock ===
Lock
=== Flight Reference Delta (2018-10) Validation Confidence ===
High
```

```
Med
Med
Med
Med
Med
=== Flight Reference Delta (2018-10) Validation Rationale ===
1
Orbit determination techniques utilized in modeling entry flight path angle are utilized by all planetary missions and thus providing numerous validation opportunities.
1
Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.
1
Validation is limited by infrequency of flights to Mars via flight reconstruction that is primarily derived rather than directly measured.
1
Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
1
Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
1
Validation by flight reconstruction where atmosphere characteristics cannot be completely separated from vehicle aerodynamics.
=== Flight Reference Delta (2018-10) Model Parameter Lock ===
Lock
=== Flight Reference Delta (2018-10) Model Parameter Confidence ===
High
High
High
Med
Med
Low
=== Flight Reference Delta (2018-10) Model Parameter Rationale ===
1
Parameters have high pedigree, completeness and relevance for the application.
1
All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
1
All entry vehicle aerodynamic coefficients have high pedigree, are complete, relevant and appropriate for their function.
1
Temperature is extrapolated at altitudes below 5 km based on surface temperature assumptions.
1
Temperature is extrapolated at altitudes below 5 km based on surface temperature assumptions.
1
At altitudes above 100 km the appropriateness and relevance of the density model parameters becomes low.
```