Acta Astronautica 213 (2023) 725–740

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

Simultaneous navigation and mascon gravity estimation around small bodies

Julio C. Sanchez [a,b,*], Hanspeter Schaub [b]

[a] Departamento de Ingenieria Aeroespacial y Mecanica de Fluidos, Universidad de Sevilla, 41092, Sevilla, Spain
[b] Department of Aerospace Engineering Sciences, University of Colorado Boulder, 80309, Boulder, United States

ARTICLE INFO

Keywords:
Small body exploration
Spacecraft autonomy
Deep space navigation
Gravity field modeling

ABSTRACT

This manuscript develops a simultaneous navigation and gravity estimation strategy around a small body. The scheme combines dynamical model compensation with a mascon gravity fit. Dynamical compensation adds the unmodeled acceleration to the filter state. Consequently, the navigation filter is able to generate an on-orbit position-unmodeled acceleration dataset. The available measurements correspond to the landmarks-based navigation technique. Accordingly, an on-board camera is able to provide landmark pixels. The aforementioned position-unmodeled acceleration dataset serves to train a mascon gravity model on-board while in flight. The training algorithm finds the optimal mass values and locations using Adam gradient descent. By a careful choice of the mascon variables and constraints projection, the masses are ensured to be positive and within the small body shape. The numerical results provide a comprehensive analysis on the global gravity accuracy for different estimation scenarios.

1. Introduction

Small bodies exploration is of great scientific interest because these objects are able to explain Solar System formation processes [1]. They could also collide with Earth, thus it is pertinent to design and test asteroid deflection techniques [2]. There are several small body missions planned in the 2020s decade. Two prominent examples include Psyche [3] which will explore the metallic asteroid of the same name and Hera [4] which will analyze in-situ the DART [5] impact on Dydimos's moon. Currently, OSIRIS-REx mission [6] is returning to Earth with collected samples from asteroid Bennu.

In the vicinity of a small body, the spacecraft motion may be highly perturbed from a Keplerian orbit. This is due to the strong effects of inhomogeneous gravity and solar radiation pressure [7]. If unaccounted for these perturbations can lead to escape or collision trajectories. Therefore, they have to be taken into account in order to ensure safe flight operations. However, before a small body mission launches, only coarse information of the body shape can be inferred through Earth-based radar and telescope measurements. While this is not a concern for solar radiation pressure, it certainly is for the small body gravity field determination. As a matter of fact, the inhomogeneous gravity is only observable in the close neighborhood of the small body. A practical example is the NEAR mission around the asteroid Eros. The Eros spherical harmonics gravity was observable, relative to their uncertainty, up to 10th degree [8]. The gravity determination process largely relied on processing Deep Space Network (DSN) range and range-rate with long (up to 30 days) optical observations arcs. Although this methodology reduces the needs for on-board computation, it slows down the mission timeline due to long data arcs, signal delays and DSN constrained accessibility. Consequently, automating the gravity field determination process has the potential to speed up the transition to low altitude operations (with high scientific return). Moreover the dependence on costly and constrained Earth-based infrastructure is reduced.

In the last decade, several publications [9–17] are proposing autonomous gravity estimation strategies. Refs. [9,10] revisited NEAR mission orbit determination and gravity field estimation by using linked autonomous interplanetary satellite orbit navigation (LiAISON). This technique exploits the gravity field asymmetries along with relative satellite-to-satellite range and range-rate measurements to provide precise orbit determination. In particular, [10] demonstrates that Eros gravity field is theoretically observable up to 9th degree by using a suitable beacon-satellite configuration. Refs. [11,12] focus on close formation satellites (namely swarm) for simultaneous navigation and asteroid characterization. These works fuse landmarks-based and inter-satellite ranging measurements within a centralized unscented Kalman filter (UKF). The UKF estimates the swarm state and spherical harmonics coefficients (amongst other variables of interest). Second-order gravity and solar sail reflection coefficient degradation are inferred in [13]. The parameters estimation is done through recursive least-squares fitting of the unmodeled acceleration (which is inferred with an

[*] Corresponding author at: Departamento de Ingenieria Aeroespacial y Mecanica de Fluidos, Universidad de Sevilla, 41092, Sevilla, Spain.
E-mail address: jsanchezm@us.es (J.C. Sanchez).

https://doi.org/10.1016/j.actaastro.2023.09.023
Received 9 May 2023; Received in revised form 21 August 2023; Accepted 17 September 2023
Available online 6 October 2023

extended state observer). However, the simulations lack realism as perfect knowledge of the spacecraft position is assumed. Ref. [14] develops a gravity model learning-based predictive control for orbit-attitude station-keeping. A spherical harmonics gravity model is learned by averaging UKF individual estimates of a satellite constellation (which uses landmarks and laser-based navigation). Multiple satellites are able to mitigate outliers and augment the convergence rate with respect to a single satellite. The results also show the positive impact of model learning in orbit control accuracy and torque demands. Two different machine-learning approaches are used in [15,16]. Ref. [15] utilizes Hopfield neural networks for spherical harmonics gravity estimation. The results show that the neural network estimation is able to provide similar performance as extended Kalman filtering (EKF) with a lower computational cost. However, the navigation component is simplified by assuming zero-mean white Gaussian noise of position and velocity. Differently, [16] couples a dynamical model compensated filter (which estimates a position-unmodeled acceleration dataset) with a physics-informed neural network gravity representation. While marginal improvement with respect to a point-mass model is achieved, it seems a promising data-driven technique for gravity field determination.

The autonomous gravity estimation literature shows a clear preference for the spherical harmonics model [9–15] (except Ref. [16]). The spherical harmonics model is generally valid only outside the body circumscribing sphere (namely Brillouin sphere). This is of great concern when planning low altitude operations (descent, landing or touch and go) because the gravity prediction may be flawed in that domain. In order to avoid spherical harmonics Brillouin sphere divergence, alternative gravity models [18–23] (amongst others) can be employed. For on-board execution and estimation, several considerations have to be analyzed (e.g. computational efficiency) besides accuracy. The polyhedron [18] and novel neural density fields [23] models provide a high global accuracy but are expensive computationally. The ellipsoidal harmonics model [19] reduces the size of the spherical harmonics divergence domain but does not avert it. The mascon model [20] can be adjusted to offer a balance between accuracy and computational load. Machine-learning based models (also [23]) are recently proposed in the form of Gaussian processes [21] and physics informed neural networks [22]. These are a suitable data-driven options but are on an early stage in terms of convergence outside the dataset domain (e.g. lack of data close to the surface).

Under the previous considerations, this work explores the mascon model for autonomous gravity estimation. This model represents the gravity field as the joint contributions of multiple simple shapes (e.g. spheres) discretizing the small body. A typical research direction consists on using mascon distributions that approximate the constant density polyhedron model [24]. Alternatively, [20,25,26] use optimization techniques to find the mascon distribution that fits a dataset. Ref. [20] fits the Earth's gravity field beyond J_2 by assuming the point masses are fixed a-priori. In [25], a genetic algorithm finds the optimal point masses locations and its values. In a similar spirit, [26] employs a Newton–Raphson algorithm to compute the optimal distribution of a low number of point masses. The previous works assume true position-acceleration (or potential in [20]) datasets for their mascon trainings. Moreover, [20,25] datasets contain abundant and homogeneously distributed samples across the spatial domain while [26] relies on low altitude points. While these conditions are acceptable for gravity field modeling analysis, they are possibly unfeasible during the on-orbit scenario. Specifically, the data may be highly concentrated on a specific orbital regime (though low altitude samples may be obtained with additional hardware [27] or ejecta observations). Even more crucial, the dataset is also corrupted by navigation uncertainties. Following the previous consideration, the on-orbit generation of the gravity acceleration data poses a question on itself. To this end, the concept of dynamical model compensation (DMC) could be tailored to this specific application. The DMC key idea is to estimate the unmodeled dynamics component (as an acceleration) along with the spacecraft state. Operating as such, the persistent unknown dynamical bias is removed from the filter process which is of high interest for precise orbit determination [28,29].

This manuscript combines a dynamical model compensated unscented Kalman filter (DMC-UKF) with the training of a mascon gravity model. Following previous works in small body navigation [9–11,14, 30,31], the spacecraft is assumed to be equipped with a camera that tracks landmarks on the small body surface. The identified landmark pixels on the image plane (which is out of the scope of this work) constitute the DMC-UKF measurements. The DMC-UKF generates an on-orbit position-unmodeled acceleration dataset over time. Then, this dataset serves to fit a mascon gravity model using Adam gradient descent. This paper extends and completes the authors initial communication of [14]. The major breakthrough is that the mascon training has now the feature to find the optimal masses spatial distribution (instead of just the masses values). Moreover, physical constraints such as masses positiveness and its containment within the body shape are explicitly accounted for. To resume, the main contributions of this work are: (1) the autonomous gravity estimation of a mascon model which does not diverge within Brillouin sphere; (2) the resulting mascon distribution is physically consistent (positive masses within the body shape) and, (3) the potential flexibility (to other gravity models) of the proposed strategy since the dataset generation is decoupled from gravity estimation.

The structure of the paper is as follows. Section 2 describes spacecraft dynamics around a small body. Section 3 presents the camera model and DMC-UKF algorithm. Section 4 develops the mascon gravity training algorithm and its integration with DMC-UKF. Section 5 shows numerical results of interest. Section 6 concludes this paper and states future research directions.

2. Dynamics around a small body

A spacecraft in the proximity of a small body is perturbed by the gravity field, the Sun's third body gravity and the solar radiation pressure as

$$\ddot{\mathbf{r}}^N = (\mathbf{R}^A_N)^T\mathbf{a}_{\text{poly}} + \mathbf{a}^N_{\odot} + \mathbf{a}^N_{\text{SRP}}, \tag{1}$$

where $\mathbf{r}^N$ is the spacecraft position expressed in an inertial small body centered frame $N \equiv \{\mathbf{0} : \mathbf{i}_N, \mathbf{j}_N, \mathbf{k}_N\}$ where $\mathbf{0}$ is the small body center of mass and $\mathbf{k}_N$ its major inertia axis. The Sun's third body gravity perturbation is $\mathbf{a}^N_{\odot}$ and the solar radiation pressure acceleration is $\mathbf{a}^N_{\text{SRP}}$. The term $\mathbf{a}_{\text{poly}}$ is the small body polyhedron gravity acceleration [18] which is assumed as the ground truth (it is further detailed in paragraph 2.2.1). This vector is expressed in a rotating small body centered frame $A \equiv \{\mathbf{0} : \mathbf{i}_A, \mathbf{j}_A, \mathbf{k}_A\}$. Let us consider $\mathbf{k}_A \equiv \mathbf{k}_N$, thus the equatorial plane is defined by $\mathbf{i}_A\mathbf{j}_A$. In order to ease the notation, the superscript A is omitted for any vector expressed in the rotating frame. The N and A frames are related through the direction cosine matrix $\mathbf{R}^A_N$. By assuming the small body rotates at a constant rate ω_A around its major inertia axis (as is the usual case for small bodies), the direction cosine matrix $\mathbf{R}^A_N$ is

$$\mathbf{R}^A_N = \begin{bmatrix} \cos(\text{LST}_0 + \omega_A t) & \sin(\text{LST}_0 + \omega_A t) & 0 \\ -\sin(\text{LST}_0 + \omega_A t) & \cos(\text{LST}_0 + \omega_A t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where LST_0 is the small body initial local sidereal time and t is the elapsed simulation time.Following [10,14,30], the small body rotational state $\mathbf{R}^A_N$ is assumed as known in this paper. This answers to the purpose of purely assessing the relation between spacecraft state and gravity field estimation. The reference frames are illustrated in Fig. 1.

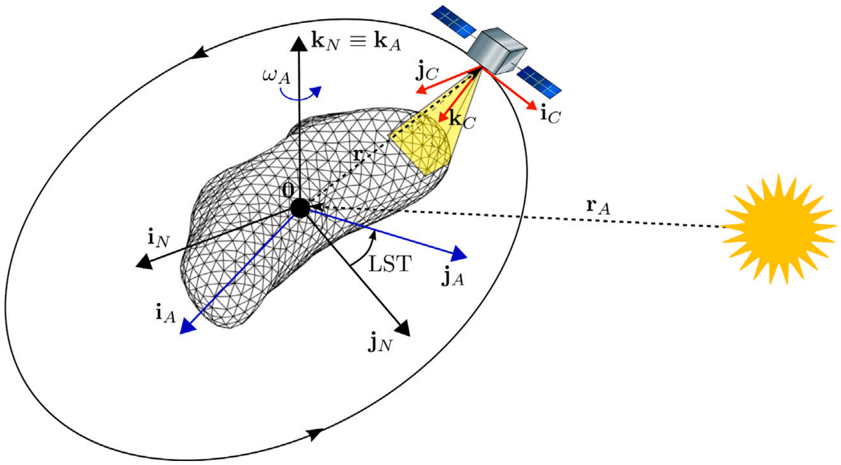

Fig. 1. Reference frames.

2.1. Solar perturbations

The Sun's third body gravity $\mathbf{a}_{\odot}^N$ and the solar radiation pressure $\mathbf{a}_{\mathrm{SRP}}^N$, are described as

$$\mathbf{a}_{\odot}^N = -\mu_{\odot}\left(\frac{\mathbf{r}_A^N + \mathbf{r}^N}{\|\mathbf{r}_A^N + \mathbf{r}^N\|^3} - \frac{\mathbf{r}_A^N}{r_A^3}\right), \quad \mathbf{a}_{\mathrm{SRP}}^N = \frac{C_R S W_{\oplus} r_{\oplus}^2}{mc\|\mathbf{r}_A^N + \mathbf{r}^N\|^3}(\mathbf{r}_A^N + \mathbf{r}^N), \tag{3}$$

where $\mathbf{r}_A^N$ is the small body relative position with respect to the Sun and $\mu_{\oplus} = 1.3271244 \cdot 10^{20}$ m^3/s^2 is the Sun's standard gravity parameter. The expression of the solar radiation pressure corresponds to the simple cannonball model. The term m is the spacecraft mass, C_R its reflection coefficient, S is the exposed surface to the photons, $W_{\oplus} = 1366$ W/m^2 is the mean energy flux received from the Sun at the mean orbital distance of $r_{\oplus} = 1$ AU and $c = 3 \cdot 10^8$ m/s^2 is the speed of light.

2.2. Small body gravity field

Throughout this paper, two models are used to characterize the small body gravity. As previously mentioned, the polyhedron model is assumed as ground truth (for validation) while mascon models are utilized for gravity estimation.

2.2.1. Polyhedron model

Ref. [18] expresses the exterior gravity field generated by a constant density polyhedron (described by faces and vertexes) as

$$\mathbf{a}_{\mathrm{poly}} = -\frac{\mu}{V}\left(\sum_{e \in \mathrm{edges}} \mathbf{E}_e \cdot \mathbf{r}_e L_e - \sum_{f \in \mathrm{faces}} \mathbf{F}_f \mathbf{r}_f w_f\right), \tag{4}$$

where V is the body volume, $\mathbf{r}_e$ is the relative position of the evaluation point with respect to the edge origin, $\mathbf{E}_e$ is the dyad product resulting from the edge and face normals, L_e is the potential of the edge as a 1D wire, $\mathbf{r}_f$ is the relative position of the evaluation point with respect to a vertex on a face, $\mathbf{F}_f$ is the outer product of the face normal vector and w_f is the solid angle of the face as viewed from the evaluation point. The explicit details of the derivation can be consulted in Ref. [18]. This model is useful for validation because it is globally accurate for the given shape with constant density. Although the constant density assumption may not hold in a real scenario, it is considered acceptable for synthetic simulations.

2.2.2. Mascon model

The mascon model characterizes the gravity field by adding the contributions of several simple volume elements (namely mascons). These can be homogeneous spheres with different densities. In this work only the exterior (outside the body volume) component of the gravity field is of interest, thus the mascon elements are denoted as point masses. Then, the mascon gravity is

$$\mathbf{a}_M = -\sum_{k=0}^{n} \mu_{M_k} \frac{\mathbf{r} - \mathbf{r}_{M_k}}{\|\mathbf{r} - \mathbf{r}_{M_k}\|^3}, \tag{5}$$

where $n + 1$ is the number of point masses while μ_{M_k} and $\mathbf{r}_{M_k}$ are, respectively, the standard gravity parameter and position of each one. In order to avoid singularities in the external gravity field evaluation, the point masses shall be within the small body shape.

3. Dynamical model compensated UKF

This section starts describing the pinhole camera model which maps 3D points to pixels in the image plane. Then, the dynamical model compensated unscented Kalman filter is stated. The DMC-UKF has the task of generating a position-unmodeled acceleration dataset for gravity estimation.

3.1. Pinhole camera model

On-board optical navigation is a state of the art technique to acquire relative measurements with respect to a small body. From a high-level perspective, the underlying process can be simulated using a pinhole camera model. This model provides algebraic expressions that map 3D points to pixels. Let us define the camera reference frame as $C \equiv \{\mathbf{r} : \mathbf{i}_C, \mathbf{j}_C, \mathbf{k}_C\}$ which, for simplicity, is centered at the spacecraft center of mass $\mathbf{r}$ (since the offset between camera aperture and center of mass is a known constant) and $\mathbf{k}_C$ is the optical axis pointing towards the viewing direction (see Fig. 1). Then, the projection of a 3D point (x_C, y_C, z_C), expressed in the C frame, to virtual image plane coordinates (u, v) is as follows

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z_C}\begin{bmatrix} x_C \\ y_C \end{bmatrix}, \tag{6}$$

where f is the camera focal length. Since the image is digital, the 3D point maps to a pixel (p_x, p_y) as

$$p_x = \begin{cases} \mathrm{ceil}(u/w_p) - 0.5 & \text{if} \quad u \geq 0, \\ \mathrm{floor}(u/w_p) + 0.5 & \text{if} \quad u < 0, \end{cases} \qquad p_y = \begin{cases} \mathrm{ceil}(v/w_p) - 0.5 & \text{if} \quad v \geq 0, \\ \mathrm{floor}(v/w_p) + 0.5 & \text{if} \quad v < 0. \end{cases}$$

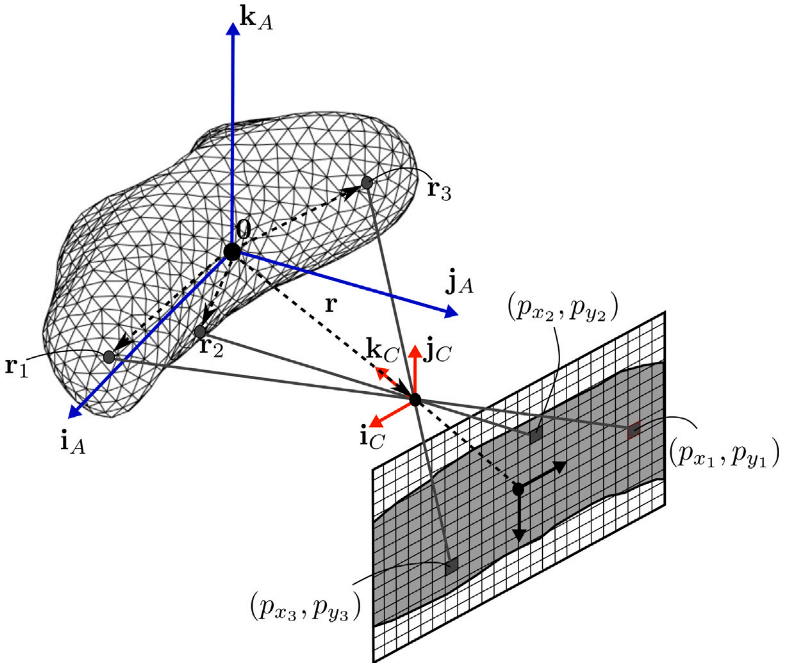

Fig. 2. Illustration of the pinhole camera model.

Although the pixel should be an integer variable, its center is taken as the practical measurement in order to reduce numerical dispersion. The term w_p is the pixel width which is determined by the camera sensor size and its resolution. An important camera parameter is the field of view FOV which quantifies how much is visible through the lens. The FOV is characterized by horizontal and vertical angles related to the focal length and sensor size as

$$\text{FOV} \equiv 2\arctan\left(\frac{n_{p_x} w_p}{2f}\right) \times 2\arctan\left(\frac{n_{p_y} w_p}{2f}\right), \tag{7}$$

where n_{p_x} and n_{p_y} are, respectively, the horizontal and vertical number of pixels. In Eq. (7), the focal length f is the varying parameter that controls the FOV size. The camera pinhole model geometry is illustrated in Fig. 2.

3.2. DMC-UKF

The DMC-UKF jointly estimates the spacecraft state (position and velocity) and the unmodeled perturbing acceleration. The dynamical model compensation concept directly tracks the unknown dynamics component through estimation of the unmodeled acceleration signal [28,29]. In this work, the previous concept is embedded in an unscented Kalman filter [32]. The UKF is a suboptimal non-linear filter for Gaussian distributions. Its main feature is the unscented transform (UT) which approximates the result of applying a non-linear function to a Gaussian distribution. To this end, the UT creates a symmetrical set of samples (namely sigma points $\boldsymbol{\chi}_{\mathbf{x}}$) around the mean $\hat{\mathbf{x}}$ of the initial Gaussian distribution $\mathbf{x} \sim N_{n_x}(\hat{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{xx}})$ as

$$\boldsymbol{\chi}_{\mathbf{x}}^{[i]} = \hat{\mathbf{x}} + \text{sgn}(i) \cdot \left(\sqrt{(n_x + \lambda)\boldsymbol{\Sigma}_{\mathbf{xx}}}\right)_{|i|}, \quad i = -n_x \dots 0 \dots n_x. \tag{8}$$

Note that N_{n_x} denotes a Gaussian distribution of dimension n_x. Then, in order to approximate the distribution $\mathbf{y} = \mathbf{f}(\mathbf{x})$, the UT passes each sigma point through the function $\mathbf{f}$. The sigma points outcome reconstructs the final distribution in a Gaussian form $\mathbf{y} \sim N_{n_y}(\mathbf{y}, \boldsymbol{\Sigma}_{\mathbf{yy}})$ as

$$\hat{\mathbf{y}} = \sum_{i=-n_x}^{n_x} w_m^{[i]} \mathbf{f}(\boldsymbol{\chi}_{\mathbf{x}}^{[i]}), \qquad \boldsymbol{\Sigma}_{\mathbf{yy}} = \sum_{i=-n_x}^{n_x} w_c^{[i]} (\hat{\mathbf{y}} - \mathbf{f}(\boldsymbol{\chi}_{\mathbf{x}}^{[i]}))(\hat{\mathbf{y}} - \mathbf{f}(\boldsymbol{\chi}_{\mathbf{x}}^{[i]}))^T, \tag{9}$$

where, following [32], the weights are defined as $w_m^{[0]} = \lambda/(\lambda + n_x)$, $w_c^{[0]} = w_m^{[0]} + 1 - \alpha^2 + \beta$ and $w_m^{[i]} = w_c^{[i]} = 1/(2n_x + 2\lambda)$, $i \neq 0$. The variables $\{\alpha, \beta, \lambda\}$ are tuning parameters controlling the spread of sigma points (see Eq. (8)) and weights. Subsequently, the process propagation, state to measurements mapping and the complete DMC-UKF algorithm are presented.

3.2.1. Process propagation

Let us express the DMC-UKF state in the inertial small body centered frame N as $\mathbf{x} = [(\mathbf{r}^N)^T, (\mathbf{v}^N)^T, (\mathbf{a}^N)^T]^T$. The term $\mathbf{a}^N$ is the unmodeled acceleration acting on the spacecraft. Accordingly, the DMC-UKF process dynamics is

$$\frac{d}{dt}\begin{bmatrix}\mathbf{r}^N \\ \mathbf{v}^N \\ \mathbf{a}^N\end{bmatrix} = \begin{bmatrix}\mathbf{v}^N \\ (\mathbf{R}_N^A)^T \mathbf{a}_M(\mathbf{r}) + \mathbf{a}_{\odot}^N(\mathbf{r}) + \mathbf{a}_{\text{SRP}}^N(\mathbf{r}) + \mathbf{a}^N \\ \mathbf{0}\end{bmatrix}, \tag{10}$$

where the Sun's third-body gravity and solar radiation pressure parameters are assumed to be known. Accordingly, the unknown dynamics corresponds to the small body gravity (unmodeled components in $\mathbf{a}_M$) which is compensated by the unmodeled acceleration $\mathbf{a}^N$. Its time variation is assumed as null ($\dot{\mathbf{a}}^N = \mathbf{0}$), thus representing a zeroth-order Gauss Markov process (white noise). This implies that the measurements sampling rate should be fast enough in order to keep track of the inhomogeneous gravity signal. Other works in the literature have embedded first [28] and second-order [29] Gauss Markov processes. However, the scope of these works is different from the present one. They indirectly account for the unmodeled gravity component by fitting an acceleration signal while here the modeling error is directly tackled by recursive updates of the gravity model.

The outcome of the process is the a-priori DMC-UKF state which is obtained by propagating Eq. (10) using a simple forward Euler integration rule

$$\mathbf{x}^- = \mathbf{x}_0 + \sum_{k=0}^{N_{\text{int}}-1} \dot{\mathbf{x}}(t_0 + k\Delta t_{\text{int}})\Delta t_{\text{int}}, \tag{11}$$

where N_{int} is the number of integration points and Δt_{int} the time step.

3.2.2. State to measurement mapping

The available measurement $\mathbf{z}$ comprises the n_L visible landmarks pixels at the observation epoch

$$\mathbf{z} = [p_{x_1}, p_{y_1}, \dots, p_{x_{n_L}}, p_{y_{n_L}}]^T. \tag{12}$$

In order to map the DMC-UKF state to the measurement, the relative positions (expressed in the camera frame C) between spacecraft and visible landmark is computed as

$$\Delta\mathbf{r}_l^C = \mathbf{R}_A^C(\mathbf{r}_l - \mathbf{R}_N^A\mathbf{r}^N), \tag{13}$$

where $\mathbf{R}_N^A$ is the direction cosine matrix from inertial to the rotating small body centered frame (see Eq. (2)). The direction cosine matrix $\mathbf{R}_A^C$ represents the orientation of the camera with respect to the rotating small body frame. Because this work does not account for attitude determination, the matrix $\mathbf{R}_A^C$ is assumed to be perfectly known (provided by the attitude determination system). Finally, each landmark pixel can be mapped as

$$\begin{bmatrix} p_{x_l} \\ p_{y_l} \end{bmatrix} = \frac{f}{w_p}\begin{bmatrix} \Delta x_l^C/\Delta z_l^C \\ \Delta y_l^C/\Delta z_l^C \end{bmatrix}. \tag{14}$$

where the pixel is not discretized in order to better quantify the difference with respect to the actual discrete measurement.

3.2.3. DMC-UKF algorithm

The implementation of the DMC-UKF step follows [32] and is described in Algorithm 1. Let us consider the current DMC-UKF state Gaussian distribution $\mathbf{x}_0 \sim N_9(\hat{\mathbf{x}}_0, \boldsymbol{\Sigma}_{\mathbf{xx}_0})$. Let us also compress the Eq. (11) process integration and Eq. (13)–(14) state to measurement map as $\mathbf{f}$ and $\mathbf{g}$ respectively. When measurements are available, the DMC-UKF state is updated as follows. Step 2 computes the a-priori state estimate $\mathbf{x}^- \sim N_9(\hat{\mathbf{x}}^-, \boldsymbol{\Sigma}_{\mathbf{xx}}^-)$ by doing the UT of the initial distribution with the process. Step 3 inflates the a-priori state uncertainty with the process noise covariance $\boldsymbol{\Sigma}_{\mathbf{ff}}$ (it quantifies the mismatch between truth and process dynamics). This uncertainty is a tuning parameter of the algorithm. Step 4 computes an a-priori measurement distribution $\mathbf{z}^- \sim N_3(\hat{\mathbf{z}}^-, \boldsymbol{\Sigma}_{\mathbf{zz}}^-)$ by mapping the a-priori state into measurement space with the UT. Step 5 inflates the a-priori measurement covariance with the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{zz}}$. This matrix accounts for measurement noise. The step 6 computes the cross-correlation matrix $\mathbf{H}_{\mathbf{xz}}$ between state and measurement. The cross-correlation matrix $\mathbf{H}_{\mathbf{xz}}$ is used in step 7 to compute the Kalman gain $\mathbf{K}$. Finally, the incoming measurement $\mathbf{z}$ is used in the Kalman update linear equation (see step 8) to compute the a-posteriori DMC-UKF state distribution $\mathbf{x} \sim N_9(\hat{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{xx}})$. This procedure is sequentially repeated each time measurements are available.

Algorithm 1: DMC-UKF step

```
1 begin
2     Apply the UT (see Eq. (8)-(9)) to Eq. (11) process:
        $\mathbf{x}^- \sim N_9(\hat{\mathbf{x}}^-, \boldsymbol{\Sigma}_{\mathbf{xx}}^-) \equiv \mathbf{f}(N_9(\hat{\mathbf{x}}_0, \boldsymbol{\Sigma}_{\mathbf{xx}_0}))$;
3     Add the process uncertainty: $\boldsymbol{\Sigma}_{\mathbf{xx}}^- \leftarrow \boldsymbol{\Sigma}_{\mathbf{xx}}^- + \boldsymbol{\Sigma}_{\mathbf{ff}}$;
4     Apply the UT (see Eq. (8)-(9)) to Eq. (13)-(14) state to
        measurement mapping: $\mathbf{z}^- \sim N_3(\hat{\mathbf{z}}^-, \boldsymbol{\Sigma}_{\mathbf{zz}}^-) \equiv \mathbf{g}(N_9(\hat{\mathbf{x}}^-, \boldsymbol{\Sigma}_{\mathbf{xx}}^-))$;
5     Add the measurement uncertainty: $\boldsymbol{\Sigma}_{\mathbf{zz}}^- \leftarrow \boldsymbol{\Sigma}_{\mathbf{zz}}^- + \boldsymbol{\Sigma}_{\mathbf{zz}}$;
6     Compute the cross-correlation matrix between state and
        measurements: $\mathbf{H}_{\mathbf{xz}} = \sum_{i=-9}^{9} w_c^{[i]} \left(\boldsymbol{\chi}_{\mathbf{x}}^{[i]} - \hat{\mathbf{x}}^-\right)\left(\boldsymbol{\chi}_{\mathbf{z}}^{[i]} - \hat{\mathbf{z}}^-\right)^T$;
7     Compute the Kalman gain: $\mathbf{K} = \mathbf{H}_{\mathbf{xz}}\boldsymbol{\Sigma}_{\mathbf{zz}}^{-1}$;
8     Update the state with incoming measurements from
        Eq. (B.5):
        $\mathbf{x} \sim N_9(\hat{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{xx}}) \equiv N_9(\hat{\mathbf{x}}^- + \mathbf{K}(\mathbf{z} - \hat{\mathbf{z}}^-), \boldsymbol{\Sigma}_{\mathbf{xx}}^-(\mathbf{I} - \mathbf{H}_{\mathbf{xz}}\mathbf{K}^T))$;
9 end
```

4. Mascon gravity estimation

The mascon gravity model estimation process is described in this section. The goal is to minimize the mean squared percent error (MSE) of the fitted model with respect to the acceleration dataset. This should be achieved while taking into account mascon physical constraints. To do so, Adam gradient descent is combined with a constraints projection step.

4.1. Gravity fitting problem

This paragraph begins by defining a loss function based on the available dataset. Then, the mascon model configuration and the constraints projection step are described.

4.1.1. Dataset and loss function

The mascon model is fitted using the on-orbit position-acceleration dataset generated by the DMC-UKF. Let us stack the DMC-UKF estimates in the vectors $\mathbf{r}_{\mathbf{S}}^{\text{data}}$ and $\mathbf{a}_{\mathbf{S}}^{\text{data}}$ as

$$\mathbf{r}_{\mathbf{S}}^{\text{data}} = \begin{bmatrix} \hat{\mathbf{r}}_1 \\ \vdots \\ \hat{\mathbf{r}}_m \end{bmatrix}, \quad \mathbf{a}_{\mathbf{S}}^{\text{data}} = \begin{bmatrix} \mathbf{a}_M(\hat{\mathbf{r}}_1) + \hat{\mathbf{a}}_1 \\ \vdots \\ \mathbf{a}_M(\hat{\mathbf{r}}_m) + \hat{\mathbf{a}}_m \end{bmatrix}. \tag{15}$$

The position and unmodeled acceleration are rotated from the inertial to the rotating small body frame as $\hat{\mathbf{r}}_j = \mathbf{R}_N^A(t_j)\hat{\mathbf{r}}_j^N$ and $\hat{\mathbf{a}}_j = \mathbf{R}_N^A(t_j)\hat{\mathbf{a}}_j^N$. The acceleration dataset comprises the sum of the unmodeled one with the current mascon model prediction at the position estimate. In order to ease the notation, let us use $\mathbf{r}_{\mathbf{S}}^{\text{data}} = [\mathbf{r}_1^T, \dots, \mathbf{r}_m^T]^T$ and $\mathbf{a}_{\mathbf{S}}^{\text{data}} = [\mathbf{a}_1^T, \dots, \mathbf{a}_m^T]^T$ from now on to denote the individual components of the dataset.

The loss function L is to be defined in terms of the discrepancy between the mascon prediction and the dataset. In [33], it is highlighted that the use of percent errors (instead of absolute residuals) helps to regularize the gravity solution (as samples are accounted for relative to their magnitude). Following that logic, let us define the gravity percent error δa as

$$\delta a_j = \frac{\|\mathbf{a}_M(\mathbf{r}_j) - \mathbf{a}_j\|}{\|\mathbf{a}_j\|}, \tag{16}$$

where $\mathbf{a}_M$ is the mascon model prediction. Then, a straightforward choice of loss function is the mean squared gravity percent errors

$$L = \frac{1}{m}\sum_{j=1}^{m} \delta a_j^2. \tag{17}$$

4.1.2. Mascon model setup

The mascon distribution should fulfill two basic physical constraints. The masses shall be enclosed within the body shape (interior constraint) and their values should be positive. The first one avoids singularities in the exterior gravity field evaluation. The second ensures that the gravity field is always attractive, thus avoiding local repulsive regions near negative masses (as it is the case in the preliminary work [17]). Although these conditions may seem simple, they are not necessarily guaranteed in a numerical optimization framework. Another additional condition emerges if the total mass is to be known (as it is assumed in this work).

It is more convenient to implicitly encode constraints within the mascon model if possible (rather than explicitly tackling them in the optimizer). In particular, the masses positiveness and total mass can be directly enforced by choosing adequate decision variables. The consistency of the mascon distribution with the total mass is ensured by clearing one of the mascon masses from the optimization as

$$\mu_{M_0} = \mu - \sum_{k=1}^{n} \mu_{M_k}. \tag{18}$$

The masses positiveness is encoded by changing the decision variable from the gravity parameter μ_{M_k} to its square-root $\sqrt{\mu_{M_k}}$. It should be noted that this does not guarantee $\mu_{M_0} \geq 0$ as it is cleared from the optimization (see Eq. (18)). This issue has to be tackled within the mascon fitting algorithm along with the interior constraint.

According to the previous development, the mascon distribution variables can be stacked in the vector $\mathbf{y}_{\mathrm{S}_M}$ as

$$\mathbf{y}_{\mathrm{S}_M} = \begin{bmatrix} \sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M} \\ \mathbf{r}_{\mathrm{S}_M} \end{bmatrix}, \quad \sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M} = \begin{bmatrix} \sqrt{\mu_{M_1}} \\ \vdots \\ \sqrt{\mu_{M_n}} \end{bmatrix}, \quad \mathbf{r}_{\mathrm{S}_M} = \begin{bmatrix} \mathbf{r}_{M_1} \\ \vdots \\ \mathbf{r}_{M_n} \end{bmatrix}, \tag{19}$$

where $\sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M}$ and $\mathbf{r}_{\mathrm{S}_M}$ stack the n square-root masses and positions of the distribution. The position of the 0th mass $\mathbf{r}_{M_0}$ is also cleared from the optimization, thus being a fixed parameter. A convenient placement may be $\mathbf{r}_{M_0} = \mathbf{0}$ (which is the one used). Using the previous stacked variables, a compact prediction of the dataset is obtained as

$$\mathbf{A}_M (\sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M})^2 + \mathbf{a}_{\mathrm{S}_{M_0}} = \mathbf{a}_{\mathrm{S}_M}, \tag{20}$$

where the squared vector denotes the element-wise product $(\sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M})^2 = \sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M} \odot \sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M}$. The matrix $\mathbf{A}_M$ exclusively depends on the relative distance between the evaluation points and each mass. The vector $\mathbf{a}_{\mathrm{S}_{M_0}}$ is an offset arising from clearing the 0th mass and assuming its location fixed. The terms $\mathbf{A}_M$ and $\mathbf{a}_{\mathrm{S}_{M_0}}$ are

$$\mathbf{A}_M = \begin{bmatrix} -\dfrac{\Delta\mathbf{r}_{11}}{\Delta r_{11}^3} + \dfrac{\Delta\mathbf{r}_{10}}{\Delta r_{10}^3} & \cdots & -\dfrac{\Delta\mathbf{r}_{1n}}{\Delta r_{1n}^3} + \dfrac{\Delta\mathbf{r}_{10}}{\Delta r_{10}^3} \\ \vdots & \ddots & \vdots \\ -\dfrac{\Delta\mathbf{r}_{m1}}{\Delta r_{m1}^3} + \dfrac{\Delta\mathbf{r}_{m0}}{\Delta r_{m0}^3} & \cdots & -\dfrac{\Delta\mathbf{r}_{mn}}{\Delta r_{mn}^3} + \dfrac{\Delta\mathbf{r}_{m0}}{\Delta r_{m0}^3} \end{bmatrix}, \quad \mathbf{a}_{\mathrm{S}_{M_0}} = -\mu \begin{bmatrix} \dfrac{\Delta\mathbf{r}_{10}}{\Delta r_{10}^3} \\ \vdots \\ \dfrac{\Delta\mathbf{r}_{m0}}{\Delta r_{m0}^3} \end{bmatrix},$$

where $\Delta\mathbf{r}_{jk} = \mathbf{r}_j - \mathbf{r}_{M_k}$ is the relative position between the jth data point and kth mass. Note that the vector $\mathbf{a}_{\mathrm{S}_{M_0}}$ does not depend on any element of the mascon distribution decision variable $\mathbf{y}_{\mathrm{S}_M}$.

4.1.3. Mascon constraints projection

The remaining constraints are the masses interiority within the body shape and the positiveness of μ_{M_0}. In order to handle these constraints, if the loss is convex, it is possible to solve for the unconstrained problem and project the solution to feasible space. The idea is that the closest feasible point to the unconstrained minimum is the optimal solution to the constrained problem. This theory can be formally expressed as

$$\begin{aligned} \mathbf{y}' &= \underset{\mathbf{y}}{\operatorname{argmin}}\, L(\mathbf{y}), \\ \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmin}}\, L(\mathbf{y}) &= \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmin}}\, \|\mathbf{y} - \mathbf{y}'\|, \end{aligned} \tag{21}$$

where $\mathbf{y}'$ denotes the unconstrained minimum and $\mathcal{Y}$ is the feasible domain. When related to the problem under consideration, it is very likely that Eq. (17) loss is not convex in $\mathbf{y}_{\mathrm{S}_M}$ space. However, if small steps are given towards minimum, the projection of Eq. (21) can be applied after each step being valid since the local minimum neighborhood is possibly convex.

Depending on the problem, Eq. (21) projection can be as complex as directly solving the constrained optimization. Nonetheless, it is very useful for the mascon model under consideration. This is due to the fact that the remaining constraints are highly decoupled in terms of decision variables. Let us consider the positiveness of the 0th point mass. This mass becomes negative if the sum of the remaining n masses exceeds μ. It is more intuitive to visualize the situation in space in $\sqrt{\mu_{M_k}}$ because the constraint represents a hypersphere of radius $\sqrt{\mu}$. Then, each time the mass distribution abandons that hypersphere, the projection sends the distribution back to the closest point lying on its surface. This is expressed as

$$\sqrt{\boldsymbol{\mu}}_{\mathrm{S}_M} = \begin{cases} \sqrt{\boldsymbol{\mu}}'_{\mathrm{S}_M} & \text{if} \quad \mu_{M_0} \geq 0, \\ (\sqrt{\mu}/\|\sqrt{\boldsymbol{\mu}}'_{\mathrm{S}_M}\|)\sqrt{\boldsymbol{\mu}}'_{\mathrm{S}_M} & \text{if} \quad \mu_{M_0} < 0, \end{cases} \tag{22}$$

which is equivalent to reduce all masses in the same proportion until $\mu_{M_0} = 0$.

The interior constraint is more difficult to tackle given the complex shape of a small body. Both an interior condition and a function computing the closest point on the surface are needed. These can be derived from a small body shape polyhedron model. From [18], the polyhedron normalized Laplacian $\nabla^2 U_{\text{poly}} = -\sum_{f\in\text{faces}} \omega_f$ expression determines if a point is interior to the shape. If the point is interior the normalized Laplacian equals -4π while it vanishes for an exterior point. The term ω_f is the solid angle of a polyhedron face as viewed from the evaluation point. When a violation occurs, the closest point on the polyhedron surface can be computed in a discretized form. Since the polyhedron shape is characterized by discrete surface features based on vertexes and faces, the distance of the exterior point with respect to them (or a derived variable) can be computed. Then, the closest point on the surface can be extracted. In this case, the projection condition is

$$\mathbf{r}_{M_k} = \begin{cases} \mathbf{r}'_{M_k} & \text{if} \quad \displaystyle\sum_{f\in\text{faces}} \omega_f(\mathbf{r}'_{M_k}) = 4\pi, \\ \underset{\mathbf{r}_f}{\operatorname{argmin}}\, \|\mathbf{r}_f - \mathbf{r}'_{M_k}\| & \text{if} \quad \displaystyle\sum_{f\in\text{faces}} \omega_f(\mathbf{r}'_{M_k}) = 0, \end{cases} \tag{23}$$

where $\mathbf{r}_f$ refers to the center of each polyhedron face (such to only loop over the faces). Since evaluating all the polyhedron faces is computationally expensive, a low resolution polyhedron shape model may be used.

4.2. Mascon optimization

A first-order technique to find the mascon distribution that minimizes Eq. (17) is gradient descent. It also works well with the constraints projection of Eqs. (22)–(23) if these are checked after each update. Gradient descent requires an initial mascon distribution that is detailed below. It is also beneficial to adapt the training process, thus Adam gradient descent is tailored for this application.

4.2.1. Initial mascon distribution

In order to start the training process, gradient descent needs an initial mascon distribution. In the authors previous work [17], where only the mass value is fitted, the body shape is divided into eight octants. Then, an equal number of masses $(n+1)/8$ is randomly placed within each octant. This initial distribution is also used in this work (see Fig. 3). Its main advantage is that the points tend to distribute well within the shape. Regarding the masses values, it is decided to start by concentrating the total mass at the origin (yellow point in Fig. 3). Accordingly, the remaining n masses are null. This may seem as a non-sensible choice, compared to assign $\mu/(n+1)$, but it provides consistency for trainings with different n. Additionally, it demonstrates how the gravity determination evolves from a basic Keplerian model.

4.2.2. Adam gradient descent

The key idea behind gradient descent is to reach the local minimum by taking repeated steps opposed to the loss gradient. Still, how this basic concept is implemented has a huge impact in the solution convergence. The recent need for efficient neural network training (which is an optimization process) has led to modified versions of the classic gradient descent algorithm. One approach that has spread out within the machine-learning community is the Adam optimizer [34]. The Adam optimizer adapts the per-parameter learning rate while also adding a momentum alike term. The per-parameter adaptation is very

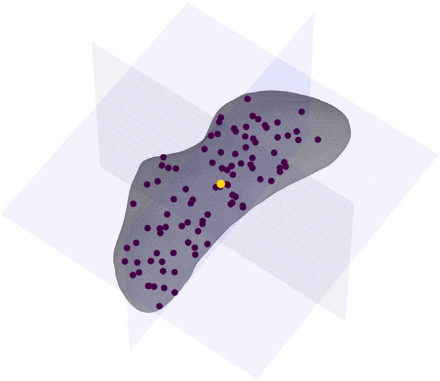

Fig. 3. Initial mascon distribution for $n = 100$.

convenient in this application because μ_M and $\mathbf{r}_M$ have a different relative impact on the acceleration prediction.

A vital step to use gradient descent is the loss function gradient ∇L expression with respect to the decision variables. The derivation of the term ∇L, for Eq. (17) loss, is detailed in Appendix A. Then, the Adam training loop is detailed in Algorithm 2 (see [34] for specific details). The Adam optimizer estimates the loss gradient first $\mathbf{m}$ and second $\mathbf{v}$ moments in step 6–7. The update rule in step 8 effectively adapts the learning rate per each parameter based on the gradient moments. Note that step 8 is an element-wise division. The adaptation allows to explore parameters that may not receive updates in classic gradient descent. The momentum component appears by the gradient moving average in the update. The mascon constraints projection Eqs. (22)–(23) is checked after each update. If the constraints are violated, the projection step sends the decision variables to feasible space. Adam gradient descent uses the hyperparameters $\{\eta, \beta_1, \beta_2, \epsilon\}$. The variable η is the classic learning rate. The terms $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of the gradient moments. Lastly, ϵ is a small number that ensures numerical stability in the update rule (it precludes divisions by zero).

Algorithm 2: Adam optimizer with constraints projection

```
1  begin
2      $\mathbf{y}_M^{[0]} \leftarrow \mathbf{y}_{M_0}$ (Initialize the decision variable);
3      $\mathbf{m}^{[0]} \leftarrow \mathbf{0}$, $\mathbf{v}^{[0]} \leftarrow \mathbf{0}$ (Initialize biased first and second-order
         gradient moments);
4      for $i \leftarrow 1$ to $i_{\max}$ do
5          $\nabla L^{[i-1]} \leftarrow \nabla L(\mathbf{y}_M^{[i-1]})$ (Compute loss gradient);
6          $\mathbf{m}^{[i]} \leftarrow \beta_1 \mathbf{m}^{[i-1]} + (1-\beta_1)\nabla L^{[i-1]}$,
             $\mathbf{v}^{[i]} \leftarrow \beta_2 \mathbf{v}^{[i-1]} + (1-\beta_2)(\nabla L^{[i-1]})^2$ (Update biased
             moments);
7          $\hat{\mathbf{m}}^{[i]} \leftarrow \mathbf{m}^{[i]}/(1-\beta_1^i)$, $\hat{\mathbf{v}}^{[i]} \leftarrow \mathbf{v}^{[i]}/(1-\beta_2^i)$ (Correct bias);
8          $\mathbf{y}_M^{[i]} \leftarrow \mathbf{y}_M^{[i-1]} - \eta\hat{\mathbf{m}}^{[i]}/(\sqrt{\hat{\mathbf{v}}^{[i]}} + \epsilon)$ (Update decision variable);
9          Apply Eq. (22) (0th mass positiveness);
10         for $k \leftarrow 1$ to $n$ (Loop through masses) do
11             Apply Eq. (23) (Interior constraint);
12         end
13     end
14 end
```

It is also convenient (for the sake of numerical stability) to internally normalize some training variables. In particular, the masses are divided by $\mu/(n+1)$. To normalize the position, a tenth of the body shape elongation on each direction is used. Note that each position coordinate is scaled differently (which helps to treat elongated small bodies). Finally, the gravity acceleration for each jth sample is normalized by its own norm $\|\mathbf{a}_j\|$. Then, its internal mascon prediction and data are $\mathbf{a}_M(\mathbf{r}_j)/\|\mathbf{a}_j\|$ and $\mathbf{a}_j/\|\mathbf{a}_j\|$ respectively. This easily allows to obtain the gravity percent error of Eq. (16) by computing the norm of the discrepancy between the previous vectors.

4.3. Simultaneous navigation and gravity estimation scheme

The simultaneous navigation and mascon gravity estimation scheme links the previous paragraphs with Section 3 DMC-UKF. Recall that the first step is to generate a position-unmodeled acceleration dataset using Algorithm 1 DMC-UKF. Then, the dataset is used to fit a mascon distribution using Adam Algorithm 2. Still, how to manage this strategy over time is to be chosen. In this work, it is chosen to process the data in 1-orbit batches. This means that the DMC-UKF fills the dataset during one orbit. At the end of this orbit, the mascon distribution is fitted with the previous data. Subsequently, the trained mascon distribution is uploaded to the DMC-UKF and the data batch is emptied. This process is repeated until the final orbit as shown in Fig. 4.

The strategy is implemented using the Basilisk[1] (BSK) software. BSK is an open source astrodyn cacta amics simulation tool [35] composed of Python modules coded in C/C++. This offers the convenience of managing the simulation with Python scripts while benefiting of the background C/C++ execution speed. The BSK implementation of the simultaneous navigation and mascon gravity estimation is illustrated in Fig. 5. It consists of a ground truth simulator and a flight software process. The ground truth simulator uses a fixed time step 4th order Runge–Kutta method to integrate Eq. (1) spacecraft dynamics. The flight software process encompasses the landmarks pixel generator (pinholeCamera), the DMC-UKF and the mascon optimizer. The pixel generator and DMC-UKF are processed at the same frequency while the mascon optimizer is executed after each orbit.

5. Results

This section tests the simultaneous navigation and mascon gravity estimation scheme through numerical simulations. The results are divided into three subsections. In the first place, only the mascon optimizer with true data. This validates the fitting of the mascon distribution. Subsequently, the simultaneous navigation and gravity estimation results are presented. Finally, a propagation analysis is done with the results of previous subsections.

Due to the extensive data available from NEAR mission, the asteroid 433 Eros is chosen as the target small body. Eros has a standard gravity parameter $\mu = 4.4627547 \cdot 10^5$ m^3/s^2 and a rotational period $T_A = 5.27$ h. Its ground truth gravity field is modeled using a polyhedron shape with 7790 triangular faces.[2] Eros heliocentric orbital parameters are $\{a = 1.4583\ \text{AU}, e = 0.2227, i = 10.829°, \Omega = 304.4°, \omega = 178.9°, \nu_0 = 246.9°\}$ and its orientation is defined by $\{\text{RA} = 11.369°, \text{dec} = 17.227°, \text{LST}_0 = 0°\}$. The orbiting spacecraft is physically characterized by a mass $m = 750$ kg, a reflection coefficient $C_R = 1.2$ and a solar radiation pressure exposed area $S = 1.1$ m^2.

5.1. On-orbit data impact on gravity estimation

This paragraph analyzes the mascon gravity estimation algorithm of Section 4. The differences between a dense and an on-orbit dataset are emphasized. Using the previous data, the global gravity accuracy of a complete mascon training and a static one (just fit the masses) are stated.

[1] http://hanspeterschaub.info/basilisk/

[2] https://sbnarchive.psi.edu/pds3/near/NEAR_A_5_COLLECTED_MODELS_V1_0/data/msi/

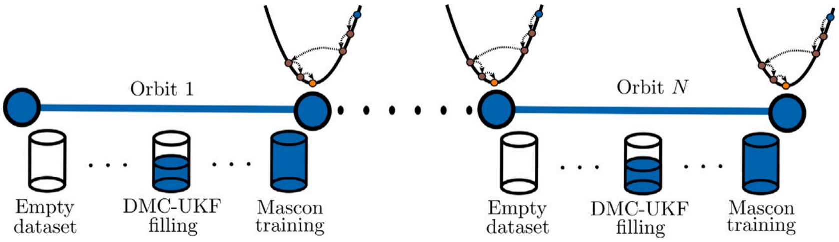

Fig. 4. Illustration of the gravity estimation process.

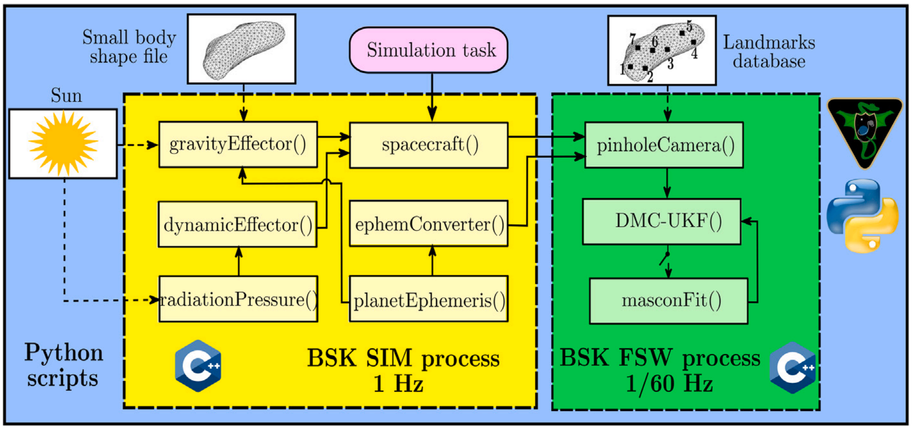

Fig. 5. Diagram of simultaneous navigation and gravity estimation in the Basilisk simulation framework.

Recall that the Adam optimizer (see Algorithm 2) requires choosing four hyperparameters and the number of iterations. The hyperparameters are tuned ad hoc as $\{\eta = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 10^{-6}\}$ while 1000 iterations are used. These values are used throughout the rest of the manuscript.

5.1.1. Training datasets

The dataset plays a fundamental role in the mascon gravity accuracy. Just using on-orbit data is challenging because the relevant low altitude features are missed. This may distort the maximum potential accuracy of the mascon gravity model. In order to obtain a bigger picture, a dense dataset homogeneously covering the small body is also considered. Furthermore, the accuracy gap between the dense and on-orbit dataset can be shrunk by the inclusion of low altitude data. This could be associated to ejecta (or gravity trackers [36]) measurements though these are not simulated realistically.

The on-orbit dataset is based on a spacecraft with initial orbital elements as $\{a_0 = 34\text{ km}, e_0 = 0.001, i_0 = 45°, \Omega_0 = 48.2°, \omega_0 = 347.8°, \nu_0 = 85.3°\}$. This dataset is composed of position-acceleration components sampled each 60 s along 10 orbital periods ($\sim$ 982 data points per orbit). Since the idea is to mimic conditions to be found in the subsequent paragraph 5.2, two considerations are made. The first one is that the gravity estimation process follows the sequential approach of Fig. 4. This means that each orbit batch is fitted sequentially by using the solution of the previous orbit as the starting point for the next one. The second consideration is the removal of points where no landmarks are visible due to lighting conditions ($\approx 20\%$ reduction of the samples). Alternatively, the dense dataset is generated by randomly placing data (same number of samples as the on-orbit) between the surface and up to 30 km radius. The training with the dense dataset also follows the sequential batch processing.

The dense and on-orbit datasets are depicted in Fig. 6. The aforementioned low altitude samples that may complement the on-orbit dataset are also shown. The low altitude data is generated randomly but its spatial diversity is limited. In particular, the low altitude samples are related to the spacecraft first orbit arc. A sample is randomly drawn for an orbit point which defines its sample latitude and longitude. Then, a random radius ranging from the surface up to 18 km is assigned. When considering this additional data, 50 samples ($\approx$6% of an orbit batch) are assumed available since the beginning of the estimation process.

5.1.2. Global gravity accuracy

It is of interest to evaluate the fitted mascon distributions, using Fig. 6 datasets, gravity accuracy. For this purpose, 55 166 evaluation points are placed between the small body surface up to a radius of 50 km. These evaluation points are uniformly distributed within altitude bands of 1.2 km (containing each $\sim$ 1400 points). Then, the mean gravity percent errors (see Eq. (16)) for these altitude bands are used as a global gravity accuracy metric. An example of this metric is shown in Fig. 7. In order to create that figure, a mascon distribution with $n = 100$ masses has been trained under different datasets (dense, on-orbit and on-orbit with low altitude samples). The low altitude samples are referred as ejecta in the legends. Furthermore, static mascon distributions (just fitting $\sqrt{\mu_M}$) are also trained. The error with a simple Keplerian gravity is included as a reference. Several conclusions can be derived from Fig. 7. The mascon training under the dense dataset is highly accurate with a maximum mean gravity error of $\approx 1\%$ very close to the surface. Instead, the use of the on-orbit dataset increase the altitude bands error in an order of magnitude with respect to the dense dataset. The inclusion of the low altitude samples has a positive effect in driving down the on-orbit errors. Lastly, the trainings using static mascon distributions show to be less accurate than the complete ones. For the static distributions, the inclusion of low altitude samples increase accuracy in that domain but degrades the high altitudes. When these low altitude vanishes (on-orbit), the complete and static mascon distribution achieve similar accuracy in the low altitude domain. This also highlights that the complete mascon training requires more information to acquire its maximum potential. Each line in Fig. 7 compresses considerable information. To illustrate what these lines are representing, Fig. 8 plots the gravity error with respect to each evaluation point altitude. For the sake of clarity, this

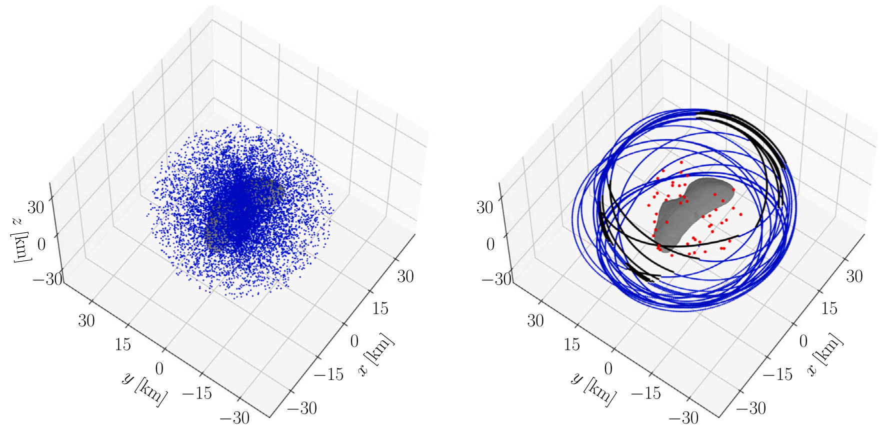

Fig. 6. Dense (*left*) and on-orbit (*right*) datasets. Black ≡ points removed; red ≡ low altitude samples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

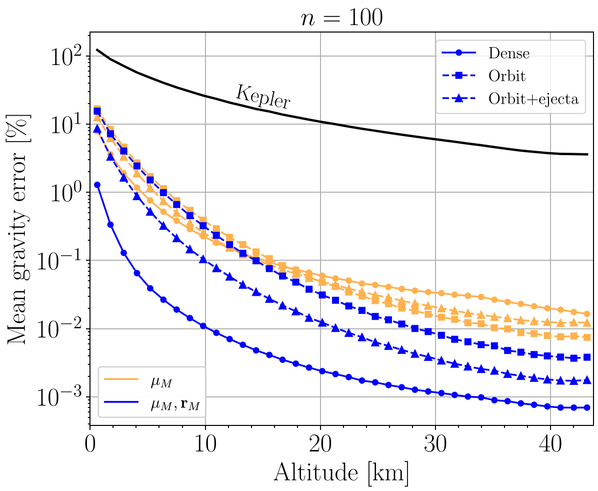

Fig. 7. Mean gravity error with respect to altitude for dense and on-orbit datasets.

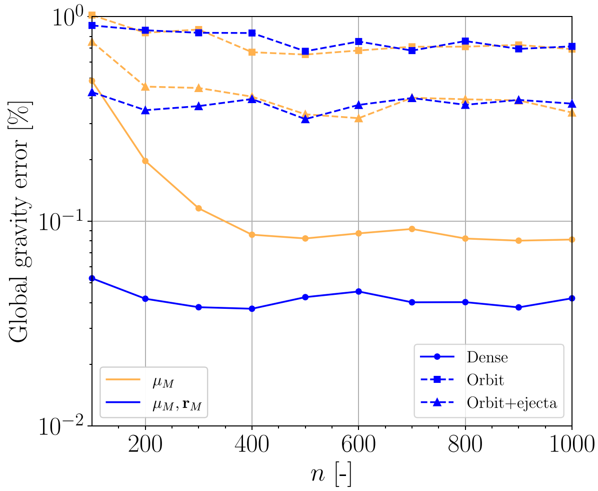

Fig. 9. Global gravity error with respect to number of masses for different mascon setups and training datasets.

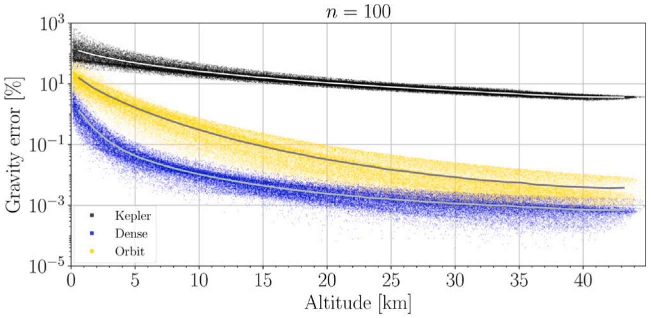

Fig. 8. Gravity error with respect to altitude for dense and on-orbit datasets.

is only done for the complete mascon distribution trainings with the dense and on-orbit datasets (and the Kepler model).

Another question that emerges is the effect of the number of masses n on the gravity accuracy. To answer this query, n is varied between 100 and 1000 under the previous training conditions. Since a new variable (n) is added to the analysis, the gravity error is compressed to the global one (mean gravity for the entire evaluation set). The result is shown in Fig. 9. It can be observed that the training varying both the mascon distribution position and mass is relatively indifferent to n. On the contrary, n seems to be relevant (up to $n = 400$) when only the mascon masses are trained. This occurs for both the dense dataset and the on-orbit one with low altitude samples. Additionally, training the mascon distribution positions does not seem to grant a distinctive advantage for the pure on-orbit dataset.

5.2. Simultaneous navigation and gravity estimation

Next the results with the simultaneous navigation and mascon gravity estimation scheme (see Fig. 5) are presented. The simulated orbit corresponds to the one shown in Fig. 6 (*right*). In this case, several parameters for the DMC-UKF are to be provided.

The DMC-UKF tuning parameters follow the ones used in [32] which are $\{\alpha = 0, \beta = 2, \lambda = 10^{-3}\}$. The DMC-UKF is initialized with true initial position and velocity while the unmodeled acceleration is assumed null. Moreover, the unmodeled acceleration is reinitialized to a null value after a gravity fit is completed or a measurement outage arises. The initial state covariance and process noise are

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{xx}}(t_0) &= \begin{bmatrix} 10^2\mathbf{I}\ (\text{m})^2 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & 10^2\mathbf{I}\ (\text{mm/s})^2 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & 1^2\mathbf{I}\ (\mu\text{m/s}^2)^2 \end{bmatrix}, \\ \boldsymbol{\Sigma}_{\mathbf{ff}} &= \begin{bmatrix} 0.1^2\mathbf{I}\ (\text{m})^2 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & 1^2\mathbf{I}\ (\text{mm/s})^2 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & 2^2\mathbf{I}\ (\mu\text{m/s}^2)^2 \end{bmatrix}.\end{aligned} \tag{24}$$

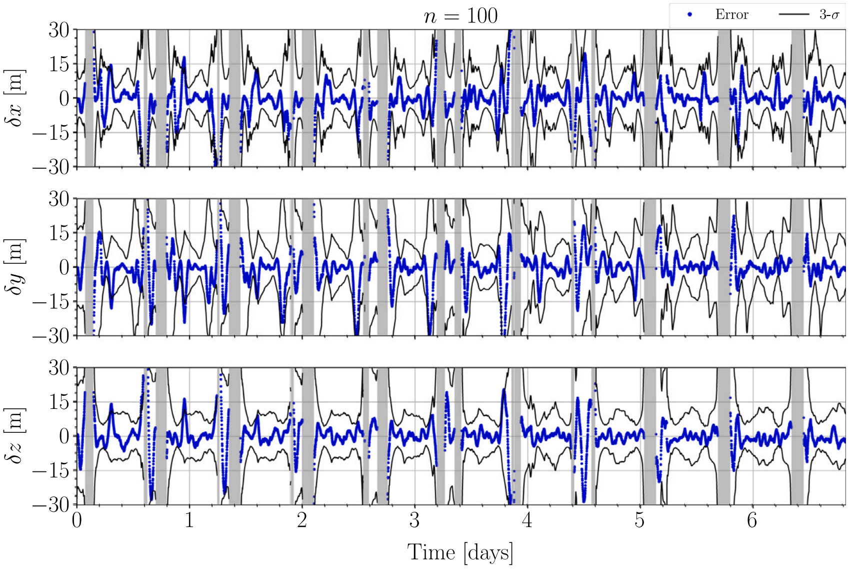

Fig. 10. DMC-UKF position error and uncertainty estimation for scenario B2. Gray area ≡ measurement outage.

In Kalman filters, adequate tuning of the process noise is key for accuracy. In this case, $\boldsymbol{\Sigma}_{\mathbf{ff}}$ is determined ad hoc via experimentation. It is noticed that the process uncertainty associated to $\mathbf{a}$ is the parameter that requires a finer tuning.

Regarding measurements, navigation data of 100 surveyed landmarks is assumed available. These are placed at the face centers of Eros polyhedron model. The landmarks are tracked by the on-board camera with 4:3 aspect ratio, a sensor size of 17.3×13 mm and a resolution of 2048×1536 px. Consequently, the pixel width is $w_p = 8.447$ μm. The camera focal length is $f = 25$ mm which is derived by a static position determination analysis in Appendix B. Landmark measurements are sampled each 60 s. The DMC-UKF uncertainty on each measurement pixel is tuned to be the unity, thus

$$\boldsymbol{\Sigma}_{\mathbf{zz}_j} = \begin{bmatrix} 1^2\ (\mathrm{px})^2 & 0 \\ 0 & 1^2\ (\mathrm{px})^2 \end{bmatrix}, \quad \boldsymbol{\Sigma}_{\mathbf{zz}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{zz}_1} & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 2} & \cdots & \boldsymbol{\Sigma}_{\mathbf{zz}_{n_L}} \end{bmatrix}, \tag{25}$$

where the measurement noise matrix $\boldsymbol{\Sigma}_{\mathbf{zz}}$ dimension changes according to the number of visible landmarks n_L.

5.2.1. DMC-UKF estimation

Let us assess the accuracy of the DMC-UKF estimates which compose the subsequent mascon training dataset. The results of this paragraph concerns a complete mascon distribution fitting with $n = 100$. The DMC-UKF accuracy greatly depends on the simulation conditions. To test this, four different scenarios (namely A1, A2, B1 and B2) are considered. Scenarios A do not account for the lighting constraint (thus no measurements outages arise) while B ones do. The numeric label “1” refers to a simulation without landmarks uncertainty. Alternatively, the “2” means that the filter landmark database is perturbed with a 5 m standard deviation Gaussian error. These conditions are summarized in Table 1. Additionally, the position and unmodeled acceleration root mean square errors (RMSE) are reported. This gives an idea on the severity of the different scenario conditions. As expected, scenario A1 is the most favorable one as the filter has continuous measurements and knows the exact landmark locations. The most severe conditions arise when measurement outages happen (scenarios B). This is mainly because the unmodeled acceleration has to converge again after each of these gaps.

In order to illustrate the DMC-UKF signal, scenario B2 position error and inhomogeneous gravity estimate are shown in Figs. 10–11. The inhomogeneous gravity component is most representative as it avoids the dominant Keplerian term. In Fig. 10, it can be observed that the loss and recapture (gray areas) of landmark measurements produce sharp errors. The position uncertainty seems to be bounded and the filter adapts itself to changes on it. The DMC-UKF is capable to keep track of the inhomogeneous gravity signal (which is the primary goal) as shown in Fig. 11. The plotted estimate is the addition of the unmodeled acceleration estimate $\hat{\mathbf{a}}$ plus the inhomogeneous gravity component of the current filter model, $\mathbf{a}_M(\hat{\mathbf{r}}) + \mu\hat{\mathbf{r}}/\hat{r}^3$. It can be observed that the DMC-UKF signal matches more closely the truth one after a couple of mascon trainings. Furthermore, gravity spikes arise (during the second half of the simulation) due to low altitude arcs caused by the accumulated effect of inhomogeneous gravity and solar perturbations in the spacecraft trajectory. These spikes are shown to be captured to a certain extent.

Table 1
Summary of scenario conditions and position-unmodeled acceleration RMSE.

Scen.	Lighting	Landmark error	RMSE(r) [m]	RMSE(a) [%]
A1	Full	No	2.164	1.202
A2	Full	Yes	3.765	1.367
B1	Partial	No	5.320	1.769
B2	Partial	Yes	8.448	2.042

5.2.2. On-orbit global gravity accuracy

The on-orbit gravity accuracy for Table 1 scenarios is now assessed. The evaluation set is the same as in paragraph 5.1.2. For $n = 100$, the mean gravity error with respect to altitude is shown in Fig. 12 along with the Kepler model and the orbit fit with a true dataset (for reference purposes). The on-orbit training seems effective as the initial Keplerian model is improved. Nonetheless, significant degradation arises with respect to the true on-orbit dataset. The errors tends to correlate with the severity of the scenarios in Table 1. In Fig. 12, the complete or the static mascon distribution does not seem to significantly differ from each other. It can be noticed that the complete training seems to perform better at low altitudes while the static one fits slightly better the high altitude domain. The analysis is complemented with Fig. 13 which shows the global error with respect to the number of masses. Except from the static mascon training under scenario A1, no clear correlations are derived in terms of global error with respect to n. It also shows that in terms of global error the complete mascon trainings seems to be better, overall, than the static ones. However, awareness should

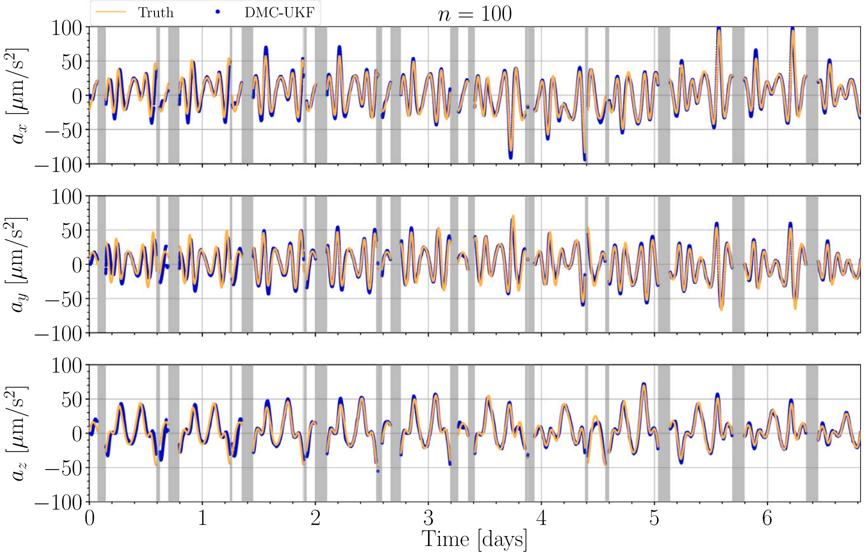

Fig. 11. DMC-UKF inhomogeneous gravity estimation for scenario B2. Gray area ≡ measurement outage.

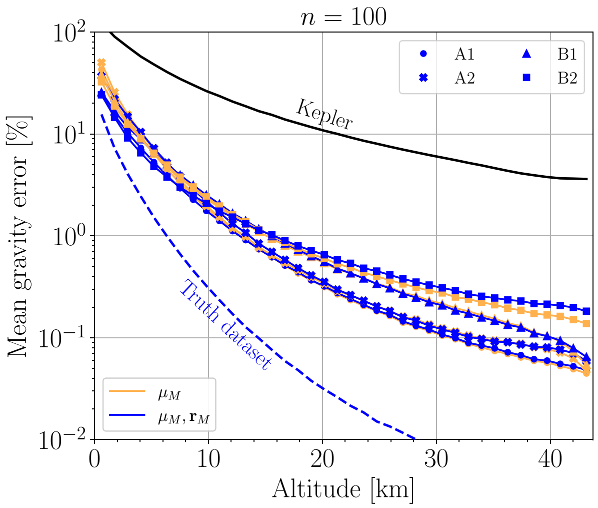

Fig. 12. Mean gravity error with respect to altitude with simultaneous navigation and gravity estimation.

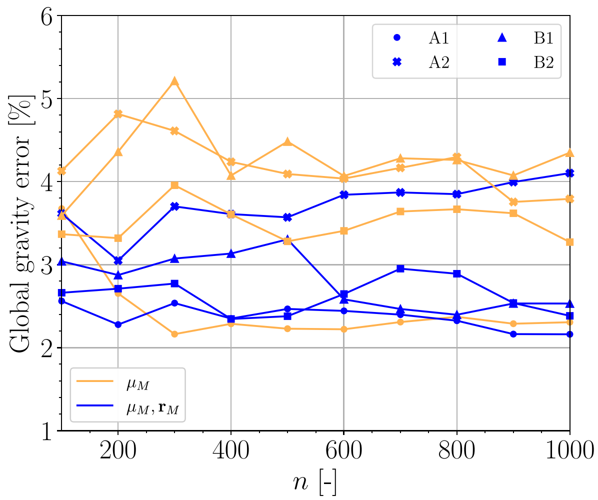

Fig. 13. Mean gravity error with respect to altitude for different mascon setups and training datasets.

be raised that the global gravity error favors low errors in the low altitude domain (since low altitude errors are higher in absolute terms). This is why scenario A2 (see low altitude domain error in Fig. 12), which does not have the most severe conditions, looks disfavorable in Fig. 13.

It may be also worth to look the trained mascon distributions. Fig. 14 shows the resulting distributions of trainings for $n = 100, 1000$ under scenario B2. The average gravity parameters of these distributions are 4418 and 445.8 m^3/s^2 respectively. It is found that some masses tend to negligible values (<1 m^3/s^2), especially near the most negative x axis, constituting a 14.8% ($n = 100$) and a 6.89% ($n = 1000$) of the total number. Along the training, the masses positions vary in average 2.307 and 2.741 km with respect to the initial distribution.

5.2.3. Inclusion of low altitude samples

Following paragraph 5.1.2, the positive impact of including low altitude samples is to be analyzed. The same low altitude data of Fig. 6 (*right*) are used. In a real mission operation, these data samples are to be acquired while on the fly (which is out of the scope of this work). In order to account for that, a 5% Gaussian error is added on each acceleration component. For scenario A1, 2D gravity error maps with and without low altitude samples (namely ejecta) are shown in Fig. 15. It can be easily observed that these additional samples help to reduce significantly the error in the low altitude domain. The fact that the ejecta data is generated randomly but concentrated on specific spatial regions can be deduced from the yz map. Table 2 compares the global gravity error with and without ejecta for the on-orbit scenarios. For all the simulated scenarios (both with the complete and static trainings), the global gravity error is reduced significantly (≈50%).

5.2.4. Computational efficiency

Let us conclude this subsection by evaluating the simultaneous navigation and gravity estimation computational efficiency. This paper simulations are executed in a M1 Max (2.06–3.22 GHz) processor using a single thread. Recall that the algorithms are coded in C++ under the Basilisk simulation framework (see Fig. 5). The execution times of the DMC-UKF step and each mascon optimization are reported in

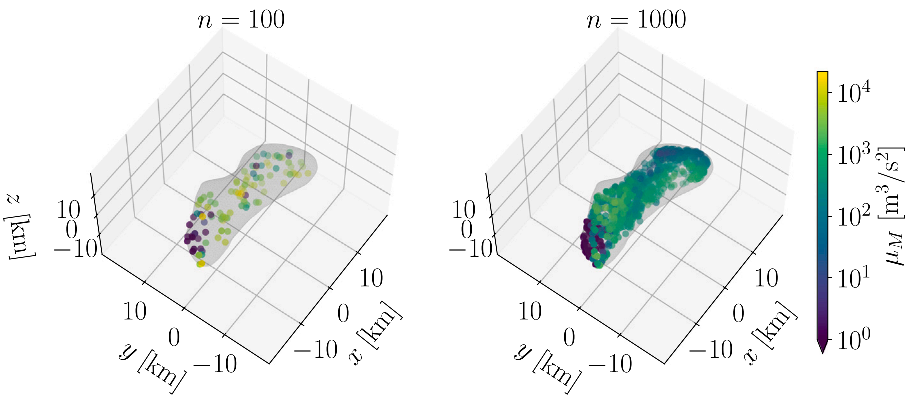

Fig. 14. Trained mascon distribution for scenario B2.

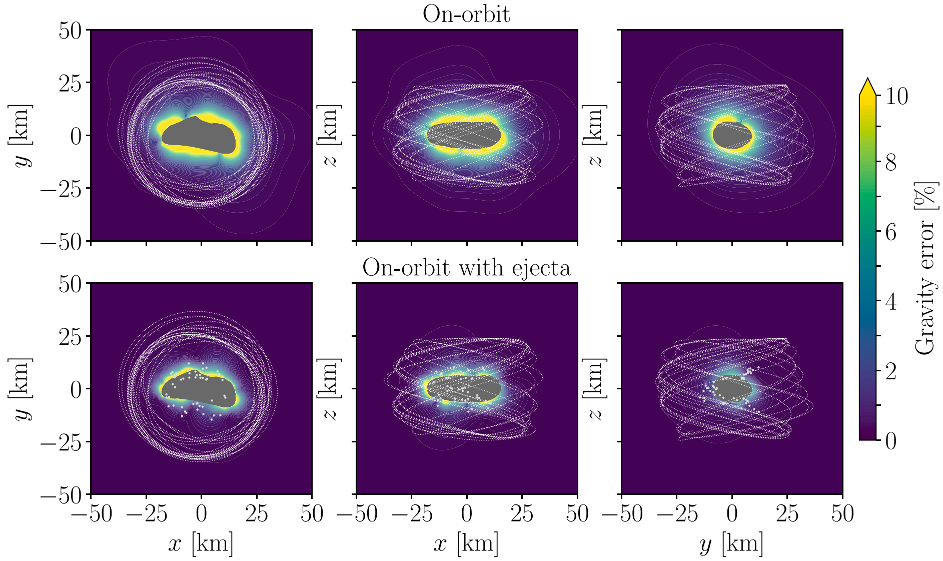

Fig. 15. Gravity error maps with and without low altitude samples for scenario A1 and $n = 100$.

Table 2
Global gravity percent error with and without low altitude samples for $n = 100$.

Scen.	Orbit		Orbit with ejecta	
	μ_M	$\mu_M, \mathbf{r}_M$	μ_M	$\mu_M, \mathbf{r}_M$
A1	3.674%	2.561%	1.424%	0.956%
A2	4.129%	3.614%	1.709%	1.572%
B1	3.591%	3.040%	1.542%	1.461%
B2	3.367%	2.661%	1.681%	1.624%

Table 3
Computational times of simultaneous navigation and gravity estimation for several models. M ≡ masses fit; MP ≡ masses and positions fit.

n [–]	DMC-UKF step		Mascon training	
	Mean [ms]	Max. [ms]	Mean [s]	Max. [s]
100 (M)	0.289	0.779	1.099	1.146
500 (M)	1.194	2.742	5.890	6.091
1000 (M)	2.327	3.648	12.53	12.69
100 (MP)	0.283	0.542	10.91	10.94
500 (MP)	1.192	2.461	56.56	56.96
1000 (MP)	2.327	4.008	112.5	112.6

Table 3. The number of masses ranges from 100 to 1000. The DMC-UKF execution is in the order of milliseconds or less. The mascon training ranges between 1 to 10 s for the static setup and 10–120 for the complete setup. Since the time span between measurements is 60 s and one orbit lasts ≈ 15 h, the computation times show promise in terms of on-board autonomy.

5.3. Propagation analysis

The intended purpose of the fitted mascon models is to serve as propagators for subsequent mission phases. Consequently, a propagation analysis is done in this subsection. The orbits to propagate vary their initial semi-major axis a_0 and inclination i_0 being $\{e_0 = 0.001, \Omega_0 = 48.2°, \omega_0 = 347.8°, \nu_0 = 85.3°\}$ fixed. The analyzed set of inclinations is $i_0 = \{0°, 45°, 90°, 180°\}$. The semi-major axis is varied between 28 to 46 km which ensures the orbit stability for the propagation duration of 12 h. The propagation accuracy is measured by the position RMSE evaluated each 60 s.

The models under consideration are the ones that fit the complete mascon distribution. Amongst them, the analyzed models correspond to A1 and B2 scenarios (simultaneous navigation and gravity estimation), on-orbit and dense datasets (true data). The fits with low altitude samples are also considered. For these scenarios, the final position RMSE is shown in Fig. 16. For the most severe scenario B2, the expected error ranges between 0.1–1 km in the majority of cases. For the less stringent

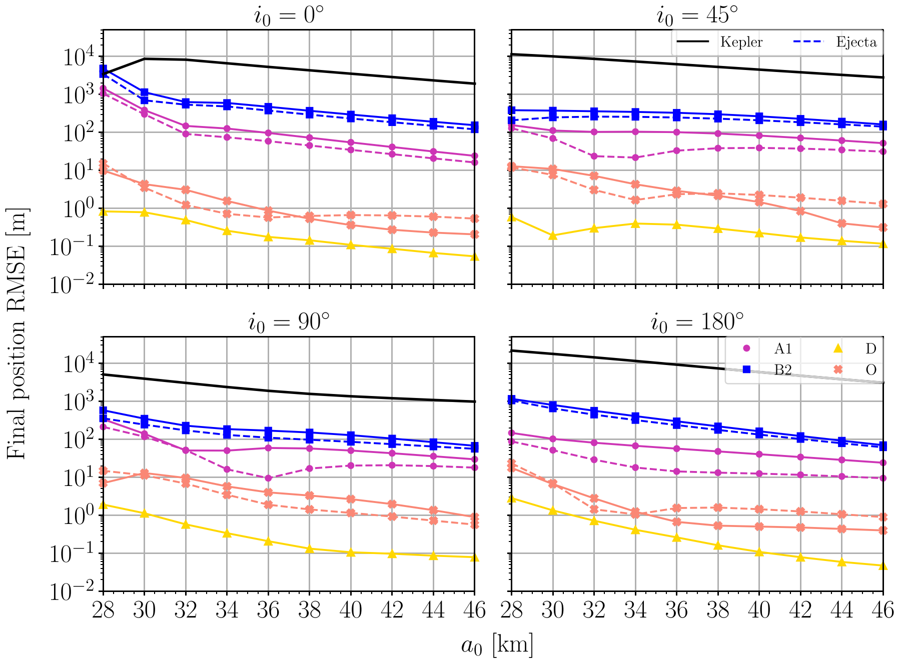

Fig. 16. Final position RMSE after 12 h for different initial orbits with $n = 1000$ mascon models.

scenario A1, with the low altitude samples, the error can be typically drive down to 10–100 m. This may preclude the use of the fitted on-orbit mascon models in long-term propagations. However, by training a truth on-orbit dataset, the errors are reduced to 1–10 m. This demonstrates that the on-orbit DMC-UKF dataset can be possibly improved with precise orbit determination. Finally, the mascon distribution trained with a dense dataset achieves a high level of accuracy (0.1–1 m RMSE). This validates the effectiveness of the mascon model for small body propagations.

The results in Fig. 16 are shown for $n = 1000$. Although the number of masses seems to not impact significantly the gravity errors (recall Fig. 9 and Fig. 13), high n values may be more robust to evaluation mismatches. This refers to the fact that in a propagation, the evaluation point diverges from the truth one over time. This intuition is confirmed by Fig. 9 where for a specific propagation, n is varied. It can be observed that $n = 1000$ provides the best results in the majority of cases. Although it is concluded that the on-orbit mascon fits are not suitable for long term propagations, they may be useful for short-term ones. For example, Fig. 17 shows that after 2 h, scenario B2 error is ≈ 20 m which greatly improves the Keplerian error of ≈ 300 m. This could be of use for navigation or control applications with high update frequencies.

Finally, the computational efficiency of the mascon propagations is evaluated in Table 4. It can be easily observed that mascon models are competitive in terms of propagation speed (without any parallelization). The computational cost is only five times higher than a single Keplerian model for the most heavy model. At the same time, that model is 25 times more efficient than the polyhedron model.

6. Conclusions

This manuscript presents a simultaneous navigation and mascon gravity estimation strategy around a small body. The scheme fuses the concept of dynamical model compensation with a mascon gravity optimizer. The dynamical model compensated unscented Kalman filter generates an on-orbit position-unmodeled acceleration dataset. The measurements are obtained by an on-board camera tracking surface landmarks. Then, the mascon distribution that best fits the on-orbit dataset is found. The mascon optimizer uses Adam gradient descent and ensures physical constraints through a projection step. Numerical simulations around asteroid Eros validate the proposed methodology. Even under different scenario conditions, the trained mascon models always improve the Keplerian model in the analyzed altitude regime. A half day propagation analysis reveals that the trained mascon distribution with the most severe conditions (measurement outages and landmarks

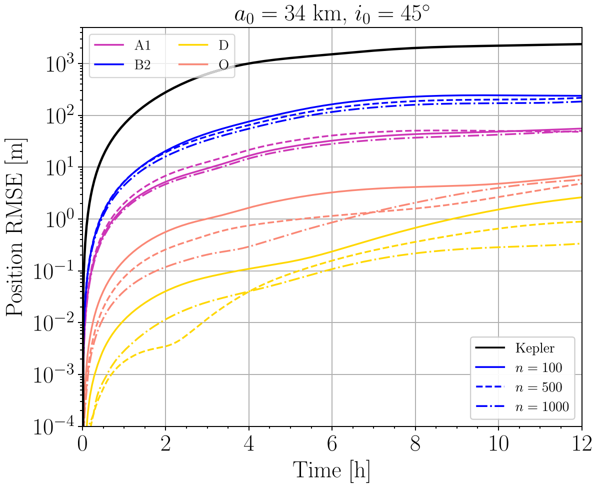

Fig. 17. Position RMSE evolution with respect to number of masses.

Table 4
Computational times of Kepler, polyhedron and mascon models for 12 h orbit propagation.

Model	Mean [s]	Max. [s]
Kepler	0.546	0.555
Poly. 7790 faces	68.11	68.31
Mascon $n = 100$	0.779	0.792
Mascon $n = 500$	1.668	1.687
Mascon $n = 1000$	2.734	2.801

uncertainty) yields 100–1000 m root mean square error. Still, there is room for future improvement since training under the truth on-orbit dataset reduces the propagation error to 1–10 m.

Future work may focus on reducing the accuracy gap between the realistic on-orbit scenario and its equivalent truth dataset. To this end, several ideas has been preliminary tested in this paper. In particular, scenarios without lighting constraint (thus no measurements outages) and the addition of low altitude samples have a positive effect in the mascon accuracy. As an example, the aforementioned propagation errors are 10–100 m under continuous measurements. In the future, it is planned to look deeper into these ideas. For example, measurement outages can be precluded by a multi-satellite configuration which enables the use of inter-satellite ranging. Apart from the signal continuity, relative ranging is a highly accurate measurement that may facilitate precise orbit determination. The detailed collection of low altitude samples is to be further explored. One possible venue is the consideration of a realistic data acquisition from ejecta particles. Another potential method is the design of low altitude fly-overs by the spacecraft (which allows to directly use the proposed DMC-UKF algorithm).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101025257.

Appendix A. Loss function gradient

This appendix provides the expression of the loss function (see Eq. (17)) gradient. The chain rule is exploited to separate the model derivatives from the loss choice.

A.1. Mascon model Jacobian

The Jacobian of the mascon model, for a dataset as Eq. (15), has the following structure

$$\mathbf{J}_M = [\mathbf{J}_{\mu_M} \;\; \mathbf{J}_{\mathbf{r}_M}], \tag{A.1}$$

where $\mathbf{J}_{\mu_M}$ is Jacobian of the masses and $\mathbf{J}_{\mathbf{r}_M}$ refers to the spatial distribution. These can be expanded as

$$\mathbf{J}_{\mu_M} = 2\mathbf{A}_M \mathrm{diag}(\sqrt{\boldsymbol{\mu}_{\mathbf{S}_M}}), \quad \mathbf{J}_{\mathbf{r}_M} = \begin{bmatrix} \left.\frac{\partial \mathbf{a}_M}{\partial \mathbf{r}_{M_1}}\right|_{\mathbf{r}_1} & \cdots & \left.\frac{\partial \mathbf{a}_M}{\partial \mathbf{r}_{M_n}}\right|_{\mathbf{r}_1} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial \mathbf{a}_M}{\partial \mathbf{r}_{M_1}}\right|_{\mathbf{r}_m} & \cdots & \left.\frac{\partial \mathbf{a}_M}{\partial \mathbf{r}_{M_n}}\right|_{\mathbf{r}_m} \end{bmatrix}, \tag{A.2}$$

where diag() indicates a diagonal matrix. The Jacobian $\mathbf{J}_{\mathbf{r}_M}$ elements are

$$\left.\frac{\partial \mathbf{a}_M}{\partial \mathbf{r}_{M_k}}\right|_{\mathbf{r}_j} = \mu_{M_k}\left(\frac{1}{\Delta r_{jk}^3}\mathbf{I} - \frac{3\Delta \mathbf{r}_{jk}\Delta \mathbf{r}_{jk}^T}{\Delta r_{jk}^5}\right). \tag{A.3}$$

Note that Eq. (A.1) structure is flexible. For example, if is decided to just fit $\sqrt{\mu_{M_k}}$ being $\mathbf{r}_{M_k}$ fixed, the spatial distribution Jacobian $\mathbf{J}_{\mathbf{r}_M}$ vanishes from the expression. Then, the Jacobian operations are very efficient since the matrix $\mathbf{A}_M$ needs only to be computed once (as it only depends on $\mathbf{r}_M$ which is fixed in this case).

A.2. Gradient with respect to acceleration

The loss function and the mascon model are related by the gravity acceleration. The loss function gradient with respect to the mascon acceleration prediction is to be computed. Since the loss (see Eq. (17)) is chosen to be the mean squared percent error, this gradient is

$$\nabla L_{\mathbf{a}_M} = \frac{1}{m}\begin{bmatrix} 2(\mathbf{a}_M(\mathbf{r}_1) - \mathbf{a}_1)/a_1^2 \\ \vdots \\ 2(\mathbf{a}_M(\mathbf{r}_m) - \mathbf{a}_m)/a_m^2 \end{bmatrix}. \tag{A.4}$$

Any other choice of loss function is admitted as long as its gradient with respect to the acceleration mascon prediction can be obtained.

A.3. Chain rule

By applying the chain rule, the loss function gradient with respect to the mascon distribution variables is

$$\nabla L = \begin{bmatrix} \mathbf{1}^T\left(\nabla L_{\mathbf{a}_M} \oplus \frac{d\mathbf{a}_M}{d\sqrt{\mu}_1}\right) \\ \vdots \\ \mathbf{1}^T\left(\nabla L_{\mathbf{a}_M} \oplus \frac{d\mathbf{a}_M}{dx_{M_n}}\right) \\ \mathbf{1}^T\left(\nabla L_{\mathbf{a}_M} \oplus \frac{d\mathbf{a}_M}{dy_{M_n}}\right) \\ \mathbf{1}^T\left(\nabla L_{\mathbf{a}_M} \oplus \frac{d\mathbf{a}_M}{dz_{M_n}}\right) \end{bmatrix}, \tag{A.5}$$

where $\mathbf{1}$ is a vector of ones that helps expressing the summation of the right side vector components. Although the expression may seem complex, the vector within the parentheses is nothing more than the element-wise product of $\nabla L_{\mathbf{a}_M}$ with each column of the mascon model Jacobian matrix $\mathbf{J}_M$.

Appendix B. Static position determination

This appendix provides a fast analysis tool for the landmarks-based navigation errors. To this end, a static landmarks-based position determination algorithm is described. Then, the position error distribution with respect to the camera focal length is analyzed.

B.1. Landmark-based position determination

A position fix can be approximately determined by solving the underlying landmarks-based navigation geometrical problem. If no errors (attitude, landmarks position, pixelation) are present, the lines between a surface landmark and its projection on the image plane uniquely intersect in the camera aperture center (assumed as spacecraft center of mass for simplicity). This is deduced from Fig. 2. When errors are present, the intersection is not unique. Alternatively, the position fix can be approximated by the closest point with respect to all lines in terms of the mean squared distance. In order to describe that logic, let us detail how the line between the landmark and its pixel can be constructed:

1. Transform the center pixel to 2D image coordinates: $(u_l, v_l) = w_p(p_{x_l}, p_{y_l})$.
2. Compute the line-of-sight vector $\mathbf{w}_l^C$ of the landmark from the spacecraft:

$$\mathbf{w}_l^C = \frac{1}{\sqrt{u_l^2 + v_l^2 + 1}}\begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix}.$$

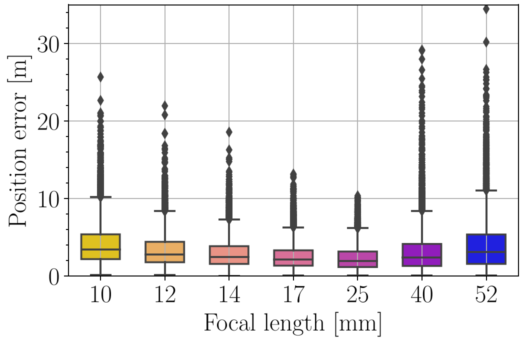

Fig. B.18. Distribution of position errors with respect to focal length.

3. Project the line-of-sight vector into the rotating small body centered frame: $\mathbf{w}_l = (\mathbf{R}_A^C)^T \mathbf{w}_l^C$.
4. Construct the parametric equation of the line: $s\mathbf{w}_l = \mathbf{s} - \mathbf{r}_l$.

Note that s is the independent variable and $\mathbf{s}$ is the set of points sweeping along the line. The distance d_l of a point $\mathbf{r}'$ to a line can be computed as

$$d_l = \|(\mathbf{r}' - \mathbf{r}_l) \times \mathbf{w}_l\|. \tag{B.1}$$

Recall that $\mathbf{w}_l$ is a unit vector. Then, the goal is to find the point $\hat{\mathbf{r}}$ that minimizes the sum of the squared distances D for a number n_L of landmarks

$$\hat{\mathbf{r}} \approx \underset{\mathbf{r}'}{\text{argmin}}\ D = \underset{\mathbf{r}'}{\text{argmin}} \sum_{l=1}^{n_L} d_l^2, \tag{B.2}$$

where the sum of the distances can be expanded as

$$\begin{aligned}\sum_{l=1}^{n_L} d_l^2 &= \sum_{l=1}^{n_L} [(\mathbf{r}' - \mathbf{r}_l) \times \mathbf{w}_l]^T [(\mathbf{r}' - \mathbf{r}_l) \times \mathbf{w}_l] \\ &= \sum_{l=1}^{n_L} (\mathbf{r}' - \mathbf{r}_l)^T (\mathbf{r}' - \mathbf{r}_l) - [(\mathbf{r}' - \mathbf{r}_l) \cdot \mathbf{w}_l]^T [(\mathbf{r}' - \mathbf{r}_l) \cdot \mathbf{w}_l].\end{aligned} \tag{B.3}$$

Note that the vector identity $(\mathbf{a} \times \mathbf{b})^T (\mathbf{a} \times \mathbf{b}) = (\mathbf{a}^T \mathbf{a})(\mathbf{b}^T \mathbf{b}) - (\mathbf{a}^T \mathbf{b})(\mathbf{b}^T \mathbf{a})$ is used. The spacecraft position estimate $\hat{\mathbf{r}}$, can be computed by taking the derivative of D with respect to $\mathbf{r}'$ and equaling the resulting expression to zero

$$\left.\frac{dD}{d\mathbf{r}'}\right|_{\mathbf{r}'=\hat{\mathbf{r}}} = \sum_{l=1}^{n_L} 2(\hat{\mathbf{r}} - \mathbf{r}_l) - 2[(\hat{\mathbf{r}} - \mathbf{r}_l) \cdot \mathbf{w}_l]\mathbf{w}_l = \mathbf{0}. \tag{B.4}$$

After rearranging terms, $\hat{\mathbf{r}}$ can be cleared as

$$\left(n_L \mathbf{I} - \sum_{l=1}^{n_L} \mathbf{w}_l (\mathbf{w}_l)^T \right) \hat{\mathbf{r}} = \sum_{l=1}^{n_L} \mathbf{r}_l - (\mathbf{r}_l \cdot \mathbf{w}_l)\mathbf{w}_l, \tag{B.5}$$

which is a 3 × 3 system of linear equations that can be solved by inverting the left-side matrix. Note that $\mathbf{I}$ denotes the identity matrix. There are two singular cases for Eq. (B.5) system. The first one is $n_L < 2$ where it is evident that there are not enough lines to determine an intersection. The other case is when all the lines are parallel which may occur if the visible landmarks are too close (thus projecting to the same pixel on the image).

B.2. Position error with respect to focal length

The static position determination algorithm of Appendix B.1 allows a preliminary analysis of navigation errors. The main advantage is that this analysis is purely geometrical and free of dynamics. In the case under consideration, it is of interest to find a suitable camera focal length f (which controls how much is visible and its resolution) for Section 5.2. To this end, several focal lengths ranging the diagonal FOV evenly between 23° and 94° are analyzed for the orbit under consideration. The distribution of position errors is shown in Fig. B.18. The simulation does not include the lighting constraint in order to avoid an excessive number of outliers. After examining Fig. B.18, the value $f = 25$ mm is used in this manuscript.

References

[1] J.C. Castillo-Rogez, M. Pavone, J.A. Hoffman, I.A.D. Nesnas, Expected science return of spatially-extended in-situ exploration at small solar system bodies, in: IEEE Aerospace Conference, 2012, pp. 1–15, http://dx.doi.org/10.1109/AERO.2012.6187034.

[2] A. Cheng, A. Rivkin, P. Michel, J. Atchison, O. Barnouin, L. Benner, N. Chabot, C. Ernst, E. Fahnestock, M. Kueppers, P. Pravec, E. Rainey, D. Richardson, A. Stickle, C. Thomas, AIDA dart asteroid deflection test: Planetary defense and science objectives, Planet. Space Sci. 157 (2018) 104–115, http://dx.doi.org/10.1016/j.pss.2018.02.015.

[3] W. Hart, G.M. Brown, S.M. Collins, M. De Soria-Santacruz Pich, P. Fieseler, D. Goebel, D. Marsh, D.Y. Oh, S. Snyder, N. Warner, G. Whiffen, L.T. Elkins-Tanton, J.F. Bell, D.J. Lawrence, P. Lord, Z. Pirkl, Overview of the spacecraft design for the psyche mission concept, in: IEEE Aerospace Conference, Big Sky, United States, 2018, pp. 1–20, http://dx.doi.org/10.1109/AERO.2018.8396444.

[4] P. Michel, M. Küppers, A.C. Bagatin, B. Carry, S. Charnoz, J. de Leon, A. Fitzsimmons, P. Gordo, S.F. Green, A. Hérique, M. Juzi, Ö. Karatekin, T. Kohout, M. Lazzarin, N. Murdoch, T. Okada, E. Palomba, P. Pravec, C. Snodgrass, P. Tortora, K. Tsiganis, S. Ulamec, J.-B. Vincent, K. Wünnemann, Y. Zhang, S.D. Raducan, E. Dotto, N. Chabot, A.F. Cheng, A. Rivkin, O. Barnouin, C. Ernst, A. Stickle, D.C. Richardson, C. Thomas, M. Arakawa, H. Miyamoto, A. Nakamura, S. Sugita, M. Yoshikawa, P. Abell, E. Asphaug, R.-L. Ballouz, W.F. Bottke, D.S. Lauretta, K.J. Walsh, P. Martino, I. Carnelli, The ESA hera mission: Detailed characterization of the DART impact outcome and of the binary asteroid (65803) didymos, Planet. Sci. J. 3 (7) (2022) 160, http://dx.doi.org/10.3847/PSJ/ac6f52.

[5] A.S. Rivkin, N.L. Chabot, A.M. Stickle, C.A. Thomas, D.C. Richardson, O. Barnouin, E.G. Fahnestock, C.M. Ernst, A.F. Cheng, S. Chesley, S. Naidu, T.S. Statler, B. Barbee, H. Agrusa, N. Moskovitz, R.T. Daly, P. Pravec, P. Scheirich, E. Dotto, V.D. Corte, P. Michel, M. Küppers, J. Atchison, M. Hirabayashi, The double asteroid redirection test (DART): Planetary defense investigations and requirements, Planet. Sci. J. 2 (5) (2021) 173, http://dx.doi.org/10.3847/PSJ/ac063e.

[6] D.S. Lauretta, S.S. Balram-Knutson, E. Beshore, W.V. Boynton, C. Drouet d'Aubigny, D.N. DellaGiustina, H.L. Enos, D.R. Golish, C.W. Hergenrother, E.S. Howell, C.A. Bennett, E.T. Morton, M.C. Nolan, B. Rizk, H.L. Roper, A.E. Bartels, B.J. Bos, J.P. Dworkin, D.E. Highsmith, D.A. Lorenz, L.F. Lim, R. Mink, M.C. Moreau, J.A. Nuth, D.C. Reuter, A.A. Simon, E.B. Bierhaus, B.H. Bryan, R. Ballouz, O.S. Barnouin, R.P. Binzel, W.F. Bottke, V.E. Hamilton, K.J. Walsh, S.R. Chesley, P.R. Christensen, B.E. Clark, H.C. Connolly, M.K. Crombie, M.G. Daly, J.P. Emery, T.J. McCoy, J.W. McMahon, D.J. Scheeres, S. Messenger, K. Nakamura-Messenger, K. Righter, S.A. Sandford, OSIRIS-REx: Sample return from asteroid (101955) bennu, Space Sci. Rev. 212 (1) (2017) 925–984, http://dx.doi.org/10.3847/PSJ/ac6f52.

[7] D.J. Scheeres, Orbit mechanics about asteroids and comets, J. Guid. Control Dyn. 35 (3) (2012) 987–997, http://dx.doi.org/10.2514/1.57247.

[8] J.K. Miller, A.S. Konopliv, P. Antreasian, J.J. Bordi, S. Chesley, C.E. Helfrich, W.M. Owen, T.C. Wang, B.G. Williams, D.K. Yeomans, D.J. Scheeres, Determination of shape, gravity, and rotational state of asteroid 433 eros, Icarus 155 (2002) 3–17, http://dx.doi.org/10.1006/icar.2001.6753.

[9] J. Leonard, B. Jones, E. Villalba, G. Born, Absolute orbit determination and gravity field recovery for 433 eros using satellite-to-satellite tracking, in: AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, United States, 2012, http://dx.doi.org/10.2514/6.2012-4877.

[10] S. Hesar, J. Parker, J. McMahon, G. Born, Small body gravity field estimation using LiAISON supplemented optical navigation, in: 36th AAS Guidance and Control Conference, Breckenridge, United States, 2015.

[11] N. Stacey, S. D'Amico, Autonomous swarming for simultaneous navigation and asteroid characterization, in: AAS/AIAA Astrodynamics Specialist Conference, Snowbird, United States, 2018.

[12] K. Dennison, N. Stacey, S. D'amico, Autonomous asteroid characterization through nanosatellite swarming, IEEE Trans. Aerosp. Electron. Syst. (2023) 1–22, http://dx.doi.org/10.1109/TAES.2023.3245997.

[13] J.D. Biggs, E. Ciccarelli, In-situ tracking of a solar sail's characteristic acceleration using a robust active disturbance estimator, in: 5th International Symposium on Solar Sailing, Aachen, Germany, 2019.

[14] J.C. Sanchez, R. Vazquez, J.D. Biggs, F. Bernelli-Zazzera, Orbit-attitude predictive control in the vicinity of asteroids with in situ gravity estimation, J. Guid. Control Dyn. 45 (3) (2022) 262–279, http://dx.doi.org/10.2514/1.G005572.

[15] A. Pasquale, S. Silvestrini, A. Capannolo, P. Lunghi, M. Lavagna, Small bodies non-uniform gravity field on-board learning through hopfield neural network, Planet. Space Sci. 212 (105425) (2022) 1–14, http://dx.doi.org/10.1016/j.pss.2022.105425.

[16] J.R. Martin, H. Schaub, Preliminary analysis of small-body gravity field estimation using physics-informed neural networks and Kalman filters, in: International Astronautical Congress, Paris, France, 2022.

[17] J.C. Sanchez, H. Schaub, Small body navigation and gravity estimation using Kalman filter and least-squares fitting, in: 33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, United States, 2023.

[18] R.A. Werner, D.J. Scheeres, Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 castalia, Celestial Mech. Dynam. Astronom. 65 (1996) 313–344, http://dx.doi.org/10.1007/BF00053511.

[19] G. Romain, B. Jean-Pierre, Ellipsoidal harmonic expansions of the gravitational potential: Theory and application, Celestial Mech. Dynam. Astronom. 79 (4) (2001) 235–275, http://dx.doi.org/10.1023/A:1017555515763.

[20] R.P. Russell, N. Arora, Global point mascon models for simple, accurate, and parallel geopotential computation, J. Guid. Control Dyn. 35 (5) (2012) 1568–1581, http://dx.doi.org/10.2514/1.54533.

[21] A. Gao, W. Liao, Efficient gravity field modeling method for small bodies based on Gaussian process regression, Acta Astronaut. 157 (2019) 73–91, http://dx.doi.org/10.1016/j.actaastro.2018.12.020.

[22] J.R. Martin, H. Schaub, Physics-informed neural networks for gravity field modeling of small bodies, Celestial Mech. Dynam. Astronom. 134 (46) (2022) http://dx.doi.org/10.1007/s10569-022-10101-8.

[23] D. Izzo, P. Gomez, Geodesy of irregular small bodies via neural density fields, Commun. Eng. 1 (48) (2022) http://dx.doi.org/10.1038/s44172-022-00050-3.

[24] T.G.G. Chanut, S. Aljbaae, V. Carruba, Mascon gravitation model using a shaped polyhedral source, Mon. Not. R. Astron. Soc. 450 (4) (2015) 3742–3749, http://dx.doi.org/10.1093/mnras/stv845.

[25] A. Colagrossi, F. Ferrari, M. Lavagna, K. Howell, Dynamical evolution about asteroids with high fidelity gravity field and perturbations modeling, in: AAS/AIAA Astrodynamics Specialist Conference, Vail, United States, 2015.

[26] J. Garcia-Bonilla, P. Machuca, M. Sanjurjo-Rivo, Small-body gravitational modeling for on-board operations and mass distribution estimation: Trade-off analysis and novel approach, in: International Astronautical Congress, Dubai, United Arab Emirates, 2021.

[27] B. Hockman, J. Villa, A. French, S. Chesley, D.J. Scheeres, J. McMahon, Gravity Poppers: Hopping probes for the interior mapping of small solar system bodies, in: IEEE Aerospace Conference, 2022.

[28] K.A. Myers, B.D. Tapley, Dynamical model compensation for near-earth satellite orbit determination, AIAA J. 13 (3) (1975) 343–349, http://dx.doi.org/10.2514/3.49702.

[29] J.M. Leonard, F.G. Nievinski, G.H. Born, Gravity error compensation using second-order Gauss-Markov processes, J. Spacecr. Rockets 50 (1) (2013) 217–229, http://dx.doi.org/10.2514/1.A32262.

[30] M. Vetrisano, M. Vasile, Autonomous navigation of a spacecraft formation in the proximity of an asteroid, Adv. Space Res. 57 (8) (2016) 1783–1804, http://dx.doi.org/10.1016/j.asr.2015.07.024.

[31] S. Zhu, D. Liu, Y. Liu, P. Cui, Observability-based visual navigation using landmarks measuring angle for pinpoint landing, Acta Astronaut. 155 (2019) 313–324, http://dx.doi.org/10.1016/j.actaastro.2018.10.011.

[32] E. Wan, R. Merwe, The unscented Kalman filter for nonlinear estimation, in: Adaptive Systems for Signal Processing, Communications, and Control Symposium, Alberta, Canada, 2000.

[33] J.R. Martin, H. Schaub, Physics-informed neural network gravity model revisited, model generation III, in: 33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, United States, 2023.

[34] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference for Learning Representations, San Diego, United States, 2015, http://dx.doi.org/10.48550/arXiv.1412.6980.

[35] P.W. Kenneally, S. Piggott, H. Schaub, Basilisk: A flexible, scalar and modular astrodynamics simulation framework, J. Aerosp. Inf. Syst. 17 (9) (2020) 496–507, http://dx.doi.org/10.2514/1.I010762.

[36] J. Villa, A. French, J. McMahon, D. Scheeres, B. Hockman, Gravity estimation at small bodies via optical tracking of hopping artificial probes, in: AAS/AIAA Astrodynamics Specialist Conference, Big Sky, United States, 2021.