Exploration of Safing Event Models for Interplanetary Spacecraft

Swapnil Pujari
Georgia Institute of Technology
270 Ferst Drive
Atlanta, GA 30332
678-654-2569
spujari21@gatech.edu

Travis Imken
Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91109
818-354-5608
Travis.Imken@jpl.nasa.gov

Glenn Lightsey
Georgia Institute of Technology
270 Ferst Drive
Atlanta, GA 30332
404-385-4146
glenn.lightsey@gatech.edu

***Abstract*— Unexpected spacecraft failures and anomalies may prompt on-board systems to change a spacecraft's state to a safe mode in order to isolate and resolve the problem. The motivation for this paper is to investigate methods to tailor the impact of safing events for spacecraft of different classes, destination, duration, and other categories of interest. Modeling spacecraft inoperability due to a spacecraft entering safe mode could enable mission planners to more effectively manage spacecraft margins and shape design and operations requirements during the conceptual design phase. This paper contributes to the area of safing event modeling by using available datasets to develop various distributions of frequency and recovery durations of safing events for interplanetary spacecraft missions.**

A safing event dataset compiled by JPL is first split into multiple subsets based on various mission classifiers. Using a previously developed mission simulation framework, a distribution of the likelihood of inoperability rates is computed through a Monte Carlo simulation. Three main safing event model types are formulated, implemented, and compared in this paper: a single Weibull distribution, a mixture of two Weibull distributions, and a Gaussian Process model. For each model type, two distributions are incorporated into the mission simulation framework: time-between-events and the recovery duration of a safing event. By specifying appropriate parameters in the mission simulation framework and Gaussian Process model, a Monte Carlo simulation is conducted for a solar-electric Mars orbiter similar to the proposed Next Mars Orbiter. Mission implications from simulated outage times and safing events by each model could motivate greater operability, faster fault resolution by operations teams, and greater system margins.

By incorporating Gaussian Process models into a mission simulation framework, a process is established by which historical mission data may be incorporated and used to model future safing events for interplanetary mission concepts. This enables mission planners to make more informed decisions during spacecraft development.

TABLE OF CONTENTS

978-1-5386-6854-2/19/$31.00

1. INTRODUCTION & MOTIVATION

Advances in on-board computing in the last few decades have enabled robotic spacecraft missions to take control of tasking responsibilities with fewer interactions from the ground. Although engineers thoroughly design and test a variety of conditions faced by the spacecraft, unexpected failures and anomalies may still arise during the mission lifetime. Rather than letting the spacecraft operate in such a state, a 'safe' mode can be implemented where the spacecraft's systems are preserved until the ground can diagnose and recover from the situation. Safe mode is typically defined as the state in which non-essential components and subsystems are powered off, while the spacecraft maintains an attitude such that it is power positive, thermally stable, and commandable by ground operators [1]. Better modeling operability can enable the development of new and complex mission architectures.

The proposed Next Mars Orbiter (NeMO) mission concept is one such example and the mission concept evaluated in this paper. NeMO may support relay & telecommunications in the Martian relay network, perform remote sensing of Mars, and partake in the Mars Sample Return campaign [2]. It may include high-power, high-Isp solar electric propulsion (SEP) to increase the overall capability of the mission. NeMO is not the first SEP interplanetary mission; the Jet Propulsion Laboratory (JPL) has flown SEP on Deep Space 1 (DS1) and Dawn, and the Japanese Aerospace Exploration Agency (JAXA) has flown this technology on Hayabusa. Furthermore, JPL has baselined SEP technology on the Pysche mission that is planned to launch in 2022.

During NeMO's interplanetary transfer to Mars, the SEP engines will need to operate at a high duty cycle to achieve the necessary ΔV. Multiple thrusting segments lasting weeks to months may be necessary during the interplanetary cruise phase of the mission. This requires the spacecraft to remain operational during this extended maneuver. If the spacecraft enters safe mode, those safing events have the effect of reducing overall operability. The frequency and recovery time of safing events may lengthen the mission, potentially reducing available margins and increasing risk in the ability to fulfill its full mission success criteria. A characteristic typical to all types of missions, inoperability is a metric that can impact the design and margins of a spacecraft. Typical inoperability values have been successfully estimated using best engineering practices; however, developing a more rigorous analysis and predictive methodology provides an additional perspective on the likelihood and effects of safing events on spacecraft operability.

An interplanetary spacecraft safe mode analysis was first done by Imken et al. [3]. A database of 240 safe mode entries from 21 interplanetary spacecraft was collected through a va-

riety of sources including JPL, NASA's Goddard Space Flight Center (GSFC), NASA's Ames Research Center (Ames), and Johns Hopkins University Applied Physics Laboratory (JHUAPL). This database contains missions starting with the Galileo mission, launched in 1989, and continues to present day with active missions. It not only includes when the safing event occurred but also mission statistics, root cause of the event, event recovery timeline, and other relevant data. The definitions of time-between-events, recovery duration, and inoperability period developed by Imken et al. are used in this paper in the same manner. Imken et al. also developed a Monte Carlo simulation to simulate the likelihood of realizing an inoperability rate for future missions using the interplanetary safing event dataset [3]. The simulation is modified to include the models developed in this paper to generate bounding frequency and recovery duration distributions. In order to generate bounding frequency and recovery duration distributions, this simulation is modified to include the models developed in this paper.

The modeling fitting work done by Imken et al. indicates that the Weibull distribution is a good candidate for the time-between-events and recovery duration datasets [3]. Due to its flexibility in describing a dataset with just two parameters, a Weibull distribution is commonly used in reliability models. Castet and Saleh modeled satellite reliability for approximately 1600 Earth-orbiting satellites using both nonparametric and parametric models [4]. A Weibull parametric model was shown to best fit the nonparametric satellite failure data. Mixed Weibull distributions, a linear combination of two Weibull distributions, can also provide modeling nonparametric satellite reliability with greater accuracy, as was done by Dubos et al. [5].

Predictive analytics is an area of statistics that deals with obtaining data about a system and using it to model future trends for a particular application. Predictive analytics can be defined as, "Technology that learns from experience (data) to predict the future behavior of individuals in order to drive better decisions" [6]. The approach taken in this paper to model safing events lends itself from machine learning. Evolved from computation learning theory in artificial intelligence, it enables computers to automate learning and making predictions from data. The foundation for the work done in this paper starts with the dataset collection, data modeling, and simulation efforts done by Imken et al. [3] and Master's thesis work by Pujari and Lightsey [7].

This paper contributes to the study of interplanetary spacecraft safing events by developing models as a means to construct a predictive framework for the inoperability rate of future spacecraft. The formulation of the first type of safing event model is developed by adapting a single and mixed Weibull distribution as a parametric class of models. Then, a generalized model using Gaussian Process (GP) models with varying mission inputs is trained and tested by selecting an appropriate covariance function, inference method, training data ratio, and noise parameter. Each model is then adapted and implemented into the mission simulation framework, whereby mission specific inputs are defined for the GP model. The results from the inoperability rates are then compared amongst the presented models for the Next Mars Orbiter mission concept. Thus, based on the frequency and outage time of a safing event simulated for a mission by the models, mission designers can gain insights on inoperability, potentially helping shape tracking, safing recovery, and missed thrust requirements.

2. SIMULATION & MODEL ARCHITECTURE

In order to best quantify safing events, a simulation architecture that can simulate inoperability rates for missions concepts is developed. Each simulation includes a model that provides prediction for the time-between-events and recovery durations of a safing event. Two main models are considered in this paper: a Gaussian Process model and a Weibull Distribution model. The dataset and architectures for the overall simulation and architectures for each model is detailed in this section.

Safing Event Dataset

The safe mode event database containing *Time-Between-Events* (TBE) and *Recovery Duration* (RD) collected by Imken et al. is utilized in the same manner with the same set of assumptions: no cascading safing events, recovery durations from Galileo discarded, all events from the same population, and others included by Imken et al. [3]. One important assumption is that the time-between-events and recovery durations for each safing event are assumed to be independent and identically distributed (iid). The rationale for this assumption is that it simplifies the analysis for a first investigation, although this may not be completely realistic if cascading safes are included. Generally, this assumption enables the use of classical statistical methods to analyze the dataset and subsets and make predictions. Additionally, no data is assumed to be censored.

Each mission and its associated safing events are categorized by four mission classifiers: Mission Class/Category, Mission Destination, Mission Duration, and SEP as seen in Table 7 in the Appendix. Each safing event is further classified by the safing event cause and by the location of the safing event in mission phase. The following list (including abbreviations) shows all the possibilities that a safing event can be classified under, and Figure 1 shows a histogram of the number of safing events for each classifier. The reason the number of valid safing events differ for time-between-events and recovery duration is due to the fact that certain recovery durations were not located; the assumptions for omission are given by Imken et al. [3]. Note that TBE and RD are independent of one another.

(1) **Mission Class**: Small, Medium, Large;

(2) **Mission Destination**: Asteroid / Comet, Heliophysics / Exoplanet, Kuiper Belt Object, Moon, Mars, Saturn, Jupiter;

(3) **Mission Duration [years]**: 0-5, 5-10, 10-15, 15-20;

(4) **Solar Electric Propulsion**: Yes, No;

(5) **Safing Event Cause**: Environmental, Hardware, Operations, Software, Unknown; and

(6) **Safing Event Mission Phase**: Cruise-Primary, Cruise-Extended, Orbit-Primary, Orbit-Extended.

The motivation to categorize the data into such subsets is two-fold; one to enable the statistical and parametric analysis as done by Pujari and Lightsey [7], and two to use these as general inputs to the predictive model. A disadvantage of specializing the data in this manner is that it reduces the sample size for that mission classifier. By already having a limited dataset due to few interplanetary missions, creating subsets of the data with a low numbers of events could give larger uncertainty to certain models. Many machine learning algorithms require large number of datasets to train them successfully.

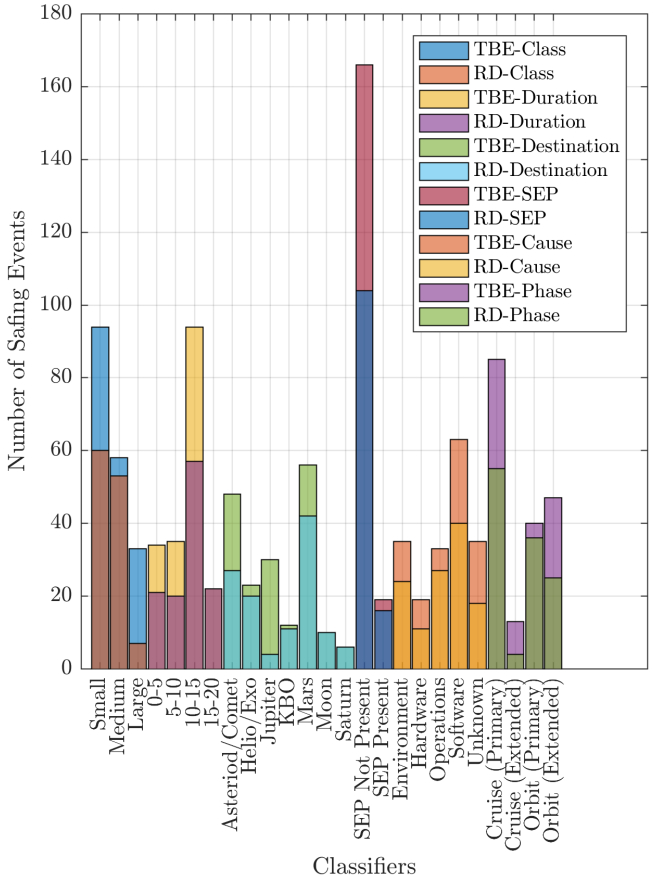

Figure 1. **Mission Classifier Safing Event Histogram**

Specific criteria are used to group missions into each classifier. For mission class, typical mission cost & mass are factored into categorizing missions. For mission destination, seven total destination categories are created based on the typical mission environment. All categories except three have more than one mission per destination category; the Moon, Saturn, and Kuiper Belt Object had only one mission's safing event for those categories.

The mission duration is categorized based on their launch date until the end-of-life or current date. They included both primary and extended mission phases. Three Mars landers and one failed Mars Orbiter only included the cruise phase of their mission as part of the safing event database. For solar electric propulsion, the category simply stated whether SEP technology is baselined as part of the mission or not. For safing event cause and mission phase, the bins are determined based on the entry logs of safing events as determined by Imken et al. [3].

Mission Simulation Architecture

A Monte Carlo analysis simulating safing events helps simulate the duration of a safing event and how much time would pass between each event for future mission concepts. The type of model specified impacts the distribution predicted. For this paper, two main classes of models are considered: a class of Weibull distributions (discussed in a future subsection) and a trained Gaussian Process model (discussed in the following subsection). Using the established mission classifiers, certain inputs are fixed for that mission, while others can use elapsed time of flight if inputs vary over time. This simulation bounds the impacts of safing events through the use parametric and supervised-learning models that use various mission inputs in order to best predict time-between-safing-events and recovery durations. By leveraging the simulation work done by Imken et al., the prediction model framework is incorporated into the existing simulation shown in Figure 2.

The maximum inoperability rate (MIR) is the primary quantitative result from a single Monte Carlo simulation. As defined by Imken et al. and utilized with the same set of assumptions in this paper, inoperability rate captures how long a spacecraft enters an unplanned shutdown due to an anomaly (inoperability period) and the percentage of time it occupies during a thrust arc duration. The inoperability period is defined as the time when the safe mode event occurs, according to spacecraft's mission elapsed timer, and when the spacecraft has resumed nominal operations after exiting from safe mode. This period includes the discovery delay that arises from a set ground pass cadence, investigation, analysis, and corrective action once a safing is realized, any extra human factor delays, and finally the time-of-flight necessary for a command to reach back to the spacecraft. A convolution of the outage time and number of events, the inoperability rate for each Monte Carlo run is combined to generate a probability distribution of the maximum inoperability rate. The MIR is reported as a percentile pulled from this distribution, usually as the 95th, 99th, or 99.7th percentile, as dictated by mission architecture risk.

Gaussian Process Model Architecture

The first type of model utilized in predicting time-between-safing-events and recovery durations independently is the GP model. In a generalized manner, this section discusses the data preparation required, overall architecture, and inputs of the GP model.

Dataset Conversion for GP Model—In order for the mission classifier inputs to be correctly interpreted by the GP model, they must be converted from the categorical string inputs to numerical values. The six categories defined earlier plus the mission elapsed percentage (MEP) are encoded into a total of 25 mission classifiers into a binary format. First, each mission classifier category is split up based on the number of mission classifiers. Since there is no ordinal relationship between each mission classifier in a category, the one-hot encoding methodology is applied. This is the case where a new binary variable is added for each unique value. Integer encoding is employed when sequential integers are applied to a particular category. By assuming a natural ordering between classifiers, poor performance or invalid results such as predictions between classifiers, resulting in a non-integer value could occur by utilizing integer encoding. Therefore, one-hot encoding is applied to each mission classifier category and then those binary numbers are concatenated together to form a 'chromosome' where all inputs are specified in a binary format. For example, the mission class category is encoded as seen in Table 1.

Table 1. **Mission Class One-Hot Binary Encoding**

Mission Class	Binary
Small	[1 0 0]
Medium	[0 1 0]
Large	[0 0 1]

A similar encoding scheme is included for all other categories (mission duration, mission destination, safing event cause,

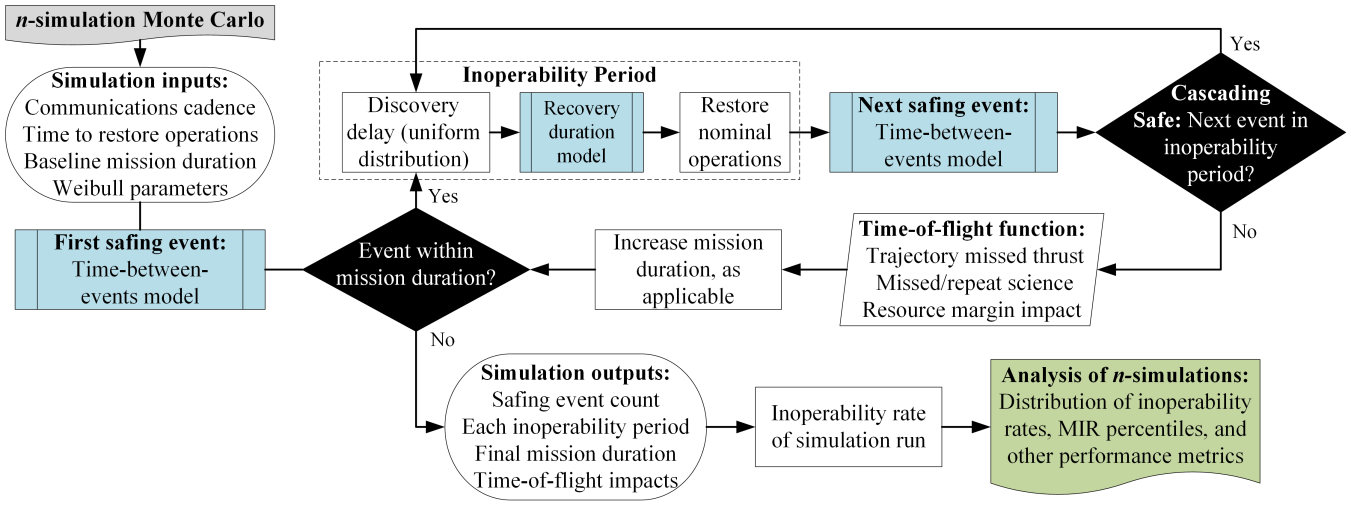

Figure 2. Safing Event Monte Carlo Simulation Block Diagram. Blue boxes indicate subprocesses, green is main simulation output, black boxes are decisions, and gray is the simulation start

and safing event mission phase). For the solar electric propulsion category, a single number is used to represent whether a mission had SEP on-board or not: 1 or -1, respectively. Prediction performance may be better handled with a nonzero binary representation for only two categorical inputs. The mission elapsed percent is also included as the last category as part of the chromosome for input purposes. This is a continuous, positive real number valued from 0 to 1 and thus did not need to be converted to binary. Concatenating each category's representation together, a total of 25 numbers (24 binary and 1 real-valued) represented the input space that is used as inputs to the GP model, shown in Equation 1.

$$\begin{aligned} &\text{GP input} = chromosone \\ &= [\text{Class}, \text{Destination}, \text{Duration}, \text{SEP}, \text{Cause}, \text{Phase}, \text{MEP}] \\ &= [1:3, 4:10, 11:14, 15, 16:20, 21:24, 25] \end{aligned} \tag{1}$$

GP Model Simulation Architecture—Due to the flexibility in the simulation framework, the trained GP prediction model framework is incorporated into the existing simulation shown in Figure 2. Figure 3 shows the developed GP model framework such that it would be very easy to incorporate into the existing simulation. The blue boxes in Figure 2 show where the GP model framework is incorporated into the full mission simulation in a "plug-and-play" manner. From the overall simulation to the GP model, the current mission elapsed percent (MEP) for that particular iteration only needs to be passed as an input. Then, using the MEP and a few other fixed inputs, the GP model's categorical inputs are created. Using the one-hot encoding scheme described earlier, the conversion from categorical to binary inputs is made. Once the training of the particular GP model (whether it is is for TBE or RD) is done, the training data and optimized hyperparameters are used to generate a prediction with a certain mean and variance. From that, a time is randomly generated using the computed mean and variance from a normal distribution (*normrnd* in MATLAB). Since the GP model is not bounded to be strictly positive, it is possible to obtain negative time values; thus the normal distribution is resampled until a positive value is obtained. Thus, a portion of the distribution to obtain a valid sample gets smaller due to the skewed distribution computed. This is discussed in further detail in the implementation section of

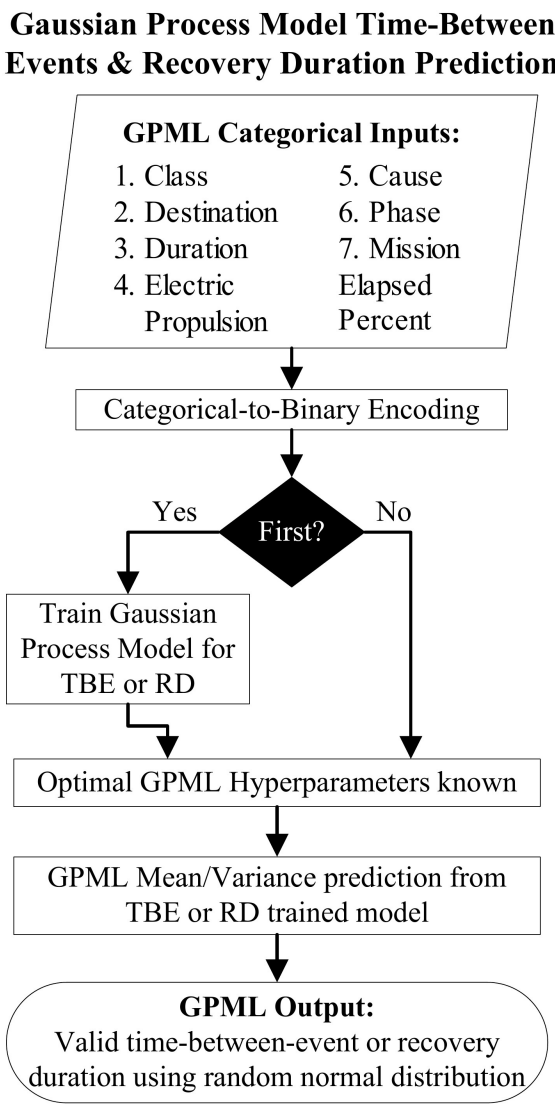

Figure 3. Gaussian Process Model Block Diagram

the GP model. In its current form, the GP model shows that the percentage of probability mass that can be predicted as negative is small and thus this assumption enables the use of a traditional covariance and likelihood functions which have better convergence properties.

The seven categorical inputs that the GP model requires are listed in the inputs parallelogram in Figure 3. The first four inputs are fixed and constant based on the candidate mission that is be simulated. Safing event cause and safing event mission phase are a function of the mission elapsed percent. Once a safing event cause is determined in the first call to the time-between-events GP model, it is passed along to the recovery duration GP model such that the cause remains the same for each event. The safing event mission phase is also fixed based on the duration of user-supplied mission phases. These parameters are easily tunable for each mission and more complex logic and time-varying inputs can be applied.

Weibull Distribution Model Architecture

The other type of model used in the simulation framework is a class of Weibull distributions. Imken et al. discusses the use of a Weibull distribution to parametrically represent the safing event dataset [3]. Similar to the GP model, two Weibull distributions are created: one for the time-between-events and the other for recovery durations.

Shown in Figure 4, the Weibull model flowchart shows how a time-between-event or recovery duration is computed. This model also fits in a "plug-and-play" manner into the mission simulation. The input describes what type of Weibull model is utilized prior to the Monte Carlo run. During the training portion of the model, each dataset and the number of Weibull parameters are specified, and the optimal Weibull parameters are computed. This training portion is only required for the first time when a new dataset or a different number of Weibull parameters are specified since the optimal values will not change. Then, regardless of the type of Weibull distribution used and given the optimal parameters, a single random value from a uniform distribution is selected and the Weibull inverse CDF is applied for each run. Essentially the model is a pseudo-random number generator that returns a Weibull distributed time-between-event and recovery duration based on the scale and shape parameter(s).

In the architecture considered for this paper, a total of four Weibull distributions are defined based on the two datasets (TBE and RD) and the two definitions of the Weibull distributions based on number of parameters (two params. & five params.). One key difference from the GP model, is that the Weibull distribution model is agnostic to mission inputs and is only sensitive to the dataset. Thus, for any mission concept, the full safing event dataset is considered applicable. The underlying assumption in this model is that all events are assumed to be equal in nature.

3. GAUSSIAN PROCESS MODEL

While the GP model and Weibull model architectures presented in Section 3 show how they work at high level, formulating and adapting the models to the appropriate dataset significantly shapes the predicted inoperability rates. Furthermore, the training portion of the model involves the selection of various parameters, data, and functions, which is the most important step to creating a successful predictive model. This section explores the formulation, adaptation, training, and assumptions for the GP model.

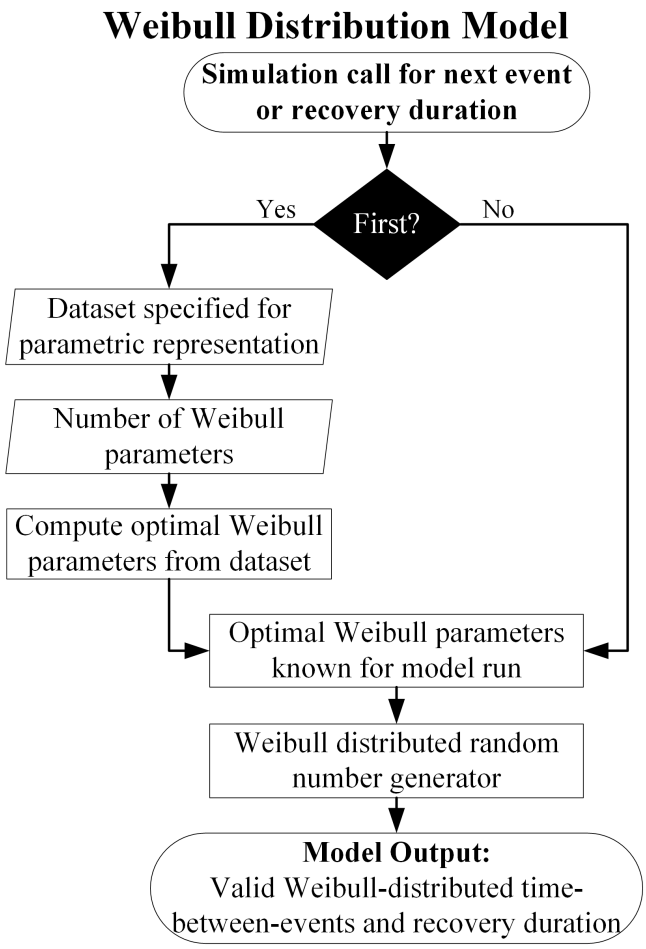

Figure 4. **Weibull Distribution Model Block Diagram**

Supervised learning algorithms, a specific class of machine learning, infer a mapping function based on user-provided input/output training data to predict new outputs given a certain input. A few supervised learning algorithms are considered before settling upon the use of a Gaussian Process model. These algorithms are typically divided into classification, clustering, or regression problems; modeling TBEs and RDs is a classic regression problem. The algorithms considered include artificial neural networks, Gaussian Process models, and regression trees. Due to the number of mission classifiers and possible permutations of each category's classifier exceeding the available data, a regression tree would be too expansive to fully capture all possible scenarios. Rather than a deterministic output that an artificial neural network produces, a GP model gives a mean and variance based on the confidence the model has for a new prediction. Adapted in this paper, the GP model uses a nonparametric kernel-based probabilistic models to take a prior distribution for a given training dataset and obtain a posterior distribution for a set of new inputs [8] [9]. A GP model can perform better with lower amounts of data because of the flexibility in adopting various functions in its computation. Furthermore, it showed promise as a regression algorithm due to its Bayesian framework rather than 'black-box' approach of neural networks. Therefore, by modeling time-between-events and recovery durations as a stochastic process that gives a posterior probability distribution, a GP model is chosen to learn and simulate data for future safing events.

Theory & Assumptions

One key assumption is that the arbitrary set of inputs, either TBEs or RDs, evaluated over a function is one sample from a multi-variate Gaussian distribution. In mathematical terms, this is defined using Bayes Theorem as seen in Equation 2 where tr refers to the training data [9].

$$P(Y|X, X_{tr}, Y_{tr}) \sim \mathcal{N}\big(Y_{tr}K(X_{tr}, X_{tr})^{-1}K(X_{tr}, X), \\ K(X, X) - K(X, X_{tr})K(X_{tr}, X_{tr})^{-1}K(X_{tr}, X)\big) \tag{2}$$

where $K(x, x^*)$ is the kernel function that maps an input from x to x^*. The impact of this assumption is how the posterior distribution of data is modeled. While typical reliability analyses have shown Weibull distributions best modeling failure events, those analyses are not able to capture inputs as is possible by a GP model. Noise is also added on the observed target values based on the confidence of the 'measurements' for safing event TBEs and RDs. Thus, another assumption made is that the noise processes have a Gaussian distribution for each observation n, seen in Equations 3 and 4, where β is a hyperparameter representing the precision of the noise. This assumption is made off the basis of the central limit theorem; where noise is assumed to be random and the aggregate of many random events tends to reflect a normal distribution.

$$t_n = y_n + \epsilon_n \tag{3}$$

$$P(t_n|y_n) \sim \mathcal{N}(t_n|y_n, \beta^{-1}) \tag{4}$$

During training there are a few fixed inputs and assumptions made to keep training computationally manageable. The maximum number of conjugate gradient steps during each minimization is limited to 10,000. For all cases, a standard deviation of two is observed because it encompasses 95.5% of all possible values in a normal distribution. A 2σ value seemed adequate as a starting point to capture most scenarios for safing event predictions. Although certain training methodologies include a validation step to further tune the model while performing the minimization, no such validation is done. The safing event dataset is much smaller when compared to conventional supervised learning datasets. Therefore, having another subset of the total data go towards validation, in addition to the testing dataset, would reduce the available data for the training subset thus limiting the accuracy of training the GP model. For that reason, no validation work has been performed on the GP model presented here and will be investigated further in the future.

Training the GP Model

Rasmussen and Nickisch developed a MATLAB toolbox that enables users to train, predict, and deploy Gaussian process models [10]. A library of various covariance functions, mean functions, inference methods, and likelihood functions are available enabling easier implementation of a GP model [11]. The GP model is first trained by taking the full data set and randomly dividing it up into training and testing data. Then, it is initialized by setting the maximum number of conjugate gradient steps, the mean function, the covariance function, the likelihood and inference functions, and the initial values for the hyperparameters (covariance, mean, and likelihood). Selection of each of these values is very important, as it dictates how the GP model will learn the safing event data. The intricacies of the selection process are detailed by Pujari and Lightsey, but a summary is included in subsections below [7]. The optimized hyperparameters are computed by minimizing the negative log-marginal-likelihood based off the training data. Using those hyperparameters, the testing data is provided into the GP model in order to compute the regression loss between the testing data and the predicted outputs. Through iteration and mathematical intuition, appropriate functions are selected, training data extracted, and hyperparameters initialized as to minimize the overall regression loss. Two separate GP models are developed: one for the time-between-events and another for recovery durations.

Automatic Relevance Detection—When training a GP model in order to find the optimal hyperparameters, the maximum likelihood function is computed to find the correlation length-scale parameter [8]. Rasmussen and Williams [9] extended this by incorporating a separate length-scale parameter for each input variable. While computing the optimal parameters, the relative importance of different inputs can be inferred from the data based on the value of the length-scale parameter. This methodology is called automatic relevance detection (ARD). Thus, it is possible to detect whether certain input variables will have a large or small effect on the predictive distribution because the 'weight' parameter is correlated with the normalized relative importance. The ARD framework is easily incorporated into various kernel functions. For safing events, this framework mathematically helps identify whether certain mission classifiers have a greater importance on predicting future safing event TBEs and RDs.

Minimization Criteria—The criteria used to evaluate differences between models during training are the mean square regression losses. Two main figures of merit are computed: mean regression loss and variance regression loss. Minimizing the distance between the predicted mean value and the actual testing data is denoted as the mean regression error, as shown in Equation 5.

$$err_{mean} = Y_{test} - \mu_{test} \tag{5}$$

Obtaining the smallest variance away from the predicted mean is denoted as the variance regression error, as shown in Equation 6.

$$err_{var} = (\mu_{test} + \sigma \times \sqrt{Var_{test}}) - Y_{test} \tag{6}$$

Then, the mean square error is computed for both mean and variance errors as shown in Equation 7 where j is either the mean or variance, i is the testing data number, and N_{test} is the total number of testing data points evaluated.

$$MSE_j = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (err_j)^2 \tag{7}$$

Having a low mean regression loss indicates that the center of the predicted posterior distribution matches with the supplied testing output. A low variance regression loss indicates that the confidence of the GP model for a particular set of inputs is high.

Selection of Noise Parameter—In many instances, the data collected may not be perfectly captured and therefore may have some uncertainty associated with its values. By including noise on the observed target values as seen in Equation 3, the uncertainty can be accounted for each time the spacecraft enters safe mode and how long it stays in safe mode. Thus, the stochastic noise process in a GP can be thought of as a signal-to-noise parameter of the observations for the safing

event data collected; if greater uncertainty existed for a particular measurement, a smaller SN ratio is used. In a GP model, the noise standard deviation parameter (SN) is incorporated into the likelihood function as a hyperparameter.

For both the time-between-events GP model and recovery duration, SN values sampled ranged from 0.01 to 100 days. Since there is a good amount of confidence with the data collected and its sources, it is deemed that a noise standard deviation value for time-between-events would be 0.1 days. However, the confidence in the observations for when a spacecraft entered and exited is lower than for time-between-events. This is due to the fact that recovery periods documented include both subjective and objective values and certain values are not well documented. Therefore, a noise standard deviation value of 1 hour is deemed appropriate for the level of confidence in the duration values collected.

Selection of Training Ratio—For the GP model, it is assumed no validation dataset would be used; therefore, a training ratio is selected, and the remaining percentage of data would be used for testing. Possible training ratios considered are: 50%, 60%, 70%, 80%, 90%. Since *dividerand* MATLAB function randomly splits the full dataset, 16 iterations per training ratio are computed as to determine what ratio would yield the lowest average and minimum MSE_{mean}, MSE_{var}, and negative log-marginal-likelihood (nlml) values. This is a brute-force methodology to remove the randomness associated with assigning different training data per iteration. 16 iterations are assumed to be sufficient enough for computational tractability purposes; however, more iterations could be included for future training purposes. Finding the smallest average values is more important because it showed greater consistency for that training ratio run across the 16 runs.

Thus, for the time-between-events GP model, a 70% training ratio is selected as the average MSE_{mean} and an average MSE_{var} are the lowest across different percentages. For the recovery duration GP model, an 80% training ratio is selected that had the lowest average MSE_{mean} and an average MSE_{var}. Since there are fewer valid data entries for the recovery duration dataset, it makes sense that a greater percentage of data is needed to accurately train the model.

Selection of Covariance Function—A covariance function is one of the core ways a prior distribution is determined. Since the training data are randomly selected, 16 iterations are again computed for each covariance function evaluated and the average and minimum MSE across each iteration for the mean and variance are computed. Five possibilities are considered as viable covariance functions: squared exponential, Matern with $\nu = 1/2, 3/2, 5/2$, and the rational quadratic. Note that ARD is assumed for all covariance functions since it provided a means to understand the cross-correlation in the input space and appropriately weight each input (mission classifier) while training the model.

For the time-between-events and recovery duration GP models, the Matern covariance function with ν = 3/2 with ARD distance measure is selected. For the TBE results, although the computed MSE for the mean had a median value compared to other covariance functions, the MSE_{var} is the second lowest. Other covariance functions had their strengths in either a minimal MSE_{mean} or MSE_{var}, but the Matern 3/2 gave the greatest balance between minimizing mean and variance MSE errors. For the RD results, the computed MSE_{mean} had a median value compared to other covariance functions, but the MSE_{var} is the lowest and thus selected. One reason why the Matern function also may be the optimal choice is because it contains the absolute exponential kernel, which may be able to better capture physical processes due to its finite differentiability [9].

Selection of Mean Function—A mean function typically helps specify where the expected posterior distribution's mean would lie. For both GP models, initially a mean function is not added as to not constrain the hyperparameters during minimization. The results show that having a constant mean function gives lower mean squared errors. Thus, a constant mean function with an initial value of 200 days is set before the minimization for time-between-events and for recovery duration, the initial value is set to 35 hours. A positive mean function created a non-symmetrical distribution around zero such that the probability of predicting a negative value would be far lower; essentially the posterior distribution is skewed towards positive values.

Selection of Likelihood & Inference Method—As stated by Rasmussen et al., "The likelihood function specifies the probability of the observations given the GP and hyperparameters. The inference methods specify how to compute with the model, i.e. how to infer the (approximate) posterior process, how to find hyperparameters, evaluate the log marginal likelihood, and how to make predictions" [10]. For a Gaussian likelihood function, an exact Gaussian inference method is used; however, for other likelihoods (e.g. Gamma, Weibull, etc.), a Laplace approximation to the posterior Gaussian process must be used. The likelihood functions that are evaluated included: Gaussian, Gamma, and Weibull. The latter two likelihoods are chosen over others to be evaluated because they apply to only strictly positive data, as is the case with the given time data.

For the time-between-events GP model, the Gaussian likelihood function had the second lowest MSE_{mean} and a median MSE_{var}. For the recovery duration GP model, the Gaussian likelihood function had the median MSE_{mean} and a low MSE_{var}. While the Weibull likelihood function had a lower MSE_{mean}, convergence for the algorithm is limited since the Gram matrix often became singular. The predictions from a Weibull likelihood would be invalid and thus the Gaussian likelihood and inference method is selected. Future work is necessary to adapt a Weibull likelihood function to properly converge.

Fully Trained Gaussian Process Model

A summary of the parameters and functions selected for each GP model is shown in table 2. This table also includes the performance metrics that are computed with the particular testing data. While the lowest errors are chosen when selecting parameters during training, the performance metrics still illustrate that there is a significant amount of error in prediction. This is due to a number of factors such as a limited dataset, refinement in mission classifier definition, and the various assumptions made on the dataset. Moreover, these parameters are in no means the optimal configuration for predicting safing events; this is a preliminary result to establish the framework necessary to use GP models for prediction of time-between-events and recovery durations. The usage of other likelihood functions such as a Gamma or Weibull is possible within the Gaussian Process framework, referenced as generalized linear models, and could help tackle the assumptions made on the dataset and noise. Future studies focusing on training the GP models will be required to further reduce the mean square errors and negative log-likelihood.

Table 2. GP Model Summary

Parameter / Function	Time-Between-Events	Recovery Duration
Noise Parameter	0.1 days	1 hour
Training Ratio	70%	80%
Covariance Function	Maternard: $\nu = 3/2$	Maternard: $\nu = 3/2$
Mean Function	Constant: 200 days (initial)	Constant: 35 hours (initial)
Likelihood Function	Gauss	Gauss
Inference Function	Gaussian	Gaussian
MSE_{mean}	69365	1261
MSE_{var}	202028	12869
nlml	865	514

Testing the GP Model

Once a model is trained, plots are generated to show how well the testing data is predicted by the GP model. A discrete number of testing points are evaluated by the GP model, which represented the $1 - TrainingRatio$ of the full dataset. The x-axis shows those training points numerically ordered on a linear scale; however, each point is actually a multi-dimensional representation of the input space (7 categorical/25 binary inputs). The rise and fall to the mean line shows how the GP model reacts to changes in particular inputs. The 2σ boundary shows the tail-end of the normal distribution centered around the mean; if the boundary is smaller, then the model has greater confidence in its prediction since it may have seen such testing data during training. Also, since the 2σ boundary encompasses 95.5% of all data when it is normally distributed, it is possible certain TBE or RD testing points would lie outside of that boundary.

The outputs of GP model at a particular testing data point are the predicted mean and variance. In order to make a prediction, a single random value from a normal distribution using the computed mean and variance is generated. Since the testing data y-axis is in units of time, it is not realistic to predict negative times. If the 2σ boundary can be negative and a prediction made is negative, then new predictions are made until a positive value is obtained. While this methodology may invalidate one sided tail of the distribution, the likelihood of obtaining a negative value remains low because the distribution is skewed towards positive values with a positive mean. Quantifiable estimates of the percentage mass of negative predictions is difficult to generate deterministic values since it varies on many parameters such as the inputs and testing data used. Implementation of the Gamma or Weibull likelihood function is necessary to tackle this drawback of the currently trained GP model to eliminate prediction of negative numbers.

Figure 5 shows a plot of testing dataset used for time-between-events comparing the actual output versus the predicted. The mean line significantly varies between 150 - 600 days elapsed between events as it is perturbed by the inputs the testing data contains. This indicates that the TBE is sensitive to the inputs, which is the desired effect of using a GP model. For certain test data, the variance grows and shrinks based on whether or not the GP model can make an accurate prediction based on its training data and covariance weights. There are a few points that lie outside of the 2σ boundary, but as mentioned, two standard deviations only contain 95.5% of all data. A Monte Carlo analysis would be required to better quantify how many points outside the 2σ are not sufficiently captured.

Figure 6 shows a plot of testing dataset used for recovery duration for safing events. The observed mean value ranges from 40 hours up to 130 hours as the testing data changes the predictions of the model. While there is sensitivity to the inputs, the predicted RD has a large variance and thus uncertainty that can be attributed to the high noise parameter used during training. Note that for a few testing data points, the recovery duration is close to zero, and the mean predictions also shift closer to those values. The RD model as compared with the TBE model predicts fewer changes to the mean but with a greater relative variance for each point.

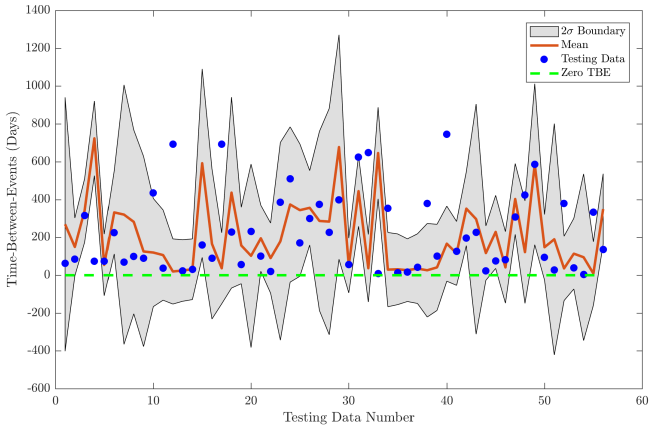

Figure 5. **Time-Between-Events GP Model using Testing Data**

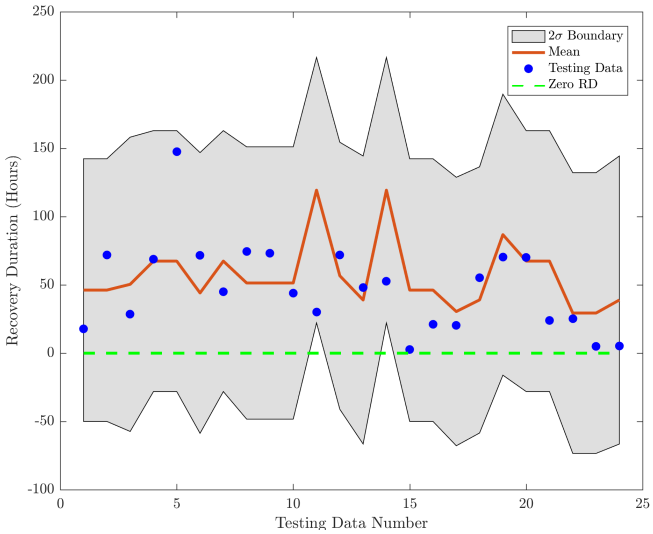

Figure 6. **Recovery Duration GP Model using Testing Data**

4. WEIBULL DISTRIBUTION MODEL

Similar to the GP model implementation section, this section explores the formulation, adaptation, training, and assumptions for the Weibull model. While the Weibull model architecture discussed in a previous section is generalized for any type of Weibull distribution to be adapted into the simulation framework, a total of three types of Weibull distributions are considered for this paper. Distributions can be sub-classified based on the number of parameters used. In this case, Weibull distributions defined by two parameters, Single Weibull, and five parameters, Weibull Mixture, are considered. Distributions can also be sub-classified based on what dataset is used for parametric representation. The Single and Mixed Weibull distributions assume the full use of any valid entries from the time-between-events and recovery

durations datasets. The Binned Weibull is a single Weibull distribution that only uses the entries for a particular mission, input, or other category that is a subset of the total dataset.

Definitions & Theory

A Weibull distribution is commonly used in reliability analyses due to its flexibility in being able to model a dataset with just two parameters: the shape, β, and scale, θ. The shape parameter is a dimensionless, positive parameter and the scale parameter is in the units of time and also positive. Equation 8 shows the reliability function and Equation 9 shows the cumulative distribution function (CDF) [12].

$$R(t) = \exp\left[-\left(\frac{t}{\theta}\right)^{\beta}\right] \tag{8}$$

$$F(t;\beta,\theta) = 1 - R(t), \quad \forall \quad t \geq 0 \tag{9}$$

A single Weibull distribution can only show a single trend from the dataset whereas more complex fits may represent interior trends in the data better, at the risk of overfitting and losing qualitative insight. Hence, a finite mixture distribution, which is a linear combination of multiple distributions, can be used.

In this analysis, a combination of two Weibull distributions with weights for each distribution are considered. This is done to understand if the data exhibited bi-modal behavior and is a better fit than a single Weibull distribution. This paper will henceforth reference the single Weibull distribution as the '1-Weibull' or 'Single Weibull' and the two Weibull mixture distribution as '2-Weibull' or 'Mixed Weibull'. Equation 10 shows the reliability function for the 2-Weibull distribution; the CDF remains the same as in Equation 9.

$$R(t) = (\alpha)\exp\left[-\left(\frac{t}{\theta_1}\right)^{\beta_1}\right] + (1-\alpha)\exp\left[-\left(\frac{t}{\theta_2}\right)^{\beta_2}\right] \tag{10}$$

where: $\quad 0 \leq \alpha \leq 1, \quad \theta_j > 0, \quad \beta_j > 0, \quad$ all $\quad t \geq 0.$

Optimal Parameters

There are multiple ways to determine whether a certain scale and shape parameter of a Weibull distribution fits the data as best as possible. A Weibull plot is one that linearizes the axes such that the data fits the estimated Weibull reliability $\hat{R}(t)$, in a linear manner. Data aligned along the $\hat{R}(t)$ line in the [ln(t); ln(-ln(R(t)))] space is considered an appropriate fit for a Weibull distribution using this graphical estimation technique.

However, a more rigorous test that is able to deduce optimal parameters is the maximum likelihood estimation (MLE) methodology. The basic concept involves formulating a likelihood function and then finding parameter(s) that maximizes that likelihood function. Saleh and Castet define the likelihood function as,"the probability of obtaining or generating the observed data from the chosen parametric distribution" [12]. The full derivation is detailed in Chapter 3 of Saleh and Castet's book [12]. For convenience, the likelihood functions formulated for the 1-Weibull and 2-Weibull are shown in the equations below. The natural logarithm of the likelihood function is taken to yield the log-likelihood equation for the 1-Weibull distribution, seen in Equation 11.

$$l(\boldsymbol{\theta}) = l(u,b) = \ln L(\boldsymbol{\theta}) = -\left(\sum_{n=1}^{n}\delta_i\right)\ln b + \sum_{n=1}^{n}(\delta_i z_i - e^{z_i}) \tag{11}$$

where:$y_i = \ln t_i, \quad u = \ln \boldsymbol{\theta}, \quad b = \beta^{-1}, \quad z_i = (y_i - u)/b$

The log-likelihood equation for the 2-Weibull is then defined as a linear combination of where a weighting parameter, α, as factored into the PDF (f) and reliability (R) functions as seen in Equation 12.

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n}[(\delta_i)\ln f(y_i,\boldsymbol{\theta}) + (1-\delta_i)\ln R(y_i,\boldsymbol{\theta})] \tag{12}$$

$$f(y_i,\boldsymbol{\theta}) = (\alpha)f_1(y_i,u_1,b_1) + (1-\alpha)f_2(y_i,u_2,b_2) \tag{13}$$

$$R(y_i,\boldsymbol{\theta}) = (\alpha)R_1(y_i,u_1,b_1) + (1-\alpha)R_2(y_i,u_2,b_2) \tag{14}$$

where: $y_i = \ln t_i, u_j = \ln\theta_j, b_j = \beta_j^{-1}, z_i = (y_i - u)/b.$

The optimal parameters, $\hat{\boldsymbol{\theta}}$, can be computed using traditional optimization methods. Maximizing $l(\boldsymbol{\theta})$, or equivalently minimizing $-l(\boldsymbol{\theta})$, is done using a quasi-Newtonian optimization algorithm − Broyden-Fletcher-Goldfarb-Shanno (BFGS), which does not require an explicit gradient formulation. The built-in MATLAB function *fminunc* is able to perform this unconstrained minimization of the log-likelihood function. Certain convergence issues can arise such as finding local minima or not converging if the initial guess is in an unstable region. The initial parameters used in the optimizer are found using trial-and-error and best-judgment. Future methodologies could include more robust ways such as shifting and scaling the data to compute initial parameters. This would improve upon convergence properties and help to find global optimal solutions.

As used in Equations 8 to 10, each t value corresponds to either TBE or RD. Thus, there are four reliability functions formulated: two for when t = TBE and two for when t = RD. Figures 7 and 8 show the CDF, Weibull probability plots, and optimal parameters for the 1-Weibull & 2-Weibull distributions for TBE and RD, respectively. The first and third subplot of each figure show the CDF; that is the cumulative probability that either a safing event will occur or if a safing event recovery is completed. The maximum likelihood estimation (MLE) methodology outlined in the theory section is utilized to compute the optimal model parameters for each CDF. The TBE CDFs show that there is a 80% probability using the 1-Weibull and a 78% probability using the 2-Weibull that the next event will have occurred in 400 days or fewer after the previous safing event or start of mission. Similarly, the RD CDFs show that there is a 71.5% probability using the 1-Weibull and a 74.5% probability using the 2-Weibull that the recovery duration of a spacecraft in safe mode will end in 72 hours or fewer. Since the 1-Weibull and 2-Weibull CDFs generally have similar predictions, certain criteria are explored in later sections to evaluate a preference between these CDFs.

The second and fourth subplots show probability plots for a 1-Weibull and 2-Weibull distribution respectively. Probability plots are used to graphically highlight how well data fits against each model. Since the Weibull distribution is linearized across its axes, if the data also is linear with the same slope, then the Weibull distribution is a good match. If there is curvature in the data away from the Weibull model line, then the probability plot indicates either a different

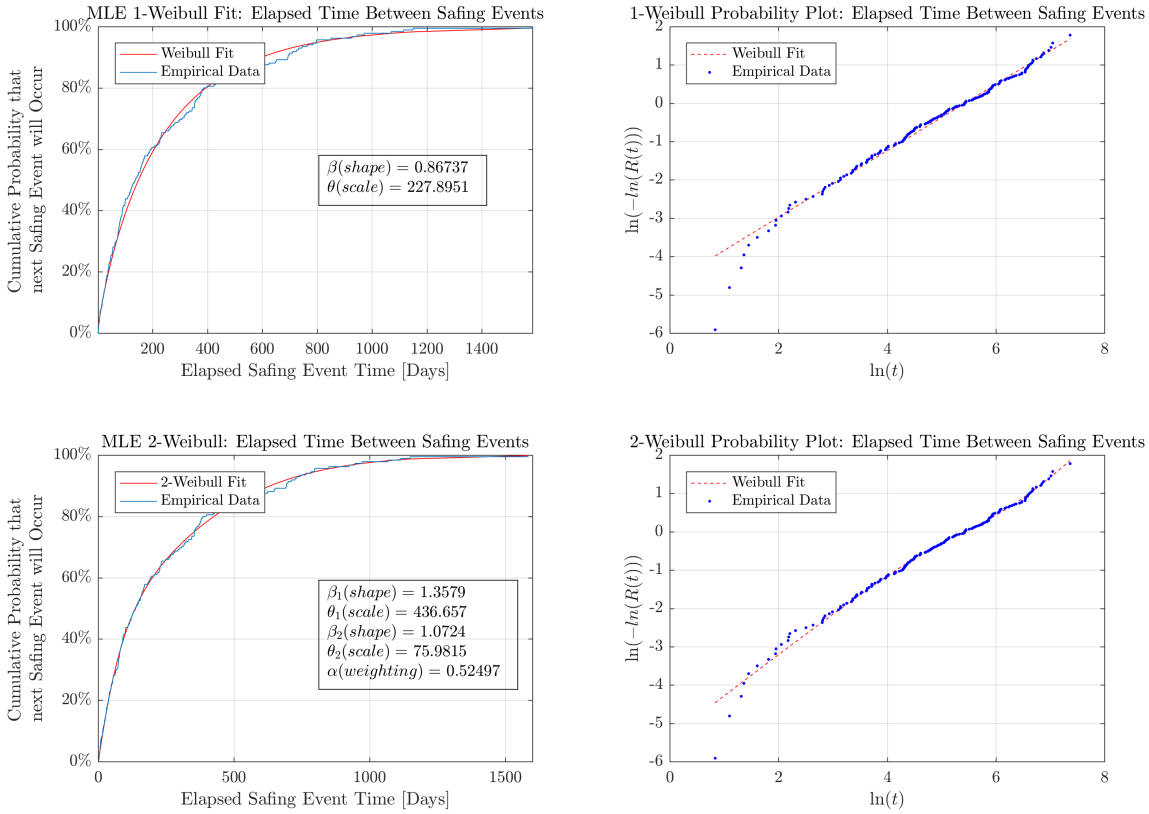

Figure 7. 1-Weibull & 2-Weibull Optimized Parameters, CDFs, and Probability Plots for Time-Between-Events

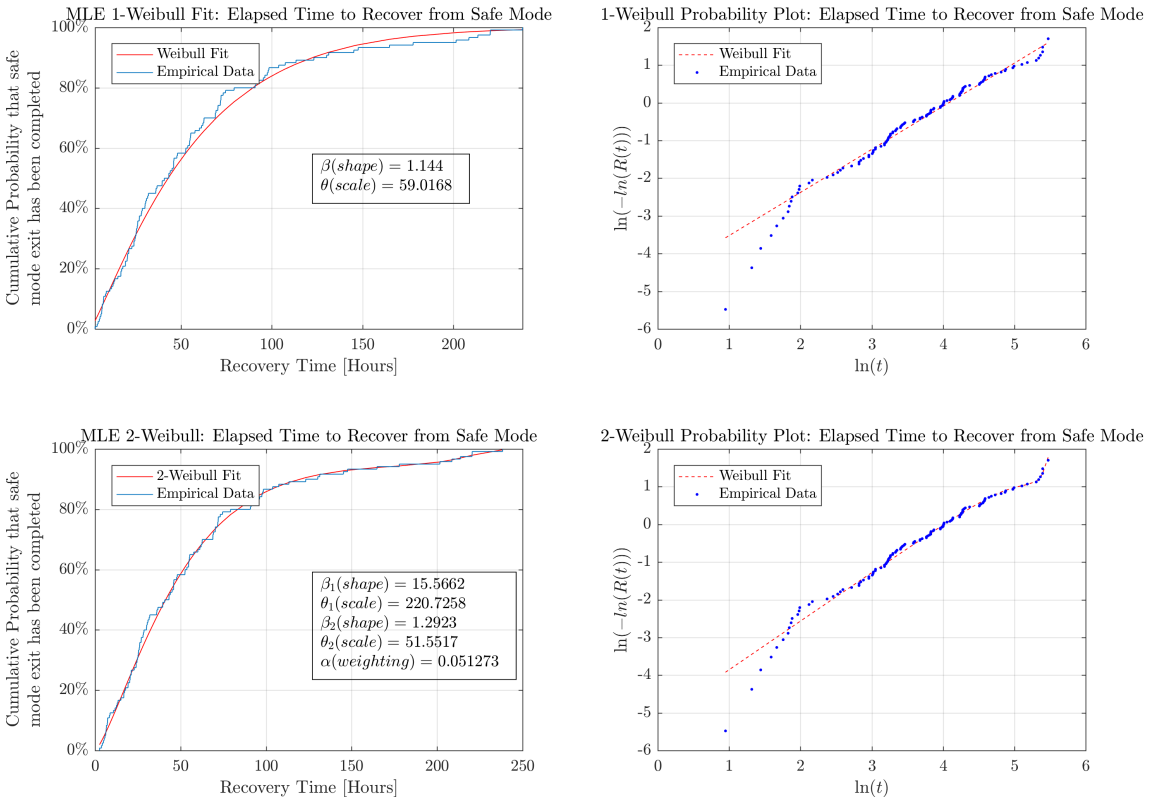

Figure 8. 1-Weibull & 2-Weibull Optimized Parameters, CDFs, and Probability Plots for Recovery Duration

distribution may fit better or a mixed distribution may be more ideal. Thus, for both 2-Weibull probability plots, the mixed distribution is better able to capture the curvature in the data from the first and second half due to the added degree of freedom. By looking at corresponding CDFs, from a graphical perspective, the 2-Weibull distributions fits the empirical CDF better. This behavior is expected since more degrees of freedom mean a better fit within the data bounds.

Verification of the 1-Weibull and 2-Weibull MLE implementation in MATLAB is conducted by using the Weibull++ software by Reliasoft Corporation. This industry standard software was chosen as it specializes in the analysis of reliability data. The percent difference between each parameter computed is no greater than 5% for all except one parameter, which had a 10% difference. This difference is not considered significant because the other goodness-of-fit parameters define how well these values fit the data. Thus, the optimal Weibull distribution parameters for both the 1-Weibull and 2-Weibull validated from the Weibull++ software give confidence in the results that the MATLAB implementation of maximizing a log-likelihood function is correct.

Goodness of Fit

Once the optimal parameters are found for both the 1-Weibull and 2-Weibull distributions as described above, a criteria is necessary in order to evaluate whether either distribution is a good fit both in an absolute and relative manner. Two criteria are used to determine the goodness of the fit to the empirical data: Mean Square Error and Akaike Information Criteria. The MSE of a predictor $\hat{Y}$ is defined as the average of the square of errors/deviations. However, in order to accurately judge the level of overfitting the model, a different criteria than MSE is needed. When estimating finite Weibull mixture distributions for reliability purposes, Elmahdy and Aboutahoun used the Akaike's Information Criteria (AIC) as a goodness-of-fit criterion [13]. AIC estimates the relative information lost in a given model that is derived from the data and trades off fit versus simplicity; the AIC value is penalized if more parameters are added to a given distribution. AIC also only reports the relative quality of one model to another but gives no warning of absolute fit. A goodness-of-fit metric is important to perform a parametric analysis because it describes how well that model fits the set of data in a statistically rigorous manner.

From Figure 9 for the time-between-events, both the 1-Weibull and 2-Weibull CDFs remain reasonably bounded to each other. Although harder to tell from the CDF plots, the probability plots show how the slope of the 2-Weibull is initially lower than that of the 1-Weibull, but also curves upwards near the end of the dataset to better approximate it. The dispersion of the residual around the empirical CDF shows that the 2-Weibull distribution is a better fit than the 1-Weibull since the 2-Weibull MSE is 3.6 times better than that of the 1-Weibull. However, computation of the relative AIC shows that the 2-Weibull distribution overfits the TBE data. Even though the MSE and AIC values produce opposing conclusions, the drawbacks of MSE hinder it from prevailing over the conclusion from AIC. Furthermore, this indicates that while the data may have some, bi-modal behavior in the data, it comes at a cost of overfitting the model and thus loses validity when choosing the 2-Weibull distribution for representing the TBE dataset.

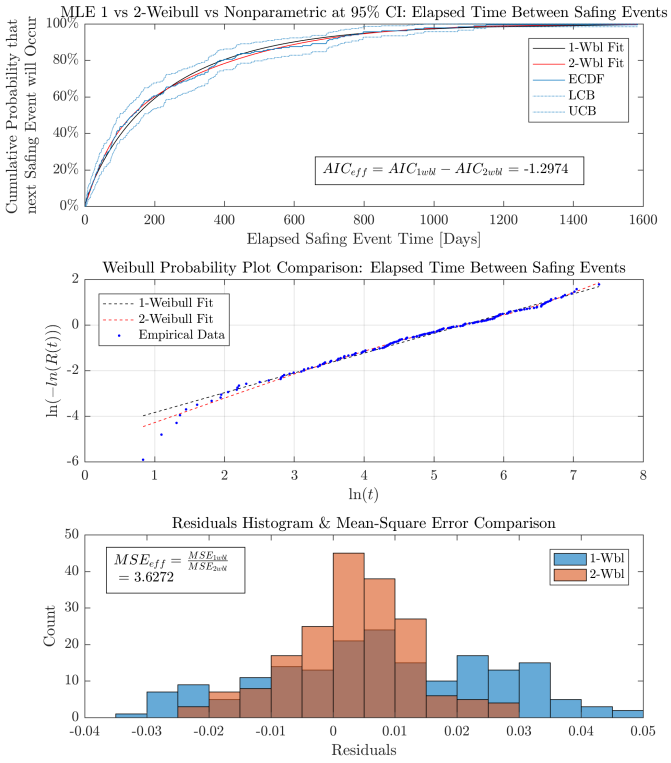

Figure 9. 1-Weibull & 2-Weibull Distributions Comparison for Time-Between-Events

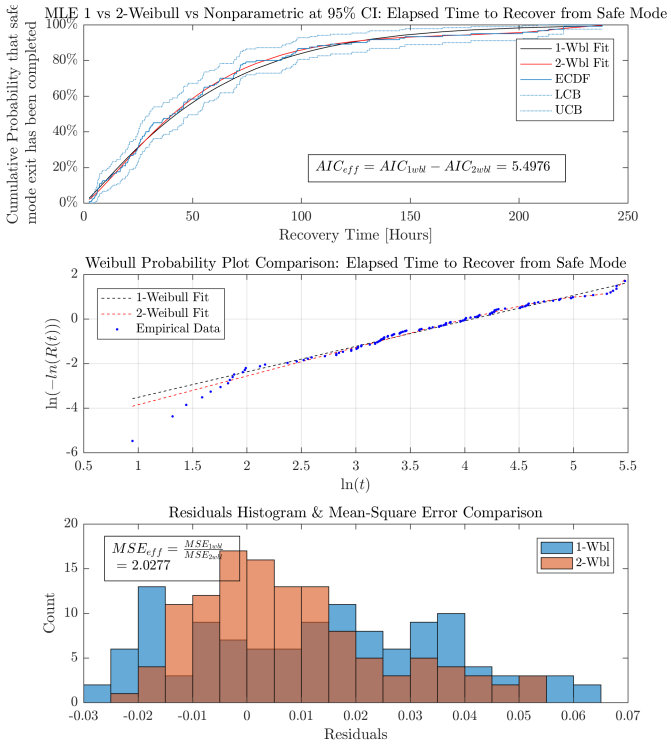

Figure 10. 1-Weibull & 2-Weibull Distributions Comparison for Recovery Durations

While not as close together as TBE, Figure 10 for the recovery duration dataset shows each CDF within reasonable bounds of the empirical dataset. The probability plot shows how the 2-Weibull is able to capture the bi-modal behavior in the data since it contains varying slopes to better fit the data. The residual subplot shows how the 2-Weibull has a more symmetric distribution of the residuals and smaller variance around 0 versus the 1-Weibull which has multiple peaks. From the goodness-of-fit computation, both the MSE ratio is greater than 1 and the AIC difference is positive, indicating that the 2-Weibull is a better representation of the recovery duration data and that it does not overfit the data. Thus, for future prediction purposes, mixed 2-Weibull distribution to predict recovery durations should yield more accurate results.

Summary of Weibull Models

Given the definitions & theory of the three types of Weibull models, how the optimal parameters were computed for each model type, and the evaluating the goodness-of-fit on the dataset, a summary of each of the three optimized Weibull Distribution models is shown below. Since this analysis constitutes the 'training' portion of the models before the first run, a random number can be generated using the known optimal Weibull parameters as discussed in the architecture. Note that these fits are subject to change as new data is added and current records refined.

Single Weibull Distribution Model—The optimal Weibull parameters computed for the time-between-events and recovery duration datasets for the 1-Weibull are used in the Single Weibull mode. These parameters can be seen in Table 3.

Table 3. **Single Weibull Distribution Model Parameters**

Dataset	Shape (β)	Scale (θ)
TBE (days)	0.86737	227.8951
RD (hrs)	1.144	59.0168

Mixed Weibull Distribution Model—Recall that for a dataset that a MSE ratio greater than 1 and a positive AIC difference implies that the 2-Weibull is a better fit than the 1-Weibull distribution. This case was true for the recovery duration dataset but not for the time-between-events dataset. Therefore, the optimal Weibull parameters for the Mixed Weibull model are computed for the time-between-events dataset from the 1-Weibull and for the recovery duration dataset from the 2-Weibull. These parameters can be seen in Table 4.

Table 4. **Mixed Weibull Distribution Model Parameters**

Dataset	β_1	θ_1	β_2	θ_2	α
TBE (days)	0.86737	227.8951	N/A	N/A	N/A
RD (hrs)	15.5662	220.7258	1.2923	51.5517	0.05127

Binned Weibull Distribution Model—A binned Weibull model is a sub-case of the Single Weibull model. What differentiates the binned model is the fact that it uses a subset of the dataset rather than the entire dataset. This subset can be defined as a particular mission or other mission classifier for comparison purposes, which is left up to the user. For the purpose of this paper and the case study for the Next Mars Orbiter mission concept, a similar Mars Orbiter mission's time-between-events and recovery durations are used in order to find the optimal 1-Weibull parameters. For the sake of mission anonymity, these values are not presented here.

5. SIMULATION RESULTS

The simulation and models' architectures & implementation have been developed agnostic to the mission for which safing events are being predicted. The framework has remained generalized such that future mission planners can use this as a starting point for various mission concepts. In order to investigate the fidelity and trends of each model from the mission simulator, the proposed Next Mars Orbiter mission concept was chosen as a case study. Thus, the results presented in this section focus on the inputs, assumptions, results of the modeled safing events for NeMO and the potential implications on the design of the spacecraft.

Inputs & Assumptions

The set of simulation parameters assumed for the proposed NeMO mission tested in this paper are shown in Table 5. The justifications for choosing these values can be found in the paper by Imken et al. [3].

Table 5. **Simulation Parameters**

Metric	Value
Discovery Delay / Cruise Pass Cadence	Pass every 3 days
Pass Length	4 hours
Time to restore nominal operations	12 hours
Time-of-flight function increase, relative to inoperability period	none

The Next Mars Orbiter mission concept is assumed to have a total mission length of 6 years, the same used by Imken et al. to enable comparisons. For this case study, no extended missions phases are incorporated. Thus, the primary cruise duration lasts 2/3 of a year, and it is assumed that for the remaining mission length, NeMO is in its primary orbit phase. The remaining categorical inputs that the GP model needs for the NeMO mission concept are as follows:

(1) **Mission Class**: Medium;
(2) **Mission Destination**: Mars;
(3) **Mission Duration [years]**: 5 - 10; and
(4) **Solar Electric Propulsion**: Yes.

Gaussian Process Model Predictions

Since the GP model framework only requires the mission elapsed percent as an external input, 25 MEP values are randomly selected from a uniform distribution, and time-between-events and recovery durations are predicted fusing the same categorical inputs as defined in the earlier section for NeMO. Figures 11 and 12 show the posterior Gaussian distributions for each MEP; this includes the mean and variance computed by the GP model. Most of the 2σ boundaries are predicted to be positive, while only certain tail ends are below the threshold. Recall that any negative values predicted are not considered, and are re-predicted from a normal distribution with the given mean and variance.

For the TBE results, it is interesting to see how the mean changes based on the mission elapsed percent. There is no clear trend that for NeMO, time-between-events increase or decrease as a function of the mission time, but there are periods where the mean predictions may be higher or lower. The volatility in the predictions is due to the type of training data the model received and how the weights are formulated. For the RD results, there is far less change in the mean values as a function of the mission elapsed percent. It seems that there are really two modes to the mean value where most

values are predicted within: 55 and 101 hours. For either TBE or RD, the validity of the results stems from the confidence achieved from the testing data, which was a subset of the full dataset. However, future work is necessary to understand how each input shapes the results. This is difficult to deduce with a machine learning algorithm due to its black-box nature.

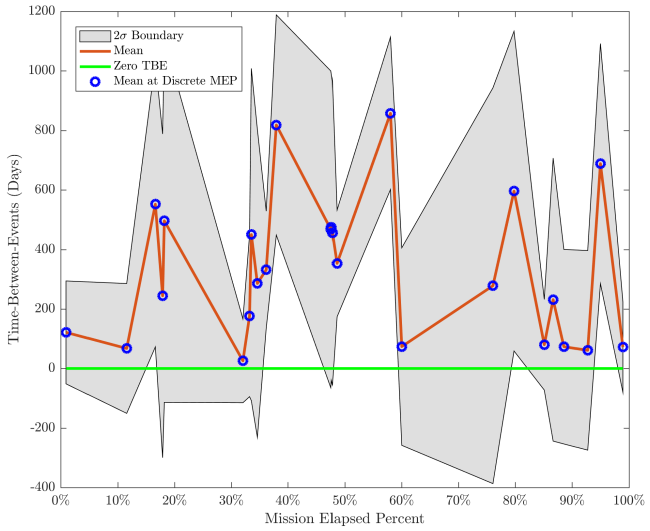

Figure 11. NeMO Sample Predictions using Trained GP Model for Time-Between-Events

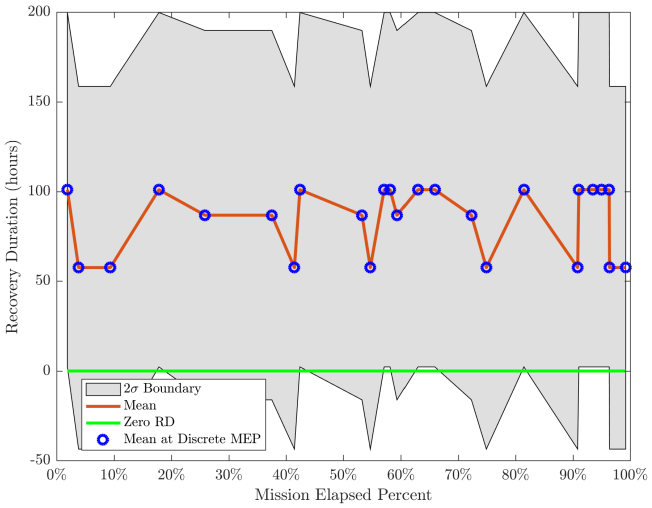

Figure 12. NeMO Sample Predictions using Trained GP Model for Recovery Durations

Monte Carlo Results

Once the training and implementation for the time-between-event and recovery duration models are complete in the simulation framework, a safing event Monte Carlo simulation for NeMO is conducted for 1 million runs. This Monte Carlo simulation is run three separate times, each with a different model: the Single Weibull Distribution model, the Mixed Weibull Distribution model, and the Gaussian Process model. The results from each Monte Carlo simulation on mission inoperability rates for the three models are listed in Table 6.

The similarity across all three models at the 99.7 percentile shows that the Monte Carlo results are within the family of one another. Assuming a 99.7% likelihood, the total number of safing events for the GP model compared to two Weibull distribution models decreases from 19 to 11 events. This result of a sharp decrease in the number of safing events implies that the simulated time-between-safing-events from the GP model is larger than what is simulated from the 1-Weibull distribution, which used by both the Single and Mixed Weibull models. Due to the TBE training data utilized as well as the covariance optimal weights, the mean predictions for the Weibull distribution models are lower than most of the mean predictions by the GP model for mission elapsed percentages. For both Weibull distribution models of a TBE, the mean is computed as 245 days between each event and the mean TBE simulated by the GP model ranges from 50 to 700 days as seen in Figure 11. Thus, the differences in the models lend to the significant difference in the simulated number of safing events for NeMO.

The simulations for each recovery duration are larger for the GP model than modeled by both Weibull distribution models. Again, in this case for the Weibull distribution models, the mean recovery duration simulated by the Weibull models is approximately 56 hours compared with the observed bi-modal result of 55 and 101 hours from the GP RD model, as seen in Figure 12. Thus, the average recovery duration for the entire mission duration simulated by a GP model is higher than what is simulated by the Weibull distribution models. Since the mixture Weibull model uses a 2-Weibull for the recovery duration as opposed to the 1-Weibull for the single Weibull model, the outage times simulated vary. The mean recovery duration simulated for the single Weibull of 56.25 hours is slightly larger than simulated by the mixture model of 56.17 hours. Thus, it makes sense at the 99.7 percentile, that each outage time simulated by the mixture model is slightly lower.

Another way to understand the results of the GP model is to compare its results with a binned Weibull model that has similar mission classifiers as itself. Since the GP model uses weights on each input to generate a predicted value, comparing those to a mission that has similar inputs may provide insight into how the supervised learning algorithm generates predictions. A total of seven categorical inputs exist for the GP model; three of those are mission agnostic, and four are mission specific. Thus, a Mars mission where three out of the four classifiers are the same as for NeMO was selected as the underlying dataset used in the binned Weibull model.

Averaging the recovery durations from the binned Weibull yields a result of 76 hours. This is longer than the simulated mean values from either Weibull distribution of ˜55 hours and falls very close to the median of GP model's simulated RD of 78 hours. And after a Monte Carlo analysis with many runs, the outage time simulated by the GP model would be larder due to the significant impact from a NeMO-like existing mission. This comparison assumes that the data from the existing Mars mission has the greatest impact on predicting safing events for NeMO, since many of the mission classifiers are the same.

For time-between-events, the average TBE for the existing Mars mission binned Weibull is 186 days compared to the average value simulated by the Weibull models as 245 days. Since this existing Mars mission does not have a SEP system on-board, a different mission was used to compare the binned Weibull to the results, a mission that currently uses SEP. The

Table 6. NeMO 99.7% Mission Inoperability Rates, Outage Times, and Number of Safing Events for a 1 million run Monte Carlo Simulation for 3 Models

Metric	Units	Single Weibull Distribution Model	Mixed Weibull Distribution Model	Gaussian Process Model
MIR	%	3.74	3.77	3.89
Total Outage Time	days	81.9	82.6	85.1
Each Outage Time	days	12.9	12.57	17.71
Total Safing Events	#	19	19	11

mean TBE for that SEP mission is 276 days compared to 245 days, as simulated by the two Weibull models. Thus, a longer mean TBE indicates a fewer safing events throughout the mission, as is shown by the GP model's predictions. This could indicate that the TBE GP model is more sensitive towards the solar electric propulsion classifier than others inputs and significantly impact the predictions. Further work is necessary to understand the sensitivities and intricacies of the GP model's wide array inputs on predicted TBE and RD. While the Single and Mixed Weibull models assume every mission as equal, creating and comparing a Binned Weibull for a mission similar to the NeMO concept enables to understand and compare the results presented by the GP model.

Combining the total number of events and each outage time, the total outage time simulated by each model is around 80 days for the NeMO mission concept. The GP model predicts a slightly higher total outage time than both Weibull models, and the mixture Weibull model predicts a day higher total outage time during the mission than the single Weibull model. Thus, it makes sense that the MIR computed for the three models fall in that order: the GP model predicts the highest value, the mixture model predicts the median MIR, and the single Weibull predicts the lowest. Each maximum inoperability rate value with a 99.7% probability is within the same percentage point. The maximum differences in MIR values corresponds to 0.15%, or equivalently for a 6 year mission, about 3.2 days of extra inoperability simulated by the GP model compared with the single Weibull model. This corresponds to 12 hours extra per year for the NeMO mission concept.

The results obtained from the GP model are similar to the Weibull model but shows certain nuanced behavior in its simulated MIR. With a result this close, it may simulate better or worse MIRs depending on the mission scenario. Validating these models against past missions and simulating other mission concepts are avenues for creating a generalized safing event prediction tool.

Implications for NeMO

There are a few implications on the design, margins, and requirements for the Next Mars Orbiter mission concept that the safing event models provide in order to reduce risk. First, the GP model predicts that recovery times for NeMO would be longer when various mission inputs are factored in. More time spent recovering the spacecraft out of safe mode means a longer percentage of time that the mission is inoperable. This may motivate the development of greater autonomy on-board NeMO such that the spacecraft bus may be able to better diagnose certain events and provide more informative health data to ground operators. It would shape the requirements on NeMO to include an increased fault checking capability and/or better data management on-board. Additionally, the pass cadence assumptions for this simulation is one DSN pass every 3 days. The maximum outage time simulated to 3σ is predicted to be 17 days; an increase in the pass cadence could decrease that outage time. Moreover, the recovery urgency based on mission risk posture per safing event could increase such that the recovery period can shorten. While the cost to the mission, from DSN time, personnel, and other resources, would increase, the resulting increased operability of the spacecraft could be worth it for the mission's success. Furthermore, new requirements could be placed on the operations team such that greater confidence and faster response time dealing with fault scenarios are implemented.

Another impact from the overall increase in MIR simulated by the GP model is the missed thrust periods. Currently, there is no time-of-flight increase implemented in the simulation; however, a 3.9% mission inoperability could affect when the mission reaches its destination. Moreover, the consequence of missing thrust maneuvers during certain segments of the trajectory may significantly lengthen the mission. Those missed thrust periods may correspond to correlations greater than 1:1 for each period. Extensions to the mission due to missing critical thrusting periods could significantly influence how margins are computed for a low-thrust mission. An increase in propellant margin would impact other margins such as mass and power, which would influence the spacecraft design considerably. In order to reduce the maximum inoperability simulated, spacecraft and operational capabilities may need to increase for the Next Mars Orbiter mission concept.

The mission inoperability rates, number of safing events, and outage times presented are mission specific to NeMO; as the inputs are changed for new missions, the results would also vary. Thus, it is up to the user to choose which model based on the given set of assumptions and confidence to predict safing events. To accommodate multiple mission inputs, the GP model enables users to factor the predictions made for the frequency and recovery duration of a safing event. The GP model results presented in this section show that the output MIRs are in family with the Weibull models' predictions. Through the simulation, mission designers would be able to quantify the likelihood of realizing the worst-case inoperability rates, and make design and operational decisions based on the results.

6. RECOMMENDATIONS & FUTURE WORK

The assumptions, analyses, and modeling reported in this paper provide a methodology for future mission planners looking to model the overall impact of safing events for a certain mission architecture. First, the user must decide what set of assumptions placed on the dataset and simplifications are acceptable for modeling purposes. Next, a safing event process model such as the Single Weibull distribution model, Mixed Weibull distribution model, or Gaussian Process model for safing event predictions is selected. When utilizing the GP model, the user must be aware of its 'black-

box' nature that occurs during the training process and that re-training may be necessary. While the models developed and implemented in this paper verify the end-to-end flow of the simulating safing events, improvements during training, tackling some of the assumptions, and validating the models can still be made.

One of the first set of simplifications is that subsets of the dataset are created using commonly defined mission classifiers. While these classifiers are made using categories that would be reasonable for a mission planner, future work could involve creating categories based on statistical significance between the data. In order to rigorously find how to split the safing time-between-events and recovery durations based on the inherent divisions within the data rather than 'arbitrary' categories, more historical mission data statistical analysis is needed. Methodologies such as classification or hypothesis tests could be useful in finding these natural boundaries in the dataset. These new categories may lend to better predictions since the weights that the GP model 'learns' would have a more statistically significant backing.

Training the GP model is the most important step to effectively utilizing a supervised learning algorithm as a predictive model. The current method of computing the mean square error for deviations from the mean and variance estimates for the testing data is a good preliminary method. However, other metrics such as mean absolute error, sum absolute error, and others can be used for evaluation purposes. Furthermore, cross-validation is another methodology during training that allows to evaluate performance on a portion of the data that is not training and testing. Subject matter experts in supervised learning algorithms could lend guidance on the selection of the noise parameters and relevant functions (e.g. covariance, likelihood, etc.) for the GP model. Greater intuition from mathematical theory on the selection of certain functions could enable better training of the model.

Based on the trained GP model from this paper, negative outputs are possible within predictions for TBE and RD. One facet of this could be due to the inherent assumption that given the multi-dimensional input space of mission classifiers, the posterior distribution is Gaussian. The successful implementation of a Weibull likelihood in the GP model is one possible way to predict non-negative values. Another possibility is constructing a new optimization problem for a GP if the Gaussian distribution assumption holds such that a positive data constraint is applied.

The lack of a large dataset for safing events inhibits the effectiveness of machine learning algorithms. One way to further infer about a population from a small dataset is by employing bootstrapping - sampling with replacement. Optimal Weibull parameters could be obtained for subsets of data by re-sampling the given subset and computing average parameters that would yield estimates to the true probability distribution. Bootstrapping could also be used in GP model estimation as a means to increase the size of the training, validation, and testing datasets. Statistical consideration must be given when employing bootstrapping on the posterior distribution the model creates.

While only parametric analyses are considered in this paper, nonparametric analysis techniques such as the Kaplan-Meir estimator can be considered to understand the true nature of the data. One key aspect of nonparametric models is the censoring of data - when failure data is incomplete. When a mission reaches the end-of-life, one could apply right-censoring since that time should not be modeled as another safing event and would be stochastic across many missions. While initial thoughts were formulated, no sufficient conclusions are made on how best to censor data. The parametric framework developed to compute 1-Weibull and 2-Weibull distributions has censoring included in the formulation; thus, it should be easy to implement and obtain new optimal Weibull distributions.

Validating the GP model and Weibull models against the past mission data is necessary to begin to understand and bound prediction errors. This validation process could attempt to calculate the actual MIR for each of the 21 missions in the database to provide a baseline for model comparisons. The initial methodology would be to remove one missions data from the dataset, recalculate the model parameters for time-between-events and recovery duration, simulate the mission profile of the removed mission, and compare the simulations outputs with the removed missions data. These comparisons could also guide which percentiles should be targeted on the MIR CDFs.

Challenges remain in architecting this validation process. Nearly every mission in the database has an event with incomplete recovery timeline data and the precise MIR may be indeterminable. While some data may be able to be located through further data mining, non-locatable records will require a new set of assumptions to be derived on how to effectively represent the missing data when calculating the MIR. The model validation process and comparison of several safing event models will be the subject of a future paper.

Finally, regardless of how accurate a model is developed, it is still limited by the data by which it is defined. As more interplanetary mission safing events occur, it is imperative to continue to collect data and store this additional information in the database. Then, when a 'significant' amount of data is added, re-training of the models may be useful to incorporate new information and re-weight accordingly.

7. CONCLUSION

Building on the work done by Imken et al., this paper lays out the architecture for multiple models to be used within the mission simulation framework, implements different predictive models, and uses the NeMO mission concept as a case study for simulating mission inoperability rates and exploring their implications. Missions that utilize solar electric propulsion such as the NeMO concept can benefit from accurately modeling safing events since long periods of continuous operations are vital for their missions.

With the collection of the safing events dataset, subsets are created based on common mission classifiers; a total of seven categories are created. To assess mission inoperability, a generalized Monte Carlo simulation is implemented to quantify the likelihood of realizing the worst-case inoperability rates. While the simulation is flexible to any type of model, a Gaussian Process model and two Weibull Distribution models are incorporated in order to simulate time-between-events and recovery durations. The architecture established for both the simulation and the models maintains simplicity for integration and a level of generality such that future enhancements are possible.

The Weibull Distribution model parametrically represents the

complete safing event database through the use of single Weibull distributions and mixtures of two-Weibull distributions. Using the maximum likelihood estimation algorithms, optimal Weibull distribution parameters are computed for the full dataset. The results indicate that the 1-Weibull is a better predictor for the time-between-events dataset while the 2-Weibull is a better model for the recovery duration dataset; this selection is used for the Mixed Weibull model. The advantage of employing Weibull distributions for modeling is the ease of implementation as it is defined by just a few parameters.

However, in order to generate predictions of safing events based on multiple user-defined inputs, a new approach for predictions is required. A Gaussian Process model met this criteria; it is a type of supervised learning algorithm that is trained and tested using the existing safing event dataset. By modeling time-between-events and recovery durations through a GP model, a posterior probability distribution can be obtained for multivariate inputs.

The GP and Weibull models presented in this paper have been verified to work with the safing event simulation architecture. Using the Next Mars Orbiter mission concept as a case study, the developed single Weibull distribution, 2-Weibull mixture distribution, and the Gaussian Process model act as predictive models, generating the likelihood of inoperability rates, outage times, and number of safing events for its simulated mission life. The binned Weibull comparisons demonstrate how certain past missions with similar classifiers as the mission of interest provide insight into the weighting that the GP model places on inputs during training. Furthermore, the results show that within a few tenths of a percent in mission inoperability rate, all three predictive models give results within the family of one another. Recommendations made for the NeMO concept include increasing spacecraft margins for missed-thrust periods as well as increasing operational and on-board fault management capabilities if the simulated inoperability is assumed to be too large.

In the area of predictive analytics, this paper uses standard statistical methodologies and supervised learning algorithms to develop, train, and test predictive models for a sample mission scenario - the Next Mars Orbiter. Not only is this work a step towards creating a more complete tool for safing event analysis and prediction, but also the results could help mission designers to consider the effects of safing events on spacecraft margins and requirements.

ACKNOWLEDGMENTS

The authors would like to thank Rob Lock and the Mars Formulation Office at the Jet Propulsion Laboratory for funding this research and Dr. David Spencer (Purdue) for enabling the funding mechanism to the Georgia Institute of Technology. The first author would like to thank Dr. Joseph Saleh (GT) for initial guidance on analyzing reliability data for spacecraft. The first author would also like to thank Marcus Pereira, a graduate student in the Autonomous Control and Decision Systems Lab under Dr. Evangelos Theodorou at Georgia Tech. His guidance on supervised learning algorithms for regressions helped formulate and train the predictive model.

This work was carried out by the Space Systems Design Laboratory at the Georgia Institute of Technology, under contract to the Jet Propulsion Laboratory, California Institute of Technology.

APPENDIX

Table 7 shows all of the missions that are contained in the safing event database. It lists how the missions are classified for the mission class, destination, duration, and solar electric propulsion categories. Tables are presented on the next page.

REFERENCES

[1] J. Wertz, D. Everett, and J. Puschell, *Space Mission Engineering: The New SMAD*, ser. Space technology library. Microcosm Press, 2011. [Online]. Available: https://books.google.com/books?id=VmQmtwAACAAJ

[2] *Emerging Capabilities for the Next Mars Orbiter*. Mars Exploration Program Analysis Group (MEPAG), February 2015.

[3] T. K. Imken, T. M. Randolph, M. DiNicola, and A. K. Nicholas, "Modeling spacecraft safe mode events," *IEEE Aerospace Conference*, March 2018.

[4] J.-F. Castet and J. H. Saleh, "Satellite reliability: Statistical data analysis and modeling," *Journal of Spacecraft and Rockets*, vol. 46, no. 5, pp. 1065–1076, Sep 2009. [Online]. Available: https://doi.org/10.2514/1.42243

[5] G. F. Dubos, J.-F. Castet, and J. H. Saleh, "Statistical reliability analysis of satellites by mass category: Does spacecraft size matter?" *Acta Astronautica*, vol. 67, no. 5-6, pp. 584–595, 2010.

[6] E. Siegel, *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*. Wiley, 2016.

[7] S. R. Pujari and E. G. Lightsey, "A statistical analysis and predictive modeling of safing events for interplanetary spacecraft," Master's thesis, Georgia Institute of Technology, Atlanta, Georgia, USA, 2018.

[8] C. Bishop, *Pattern Recognition and Machine Learning: All "just the Facts 101" Material*, ser. Information science and statistics. Springer, 2013. [Online]. Available: https://books.google.com/books?id=HL4HrgEACAAJ

[9] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptative computation and machine learning series. University Press Group Limited, 2006. [Online]. Available: https://books.google.com/books?id=vWtwQgAACAAJ

[10] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *Journal of Machine Learning Research*, vol. 11, no. Nov, pp. 3011–3015, 2010.

[11] C. Rasmussen and C. Williams, *The GPML Toolbox version 4.1*, November 2017, http://www.gaussianprocess.org/gpml/code/matlab/doc/manual.pdf.

[12] J. H. Saleh and J.-F. Castet, *Spacecraft reliability and multi-state failures: a statistical approach*. John Wiley & Sons, 2011.

[13] E. E. Elmahdy and A. W. Aboutahoun, "A new approach for parameter estimation of finite weibull mixture distributions for reliability modeling," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1800 – 1810, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0307904X12002545

Table 7. Mission Classifier Inputs

Mission Name	Class	Destination	Duration [years]	Electric Propulsion
Dawn	Small	Asteroid/Comet	10-15	Yes
Deep Impact	Small	Asteroid/Comet	5-10	Yes
Deep Space 1	Small	Asteroid/Comet	0-5	No
Genesis	Small	Heliophysics/Exoplanet	0-5	No
Lunar Reconnaissance Orbiter	Small	Moon	5-10	No
Mars Atmosphere and Volatile Evolution	Small	Mars	0-5	No
Mars Climate Orbiter [1]	Small	Mars	0-5	No
Mars Global Surveyor	Small	Mars	5-10	No
Mars Odyssey	Small	Mars	15-20	No
Mars Polar Lander [1]	Small	Mars	0-5	No
Phoenix Mars Lander [1]	Small	Mars	0-5	No
Stardust	Small	Asteroid/Comet	10-15	No
Juno	Medium	Jupiter	5-10	No
Kepler	Medium	Heliophysics/Exoplanet	5-10	No
Mars Reconnaissance Orbiter	Medium	Mars	10-15	No
New Horizons	Medium	Kuiper Belt Object	10-15	No
OSIRIS-REx	Medium	Asteroid/Comet	0-5	No
Spitzer	Medium	Heliophysics/Exoplanet	5-10	No
Cassini	Large	Saturn	15-20	No
Galileo	Large	Jupiter	10-15	No
Mars Science Laboratory [1]	Large	Mars	0-5	No

[1] **Cruise phase of mission only**

BIOGRAPHY

Swapnil R. Pujari *received a B.S. and M.S. in Aerospace Engineering from the Georgia Institute of Technology in 2016 and 2018 respectively. He currently is a systems engineer at The Boeing Company working on the SES O3b mPOWER and ViaSat-3 programs. He is involved with thermal analyses as well as thermal/vacuum testing. At his time at Georgia Tech, Swapnil was involved in small satellite missions. He was the mechanical lead and chief systems engineer of the Prox-1 Microsatellite mission, the payload lead to the Tether and Ranging (TARGIT) CubeSat Mission, and the electrical power subsystem lead on MicroNimbus, a 3U radiometer CubeSat Mission. He has interned twice with the Jet Propulsion Laboratory under the Mars Formulation Office working on the Next Mars Orbiter and Mars Sample Return Lander concept missions.*

Travis Imken *received a M.S. in Aerospace Engineering from the University of Texas at Austin in 2014 and is in the JPL Project Systems Engineering and Formulation Section. He serves as a Deployment Phase Systems Engineer for the InSight Lander, overseeing placement of the missions payloads on the Martian surface. Travis also supports the RainCube mission as the Project Systems Engineer. Past projects include Mars Sample Return, ARRM, and the Lunar Flashlight and NEA Scout deep space CubeSats. He is also involved with JPLs Innovation Foundry, serving as a systems engineer on Team X/Xc as well as a small satellite expert with the A Team.*

E. Glenn Lightsey *is a Professor in the Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology. He received his Ph.D. from Stanford University in 1997. He is the Director of the Space Systems Design Lab at Georgia Tech. His research program focuses on the technology of satellites, including: guidance, navigation, and control systems; attitude determination and control; formation flying, satellite swarms, and satellite networks; cooperative control; proximity operations and unmanned spacecraft rendezvous; space based Global Positioning System receivers; radio navigation; visual navigation; propulsion; satellite operations; and space systems engineering. He has written more than 140 technical publications. He is an AIAA Fellow, and he serves as Associate Editor-in-Chief of the Journal of Small Satellites and Associate Editor of the AIAA Journal of Spacecraft and Rockets.*