Sensitivity Analysis and Uncertainty Quantification of a Mars Ascent Vehicle Model Concept

Connor D. Noyes
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr
Pasadena, CA 91109
714-915-7445
cdnoyes@jpl.nasa.gov

Joel Benito
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr
Pasadena, CA 91109
818-354-4809
Joel.Benito.Manrique@jpl.nasa.gov

Lee D. Peterson
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr
Pasadena, CA 91109
818-354-3346
ldpeters@jpl.nasa.gov

Abstract—**The design of a conceptual Mars ascent vehicle is a challenging problem. In order to aid the vehicle and mission concept design it is important to understand the driving design parameters and the expected performance in the presence of model errors and uncertainties. An existing six degree of freedom simulation model is analyzed on a statistical basis using the methods available in the Design Analysis Kit for Optimization and Terascale Applications toolkit. The methods utilized include conventional Monte Carlo techniques, metamodeling via polynomial chaos expansions, and global variance-based sensitivity analyses. Two additional analysis methods referred to as "Monte Carlo filtering" and "Classification trees" are used to determine which uncertain parameters are driving the performance of the vehicle. Monte Carlo filtering provides a methodology to determine which parameters cause qualitatively different behavior while the classification trees use heuristics to partition the input space and assign probabilities to each partition. These methods serve as qualitative descriptors of model sensitivity while variance-based global sensitivity analysis seeks a quantitative mapping from total output variance to the variance of individual inputs. Application of these techniques to several outputs of a Mars ascent vehicle concept simulation indicates that only a select few input factors dominate their variance.**

TABLE OF CONTENTS

1. INTRODUCTION

A Mars Ascent Vehicle (MAV) is a conceptual rocket that would put into Mars orbit a container with geological and atmospheric samples from Mars. The container, called an Orbiting Sample (OS), would remain in orbit until an Earth return vehicle captures it and returns it to Earth for analysis. The design of a MAV is a challenging problem: it would have to endure a trip to Mars, including violent Earth launch and Mars landing, survive up to a year on the Mars surface at very low temperatures, and be capable of accurately injecting an OS into a designated orbit. Furthermore, it would have to present low mass and compact packaging to be able to be delivered to Mars on an MSL-class lander. The MAV concept could also be considered to be a step towards the technology developments needed for human colonization of Mars.

There are multiple MAV concept architectures and design options available: from different number of stages to propulsion technology alternatives and aerodynamic features. In addition to these vehicle design options there are mission design parameters that play an important role: launch event characteristics like the launch orientation, and target orbit being the most relevant. The objective of this work is to explore the option space and evaluate the impact of the different options on the MAV performance and design. Ultimately, statistical analyses will be a key component in refining these designs and down-selecting from the available architectures.

Uncertainty quantification (UQ) and sensitivity analysis (SA) are two closely related ideas. Here UQ is taken to be the characterization of overall uncertainty in system responses due to variability in the system's parameters.[3] SA aims to determine how this output variability can be apportioned to each of the uncertain inputs, i.e., which inputs are most important in determining the uncertainty of the responses of interest.[4] UQ presents a methodology for determining the robustness of the system in the presence of uncertain inputs; SA clarifies the input-output relationships in the model, and allows for uncertainty reduction and optimization of the system's performance by focusing on the uncertain parameters with the greatest effects on the system's output. Random sample Monte Carlo methods are the de facto standard for UQ of uncertain nonlinear dynamical systems but these methods should always be complemented or even supplanted by other techniques. Full factorial design provides a complete sensitivity analysis by sampling the model for all possible combinations of inputs. This is not practical when the model is expensive to evaluate or the uncertainty space is high-dimensional.

The paper begins with a description of the simulation model of the MAV underlying all of the analyses presented herein. The following section discusses the statistical bases for analyzing the model. The next section presents the results and the final section concludes the paper and gives an outlook on future work to be done.

2. Six Degrees of Freedom MAV Models

The trajectory simulation engine is DSENDS (Dynamics Simulator for Entry, Descent and Surface landing).[12] DSENDS is an EDL-specific extension of a JPL multi-mission simulation toolkit Darts/Dshell (see Ref. [13],[15] for details) which is capable of modeling spacecraft dynamics, devices, and subsystems, and has been used by interplanetary and science-craft missions such as Cassini, Galileo, and MSL.[12],[14] DSENDS presents a dynamics core written in C++ with a user layer written in Python. All high-level modeling is done in Python, and the low-level modeling (IMU, propulsion, aerodynamics) is done in C,C++, or Fortran. The GNC flight software is also written in C++ and interacts with the simulation through the Python layer. Each MAV architecture is modeled as a collection of bodies, actuators, and sensors. The modeled bodies include the OS, the main engine assembly, propellant tanks, and the rocket body. Actuators include the OS ejection system which is comprised of three springs, the reaction control system (RCS), the main engine. An IMU is the primary sensor; the simulation implements bias, random walk, and scale factors for the individual accelerometers and gyroscopes based on Sensonor STIM300 specifications.[8] See Figure 1 for a complete overview of the simulation model. The aerodynamics of the MAV are modeled and incorporated via Fortran software written by Langley Research Center. MarsGRAM 2010 is used as a model of the Martian atmosphere.[9]

The simulation is governed by a finite state machine. The sequence of states through which the simulation progresses is nonlinear and dynamic; the sequence through which an off-nominal simulation progresses may differ substantially from that of the baseline scenario. The simulation is run with a 2 ms integration stepsize while the GNC is run at 50 Hz. Figure 1 shows an overview block diagram of the simulation capabilities and models included. Only a single stage to orbit vehicle architecture is modeled for analysis in this work. The reader is referred to Ref. [18] for details on the simulation.

3. Statistical Analysis Methods and Inputs

The majority of the analysis is performed using DAKOTA (Design Analysis Kit for Optimization and Terascale Applications), an analysis software developed by the Sandia National Labs.[1],[2] It provides algorithms for optimization, sensitivity analysis, and uncertainty quantification. The DSENDS simulation is connected to DAKOTA via the Integrated Modeling and Uncertainty Quantification package (imuQ), a software package developed at JPL. imuQ is used to describe the analysis method and to manipulate input and output data between the simulation and DAKOTA.

The model was built with statistical analyses in mind and to that end, a large number of its parameters are governed by probabilistic distributions to be sampled from by DAKOTA. It is currently assumed that all inputs are governed by either a uniform distribution or a Gaussian distribution though future work will utilize additional distributions as needed to model the uncertain parameters. The total number of aleatory variables is 160 which, coupled with the computation time of a single simulation, prohibits the use of a full factorial design in analyzing the system. The input factors include variability in parameters of the main engine and each of six attitude control thrusters including thrust magnitude, specific impulse, mounting and alignment errors. Further input factors include launch orientation, aerodynamic coefficients, atmospheric conditions, knowledge of initial state, IMU parameters, mass properties of individual elements, and OS ejection system parameters.

Monte Carlo

Monte Carlo (MC) methods are ubiquitous in statistical analysis and uncertainty quantification of numerical simulations largely due to its ease of implementation. The main idea behind MC methods is to run the simulation for a number of samples, each time drawing each uncertain input from its governing distribution. Statistics such as mean and variance can then be estimated from the model output. Although MC methods do not suffer from the curse of dimensionality, the number of samples required for accurate statistics may nevertheless be quite large due to their slow convergence. A number of other techniques exist that attempt to improve upon the efficiency with which uncertainty is propagated through the model, including variance-reduction methods like Latin hypercube sampling (LHS), stochastic expansions like polynomial chaos expansions, and adaptive/importance sampling to name only a few. A full review of Monte Carlo methods and other forward uncertainty quantification techniques is beyond the scope of this paper; for further details see, for example, Ref. [5]. All MC-based results presented in this work were sampled using Latin hypercube sampling (LHS) where the probability distribution of each input variable is partitioned into equally probable intervals and each interval is sampled only once. In essence, LHS ensures that the realizations of probabilistic inputs are representative of the real variability. LHS is a superior alternative to pure random sampling because it comes essentially for free in terms of both implementation and computational effort, and has the potential benefit that fewer samples may be required to converge to stable mean and variance estimates.

Monte Carlo Filtering—The finite state machine-based simulation allows for very efficient manipulation of the data obtained via MC methods. For example, samples that reach the target orbit and succeed in ejecting the OS can be termed "behavioral" cases while those that do not (for any variety of reasons) are deemed "non-behavioral" cases. Note that this is only one way in which the data can be separated and that the behavioral and non-behavioral sets can be filtered further in the same way.

Separation of the MC results in this manner allows for a form of statistical analysis to be performed via two-sample Kolmogorov-Smirnov (KS) tests between the behavioral and non-behavioral sets.[4] Conceptually, the two-sample KS test

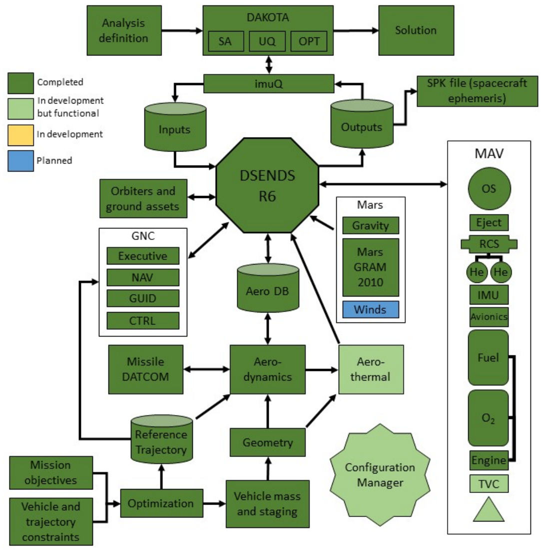

Figure 1. **A high-level overview of the MAV simulation architecture.**

is used to determine if the underlying probability distributions of two data sets differ. The KS statistic is computed

$$D_{KS} = \sup_x |F_B(x) - F_{NB}(x)|, \tag{1}$$

where F_B and F_{NB} refer to the empirical cumulative distribution functions of the behavior and nonbehavioral sets, respectively, and $\sup_x$ is the supremum function over the domain x. In order to reject the null hypothesis that the samples come from the same distribution at a confidence level α, the KS statistic must satisfy the relation

$$D_{KS} > f(\alpha)\sqrt{\frac{n_B + n_{NB}}{n_B n_{NB}}}, \tag{2}$$

where $f(\alpha)$ is a function that maps confidence levels to critical values of the two-sample test, and n_B, n_{NB} are the number of samples in the behavioral and non-behavioral sets, respectively. By performing the KS test on these sets for each input factor, it is possible to determine which statistically significant inputs are driving the difference between them. In this way, MCF provides a qualitative form of SA. One key reason why MCF is particularly useful is that, unlike many SA methods, it does not require further simulations, instead utilizing existing sampled outputs. Additionally, the method considers the entire range of values that each uncertain input can take unlike partial derivative methods that estimate sensitivities point-wise. One limitation is that the method relies on the marginal distribution of each input variable and is thus incapable of determining sensitivity to interaction effects. Figure 2 shows how Eqn. 1 is applied to a single input factor to determine its test statistic.

Classification Trees

Decision tree learning is a data mining technique utilizing machine learning. In particular, classification trees are used herein to partition the input space and assign each partition a probability estimate regarding the output variables. Tree learning is a research field in itself so the interested reader is directed to Refs. [21],[22] for information on the CART decision tree algorithm used in this work. The implementation used is the Python package *scikit-learn's* tree classifier with

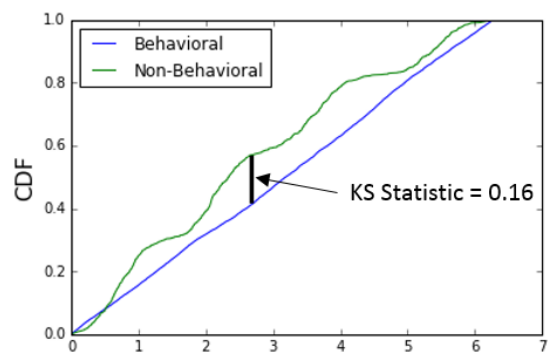

Figure 2. **Visual representation of the computation of KS test statistic.**

the information theory concept of entropy as its criterion. [20] Not unlike MCF, classification trees do not require additional simulations.

Metamodeling

DAKOTA exposes a number of metamodeling techniques that allow us to create an input-output model that approximates the results of the original simulation. These models are also sometimes referred to as "surrogate models" or "response surfaces." In general, a metamodel can be expensive to construct, especially when the original simulation model is computationally intensive to run and the input space is high dimensional. It is also difficult to verify the accuracy of such a model. If, however, a metamodel that accurately captures the complexity of the underlying model can be obtained then it can be utilized in a number of different manners. For instance, it can sampled via MC extremely rapidly relative to sampling the original model. It may also be subjected to an optimization routine or used in an optimization under uncertainty setting due to its computational efficiency. In this work, we constructed a metamodel using the polynomial chaos expansion (PCE) method. PCEs are a class of stochastic collocation methods and it is well documented that unlike

MC, PCE methods scale poorly as the number of stochastic or probabilistic inputs grows and may be unsuitable for non-smooth outputs.[16],[17] However, metamodels offer a point of comparison with MC methods and can also be utilized in the global sensitivity approach described in the following section.

Global Sensitivity Analysis by Variance-Based Decomposition

Global sensitivity analysis can be performed in DAKOTA via variance-based decomposition. The method involves the computation of the Sobol' indices. Variance-based measures of sensitivity are appealing due to their global nature (i.e. they measure sensitivity across the whole input space), and the fact that they can handle nonlinear responses.[4] Computing the Sobol' indices requires the computation of multidimensional integrations that are often performed numerically by using a sampling method such as LHS. This approach suffers from the "curse of dimensionality" in that the total number of model evaluations required scales linearly with the number of inputs $n_{evaluations} = N(n_{inputs} + 2)$ where N is chosen by the analyst to achieve a desired accuracy and is generally hundreds or thousands.[23] Alternatively, the indices are available in closed-form for several forms of response surfaces including PCEs.[6],[16],[17] The main effect sensitivity index S_i and total effect index T_i are computed as

$$S_i = \frac{Var_{x_i}(Y|x_i)}{Var(Y)} \tag{3}$$

$$T_i = \frac{Var(Y) - Var(Y|x_{-i})}{Var(Y)} \tag{4}$$

where $Y = f(x)$ is a model output, x_i is the i^{th} uncertain input variable, x_{-i} is the set of all uncertain variables except the i^{th}. The total effect index is a measure of the proportion of output Y that can be attributed to input x_i, either directly or through interaction with other inputs. The main effect sensitivity index is the fraction of the variance in Y that is due solely to input x_i.[2] For a linear, additive model, the total effect index will be equal to the main effect index for each input, and the sum of the main effect indices over all inputs will be unity. When an input's total index is larger than its main effect, it implies interaction with one or more additional inputs in the model, and the sum of either index over all inputs will not in general be unity.

Bootstrapping

A common way to assess the distributional properties of statistics generated via sampling (Monte Carlo) is a re-sampling technique called bootstrapping.[5][19] Resampling techniques are those that reuse available samples to extract further information. Bootstrapping refers to random sampling with replacement of existing samples to construct measures such as variance, mean square error, or confidence intervals for a given statistic. For example, the original sampling data results in a sample mean $\mathbb{E}$[X] and variance $\mathbb{V}$[X] for an output X. Bootstrapping then further provides a way to estimate measures of accuracy for $\mathbb{V}$[X], such as $\mathbb{V}[\mathbb{V}$[X]].

4. Results

A Monte Carlo simulation with 10,000 samples was conducted using the full MAV model for uncertainty quantification and for regionalized sensitivity analyses via the methods of sections 3-3. All CDFs (cf. Fig. 5), are drawn with 3-σ bounds estimated via bootstrapping. The percentiles given are all bootstrapped estimates as well. Additionally, a variance-based decomposition was performed on the model with the number of samples per input was taken to be 315 for a total of 12,600 model evaluations used to estimate the Sobol' indices. Although more evaluations would benefit the quality of the decomposition, the MAV simulation is sufficiently computationally expensive to preclude a greater number of total samples.

Output	1%	50%	99%
Periapse altitude (km)	382	455	473
Apoapse altitude (km)	472	486	545
Navigated periapse (km)	463	473	475
Navigated apoapse (km)	474	477	481
Inclination (deg)	92.2	93.0	93.9

Table 1. **UQ results for selected outputs concerning the final orbit.**

Model Output Selection

The MAV simulation model has many outputs so a subset of them must be chosen for analysis. The purpose of a MAV is to deliver the OS into orbit to be recaptured at some point in the future. In order to ensure that recapture is feasible, there will be requirements on orbital insertion accuracy. By first applying UQ techniques to the problem, variations in final orbit can be quantified. Then, SA techniques can be used to convert requirements on insertion accuracy into requirements on input uncertainty, essentially determining how well the most important factors must be known in order to meet the target requirements. The outputs considered are inclination, periapse and apoapse altitudes, and equivalently semi-major axis (SMA) and orbital eccentricity.

Liquid injection thrust vector control (LITVC) is modeled in the MAV simulation. The liquid injected is the same oxidizer used by the main engine. The total oxidizer carried onboard the vehicle is a fixed amount, and due to uncertainty it is not known a priori how much oxidizer LITVC will require, or how much oxidizer the total impulse to orbit will necessitate. High oxidizer consumption via LITVC is therefore a risk on the system. Thus, SA and UQ techniques are applied to study LITVC performance.

Note that in the following sections a number of input factors will be labeled with "X" and "Y" components referring to the two axes orthogonal to the MAV's roll axis and may also be referred to as the vehicle yaw and pitch axes respectively. For a vehicle that is axisymmetric about the roll axis, it is anticipated that the effects of "X" and "Y" inputs will be roughly equal.

Final Orbit

Select percentiles concerning the final orbital insertion accuracy are given in Table 1. The significant difference between the resulting orbit and the onboard navigation system's estimate of its orbit indicates that initial knowledge error about the vehicle's pitch and yaw axes is the most important contributor to the variation in periapse and apoapse. This is confirmed via Monte Carlo filtering the 10% highest and lowest apoapse cases which indicates that initial attitude errors

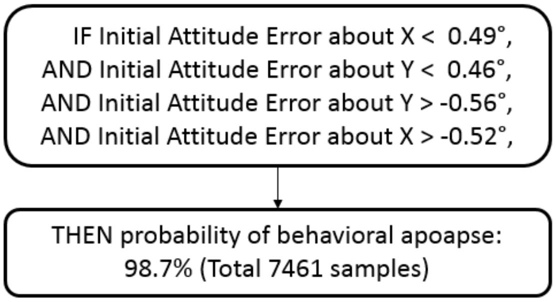

Figure 3. One leaf of a classification tree based on largest apoapse variations.

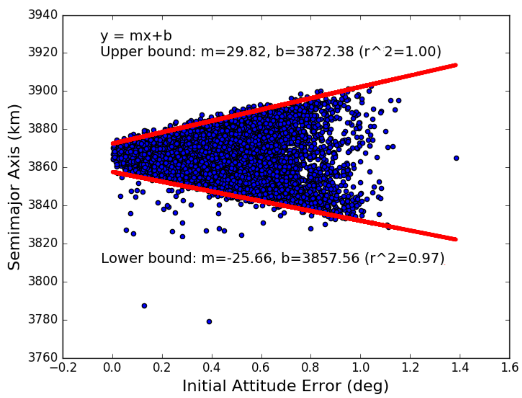

Figure 4. Semi-major axis viewed as a function of initial attitude error with first order trend lines shown.

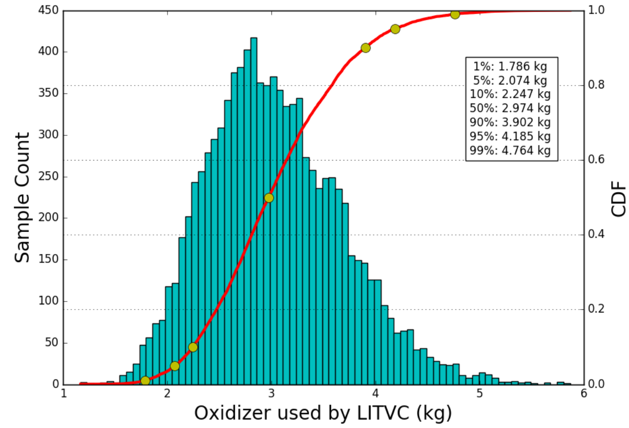

Figure 5. Combined CDF and histogram of the oxidizer mass used by the liquid injection TVC system.

Input	KS Statistic
Cm multiplier (aerodynamic pitch coefficient)	0.20
Launch azimuth	0.16
Engine misalignment X	0.15
Engine misalignment Y	0.14

Table 2. Monte carlo filtering results for high LITVC usage.

are an order of magnitude more influential than the next input factors. A classification tree was generated and the largest leaf, shown in Figure 3, indicates that if attitude error about each axis is smaller than approximately $0.5°$ in magnitude, then the apoapse will remain within the behavioral set. No other input factors are indicated in any of the other leaves of the tree. Together with small contributions from the IMU data, initial attitude errors result in significant velocity errors which are the dominant factor in the altitude variations. Inclination is not strongly affected by these factors; its total deviation across all simulations is less than $2°$, and filtering based on the largest deviations from the median reveals that attitude initialization error about the roll axis is instead the dominant factor.

Automated Linear Fitting—Using UQ samples it is possible to estimate trends in statistics as a function of a single input variable. Firstly the entire input range is divided into several equi-probable partitions (typically 5-10 is sufficient). Then, the desired statistic (a percentile, mean, etc) in each input range is computed. A linear least squares regression is then used to find the best relation between the output and input. This information allows the designer to estimate what input range is required to satisfy requirements on the output. For a given output, the input to use is typically chosen based on the results of SA, MCF, or measures of dependence.

The 0.5% and 99.5% SMA are fitted as a function of the total initial attitude error in Figure 4. From the two trend lines it can be seen that with no initial error in knowledge of the vehicle's orientation, the simulated MAV reaches its target SMA to within ± 6 km. This bound increases at a rate of approximately 26 km per degree of initial attitude error. If a requirement is imposed that a MAV must deliver the orbiting sample to within ± 25 km of the targeted SMA with 99% probability, then by subtracting the lower bound from the upper bound an estimate can be obtained. The magnitude of the attitude error must be smaller than $0.25°$ in order to achieve the required ± 25 km accuracy under the conditions in Monte Carlo. Although fitting linear relationships will not always be applicable, this example shows that it may find use even in highly nonlinear, coupled system models like a MAV concept. The method is also not restricted to first-order fitting.

Thrust Vector Control

Figure 5 shows the thrust vector control usage statistics from the LHS results. Monte carlo filtering is applied with the samples with the 10% highest usage considered the non-behavioral set, and the results, shown in Table 2, indicate that the pitch aerodynamic coefficient, launch direction, and main engine misalignments are driving the highest usage. Aerodynamic forces disturbing the MAV during atmospheric flight must be controlled via TVC. The MAV simulation considers launches in all $360°$ of azimuth. Launching opposite the desired direction will require a change in orientation that is actuated via TVC thereby increasing the amount of oxidizer expended. Similarly, engine misalignment manifests as additional perturbations that must be countered by the control system.

Input	Total Effect Index	Main Effect Index
Launch Azimuth	0.783734	0.104
Engine misalign Y	0.289098	-0.005
Engine misalign X	0.277016	0.032
Cm multiplier	0.099405	0.116
Cn additive	0.089091	0.020
Cm additive	0.080274	0.048

Table 3. **Variance-based decomposition results for LITVC usage.**

Table 3 shows the thrust vector control results from the variance-based decomposition. The Sobol' indices for the six highest contributors, ranked by their total effect index, are show and there are several items to note. The total index is larger than the main index by a substantial amount in almost all inputs, indicating that interactions between inputs account for a significant amount of the total variance. The sum over all main effects (note that only the top six are shown in the Table 3) is 0.33 from which we conclude that approximately two-thirds of the LITVC variance results from interactions between two or more inputs. The variance-based results indicate two additional aerodynamic coefficients are small contributors but otherwise the decomposition agrees well with the Monte Carlo filtering results. Recalling from Section 3 that filtering does not capture interaction effects, it is appropriate to compare filtering with main effect indices and in fact, the two produce very consistent results. Not only do their most important inputs agree, but so too do the numerical ranking of their effects as well. Note that negative values for the main index are the result of numerical errors when the index is near zero, and that increasing the number of samples will improve the quality of the decomposition. Looking again at Table 2, it is natural to wonder if the combined effects of X and Y place engine misalignment higher in the ranking. The VBD results in Table 3 allow us to answer that question, however, showing that azimuth actually has an even stronger interaction so it remains at the top, but the misalignments do overtake the aerodynamic coefficients once interactions are considered.

5. Concluding Remarks

Classical Monte Carlo methods are essential to uncertainty quantification but additional techniques can augment these methods to glean more information from sampled data. Monte Carlo filtering provides an effective way to uncover the factors driving certain simulation phenomena, and in some situations can approximate a type of qualitative sensitivity analysis. A key limitation of filtering is its inability to capture coupled interactions among inputs and outputs. Classification trees provide similar analysis but allow for some interaction effects to be discovered. Three different approaches to sensitivity analysis were utilized, each based on different criteria: density (MCF), entropy (trees), and variance (VBD), but the conclusions drawn from each are largely the same.

Ultimately, the application will decide the best combination of techniques with which to analyze. The dimension of the uncertainty space and the computational cost of the model being studied are the most important factors in determining which methods to apply. There is a tradeoff between the numerical cost of each analysis method and the amount of information it yields. Monte carlo filtering is an efficient method for approximate sensitivity analysis from given data, but it does not provide numerical mapping between input and output variance. Other sensitivity analysis techniques are capable of providing this information but at significantly increased computational costs. Metamodeling can reduce the computational burden by achieving faster convergence but not all models are amenable to this approach. The high dimensionality of the MAV simulation's input space makes it a poor candidate for such methods.

Finally we note that while the MAV simulation is intended to reflect reality as closely as possible, no model is perfect and inferences drawn based on the model must be considered carefully. The techniques applied in this paper say nothing about the validity of the model itself.

Acknowledgments

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Ebeida, M.S., Eddy, J.P., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., Vigil, D.M. and Wildey T.M., "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.1 User's Manual," Sandia Technical Report SAND2014-4633, Sandia National Laboratories, Albuquerque, NM, Updated November 2014.

[2] Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Ebeida, M.S., Eddy, J.P., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., Vigil, D.M. and Wildey T.M., "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.1 Theory Manual," Sandia Technical Report SAND2014-4253 Sandia National Laboratories, Albuquerque, NM, Updated November 2014.

[3] Ghanem, R., Higdon, D., and Owhadi, H., "Handbook of Uncertainty Quantification," Springer International, 2016.

[4] Saltelli, A., Ratto, M., Andres, T., Campolongov, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., "Global Sensitivity Analysis: The Primer," Wiley-Interscience, February 2008.

[5] Kroese, D.P., Taimre, T., Zdravko, I.B., "Handbook of Monte Carlo Methods," Wiley, March 2011.

[6] Tang,G., Iaccarino,G., and Eldred,M.S., "Global sensitivity analysis for stochastic collocation expansion," In Proceedings of the 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (12th AIAA Non-Deterministic Approaches conference), Orlando, FL, April 12-15, 2010. AIAA Paper 2010-XXXX.

[7] "LN-200 FOG Family - Advanced Airborne IMU/AHRS," Northrup Grumman Systems Corporation, 2013.

[8] "STIM300 Inertia Measurement Unit Datasheet," Sensonor AS, 2016.

[9] "Mars Global Reference Atmospheric Model," http://see.mfsc.nasa.gov/tte/model_Marsgram.html

[10] Dankanich, J. and Klein, E., "Mars Ascent Vehicle Development Status, IEEE Aerospace Conference, 2012.

[11] Dux, I., Huwalt, J. and Dankanich, J., "Mars Ascent Vehicle Gross Lift-off Mass Sensitivities for Robotic Mars Sample Return, NASA Technical Report NASA/TM-2011-216968.

[12] J. Balaram, R. Austin, P. Banerjee, T. Bentley, D. Henriquez, B. Martin, E. McMahon, G. Sohl, "DSENDS - A High-Fidelity Dynamics and Spacecraft Simulator for Entry, Descent and Surface Landing", IEEE 2002 Aerospace Conf., Big Sky, Montana, March 9-16, 2002.

[13] J. Biesiadecki, D. Henriquez, and A. Jain, "A reusable, real-time, spacecraft dynamics simulator," in 6th Digital Avionics Systems Conference, Irvine, CA, Oct. 1997.

[14] Burkhart, P.D., and Casoliva, J., "MSL DSENDS Analysis and Operations," 23rd International Symposium on Space Flight Dynamics, 2012.

[15] Lim, C., and Jain, A., "Dshell++: A Component Based, Resuable Space System Simulation Framework," Third IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT 2009), Pasadena, CA, July 19-23, 2009.

[16] Crestaux, T., Le Maitre, O., Martinez J.M., "Polynomial chaos expansion for sensitivity analysis," Reliab Eng Syst Safety (2009), doi:10.1016/j.ress.2008.10.008

[17] Sudret, B., "Global sensitivity analysis using polynomial chaos expansion," Reliab Eng Syst Safety (2007), doi:10.1016/j.ress.2007.04.002

[18] Benito, J., and Noyes, C., "Mars Ascent Vehicle Model Simulation," IEEE Aerospace Conference, 2016.

[19] Efron, B., Tibshirani, R., "An Introduction to the Bootstrap" Boca Raton, FL: Chapman & Hall/CRC, 1993.

[20] Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR 12, pp. 2825-2830, 2011.

[21] Lewis, R. J. "An introduction to classification and regression tree (CART) analysis." Annual meeting of the society for academic emergency medicine in San Francisco, California. 2000.

[22] Lemon, S. C., et al. "Classification and regression tree analysis in public health: methodological review and comparison with logistic regression," Annals of behavioral medicine 26.3 (2003): 172-181.

[23] Weirs, V. Gregory, et al. "Sensitivity analysis techniques applied to a system of hyperbolic conservation laws." Reliability Engineering & System Safety 107 (2012): 157-170.

BIOGRAPHY

Connor Noyes *received his B.S. and M.S. degrees in aerospace engineering from California Polytechnic State University in 2013 and is currently a Ph.D. student at the University of California, Irvine. His research interests include optimal control, robust optimization, and trajectory design.*

Joel Benito *received a B.S. in Telecommunications and Electrical Engineering from Polytechnic University of Catalonia, Spain, an M.S. in Aerospace Engineering from SUPAERO, France and a Ph.D. in Aerospace Engineering from the University of California, Irvine. Since then, he joined JPL in the EDL Guidance and Control Systems Group, where he has worked on Mars precision landing technologies, EDL flight performance analysis, GNC systems engineering and Mars Ascent Vehicle GNC systems and trajectory design.*

Lee Peterson *is a Principal Technologist in the JPL Mechanical Systems Engineering Fabrication and Test Division. Prior to joining JPL in 2008, he was the Gary L. Roubos Professor and Department Chair of the Aerospace Engineering Sciences department at the University of Colorado at Boulder. With over 30 years experience in the mechanics and dynamics of spacecraft structures, he is author or co-author of over 120 publications. He has an SB, SM and PhD in Aeronautics and Astronautics at MIT.*