Proceedings of the ASME 2018 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2018
August 26-29, 2018, Quebec City, Canada

DETC2018-85137

MODEL VALIDATION IN EARLY PHASE OF DESIGNING COMPLEX ENGINEERED SYSTEMS

Elham Keshavarzi
Ph.D Candidate
Mechanical Engineering
Oregon State University
Corvallis, OR, USA

Kai Goebel
Intelligent Systems Division
Coordinator
Ames Research Center
Moffett Field, CA, USA

Irem Y. Tumer
Professor
Mechanical Engineering
Oregon State University
Corvallis, OR, USA

Christopher Hoyle
Associate Professor
Mechanical Engineering
Oregon State University
Corvallis, OR, USA

ABSTRACT

In design process of a complex engineered system, studying the behavior of the system prior to manufacturing plays a key role to reduce cost of design and enhance the efficiency of the system during its lifecycle. To study the behavior of the system in the early design phase, it is required to model the characterization of the system and simulate the system's behavior. The challenge is the fact that in early design stage, there is no or little information from the real system's behavior, therefore there is not enough data to use to validate the model simulation and make sure that the model is representing the real system's behavior appropriately. In this paper, we address this issue and propose methods to validate the model developed in the early design stage. First we propose a method based on FMEA and show how to quantify expert's knowledge and validate the model simulation in the early design stage. Then, we propose a non-parametric technique to test if the observed behavior of one or more subsystems which currently exist, and the model simulation are the same. In addition, a local sensitivity analysis search tool is developed that helps the designers to focus on sensitive parts of the system in further design stages, particularly when mapping the conceptual model to a component model. We apply the proposed methods to validate the output of failure simulation developed in the early stage of designing a monopropellant propulsion system design.

Keywords: Model Validation, early design phase, failure simulation, final behavior

INTRODUCTION

The question is how do we make sure that the behavior simulated by the model selected in the early design phase is correct? The approach used in this paper to answer this question is *model validation*. In discussing model validation, we will also discuss the related concepts of *model selection* and *model calibration*. The reason for this is to understand the statistical methods associated with these related concepts to determine if they have applicability in the early design phase.

Model validation studies the validity of the model to represent the real system's behavior (Sargent, 1987). There are many approaches to validate a computer model or mathematical model; however, there is not an applicable method to show how a designer can validate a model developed in the early design stage when the real system or prototype is not available. It is difficult to validate a model in early design since very little is known at this time. Validation at this phase must consider not only traditional model validation, but also suggestions for further design steps.

The objective of this paper is to propose methods to validate the simulation generated by a model developed in the early design stage, when the complete system or prototype is not available, particularly to ensure that it the model is a good representation of the real system.

Following section investigates existing methods and discusses the limitations of each method and the reasons they are not applicable to validate the result of a selected model in the early design stage.

MODEL VALIDATION METHODS

In this part, we investigate strategies to validate models based on the available data. By looking at the literature in model validation, we gain insights into approaches that can be applied or adapted for our problem. Models are tools to represent and study a real system. Popular models to represent a system are mathematical, statistical, computer, component, functional or network models. Model validation is a necessary step to make sure that the model simulation reflects the real behavior of the system. Later in this paper, we discuss the limitations of model validation in the early design. It is shown that existing validation methods are based upon strong assumptions, and therefore they are not always applicable to validate simulation in the early design stage, when little or inadequate information is available from the real system's behavior. Different approaches to assessing model validity are described.

A complex engineered system is built for a specific set of purposes or objectives, such as improving the sustainability, safety, reliability, performance etc., while managing the cost of the system; this is a challenge that engineers face from the beginning of the design process. The first step in designing such systems is to develop a model and simulate the behavior and characteristics of the system to enable the designers to study the system, and decide upon strategies to improve the design features regarding the objectives. Model validation is a necessary step to ensure that information obtained from the model is within an acceptable range of accuracy consistent with reality (Sanders, 1996; Lewis, 1992; Law, 2008; Cook and Skinner, 2005). There are many approaches that have been used to validate a model. We classify the approaches into three groups based on the amount of available true data to validate a model. For the aim of this paper, when there are enough data available to assume a probability distribution, the classification is called *adequate data*. When there are some true data available to validate the model, but not enough to assigning a probability distribution, it is called *little data*. When there is no data and the only available source to validate a model is expert knowledge, it is a *no data* situation. Table 1. represents the classification of the model validation methods based on the available data. In the first class, enough true data is available to validate the model under study. For this class, standard statistical methods can be used to test if the model under review produces the result similar to the true data by comparing the true data and output generated by the model. True data can be obtained from another valid model, historical data, prototype, or a real system. In the second class, there is not enough true data available to validate the whole model; however, observations from some subsystems are available. In this class, a prototype or real system is not manufactured or historical data of the complete system behavior is not available; however, observations from some subsystems can be applied to validate parts of the model. If the observations and model output are discrete values, the *chi-squared test* is the tool to test the validity; otherwise, for continuous values, the *K-S test* is the appropriate tool to test the similarity of the observed data from subsystems and output from the model. In the third class, there is no data available to validate the model. In this case, the only source to validate the model is expert knowledge. Methods like Failure Modes and Effect Analysis (FMEA) or evidence theory are applied to quantify the expert knowledge and validate the model. In regard to the objective of this paper to validate the output from the model developed in early design stage, class 2 or 3 in Tab. 1. is typically the case, because in the early design stage no or little data from the complete system is available.

Table 1. Model validation methods and assumptions

Class	Available Data	Source	Tools to Test Validity
1	Adequate Data	Historical Data, Another Valid Model, Real system/Prototype	Standard statistical Methods, Sensitivity Analysis
2	Little Data	Subsystems	Chi-Squared for Discrete Samples K-S Test for Continuous Samples
3	No Data	Experts knowledge	FMEA, Evidence Theory

The following sections provide more details on each class.

Adequate True Data to Validate Model

Adequate true data can be utilized to select, calibrate and validate a model with familiar statistical methods. True data might be obtained from a real system or prototype observations, or a valid model. Having enough true data is desired to develop, select, test or validate a model, however most of the time in complex engineered system design, this is not the case. In the next subsections, we will discuss techniques for model selection and model calibration, in addition to model validation, to understand how adequate data can be used in the modeling process.

Model Selection

If enough data is available from the real system's behavior, cross-validation method can be applied to select the model with closest behavior to the real system.

Cross-validation is a technique to select the model that has lowest Mean Square Error (MSE) with respect to the true data. It is applicable when there are different models and the goal is to select the one that generates output similar to the true data (Smyth, 2000; Kohavi, 1995; Stone, 1977).

Another application of the cross-validation method is to build a model when enough true data is available. This has been widely used to build predictive models e.g., sales predictions models. In this application, the dataset is divided to two parts, one part is used to build a predictive model and another part is utilized to test the model. Survey paper by Arlot and Celisse (2010) provides more details about cross-validation method.

A method to select a model among two candidate models, when true data is available is *Likelihood Ratio (LR)* test. It is assumed that a proper statistical distribution can be assigned to data generated by each one of the candidate models and there are no unknown parameters to estimate. According to Vuong (1989), the hypothesis for the likelihood ratio test specifies if there is a meaningful difference between the parameter obtained from the first model and the parameter from the second model:

$$H_0 :\ \theta_1 = \theta_2$$

$$H_A : \theta_1 \neq \theta_2$$

where θ_1 is the parameter of the first model and θ_2 is the parameter of the second model. Equation (1) demonstrates the likelihood ratio test:

$$LR = \frac{L(\theta_2|d)}{L(\theta_1|d)} \qquad (1)$$

The true dataset, d, (obtained from valid model or observation or real system) provides the maximum likelihood estimate of the parameter. The ratio selects the model that is more likely to produce true dataset d. In other words, the selected model has the parameter with smaller difference from maximum likelihood estimator (Banerjee, et al., 2008)

Bayes Factor is a popular method to decide between two models when there is true data, d. This method has been used in a wide range of fields (Wasserman, 2000; Raftery, 1995; Berger and Pericchi, 1996). It is assumed that a proper statistical distribution can be assigned to data generated by each one of the candidate models. Consider M_1 and M_2 are two candidate models and we are interested in selecting the one that is more likely to produce given true data *d*. Minhas et al. (2014) present the Bayes factor B_{12} illustrated in Eq. (2) to decide between model M_1 and M_2:

$$B_{12} = \frac{P(M_1|d)/P(M_2|d)}{P(M_1)/P(M_2)} \qquad (2)$$

Where $P(M_i)$ is the prior probability distribution of model M_i and $P(M_i|d)$ is the posterior probability distribution of the model M_i generates the data. When the Bayes factor is larger than 1, model M_1 is selected, and when the Bayes factor is less than 1, model M_2 is selected. Bayes factor equal to 1 does not provide any evidence to select one model over the other.

Model Calibration

True data can be used to calibrate the model. Calibration is widely used in computer-based models. In the calibration technique, some particular model parameters are considered fixed and unknown and using the true data, the best match of the parameters is found. In addition, a *bias factor* is typically added to the model to account for lack of fit of the model to the data (i.e., modeling error).

The limitation of the calibration technique is that in practice model parameters are not fixed and they change in each run of simulations. Xiong et al. (2009), address the limitation of using true data to calibrate a model assuming the parameters are fixed. They proposed a Maximum Likelihood Estimation (MLE) method to assign a distribution to random changing parameters of the model and to calibrate those parameters utilizing true data.

Model Validation

When there is a model developed to represent the real system's behavior and we are interested in validating the model using true data, many standard statistical tests can be applied. In this case, the validation phase focuses on comparing the true data, with the corresponding elements of the model simulation to determine whether the differences are acceptable. If the result shows that the model simulation is different from the true data, changes should be made to the model. Making changes in the model has to be continued until designers make sure that the model is a good representation of the real system, and the difference between the simulation and true data is not significant.

Another tool to validate a model using true data is sensitivity analysis. In this technique, designers evaluate the confidence in a model based on the relationship between inputs and outputs of the model simulation. This requires studying the effect of each input on the output variation. The same input-output relationship is expected to occur in the real system (Saltelli, 2000; Wagner, 1995; Triantaphyllou and Sánchez, 1997)

Sensitivity analysis requires repeating the simulation many times; therefore, it can be expensive when the simulation takes a long time to run or the model has too many input parameters. Another limitation of sensitivity analysis is the assumption of the independent input parameters. Some sensitivity analysis techniques assume independency between model inputs; however, in designing complex engineered systems, designers have to pay extra attention to the model inputs, since they can be highly dependent. Dependent inputs shows greater variation in the model output compared to an independent parameter.

Regression is a simple tool for sensitivity analysis. Regression coefficients are representation of the sensitivity of the response variable (output) to each one of the dependent variables (inputs) (Storlie et al., 2009; Hurvich and Tsai, 1989). The limitation of the regression method is that it assumes a linear relationship between the model output and the input parameters (variables). Also, it is assumed that there is only one output for the model. However, in designing complex engineered systems, there could be more than one output or a non-linear relationship between the input parameters and model output. When there is more than one output, sensitivity analysis can be performed for each output separately but, it is hard to interpret the result in a correct way. In such cases, variance-based sensitivity analysis is recommended.

The variance-based, or global, sensitivity analysis SOBOL method is a sensitivity analysis method based on probabilistic framework. It breaks the variation of the model output into fractions, each fraction is assigned to one model input (Saltelli et al., 2010; Marzban, 2013; Mara and Tarantola, 2012). The local sensitivity is a method to complement the SOBOL analysis. In this technique, the fractions are produced by taking partial derivatives of the output regards to each input $\left|\frac{\partial Y}{\partial X_i}\right|$.

Little Data to Validate Model

When there is little true data to validate a model, assigning a probability distribution is not reasonable. In this case, *non-parametric* tests are better strategies to compare the data generated by model to the true data observed from real system or prototype or a valid model, because non-parametric tests are distribution-free, it means they are not based on probability distribution assumption for the data.

If true data and simulated data are categorical, a *chi-squared test* can be used to validate the model (Mantel, 1963; Conover and Iman, 1981). The null hypothesis is that frequencies for the true data and the simulated data are equal. Schoenfeld (1980) illustrates that under the null hypothesis, the test statistic has approximately a chi-squared distribution with $n-1$ degrees of freedom. For n categories, the test statistic is calculated by Eq. (3).

$$Test\ Statistic = \sum_{i=1}^{n} \frac{(Observed_i - Simulated_i)^2}{Simulated_i} \qquad (3)$$

If true data and simulated data are continuous values, and non-categorical, the *Kolmogorov–Smirnov* test is a well-known technique to validate the model (Conover, 1972; Massey, 1951). The null hypothesis in a Kolmogorov-Smirnov test is simulation and observations are from the same distribution, the alternative is that they are not from the same distribution. If the test result is 1, the data provides enough evidence to reject the null hypothesis; if the test result is 0, we fail to reject the null hypothesis. Rejecting the null hypothesis means the simulation and true data are not from the same distribution. In this case, the model is not adequate to represent the real system.

In the early design stage, there is no information available from the real system's behavior. The prototype of the system is not manufactured yet and historical data from the complete system does not exist for a new design. In this case, the methods based on statistical distribution assumptions are generally not applicable.

The limitations and assumptions of model validation in early design stage are the main motivation for this paper. Specifically, we are interested in answering the question *how we can make sure that the failure behavior simulation by the selected model in the early design stage is a good representation of the real system's behavior*. The main challenge to answer to this question is that there is no real system or prototype manufactured in the early design stage and there is no historical data available from the complete system. Therefore, model validation for the selected model simulation must be built upon data from some subsystems and expert knowledge as the only source of information available in the early design stage. In the following section, we propose methodologies to validate the fault behavior simulated by the selected functional model for the monopropellant propulsion system in the early design stage.

No Data to Validate Model

When there is no true data to validate a model, one strategy is to use expert knowledge to validate the model. The challenge is to quantify the expert knowledge in a meaningful way to be able to compare with the model output. When probability theory and distribution assumptions are not applicable, evidence theory is a good approach. Evidence theory is also called *Dempster–Shafer* theory because Shafer (1992) developed the idea presented by Dempster. In evidence theory, the basic idea to quantify expert knowledge is using *Basic Belief Assignment (BBA). BBA* can be described as the degree of trust in an element.

An example of using BBA is provided as follow. Assume in designing a complex engineered system, $X = \{x_1, x_2, x_3, x_4, x_5\}$ where x_1 to x_5 are uncertain elements that we are interested in quantifying using expert knowledge. The elements are mutually exclusive of each other: x_1 defines system behaves nominally, x_2 to x_4 define different degraded states of the system, and x_5 is failure mode of the system. BBA quantifies expert knowledge using a mapping function m. The number $m(A)$ is in range zero to one and illustrates the portion of total expert's belief in element A, when $A \epsilon X$ (Sentz and Ferson, 2002; Zadeh, 1984). Bae et al. (2004) explained when there are two BBA, m_1 and m_2, provided by different evidence sources, e.g., different expert knowledge, Dempster's equation can be utilized to get the combined BBA presented in Eq. (4).

$$m_A = \frac{\sum_{c_i \cap c_j = A} m_1(c_i) m_2(c_j)}{1 - \sum_{c_i \cap c_j = \emptyset} m_1(c_i) m_2(c_j)} \tag{4}$$

where C_i and C_j are propositions from knowledge sources m_1 and m_2. In Eq. (4), $\sum_{c_i \cap c_j = \emptyset} m_1(c_i) m_2(c_j)$ is the conflict among two independent sources of knowledge. Dempster's rule resolves the conflict by normalization. It is more reasonable to present a range instead of a single number to quantify belief or trust or confidence of experts. In evidence theory, the range is expressed as $[Bel(A), Pl(A)]$. Equations (5) and (6) show how to obtain the range.

$$Bel(A) = \sum_{c \subset A} m(c) \tag{5}$$

$$Pl(A) = \sum_{c \cap A \neq \emptyset} m(c) \tag{6}$$

where $Bel(A)$ is the total degree of belief. It is the summation of $BBAs$ for sets including complete proposition A.

As defined by Bae et al. (2004), plausibility for A, $Pl(A)$, is the summation of $BBAs$ of propositions that do not have empty intersections with proposition A. In fact, $Pl(A)$ is the total BBAs that agree with proposition A totally or partially. Evidence theory uses the described equations to quantify the degree of belief or trust or confidence of experts to a model output and help decide about the validation of the model.

Failure Mode and Effect Analysis (FMEA) is another strategy to bring the ideas of experts take into account. FMEA describes failure of the system as well as causes and effects of the failures. The results of FMEA are used to consider design changes that may be necessary to reduce risk. It is assumed that experts are able to judge the models based on the outputs (Stamatis, 2003; Van Leeuwen, 2009; Goddard, 1993).

VALIDATION OF FAILURE SIMULATION IN EARLY DESIGN PHASE FOR A MONOPREPELLANT PROPULSION SYSTEM

In our previous paper (Keshavarzi et al., 2017), we proposed a design method in early design phase using functional models. In the proposed method, a population of functional models is generated and the potential failure scenarios are simulated; finally, applying a cost-risk model, the most resilient design is selected. In this paper, we are interested in validating the result of simulation created by a selected model in the early design phase. In case of monopropellant propulsion design, the selected model is a *functional model*. A functional model is a structured representation of the functions required to meet system requirements (Conover, 1972). The purpose of a functional model is to describe the system behavior and determine vulnerable parts of the design, resulting in potential system improvement. Figure 1. illustrates the selected functional model for the monopropellant propulsion system. In the selected design, the propellant plays the role of cooling material and the extra heat produced from propulsion helps to expand inert gas and preheat propellant for the better propulsion.

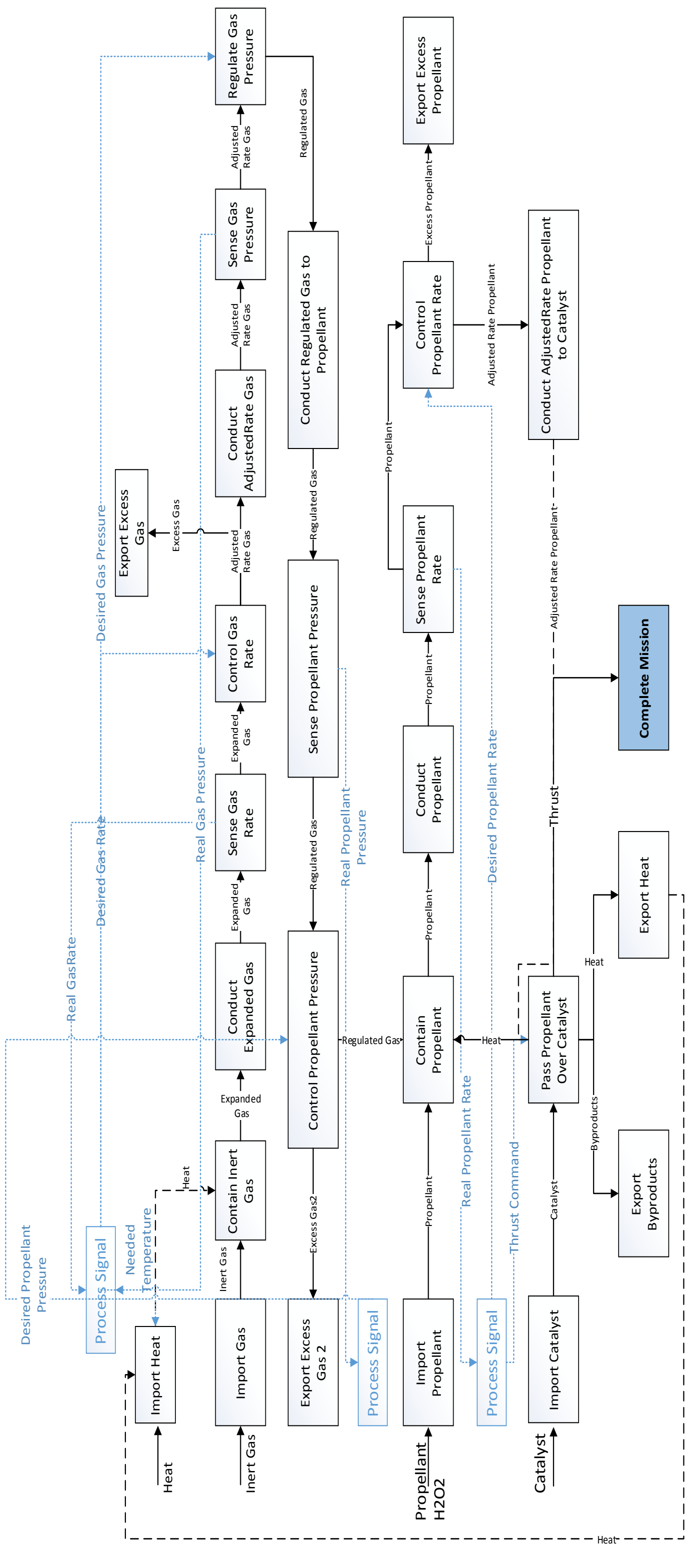

Figure 1. Selected functional model in early design stage for monopropellant propulsion system

The failure behavior simulation result using the selected function model is presented in Fig. 2. The final behavior of the system can be classified to nominal (the spacecraft complete the mission), too much thrust (the spacecraft passes the target orbit), too little thrust (the spacecraft doesn't reach to the target orbit), no thrust (the spacecraft does not move), and catastrophic behavior like explosion or crash to the space debris.

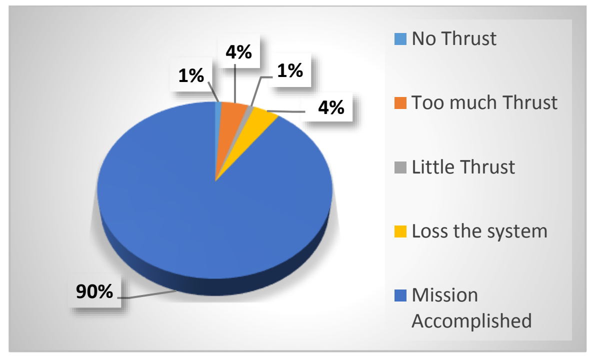

Figure 2. Failure simulation result for a monopropellant propulsion system

The question is how to validate the fault behavior simulation generated by the selected model. To answer this question, we apply ideas from model validation methods using *expert knowledge*, *historical data* from sub-systems and *sensitivity analysis* to develop three practical methodologies to guide the designers to validate the model developed in the early design stage. To apply expert knowledge to validate the model in early design stage, we propose a *FMEA* based method to quantify expert knowledge and validate the model selected in early design. Existing validation methods based on historical data are not directly applicable, since it is assumed that historical data of real system behavior are not available; therefore, we made modifications to the idea to enable the designers to validate the fault behavior generated by the selected functional model in the early design stage utilizing the observed data from one or more subsystem/s operating in other systems. For this method, we use a *chi-squared test*. The *sensitivity analysis* method is based on the partial derivatives of the cost model and it guides designers to focus on sensitive parts of the system, specifically in later design stages. These three methodologies are described in detail in the following subsections.

Apply Expert Knowledge to Validate Functional Model

One approach to validate a model is to have experts with deep knowledge to confirm the simulation results and decide about validity of the model. This strategy is commonly used when data (observations) of real system's behavior is not available.

In this method, experts should be involved in the design process to be able to provide the designers effective technical advice and validate the model through analyzing the failure simulation outputs. For example, in the monopropellant propulsion system design, experts from NASA Ames were involved in the design progress. Based on NASA Ames expert knowledge, a FMEA was conducted to validate the result from fault behavior simulated by functional model. Failure modes and effects analysis (FMEA) is a step-by-step approach for identifying all possible failures in a design of the system.

Failures are defined as any errors that affect the completion of the mission for the monopropellant propulsion system. The effects analysis part of FMEA addresses the cause and consequences of each failure. Failures are sorted considering the severity of the consequences of the failures and the chance of occurrence. FMEA is not a one-time analysis: it has to be done in the early design phase, as well as in further design steps and in the lifecycle of the system.

The main goal of the FMEA in design process and operation control is to provide improvement strategies to reduce the failures, particularly the ones with higher priority rank (Teng, and Ho, 1996; Vesely et al., 1981). In this paper, we use a FMEA to quantify expert knowledge about fault behavior of the system in the early design stage when there is no information available about complete system's behavior. The setup to quantify the expert knowledge is exploring their ideas about possible failure scenarios and probabilities of failure. The schematic developed by Tang et al., (2007) shown in Fig. 3. was applied to develop the FMEA table.

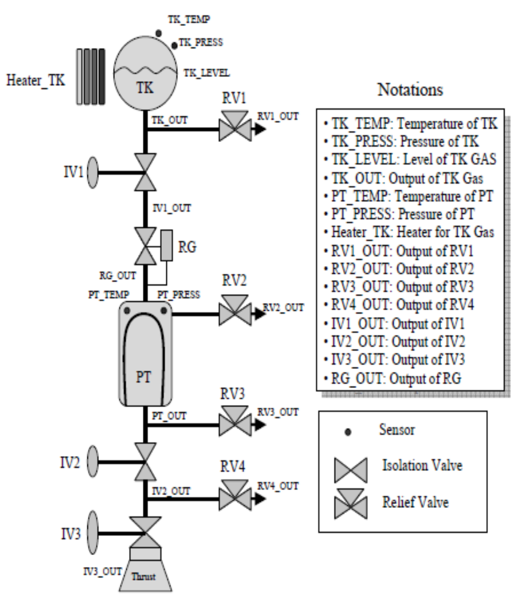

Figure 3. A monopropellant propulsion system schematic (Tang et al., 2007a).

The failures that affect system *mission accomplished* are divided into four classes, and in each class the severity and the occurrence have been decided by NASA Ames experts. Table 2. represents a developed FMEA for the monopropellant propulsion system based on NASA Ames documents and asking their experts about the occurrence of each failure mode (probability) and severity. The severity is a number from 1 to 10, where 10 is the highest and 1 is the lowest. The occurrence is a percentage which shows the number of failure occurrences every 100 missions.

Table 2. FMEA for monopropellant propulsion system

Potential Failure Mode	**Potential Effect of Failure**	**Class**	**Occur in hundred**	**Severity**
Valve IV1 stuck closed	Failure of the system to provide thrust when commanded	1	3	9
Valve IV2 stuck closed	Failure of the system to provide thrust when commanded	1		
Valve IV3 stuck closed	Failure of the system to provide thrust when commanded	1		
Leakage in IV1	Failure of the system to provide thrust when commanded	1		
Regulator failure	Failure of the system to provide thrust when commanded	1		
Low propellant level	Failure of the system to provide thrust when commanded	1		
Valve IV3 is stuck open	Continued firing after the system has been commanded off	2	2	9
Timer Relay K6 fails to disengage	Continued firing after the system has been commanded off	2		
Switch S3 failure	Continued firing after the system has been commanded off	2		
Pressure sensor on RG fails	Continued firing after the system has been commanded off	2		

Pressure Sensor TK failures	Continued firing after the system has been commanded off	2		
Pressure Sensor PT failures	Continued firing after the system has been commanded off	2		
Temperature Sensor TK failures	Continued firing after the system has been commanded off	2		
Temperature Sensor PT failures	Continued firing after the system has been commanded off	2		
Abnormal inert gas path pressure	Inadequate gas pressure	3		
Abnormal propellant gas path pressure	Inadequate gas pressure	3		
Pressure regulator failure	Inadequate gas pressure	3	4	8
Low propellant due to leakage	Inadequate gas pressure	3		
Low inert gas due to leakage	Inadequate gas pressure	3		
Catastrophic Failure	Explosion or lose control of the system	4	0.1	10

NASA Ames experts concluded that regulator and isolation valve number 4 plays a significant role in mission accomplishment, and they decided to improve the design to that shown in Fig. 4., with a redundant regulator and isolated valve. The system can switch the gas path to the redundant (IV4-RG2) path when a fault in either IV1 or RG occurs (Tang et al., 2007a, 2007b).

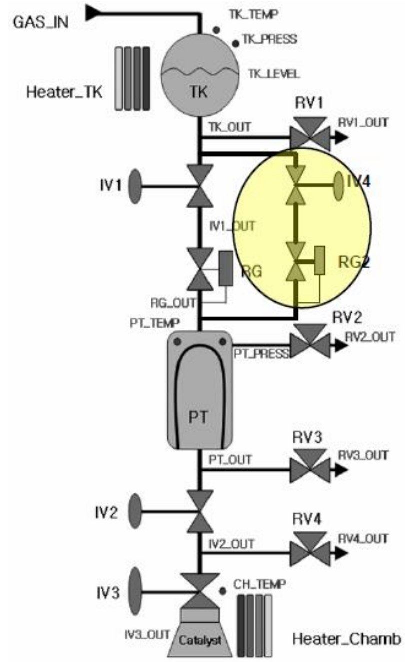

Figure 4. Parallel redundant regulator and isolation valve (Tang et al., 2007a).

The result obtained from functional model failure simulation is compared to the FMEA. As shown in Tab. 2., effects of failures can be classified into 3 categories: 1) failure of the system to provide thrust when commanded, 2) continued firing after the system has been commanded off, and 3) inadequate gas pressure. These groups are the same as the end states or final behaviors that we defined for the functional model's failure simulation. The classes of the failures, occurrence, and severity from the FMEA table matches with our developed end states and probabilities obtained from classification of failure scenarios. As shown in Tab. 3., for each category of effects from expert knowledge, the summation of the occurrence provides the probability of that type of failure.

Table 3. Proportions of each category for expert knowledge and simulation

Expert Knowledge		**Simulation**	
Failure Effect	**Proportions (Percentage)**	**Undesired End States**	**Proportions (Proportions)**
Failure of the system to provide thrust when commanded	3%	No Thrust	1%
Continued firing after the system has been	2%	Too much Thrust	4%

commanded off			
Inadequate gas pressure	4%	Little Thrust	1%
Catastrophic Failure	0.1%	Loss the system	4%
No Failures	90.9%	Mission Accomplished	90%

Table 3. provides the proportion of each category occurrence based on the simulation and expert knowledge. We are interested to know if there is any significant difference between the simulated proportions and the proportions claimed by experts for different categories. Using a *chi-squared* test to compare the probabilities quantified from expert knowledge and probabilities obtained from simulation resulted in a p-value of 0.1266 which shows statistically there is no evidence to reject the null hypothesis, both sets of proportions are similar. Also, our definition of undesired final behavior and the probability of having each undesired end state obtained from failure simulation using functional model were discussed and approved by a NASA Ames expert on September 28, 2016 at Oregon State University.

Apply Subsystem Data to Validate Functional Model

This validation phase focuses on utilizing available observed data from some subsystems to validate the simulation results when there is no data available from the entire system behavior.

There are statistical tests to specify whether the difference between observation and simulation is meaningful or not. There are parametric methods that require a distribution assumption. When observed and simulated data are discrete or categorical, and no distribution assumption is made, non-parametric methods are more precise and robust. What we obtain from a selected functional model simulation in the early design stage is the failure scenarios, which can be divided into distinct categories based on the final behavior of the system.

In the monopropellant propulsion system, there is no data from the real system available to validate the functional model simulation, and manufacturing a prototype is not technically or financially feasible in the early design phase. However, there are subsystems that are built into *other engineered systems*. A NASA Ames expert provided us the information from three years of operation of an outer space system with a similar regulating gas subsystem. In each year, the system completes three missions on average. In three years, only one time the *regulate gas* function had a malfunction and produced lower pressure than what was expected; in other observations, there were no fault behaviors related to the regulating gas. The modes in the functional model for regulating gas are:

- *Nominal Pressured Regulating*
- *Low Pressure Regulating*
- *High Pressure Regulating*
- *No Pressure Regulating*
- *No Gas to Regulate*

For the aim of this study, we extract the scenarios with "No Gas to Regulate" because this mode is caused by failure of the previous functions. Figures 4. and 5. show the histograms for the classified scenarios caused by the subsystem regulate gas failure, for the simulated functional model and the real system. The null hypothesis is the densities of the two histograms are bin by bin equal, against the alternative that they are different.

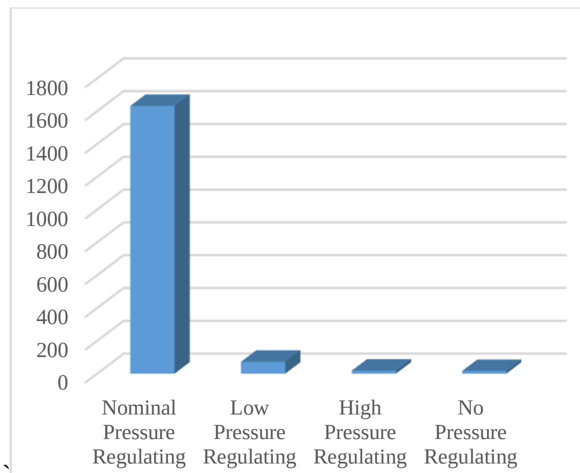

Figure 5. Simulation scenarios for selected monopropellant design

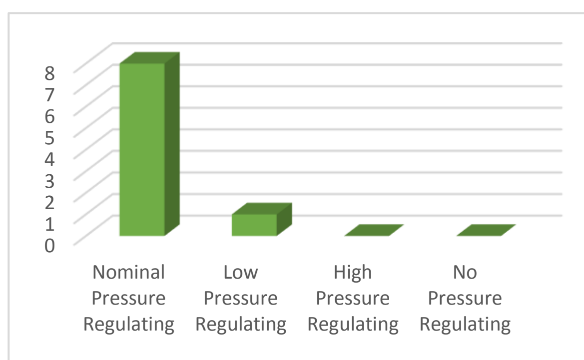

Figure 6. Observed scenarios for gas regulator in spacecraft engine

Table 4. Presents that fault behavior produced by the functional model is categorical. In other words, the simulation

only produces discrete outcomes belonging to one category of final behavior and the observations are also categorical. Therefore, a *chi-squared* test is applied to compare the observed and simulated behavior.

Table 4. Observed and simulated proportions for regulating gas

End State	Simulation	Observations
Nominal Regulating	1633	8
Low Pressure Regulating	73	1
High Pressure Regulating	18	0
No Pressure	16	0

The chi-squared critical value at 95% confidence interval for 3 degrees of freedom is 1.2115. The *P-value* is 0.7503, which shows the observed proportions are not significantly different from the expected proportions.

Local Sensitivity Analysis of Selected Functional Model

Sensitivity analysis is an unavoidable part of the validation process (Conover and Iman, 1981). Designers apply the sensitivity analysis to study the effect of each model's input on the model's output. In this section, a local sensitivity analysis methodology is proposed to find sensitive functions in the selected model for further study in the design process.

This approach is based on the mathematical derivative of the cost-risk function with respect to cost of each faulty behavior. The cost-risk function is formulated based on the operation, design, mitigation, failure cost, and probability of failure. Changes in the actual functions in selected functional models only affect the probability of the system fault behavior and consequential cost of failure. Partial derivatives of the cost function relative to C_I , C_O and C_M are constant values, and they do not provide insight about how the function influences the cost function. Equation (7) shows the derivative of the cost function to the cost of failure, which reflects the probability of having faulty behavior i.

$$f = \sum_{i=1}^{N} C_{R,i} P_{R,i} \tag{7}$$

$$\frac{\partial f}{\partial C_{R,1}} = P_{R,1} \;, \quad \frac{\partial f}{\partial C_{R,2}} = P_{R,2} \;, \dots, \; \frac{\partial f}{\partial C_{R,N}} = P_{R,N}$$

To quantify each partial derivative in Eq. (7), the classification of failure scenarios based on the final behavior of the system is used. The total number of failure scenarios that ended with a specific behavior divided by the total number of scenarios generated by the model quantifies each of the partial derivatives. For instance, the classification illustrated in Fig. 2, reflects the probability of occurrence of each final behavior, and can be used to quantify the partial derivatives of the cost-risk model. To identify the probability of failure of sensitive functions, we develop a search tool in Python to specify the top sensitive functions. The tool searches for the top repeated functions among all the unique failure scenarios. The pseudo code for the search tool to find the sensitive functions is presented as follow:

```
Read the file of unique scenarios

Put the file into a list

Capitalize all the words in the list

Create an empty dictionary

for each line in the list

    remove all the white space

    for each word in the line

        if word exists in the dictionary

            add 1 to the number of counts for this word;

        else

            add the word to the dictionary

    end (for)

end (for)
```

For the selected functional model, the developed search tool determines that "*Expanded Gas*", and "*Excess Gas*" are the most frequent words in the failure simulation results

"*Heating Gas*" causes expanded gas and excess gas issues which is the main root for undesired final behavior of the system.

In the selected design, the extra heat produced from thrust is saved to apply in the process of heating gas. Next, in the preliminary and detailed design phases, designers should investigate the strategies to enable the design to deal with the heat issue. Figure 7. shows the parts of selected functional model (red arrows) that mainly cause the undesired final behavior.

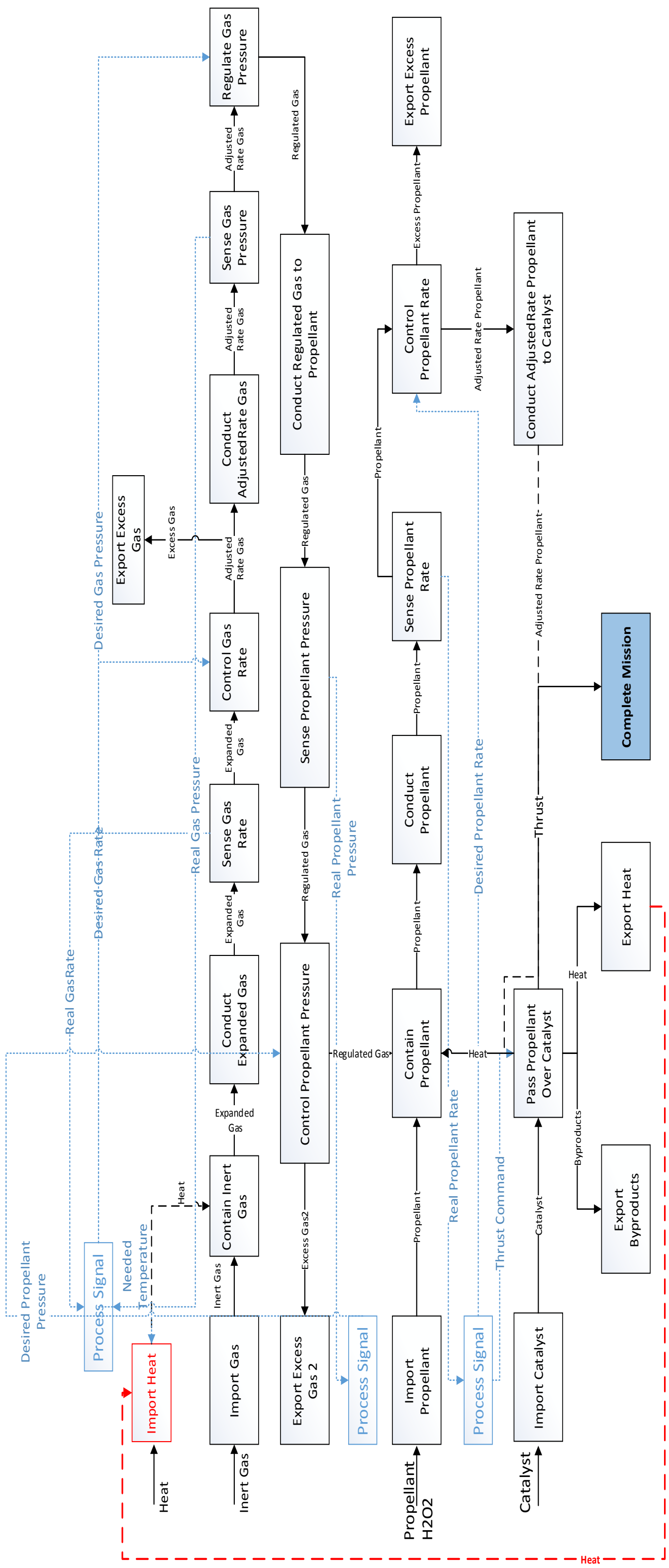

Figure 7. Sensitive parts in the monopropellant functional model

The results of the local sensitivity analysis should be consulted throughout the design process. Designers should pay more attention to sensitive functions, especially in the phase when mapping the functional model to a component model. The model validation in this way is viewed as an evolutionary process: as we go through the design process, more information is available which can be applied for further validation and implement updates if needed. Table 5. provides a summary of proposed methods in this section to validate the results of the selected model in early design stage.

Table 5. Methods to validate model selected in the early design stage

ID	Functional Model Validation Method	Tools
1	Apply expert ideas to validate functional model	Failure Probabilities, FMEA
2	Apply observed data for a subsystem to validate functional model	Failure Probabilities, Chi-Squared Test
3	Apply local sensitivity Analysis to find the most sensitive functions	Cost-Risk Model Derivatives, Sensitive Functions Search Tool

CONCLUSION

The main contribution of this paper is to provide practical methodologies to validate the simulation generated by the model selected in the early design phase. We reviewed the concept of model validation and its value in complex engineered systems. We classified the existing validation methods based on the information available from the real system's behavior in the early design phase and for each class, we proposed the tools and techniques. We discussed that some existing methods are not applicable directly to validate model developed in early design stage since there is no or inadequate information from the real system behavior. We showed that there is a lack of research on methods to validate a model when there is no or inadequate observed data. We proposed strategies to validate the fault behavior generated by a functional model in the early design stage. In the first proposed method, we showed how to quantify the expert's knowledge using a FMEA technique to decide about the validation of the failure probabilities obtained from the simulated failure scenarios. The second proposed approach applies available observed data related to a subsystem of the model, then the non-parametric chi-squared test is utilized to provide evidence of meaningful differences between the model and observation. The third strategy is based on the local sensitivity analysis. In this method we applied our developed tool to find the most sensitive functions and find the variation in the cost model caused by changes in the sensitive functions. Functions with a high effect on the cost model should receive close attention in further steps of design process, especially when mapping the functional model to a component model. We successfully applied the proposed strategies to validate the result of model simulation in early design stage for a monopropellant propulsion system.

ACKNOWLEDGMENTS

This research was funded in part by the National Science Foundation (project number CMMI 1363509) and NASA Grant and Cooperative Agreement NNX15AQ90G. The opinions, findings, conclusions, and recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsor.

REFERENCES

Arlot, S. and Celisse, A., 2010. A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*, pp.40-79.

Bae, H.R., Grandhi, R.V. and Canfield, R.A., 2004. Epistemic uncertainty quantification techniques including evidence theory for large-scale structures. *Computers & Structures*, *82*(13-14), pp.1101-1112.

Banerjee, O., Ghaoui, L.E. and d'Aspremont, A., 2008. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research*, *9*(Mar), pp.485-516.

Berger, J.O. and Pericchi, L.R., 1996. The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, *91*(433), pp.109-122.

Conover, W.J. and Iman, R.L., 1981. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician*, *35*(3), pp.124-129.

Conover, W.J., 1972. A Kolmogorov goodness-of-fit test for discontinuous distributions. *Journal of the American Statistical Association*, *67*(339), pp.591-596.

Cook, D.A. and Skinner, J.M., 2005. How to perform credible verification, validation, and accreditation for modeling and simulation. *The Journal of Defense Software Engineering*, *18*(5), pp.20-24.

Goddard, P.L., 1993, January. Validating the safety of embedded real-time control systems using FMEA. In *Reliability and Maintainability Symposium, 1993. Proceedings., Annual* (pp. 227-230). IEEE.

Hurvich, C.M. and Tsai, C.L., 1989. Regression and time series model selection in small samples. *Biometrika*, *76*(2), pp.297-307.

Keshavarzi, E., McIntire, M., Goebel, K., Tumer, I.Y. and Hoyle, C., 2017, August. Resilient System Design Using Cost-Risk Analysis with Functional Models. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp.V02AT03A043-V02AT03A043).

Kohavi, R., 1995, August. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, No. 2, pp. 1137-1145).

Law, A.M., 2008, December. How to build valid and credible simulation models. In *Simulation Conference, 2008. WSC 2008. Winter* (pp. 39-47). IEEE.

Lewis, R.O., 1992. *Independent verification and validation: A life cycle engineering process for quality software* (Vol. 11). John Wiley & Sons.

Mantel, N., 1963. Chi-square tests with one degree of freedom; extensions of the Mantel-Haenszel procedure. *Journal of the American Statistical Association*, *58*(303), pp.690-700.

Mara, T.A. and Tarantola, S., 2012. Variance-based sensitivity indices for models with dependent inputs. *Reliability Engineering & System Safety*, *107*, pp.115-121.

Marzban, C., 2013. Variance-based sensitivity analysis: An illustration on the Lorenz'63 model. *Monthly Weather Review*, *141*(11), pp.4069-4079.

Massey Jr, F.J., 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, *46*(253), pp.68-78.

Minhas, R., De Kleer, J., Matei, I., Saha, B., Janssen, B., Bobrow, D.G. and Kurtoglu, T., 2014, March. Using fault augmented modelica models for diagnostics. In *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden* (No. 96, pp. 437-445). Linköping University Electronic Press.

Raftery, A.E., 1995. Bayesian model selection in social research. *Sociological methodology*, pp.111-163.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M. and Tarantola, S., 2010. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, *181*(2), pp.259-270.

Saltelli, A., Chan, K. and Scott, E.M. eds., 2000. *Sensitivity analysis* (Vol. 1). New York: Wiley.

Sanders, P., 1996. *DoD modeling and simulation (M&S) verification, validation, and accreditation (VV&A)* (No. DODI-5000.61). Office of the Under Secretary of Defense. (Acquisition and Technology) WASHINGT ON DC.

Sargent, R.G., 1987. An overview of verification and validation of simulation models. In *Proceedings of the 19th conference on Winter simulation* (pp. 33-39). ACM.

Schoenfeld, D., 1980. Chi-squared goodness-of-fit tests for the proportional hazards regression model. *Biometrika*, *67*(1), pp.145-153.

Sentz, K. and Ferson, S., 2002. *Combination of evidence in Dempster-Shafer theory* (Vol. 4015). Albuquerque: Sandia National Laboratories.

Shafer, G., 1992. Dempster-shafer theory. *Encyclopedia of artificial intelligence*, pp.330-331.

Smyth, P., 2000. Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and computing*, *10*(1), pp.63-72.

Stamatis, D.H., 2003. *Failure mode and effect analysis: FMEA from theory to execution*. ASQ Quality Press.

Stone, M., 1977. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp.44-47.

Storlie, C.B., Swiler, L.P., Helton, J.C. and Sallaberry, C.J., 2009. Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. *Reliability Engineering & System Safety*, *94*(11), pp.1735-1763.

Tang, L., Kacprzynski, G., Goebel, K., Reimann, J., Orchard, M.E., Saxena, A. and Saha, B., 2007b, November. Prognostics in the control loop. In *Working Notes of 2007 AAAI Fall Symposium: AI for Prognostics*.

Tang, L., Saxena, A., Orchard, M.E., Kacprzynski, G.J., Vachtsevanos, G. and Patterson-Hine, A., 2007a, March. Simulation-based design and validation of automated contingency management for propulsion systems. In *Aerospace Conference, 2007 IEEE* (pp. 1-11). IEEE.

Teng, S.H. and Ho, S.Y., 1996. Failure mode and effects analysis: an integrated approach for product design and process control. *International journal of quality & reliability management*, *13*(5), pp.8-26.

Triantaphyllou, E. and Sánchez, A., 1997. A sensitivity analysis approach for some deterministic multi-criteria decision-making methods. *Decision Sciences*, *28*(1), pp.151-194.

Van Leeuwen, J.F., Nauta, M.J., De Kaste, D., Odekerken-Rombouts, Y.M.C.F., Oldenhof, M.T., Vredenbregt, M.J. and Barends, D.M., 2009. Risk analysis by FMEA as an element of analytical validation. *Journal of pharmaceutical and biomedical analysis*, *50*(5), pp.1085-1087.

Vesely, W.E., Goldberg, F.F., Roberts, N.H. and Haasl, D.F., 1981. *Fault tree handbook* (No. NUREG-0492). Nuclear Regulatory Commission Washington DC.

Vuong, Q.H., 1989. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society*, pp.307-333.

Wagner, H.M., 1995. Global sensitivity analysis. *Operations Research*, *43*(6), pp.948-969.

Wasserman, L., 2000. Bayesian model selection and model averaging. *Journal of mathematical psychology*, *44*(1), pp.92-107.

Xiong, Y., Chen, W., Tsui, K.L. and Apley, D.W., 2009. A better understanding of model updating strategies in validating engineering models. *Computer methods in applied mechanics and engineering*, *198*(15-16), pp.1327-1337.

Zadeh, L.A., 1984. Review of a mathematical theory of evidence. *AI magazine*, *5*(3), p.81.

