AAS 680

DYNAMICAL MODELS FOR SPACECRAFT LANDINGS IN LOW-GRAVITY ENVIRONMENTS USING POLYNOMIAL CHAOS.

Iosto Fodde*, Lucia Francesca Civati*, Alessia Cremasco*, and Fabio Ferrari*

Landings on the surface of small solar-system bodies (SSB) are incredibly valuable in terms of scientific return, however there is also a high risk associated to them as seen for the Philae and MINERVA probes. One of the main problems is the complex modeling of the interaction between the lander and the surface of the SSB. This work presents novel ways of combining high fidelity simulations with data-driven methods to obtain dynamical models of the spacecraft-surface interactions. A polynomial chaos method is employed, which takes as input a small set of high-fidelity simulation runs, and generates a polynomial model from it that accurately describes the input-output relation of the original model. This PCE model can then be used to perform sensitivity and reliability analyses, which for the original model is infeasible due to numerical constraints. The generated model is shown to be highly accurate and versatile, allowing mission designers to perform robust and efficient landings of a spacecraft on the surface of SSBs.

INTRODUCTION

Landings on the surface of small solar-system bodies (SSB) are incredibly valuable in terms of scientific return as the spacecraft-surface interaction provides direct information on the internal structure and material properties of the SSB while their instruments can do some in-situ measurements to characterize the SSB in more depth. Many missions have attempted, or are planning to attempt, landings on the surface of a SSB. For example, the upcoming Hera mission will attempt to land its two CubeSats on the surface of asteroids Didymos and Dimorphos,[1] while JAXA's Martian Moons eXploration (MMX) will deploy a rover on the surface of the Martian moon Phobos.[2] Even though several missions succeeded in deploying their landers safely, many others have had more problems and difficulties. For example, Philae experienced unintentional bouncing which resulted in the lander being lost on the surface for a long period of time, and the MINERVA lander, part of the Hayabusa mission, also did not succeed in landing on the surface of Itokawa. One of the main problems is the uncertainty related to the surface properties of SSBs. A large number of SSBs are thought to have a large amount of granular material, also known as regolith, and boulders on their surface.[3] The interactions between the individual regolith particles, the boulders, and the lander itself are governed by complex contact forces and hard to model on Earth due to the low-gravity environment on these SSBs.

*Department of Aerospace Science and Technology (DAER), Politecnico di Milano, Via La Masa 34, Milan, Italy

Most research on designing lander trajectories and determining the effect of bouncing on the surface use so-called hard surface methods.[4] These hard surface methods use a small amount of parameters that abstract away the different effects of the internal interactions happening within the regolith when in contact with the lander. This allows for fast and efficient modeling to perform statistical analyses and determine lander safety, but are generally not highly accurate and are thus also hard to use when doing inverse modeling to determine important surface properties. On the other hand, soft surface methods model the regolith using a large number of discrete bodies and simulate fully their interactions.[5] This increases the fidelity of the model greatly and works well when trying to estimate surface properties from observing the landing, but drastically increases the numerical complexity and thus making statistical analysis using Monte Carlo approaches impractical.

In this work, a novel approach is proposed which aims to generate an efficient dynamical model of the spacecraft-surface interaction like the hard surface models, but which retains the accuracy of the soft surface model. This is achieved using a data-driven approach where a small number of soft surface simulations are used to fit a model that can be more efficiently used. This model is created by a non-intrusive polynomial chaos method, which samples the initial conditions and system parameters in an optimal way, and takes the input-output relationship of these sample simulations to create a fit based on a family of orthogonal polynomials that represent the prior distribution of the input variables. This polynomial model can be used to perform large scale sensitivity and reliability assessments of the landing without sacrificing the accuracy greatly.

SOFT SURFACE MODELLING

A discrete element modelling (DEM) technique is used to simulate the motion of the surface particles, the lander, and their interaction. In this work, specifically the GRAINS software developed in[6] is used, which is built upon the Chrono physics engine.[7]

For the case considered here, all objects in the simulation move under a constant acceleration, equal to the surface acceleration of the chosen body. A more complex gravitational model can also be used; however, during the surface contact moment the gravitational force is not as important as the contact forces. Thus, a constant acceleration for this short time period is sufficient. At the start of each timestep a collision detection is performed to determine which bodies in the simulation are in contact and where the contact points are located. Once the contact points are found, the related contact force is calculated. A smooth contact model is chosen as this works best for cases where continuous contacts are present, as is the case for the settled surface that is modeled here. For all the found contact points, the contact force is then calculated using a visco-elastic force model as follows:

$$\mathbf{F}_n = k_n \delta_n^p \hat{n} - \gamma_n \delta_n^q \mathbf{v}_n, \tag{1}$$

$$\mathbf{F}_t = -k_t \delta_t^s \hat{\delta}_t - \gamma_t \delta_t^q \mathbf{v}_t, \tag{2}$$

where k and γ are the stiffness and damping coefficients respectively, with the subscripts denoting their normal and tangential directions. The exponents p, q, s are chosen based on the specific contact model. In this work the Hertzian model is used, for which $p = 3/2$, $q = 1/4$, and $s = 1/2$. The tangential force is calculated using the Coulomb friction condition. This condition states: $|\mathbf{F}_t| \leq \mu|\mathbf{F}_n|$ where μ is the friction coefficient. So when $|\mathbf{F}_t| > \mu|\mathbf{F}_n|$, $\boldsymbol{\delta}_t$ is scaled such that $|\mathbf{F}_t| = \mu|\mathbf{F}_n|$.

The stiffness and damping coefficients can be used to obtain the desired contact properties. Using Hertzian theory, a relation can be found between these material properties and the spring-dashpot coefficients:[8]

$$k_n = \frac{4}{3}\bar{E}\sqrt{\bar{R}\delta_n}, \tag{3}$$

$$k_t = 8\bar{G}\sqrt{\bar{R}\delta_n}, \tag{4}$$

$$\gamma_n = -2\sqrt{\frac{5}{6}}\beta\sqrt{\frac{3}{2}\bar{m}k_n}, \tag{5}$$

$$\gamma_t = -2\sqrt{\frac{5}{6}}\beta\sqrt{\bar{m}k_t}. \tag{6}$$

where $\bar{E}$ is the effective Young's modulus and $\bar{G}$ as the effective shear modulus, given by:

$$\bar{E} = \left(\frac{1-\nu_i^2}{E_i} + \frac{1-\nu_j^2}{E_j}\right)^{-1}, \tag{7}$$

$$\bar{G} = \left(\frac{2(2+\nu_i)(1-\nu_i)}{E_i} + \frac{2(2+\nu_j)(1-\nu_j)}{E_j}\right)^{-1}. \tag{8}$$

Here i and j are the two bodies in contact. β is related to the coefficient of restitution e as follows:

$$\beta = \frac{\ln e}{\sqrt{\ln^2 e + \pi^2}}. \tag{9}$$

Finally, at the point of contact the effective radius of curvature and mass are $\bar{R} = (R_i^{-1} + R_j^{-1})^{-1}$ and $\bar{m} = (m_i^{-1} + m_j^{-1})^{-1}$.

The total equations of motion of each individual object in the simulation is:

$$m_i\ddot{\mathbf{r}}_i = m_i\mathbf{a} + \sum_{k\in\mathcal{K}}(\mathbf{F}_{n,ik} + \mathbf{F}_{t,ik}), \tag{10}$$

$$\mathbf{I}_i\dot{\boldsymbol{\omega}}_i = -\boldsymbol{\omega}_i \times (\mathbf{I}_i\boldsymbol{\omega}_i) + \sum_{k\in\mathcal{K}}\mathbf{T}_{ik}, \tag{11}$$

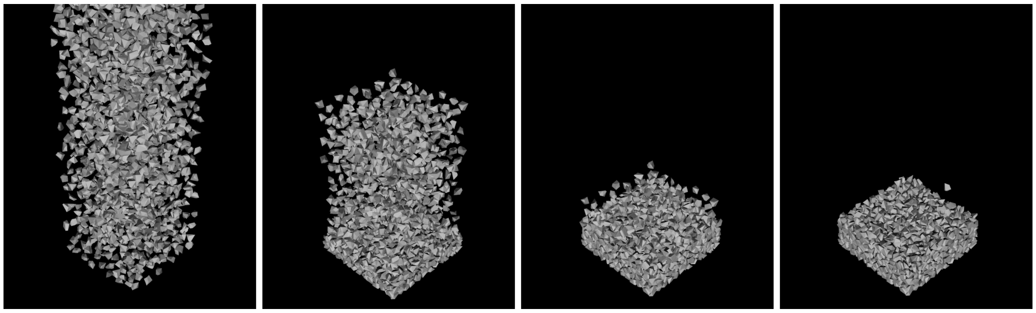


Figure 1: The creation of the SSB surface in GRAINS.

where m_i, $\mathbf{I}_i$, $\mathbf{r}_i$, and $\boldsymbol{\omega}_i$ are the mass, inertia tensor, position, and angular velocity of body i, respectively. The set of all contact points is denoted by $\mathcal{K}$. Finally, $\mathbf{T}$ is the torque of the particle given by off-center collisions and other contact related torques.

The first step in setting up the simulation environment is the creation of the surface. This process starts with the generation of a number of particles randomly placed inside a large volume. Each individual particles is created by generating a cloud of points and fitting a convex envelope around these points. The sizes of the particles are kept constant here but in case the surface is observed to have a specific size distribution then these particles can be generated as well to follow this distribution. Once the particles are generated, a container is placed below the volume and a simulation is started with a similar constant acceleration to the final landing simulation. This allows for the particles within the volume to naturally fall into the container and settle under the local surface acceleration. This process is shown in Figure 1. Once the surface has been created, the lander is simulated by generating a rigid geometry, placing it above the surface, and giving it an initial velocity magnitude and direction. The simulation is then run until the lander is far away enough of the surface that no interaction is possible anymore. An example simulation of a lander interacting with the settled surface is presented in Figure 2, where a constant acceleration of $0.5 \cdot 10^{-4}$ m/s^2 is used, representing roughly the surface acceleration of bodies like Didymos and Apophis.

POLYNOMIAL CHAOS EXPANSION

The simulation is mathematically modelled a forward model $\mathbf{f}$ defined as follows:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}(t_0), \boldsymbol{\xi}), \tag{12}$$

where $\mathbf{y} \in \mathbb{R}^m$ is a vector containing the output variables, $\mathbf{x}(t_0)$ is the initial state, and $\boldsymbol{\xi} \in \mathbb{R}^n$ is the model parameter vector. Here, the initial state $\mathbf{x}(t_0)$ is considered fixed and the model parameters are considered independent random variables; therefore the state dependency can be dropped and the model can be written as: $\mathbf{f}(\boldsymbol{\xi})$.

The main concept behind polynomial chaos expansion (PCE) is that the model can be represented using a set of orthogonal polynomials as follows:

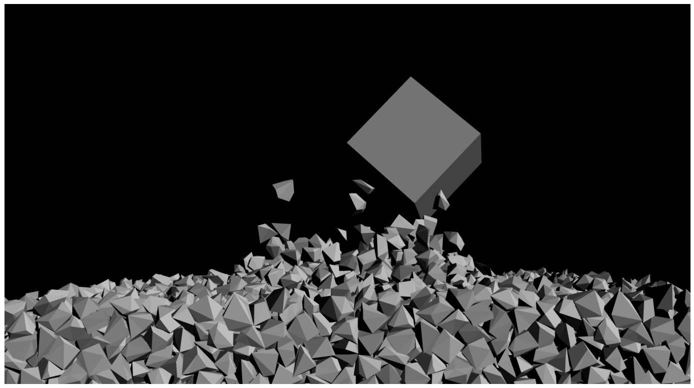


Figure 2: A snapshot of an example simulation of a landing on an SSB surface.

$$\tilde{f}(\boldsymbol{\xi}) = P_{n,d}(\boldsymbol{\xi}) = \sum_{\mathbf{i},|\mathbf{i}|\leq n} c_{\mathbf{i}} \Psi_{\mathbf{i}}(\boldsymbol{\xi}), \tag{13}$$

where d is the degree of the polynomial, n is the number of variables of the polynomial, $\mathbf{i} \in [0, d]^n \subset \mathbb{N}^n$ is a vector of indices with $\mid \mathbf{i} \mid = i = \sum_{r=1}^{n} i_r$, $\{c_{\mathbf{i}}\}$ are a set of coefficients, and $\{\Psi_{\mathbf{i}}(\boldsymbol{\xi}) = \alpha_{i_0}(\xi_0) \cdot \alpha_{i_1}(\xi_1) \cdot ... \cdot \alpha_{i_d}(\xi_d)\}$ are multivariate basis functions generated using a tensor product of univariate polynomials. The polynomial basis functions $\{\Psi_i\}$ are orthogonal under the weight function ρ:

$$\langle \Psi_n, \Psi_m \rangle = \int \Psi_n(\boldsymbol{\xi}) \Psi_m(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi} = \langle \Psi_n^2 \rangle \delta_{n,m}. \tag{14}$$

The size of the expansion, i.e. the number of terms of the polynomial, is given by $\mathcal{N} = \binom{n+d}{d}$, where d is the maximum degree of the polynomial.

The original version of this method was called homogeneous chaos, or also Wiener chaos after the author.[9] There, Hermite polynomials were used as basis functions and a Gaussian distribution was considered. It was found that the expansion exponentially converges to the true output distribution, mainly due to the fact that the weight function ρ associated with Hermite polynomials has the same functional form as a Gaussian distribution. This idea was expanded upon in[10] to other orthogonal polynomial families which have other weight functions that are also associated to different classes of probability density functions through the so-called Askey scheme. This way, non-Gaussian distributed functions are approximated with exponential convergence as well by selecting the corresponding polynomial basis from the Askey scheme.

The coefficients $\{c_i\}$ can be found in various different ways, both using intrusive and non-intrusive methods. Here, a non-intrusive point collocation method is selected due to its found accuracy. For this method a set of Q input parameters, also known as nodes, are samples and propagated forward through the model. Then, the following problem is solved:

$$\{c_{\mathbf{i}}\} = \arg\min_{c_{\mathbf{i}}} \sum_{k=0}^{Q} \left[f(\boldsymbol{\xi}_{\mathbf{k}}) - \sum_{\mathbf{i},|\mathbf{i}|\leq n} c_{\mathbf{i}} \Psi_{\mathbf{i}}(\boldsymbol{\xi}_{\mathbf{k}}) \right], \tag{15}$$

using a least squares regression technique by inverting the following system:

$$HC = Y, \tag{16}$$

where:

$$H = \begin{bmatrix} \Psi_{i_1}(\boldsymbol{\xi}_1) & \dots & \Psi_{i_s}(\boldsymbol{\xi}_1) \\ \vdots & \ddots & \vdots \\ \Psi_{i_1}(\boldsymbol{\xi}_s) & \dots & \Psi_{i_s}(\boldsymbol{\xi}_s) \end{bmatrix}, C = \begin{bmatrix} c_{i_1} \\ \vdots \\ c_{i_s} \end{bmatrix}, Y = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_s \end{bmatrix} \tag{17}$$

where $s = \mathcal{N}$, $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_s$ are the nodes, and Y the vector containing all the corresponding propagated samples $\mathbf{y}_i = \mathbf{f}(\boldsymbol{\xi}_i)$.

The samples can be chosen using various different sampling rules, each of which significantly changes the accuracy of the expansion. Most of these sampling rules lead to a rapidly growing amount of samples when the amount of variables grows, as they are extended to multivariate spaces through tensor products. This is also known as the curse of dimensionality. An alternative approach, known as the Smolyak sparse grid, was developed in,[11] and selects a set of points based on the extrema of Chebyshev polynomials. An important aspect is that this approach does not suffer from the curse of dimensionality, as the number of points grows polynomially with the dimension of the problem instead of exponentially. This property can be seen in Figure 3 where the amount of samples for a given amount of variables and polynomial degree are given for both a standard full tensor grid approach and for the Smolyak sparse grid. It can be seen that the amount of samples for the sparse grid grows much slower than the full tensor grid. The methodology behind this technique is beyond the scope of this work. A more in depth explanation of this method for uncertainty propagation is given in.[12]

There are various properties of the resulting PCE model that can be utilised. First, the PCE model allows for an analytical formulation of the moments of the output distribution through the orthogonal property of the polynomial basis:

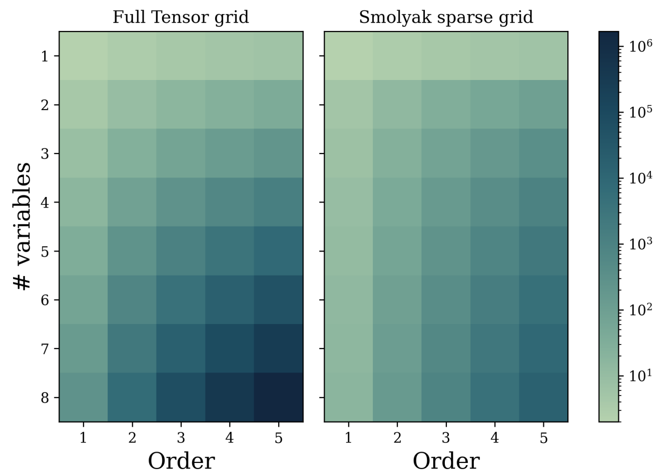


Figure 3: The amount of quadrature points for different number of input variables and polynomial order.

$$\bar{y} = \mathbb{E}[y] \approx \mathbb{E}[P_{n,d}(\boldsymbol{\xi})] = \int \left(\sum_{\mathbf{i},|\mathbf{i}|\leq n} c_{\mathbf{i}} \Psi_{\mathbf{i}}(\boldsymbol{\xi}) \right) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi} = c_{\mathbf{0}}, \tag{18}$$

where the fact that $\Psi_0 = 1$ and $\mathbb{E}[\Psi_{\mathbf{i}}] = 0,\ \forall \mathbf{i} \neq \mathbf{0}$ is used to obtain the final relation. With this result, the second moment, or the variance, is obtained:

$$\sigma^2 = \mathbb{E}[(y - \bar{y})^2] \tag{19}$$

$$\approx \int \left(\sum_{\mathbf{i},|\mathbf{i}|\neq 0} c_{\mathbf{i}} \Psi_{\mathbf{i}}(\boldsymbol{\xi})) \right) \cdot \left(\sum_{\mathbf{i},|\mathbf{i}|\neq 0} c_{\mathbf{i}} \Psi_{\mathbf{i}}(\boldsymbol{\xi}) \right) \rho(\boldsymbol{\xi}) d\xi \tag{20}$$

$$= \sum_{\mathbf{i},|\mathbf{i}|\neq 0} \langle \Psi_{\mathbf{i}}^2 \rangle \, c_{\mathbf{i}}^2. \tag{21}$$

Using this variance measure, the first order Sobol indices $\mathbf{S}_i$ can be calculated as well. These indices measure the expected reduction in the variance of the output $\mathbf{y}$ when fixing the input parameter ξ_i. Thus, they also measure the contribution of the variance of an input parameter to the output parameter. Mathematically, they are defined as:

$$\mathbf{S}_i = \frac{Var[\mathbb{E}(\mathbf{y} \mid \xi_i)]}{Var[\mathbf{y}]}. \tag{22}$$

For the PCE based sensitivity indices, the basic idea is that the variance can be computed similarly to Eq. (21).[13] First, define $\mathcal{I}_{i_1,...,i_s}$ as the set of $\mathbf{i}$ tuples such that $(i_1, ..., i_s)$ are nonzero. This would mean that $\mathcal{I}_i$ corresponds to all the terms in Eq. (13) that only depend on ξ_i. Using this notation, the Sobol indices of Eq. (22) can be written as:

$$\mathbf{S}_i = \frac{\sum_{\mathbf{i}\in\mathcal{I}_i} \langle \Psi_{\mathbf{i}}^2 \rangle \; c_{\mathbf{i}}^2}{\sum_{\mathbf{i},|\mathbf{i}|\neq 0} \langle \Psi_{\mathbf{i}}^2 \rangle \; c_{\mathbf{i}}^2}. \tag{23}$$

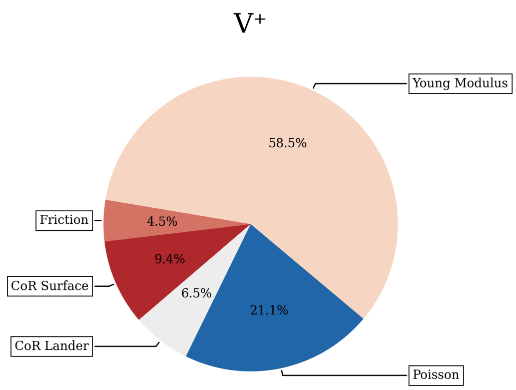
The main idea of the proposed methodology is thus to create a reduced order model (ROM) of the interaction between the lander and the SSB surface by generating a low amount of samples, running the simulation for each of these samples, calculating the resulting output variables, and then fitting the model to these input-output pairs. Then, the ROM can be used to perform various statistical analyses, as will be shown in the next section.

RESULTS

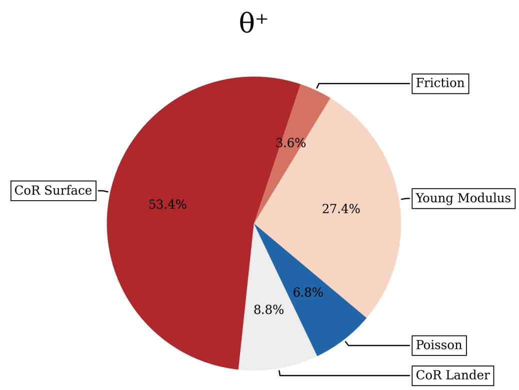
This section will show the accuracy of the presented method, and how the resulting ROM can be used for various types of analyses. For this case, a fixed initial velocity and landing direction is used. This way, the focus is mainly on the uncertain properties of the SSB surface and how they influence the motion of the lander. In real-world scenarios, generally the landing velocity direction and magnitude are also uncertain. These two variables can simply be added to the PCE model without any augmentation of the overall method. However, here it is chosen to focus on the surface parameters to keep the problem small enough to test its accuracy better, without loss of generality as the method does not change when incorporating other uncertain parameters like the initial state.

The uncertainty vector $\boldsymbol{\xi}$ contains the following variables: Young modulus Y, Poisson ratio ν, coefficient of restitution (CoR) of both the lander and surface particles e, and the friction parameter μ. Due to high number of variables considered, a two step approach is adopted. For the first step, a medium-order (3rd order) polynomial model is generated considering all variables previously mentioned. This medium-order model is used perform a sensitivity analysis using the Sobol indices of Eq. (23). For these indices the error of the model is less important due to the fact that its a relative calculation, thus a medium order model (determined to have a mean error of around 10%) is sufficient. The set of variables that are responsible for the majority of the output variance are then selected for the second step. The second step consists of generating a high-order model (5th order), using the lower dimensional uncertainty vector, which can be used for analyses where the accuracy of the model is of greater importance. As the growth of the number of required quadrature points when increasing the order and/or the number of variables is high, this two-step approach reduces significantly the amount of simulations that need to be run compared to having a high-order model with a large uncertainty vector dimension.

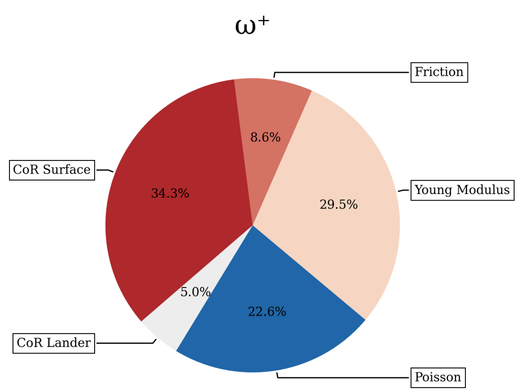
The results from the sensitivity analysis using the medium-order model are presented in Figure 4. Three output variables are considered: the outgoing velocity magnitude V^+, the outgoing velocity



(a) Outgoing velocity magnitude.



(b) Outgoing velocity direction.



(c) Outgoing angular speed.

Figure 4: Pie charts representing the Sobol index values for each input parameter, given in percentage of the total output variance, for the three output variables.

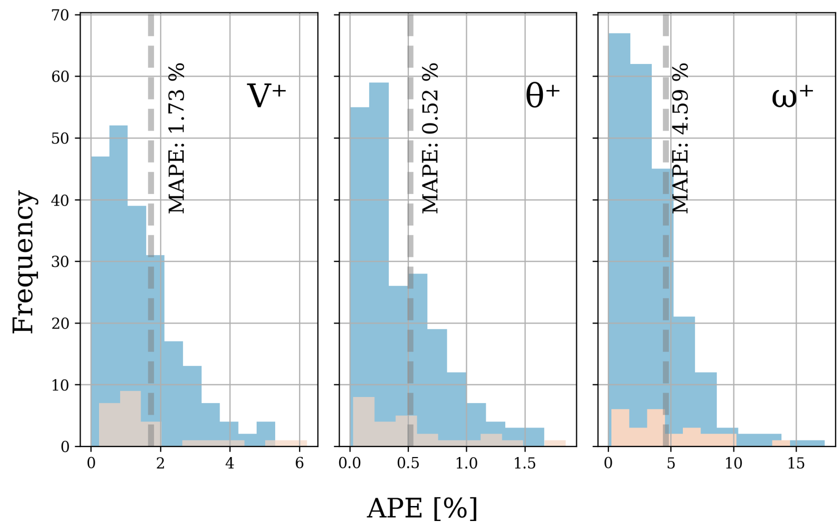


Figure 5: The absolute percentage error (APE) for the multivariate PCE model. The vertical dashed line represents the mean APE (MAPE). The blue histogram represents the training nodes of the PCE model whereas the light colored bars are from the test set.

direction θ^+, and the outgoing angular speed ω^+. For all three variables it is clear that the most important input parameters are Y and the surface ϵ. For both V^+ and ω^+ it can also be seen that ν plays a relatively important role as well. The friction μ has some influence on those parameters as well, but less than ν, and as more than 90 % of all variance is captured by the previously mentioned three variables, the friction is left out of consideration for now.

Now the second step of the approach is executed, where a 5th order polynomial is used to generate a high accuracy model. To determine the accuracy of this model, an analysis is performed where both the accuracy versus the input nodes and versus a testing set are measured. Figure 5 shows the absolute percentage error (APE), defined as:

$$\text{APE} = \frac{|\tilde{y}_i - y_i|}{y_i} \cdot 100\%, \tag{24}$$

where y_i is the true ith output variable, and $\tilde{y}$ is the PCE model output. It can be seen that in general the APE is low for both the training and testing set, with no significant increase in error when considering the testing set. The mean APE (MAPE) is shown to be 1.73 % for V^+ and 0.52 % for θ^+, sufficiently low for most applications considered here. The error in ω^+ is slightly higher, 4.59 %, which is mainly due to the fact that the absolute value of ω^+ is in general very low, thus the PCE model has a hard time fitting this variable.

Now that it has been validated that the PCE model is sufficiently accurate, several different types

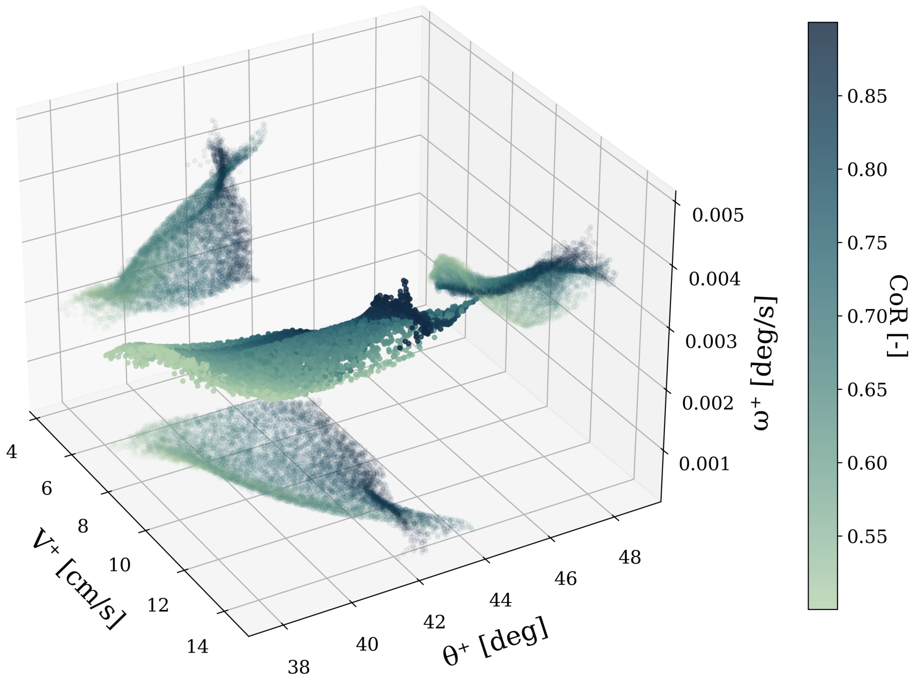


Figure 6: The outputs of $1e5$ samples generated from the PCE model. The color represents the corresponding input CoR.

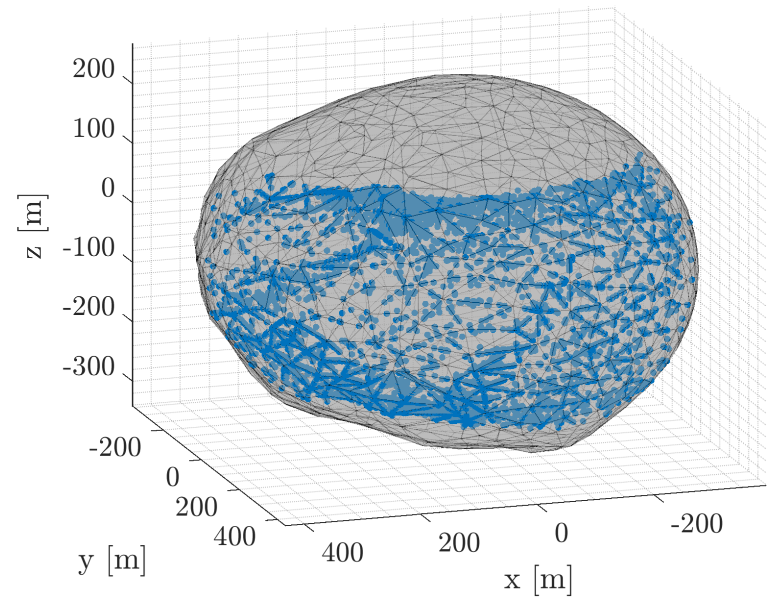


Figure 7: Distribution of landing locations.

of analyses can be performed. First, a general Monte Carlo analysis with $1e5$ points is performed to estimate the joint distribution of the output variables and how the input variables influence the distribution. It is noted that this type of analysis is practically infeasible with the original DEM model as each simulation can take between 45 minutes to an hour, whereas now simply a polynomial evaluation is performed for each sample, requiring a total computational time of the full Monte Carlo analysis in the order of minutes. Figure 6 shows the output distribution generated from the MC analysis. The non-linear nature of the system is clear from the shape of the distribution as this is far from Gaussian behavior. The corresponding input coefficient of restitution (CoR) is also shown in Figure 6, where, as expected, a correlation can be observed between higher velocities and higher CoRs. However, the specific output velocity can not be generally inferred from the input CoR. This is because the CoR of the interaction between individual surface particles is considered and not the global CoR, which is defined as the ratio between input and output velocity of the lander, which is the one used in hard surface models. This shows that the PCE model is capable of retaining the complex dynamical features of soft body models, while having a much lower computational burden.

The output distribution of Figure 6 can also be used to generate a landing dispersion estimate. The output velocity magnitudes and directions are used as initial conditions for a numerical propagator that considers the polyhedron shape based gravitational influence of the central body (here considered to be Didymos). These initial conditions are propagated forward in time until either the final stopping time of 12 hours is reached or they re-impact the surface of the asteroid. The resulting distribution of the landing locations can be seen in Figure 7, where a landing location of 0 degree latitude and 0 degree longitude is considered. It can be seen that even though the velocity of the lander is significantly reduces after the first bounce, the dispersion after the first bounce is still high. Finally, it was found that for this case only 48.26 percent of trajectories remain on the surface.

CONCLUSION

This work presents a novel approach to simulating the interaction between a spacecraft and the surface of a small Solar system body (SSB). A polynomial chaos expansion (PCE) was used to generate a model that retains the complexities of a soft surface model, but which also has the computational efficiency of a hard surface model.

A large amount of surface parameters were considered at first that characterize the geomechanical properties of the surface material. During a preliminary analysis with a medium-order PCE model, it was shown that considering just three of these parameters (the Young Modulus, coefficient of restitution, and Poisson ratio), more than 90 percent of the output variance is retained. With this a second more accurate PCE model is generated, which was shown to have errors $< 5\%$ with respect to the soft surface, DEM, model.

A statistical analysis of the output of the PCE model showed that the coefficient of restitution of the individual surface particles have a large influence on the outgoing velocity magnitude and direction. This in turn leads to a large dispersion of post-bouncing landing locations, underlying the need for accurate and efficient models for bouncing in low-gravity environments.

Concluding, the methodology presented in this work is shown to enable mission designers to perform robust and efficient landings of spacecraft on the surface of SSBs, creating novel science and exploration opportunities.

ACKNOWLEDGMENT

This work was partly carried out within the Space It Up! project funded by the Italian Space Agency (ASI) and the Italian Ministry of University and Research (MUR) under contract No. 2024-5-E.0, CUP No. I53D24000060005. Partly funded by the European Union (ERC, TRACES, 101077758). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] P. Michel, M. Küppers, A. C. Bagatin, B. Carry, S. Charnoz, and others, "The ESA Hera Mission: Detailed Characterization of the DART Impact Outcome and of the Binary Asteroid (65803) Didymos," *The Planetary Science Journal*, Vol. 3, 7 2022, p. 160, 10.3847/PSJ/AC6F52.

[2] S. Ulamec, P. Michel, M. Grott, U. Böttger, S. Schröder, H. W. Hübers, Y. Cho, F. Rull, N. Murdoch, P. Vernazza, O. Prieto-Ballesteros, J. Biele, S. Tardivel, D. Arrat, T. Hagelschuer, J. Knollenberg, D. Vivet, C. Sunday, L. Jorda, O. Groussin, C. Robin, and H. Miyamoto, "Science objectives of the MMX rover," *Acta Astronautica*, Vol. 210, 9 2023, pp. 95–101, 10.1016/J.ACTAASTRO.2023.05.012.

[3] N. Murdoch, M. Drilleau, C. Sunday, F. Thuillet, A. Wilhelm, G. Nguyen, and Y. Gourinat, "Low-velocity impacts into granular material: application to small-body landing," *Monthly Notices of the Royal Astronomical Society*, Vol. 503, 4 2021, pp. 3460–3471, 10.1093/MNRAS/STAB624.

[4] I. Fodde, J. Feng, M. Vasile, and J. Gil-Fernández, "Design of Robust Ballistic Landings on the Secondary of a Binary Asteroid," *Journal of Guidance, Control, and Dynamics*, 8 2024, pp. 1–17, 10.2514/1.G007786.

[5] F. Thuillet, P. Michel, C. Maurel, R. L. Ballouz, Y. Zhang, D. C. Richardson, J. Biele, E. Tatsumi, and S. Sugita, "Numerical modeling of lander interaction with a low-gravity asteroid regolith surface - Application to MASCOT on board Hayabusa2," *Astronomy & Astrophysics*, Vol. 615, 7 2018, p. A41, 10.1051/0004-6361/201832779.

[6] F. Ferrari, A. Tasora, P. Masarati, and M. Lavagna, "N-body gravitational and contact dynamics for asteroid aggregation," *Multibody System Dynamics*, Vol. 39, 1 2017, pp. 3–20, 10.1007/S11044-016-9547-2/FIGURES/9.

[7] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischman, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," *High Performance Computing in Science and Engineering*, Vol. 9611, Springer Verlag, 2016, pp. 19–49, 10.1007/978-3-319-40361-8_2/FIGURES/16.

[8] J. Fleischmann, "DEM-PM contact model with multi-step tangential contact displacement history," tech. rep., University of Wisconsin, Madinson, 2015.

[9] N. Wiener, "The Homogeneous Chaos," *American Journal of Mathematics*, Vol. 60, 10 1938, p. 897, 10.2307/2371268.

[10] D. Xiu and G. Em Karniadakis, "The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations," *SIAM Journal on Scientific Computing*, Vol. 24, 7 2006, pp. 619–644, 10.1137/S1064827501387826.

[11] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Doklady Akademii Nauk SSSR*, Vol. 148, No. 5, 1963, pp. 1042–1045.

[12] A. Riccardi, C. Tardioli, and M. Vasile, "An intrusive approach to uncertainty propagation in orbital mechanics based on tchebycheff polynomial algebra," *Advances in the Astronautical Sciences*, Vol. 156, Vail, Univelt Inc., 2016, pp. 707–722.

[13] B. Sudret, "Global sensitivity analysis using polynomial chaos expansions," *Reliability Engineering & System Safety*, Vol. 93, 7 2008, pp. 964–979, 10.1016/J.RESS.2007.04.002.