A gravity assist mapping for the circular restricted three-body problem using Gaussian processes

Liu, Yuxin; Noomen, Ron; Visser, Pieter

DOI
10.1016/j.asr.2021.06.054

Publication date
2021

Document Version
Final published version

Published in
Advances in Space Research

Citation (APA)
Liu, Y., Noomen, R., & Visser, P. (2021). A gravity assist mapping for the circular restricted three-body problem using Gaussian processes. *Advances in Space Research*, *68*(6), 2488-2500. https://doi.org/10.1016/j.asr.2021.06.054

Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 68 (2021) 2488–2500

ADVANCES IN SPACE RESEARCH (a COSPAR publication)

www.elsevier.com/locate/asr

A gravity assist mapping for the circular restricted three-body problem using Gaussian processes

Yuxin Liu *, Ron Noomen, Pieter Visser

Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, the Netherlands

Received 9 February 2021; received in revised form 23 June 2021; accepted 28 June 2021
Available online 9 July 2021

Abstract

Inspired by the Keplerian Map and the Flyby Map, a Gravity Assist Mapping using Gaussian Process Regression for the fully spatial Circular Restricted Three-Body Problem is developed. A mapping function for quantifying the flyby effects over one orbital period is defined. The Gaussian Process Regression model is established by proper mean and covariance functions. The model learns the dynamics of flyby's from training samples, which are generated by numerical propagation. To improve the efficiency of this method, a new criterion is proposed to determine the optimal size of the training dataset. We discuss its robustness to show the quality of practical usage. The influence of different input elements on the flyby effects is studied. The accuracy and efficiency of the proposed model have been investigated for different energy levels, ranging from representative high- to low-energy cases. It shows improvements over the Kick Map, an independent semi-analytical method available in literature. The accuracy and efficiency of predicting the variation of the semi-major axis are improved by factors of 3.3, and 1.27×10^4, respectively.

Keywords: Gravity assist mapping; Machine learning; Gaussian process regression

1. Introduction

For deep space missions, a flyby along a planet or a major moon is a common technique to save propellant. The state of the spacecraft is changed effectively due to the gravitational pull of these massive bodies. In the preliminary design of an interplanetary mission, the patched-conics model is typically used (Bate et al., 1971; Broucke, 1988; Longuski and Williams, 1991; Strange and Longuski, 2002). This model has a crucial assumption: the gravity of the flyby body (e.g. Mars) is considered if and only if the spacecraft comes into the sphere of influence of that body. The gravity effect of other objects (in particular, the Sun) can be ignored; the original problem is split up in a succession of elementary two-body problem formulations. The patched-conics model simplifies the preliminary mission design process. However, it has a limitation in describing the dynamics in more detail, in particular for low-energy cases with phenomena like invariant manifolds, which do exist in a three-body formulation. The special dynamics described by the three-body formulation has proven to be essential to design low-energy trajectories or study the movement of asteroids between Mars and Jupiter (Beutler, 2004; Campagnola et al., 2012, 2014).

In order to reduce the degree of difficulty of a most general three-body formulation, the Circular Restricted Three-Body Problem (CR3BP) can be introduced (Szebehely, 1967). Here, it is assumed that the mass of one of the three bodies can be neglected and the other massive bodies move in circular orbits about the barycenter of the system. Considering the Sun-Earth-spacecraft system, this formulation takes the gravitational attraction of the Sun and the Earth into account simultaneously. The domain of a flyby in the

* Corresponding author.
E-mail address: yuxin.liu@tudelft.nl (Y. Liu).

https://doi.org/10.1016/j.asr.2021.06.054
0273-1177/

CR3BP is expanded beyond the sphere of influence of the Earth.

The Keplerian Map (KM) is a common method to study the flyby effects in the CR3BP. Based on perturbation theory, this method was initially developed to investigate the dynamics of comets around Jupiter (Petrosky and Broucke, 1987; Chirikov and Vecheslavov, 1989; Conley, 1968; Malyshkin and Tremaine, 1999; Zhou et al., 2000). Ross and Scheeres (2007) studied the KM in the system of a planar CR3BP, which models the flyby effect over one orbital period using an energy-kick function. It works effectively for an energy level of around 3.0 for the Jacobi constant. For a quantitative investigation in the spatial framework (CR3BP), a three-dimensional KM was developed to solve initial-value problems (Peñagaricano and Scheeres, 2010). The accuracy of this method is reasonably good for low three-body energy levels only. Alessi and Sánchez (2016) expanded this method and derived a formulation to estimate the variation of orbital elements due to the perturbation of a third body. It then may be used to study the effects of a flyby. The results are obtained by applying the classical Lagrange planetary equations and performing a first iteration of Picard's method over one orbital period. This formulation assumes a sufficiently small mass ratio and may suffer from singularity problems for low eccentricity and inclination. Neves et al. (2018) analysed various ways of modelling the third-body effect and tried to obtain a fully analytical solution. They showed the accuracy of these models within certain fields of application. It is well known that the 3BP is not solvable in an analytically closed form. Many attempts have been made to find closed analytical solutions for the variation of the orbital elements of the massless body under specific conditions in the CR3BP. The techniques mentioned above are semi-analytical and employ numerical methods to generate solutions. The Flyby Map developed by Campagnola et al. (2012) is also a typical approach to study the CR3BP, which is fully numerical and valid for a wide range of energy levels. It extended the functionality of the Tisserand graph over the applicability of the patched-conics model. According to the spacecraft velocity relative to the secondary, the flybys are categorised into two types: 'direct' and 'retrograde' flybys. In terms of the variation of semi-major axis, the direct flybys are shown to be more efficient than the latter.

This paper is inspired by the works on the Keplerian Map and Flyby Map. Instead of solving the analytical forms of flyby effects in the CR3BP, we develop a Gravity Assist Mapping (GAM) based on Gaussian Process Regression (GPR) to quantify flyby effects in a full three-dimensional situation. Specifically, the flyby refers to an encounter of the massless body with the second-largest body. GPR is a supervised machine-learning method, which was developed by Krige (1951) for mine valuation. For a complete three-body problem, the technique of Deep Neural Network was used to predict the position of each body within a fixed duration (Breen et al., 2019). The prediction of velocity was not investigated. Instead of using Cartesian coordinates, the GAM models use Keplerian orbital elements similar to the Keplerian Map. The size of the training data set for a DNN model requires around 10,000 samples. Moreover, the structure, the number of layers, the number of nodes, the activation function and the pooling function, etc. need to be determined. The model selection of the GPR method is simpler than that of the DNN. In addition, the GPR models work accurately at a lower level of training data size (Rasmussen and Williams, 2006). In the field of astrodynamics, GPR has been used to evaluate the accessibility of asteroids, to model the gravity field of small bodies, and to design low-thrust trajectories (Shang and Liu, 2017; Gao and Liao, 2019; Bouwman et al., 2019). Liu et al., 2021 used GPR to evaluate flyby effects; their model was developed in a planar CR3BP and performs more efficiently and accurately than the work of Ross and Scheeres (2007). The complexity of the CR3BP increases significantly when going from a planar model to a full spatial model. In the current work, we develop a new GPR-based GAM to solve this. The Keplerian Map for approximating a particle's motion was initially developed for low three-body energies (Ross and Scheeres, 2007). Here, we aim at developing an approach which is applicable for a wider range of energy levels. The energy of the considered cases ranges from values lower than that associated with the L1 libration point to values larger than that associated with L4/L5. Several thousands of training samples are required for the GPR model to learn the dynamics. The prediction of output given an arbitrary input will be compared with a previous semi-analytical method and numerical integration methods. The computational effort is expected to be further reduced particularly when a large group of initial conditions needs to be assessed.

2. Circular restricted three-body problem

The CR3BP describes the dynamics of three point masses: two massive bodies P_1 and P_2 and a third body P_3 with masses M_1, M_2 and M_3, respectively, which meet the condition $M_1 > M_2 \gg M_3$. Throughout this paper, flybys in the Sun-(Earth + Moon)-spacecraft system are investigated and the primary P_1, the secondary P_2 and massless particle P_3 are the Sun, the Earth + Moon and the spacecraft, respectively. In the rotating reference frame, the origin is set at the barycenter of P_1 and P_2, and the X-axis is aligned with the direction to P_2. The primaries P_1 and P_2 rotate around the Z-axis. The equations of motion for the spacecraft are described as (Szebehely, 1967)

$$\begin{cases} \ddot{x} - 2\dot{y} = x - \frac{(1-\mu)}{r_1^3}(x+\mu) - \frac{\mu}{r_2^3}(x-1+\mu) \\ \ddot{y} + 2\dot{x} = y - \frac{(1-\mu)}{r_1^3}y - \frac{\mu}{r_2^3}y \\ \ddot{z} = -\frac{(1-\mu)}{r_1^3}z - \frac{\mu}{r_2^3}z \end{cases} \quad (1)$$

where x, y and z represent the normalized position coordinates in the barycentric Cartesian reference frame, and

$$\begin{cases} \mu = M_2/(M_1+M_2) \\ r_1 = \sqrt{(x+\mu)^2+y^2+z^2} \\ r_2 = \sqrt{(x-1+\mu)^2+y^2+z^2} \end{cases} \tag{2}$$

μ is defined as the mass parameter which differentiates between different implementations of the CR3BP, and is equal to 3.036×10^{-6} for the system of Sun-(Earth + Moon)-spacecraft. r_1 and r_2 are the distances between P_3 and the primaries P_1 and P_2, respectively.

A closed-form solution of Eq. 1 does not exist; numerical integration methods are typically used to solve the problem. Some attempts were made to find analytical solutions for the variation of the orbital elements of P_3 (Ross and Scheeres, 2007; Alessi and Sánchez, 2016; Neves et al., 2018). Based on the concept of KM developed by Ross and Scheeres (2007), the current work studies a new approach to quantify the flyby effects for a wide range of Jacobi energy levels.

2.1. Jacobi constant

The CR3BP has an analytical integral of motion which is constant along the trajectory of the spacecraft. The value of this Jacobi constant C_J is fully determined by the position and velocity of the spacecraft. In the rotating reference frame, the Jacobi constant is defined as (Beutler, 2004):

$$C_J = x^2+y^2+\frac{2(1-\mu)}{r_1}+\frac{2\mu}{r_2}-(\dot{x}^2+\dot{y}^2+\dot{z}^2)+\mu(1-\mu) \tag{3}$$

The Jacobi constant reflects the energy level of the trajectory of the spacecraft. A high value of C_J represents a low energy level. The region that is accessible for the spacecraft is related to the Jacobi constant for a given mass ratio. The Jacobi constant associated with the Lagrange libration points allows to characterize the accessible region for different cases. For $\mu = 3.036 \times 10^{-6}$, the values for the Jacobi constant of these points are $C_{L1} = 3.000898, C_{L2} = 3.000893, C_{L3} = 3.000003, C_{L4/L5} = 2.999997$. For some values of C_J, there are inaccessible regions which divide the accessible regions (i.e. Hill's regions) into different parts. The boundaries of these regions are zero-velocity surfaces. If the value of C_J is larger than C_{L1}, a particle remains in its initial region. When $C_{L1} > C_J > C_{L2}$, it is possible for the spacecraft to move to the vicinity of the other primary if it started in the vicinity of either of the primaries. Transitions between the interior and exterior regions are not possible yet. This becomes possible when $C_J < C_{L2}$ (for details see Beutler (2004)). The current work focuses on the motion of a spacecraft which starts from the exterior region as shown in Fig. 1.

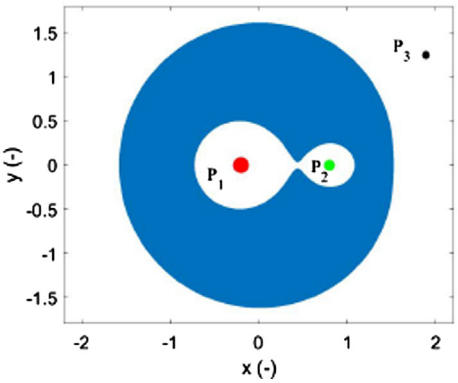

Fig. 1. Hill's region when $C_{L1} > C_J > C_{L2}$.

3. Gravity assist mapping

3.1. Mapping function

In this paper, we aim to develop a GPR model to quantify the flyby effects in the framework of a CR3BP, i.e. a GPR-based GAM (GPR-GAM). The mapping function of the GPR-GAM is described as:

$$\begin{aligned} &F_{\mathrm{GPR-GAM}} : \boldsymbol{x} \mapsto \boldsymbol{y} \\ &\text{or}[a, e, i, \omega, \phi] \mapsto [\delta a, \delta e, \delta i, \delta\omega, \delta\Omega] \end{aligned} \tag{4}$$

where the vector $\boldsymbol{x}$ represents the orbital elements of the spacecraft before a flyby and $\boldsymbol{y}$ the variation of orbital elements due to that flyby, after one orbital period of the initial osculating orbit. a, e, i, ω, Ω are osculating semi-major axis, eccentricity, inclination, argument of pericenter and longitude of the ascending node of the spacecraft with respect to the primary. Particularly, the angular phasing angle $\phi = \tan^{-1}(\tan(\omega)\cos(i)) + \Omega$ shown in Fig. 2 describes the relative position between the spacecraft and the secondary (Alessi and Sánchez, 2016). When investigating the effects of a flyby, δa and δe have an explicit response to ϕ. The reference system will be detailed below.

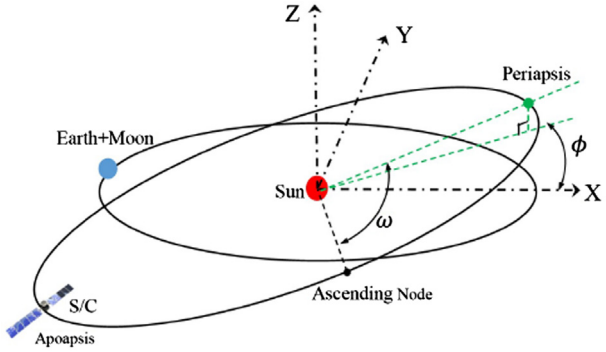

Fig. 2. Schematics of a three-body system before a flyby. The orbit of the S/C is the initial osculating orbit centered at the Sun in the inertial reference frame; it starts at apoapsis.

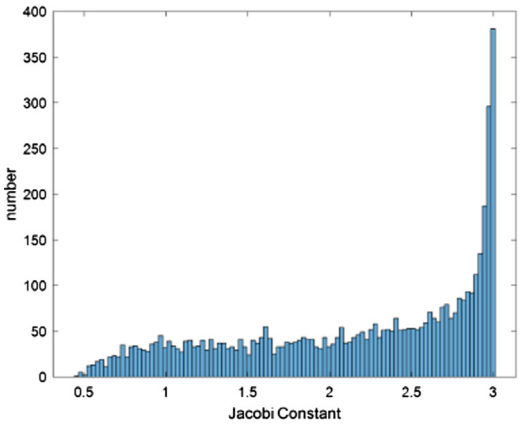

Fig. 3. Histogram of the Jacobi constant for 5000 samples based on the input space defined in Table 1.

Table 1
Boundary for the input space.

r_p [AU]	r_a [AU]	i [deg]	ω [deg]	ϕ [deg]
[1.00004464, 1.02]	[1.01, 3.03]	[0, 90]	[0, 360]	[−25, 25]

The minimum value of r_p corresponds to a minimum distance of 300 km between the spacecraft and the Earth surface when $\phi = 0$ deg, i.e. both of them arrive at the positive X-axis simultaneously.

3.2. Configuration of flyby

In Eq. 4, the input and output are defined in an inertial reference frame. Fig. 2 shows the orbit of a spacecraft before a flyby in the Sun-(Earth + Moon)-spacecraft system. The origin is located at the Sun, and the Earth revolves around it in the XY-plane, so that the reference plane is the ecliptic plane. Here, the input vector $\boldsymbol{x}$ represents the osculating orbital elements of the spacecraft at apoapsis. The angular phasing ϕ denotes the angle between the projection of the Sun-periapsis line of the spacecraft onto the XY-plane and the positive X-axis. The spacecraft starts from apoapsis with a true anomaly of $-\pi$, and the true anomaly of the Earth is initialized such as to reach the positive X-axis when the spacecraft has travelled half of its unperturbed orbital period.

The GPR is a supervised learning method which requires training samples consisting of input and output. In this paper, the boundary of the input space is defined such to obtain obvious flyby effects. In order to quantify the distance between the spacecraft and the secondary properly, we define a boundary based on the radius of periapsis r_p and the radius of apoapsis r_a, and then transform these into a and e.

The boundary is specified in Table 1. Based on this input space, the Jacobi constant ranges from 0.512176 to 3.001196, which covers energy levels from low to high (Ross and Scheeres, 2007). The value of i only takes prograde orbits into account, which is the expected geometry for normal interplanetary missions. The boundary for ϕ is defined in order to observe an obvious flyby effect; the distance in phasing should not become too large. Fig. 3 shows the histogram of the Jacobi constant of a group of 5000 randomly generated samples (in line with further calculations). In addition to this, we divide the range of ω into four quadrants:

$$\begin{aligned}\omega &= \{\omega^{Q1}, \omega^{Q2}, \omega^{Q3}, \omega^{Q4}\} \\ &= \{(0, 90], (90, 180], (180, 270], (270, 360]\} \text{ deg}\end{aligned} \tag{5}$$

Given the same a, e, i and ϕ, the output $\boldsymbol{y}$ is a periodic function in terms of ω:

$$\boldsymbol{y}(\omega|a, e, i, \phi) = \boldsymbol{y}(\omega + 180|a, e, i, \phi), \omega \in [0, 180] \text{ deg} \tag{6}$$

In order to achieve an accurate estimate of the flyby effects, two GPR models are built separately for ω^{Q1} and ω^{Q2}. Using the property of periodicity (Eq. 6), these models can be simply applied to ω^{Q3} and ω^{Q4}. After generating training inputs, the training outputs are obtained through numerical propagation using Eq. 1, as will be explained in Section 4.3.

4. Gaussian process regression model

The development of a GPR-GAM considers two major steps: training and predicting. During the training part, the GPR model is built by digesting the empirical information from a certain process. In this paper, this process is the mapping of orbital elements before the flyby to the post-flyby changes of those elements. In the prediction part, the trained model estimates the output for an arbitrary input. We develop five GPR-GAM models with a single output; each model associates the five inputs into one of the outputs in Eq. 4. This separation of constructing a GPR-GAM model simplifies the training process to a single-objective optimization problem. The mapping function of the basic GPR model is:

$$\begin{aligned} f_{\text{GPR-GAM}}: \quad & \mathbb{R}^5 \to \mathbb{R} \\ & \boldsymbol{x} \mapsto y \end{aligned} \tag{7}$$

where y represents one of the outputs $[\delta a, \delta e, \delta i, \delta\omega, \delta\Omega]$ as in Eq. 4.

The GPR-GAM model is developed in terms of the properties of a Gaussian Process (GP). A GP is a collection of random variables following their own Gaussian distribution. A key property of any GP is that any finite number of these random variables forms a joint Gaussian distribution. Instead of simple variables, the GPR-GAM method takes the function $f(\boldsymbol{x})$ as that random variable. Therefore, the development of the GPR-GAM model works in function space. A GP over the function $f(\boldsymbol{x})$ is given as:

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{8}$$

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are points in the input space; $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}')$ are the mean function and the covariance function, respectively. Both functions will be explained in Sections 4.1 and 4.2, respectively. These functions specify the properties of a GP completely through their formulation and

hyper-parameters. The hyper-parameters are actually the parameters of the mean function and the covariance function. 'Hyper-' is used to emphasize that they are characteristics of a non-parametric model (Rasmussen and Williams, 2006).

The formulation of $m(\boldsymbol{x})$ and $k(\boldsymbol{x},\boldsymbol{x}')$ is selected by the authors. Each formulation has a set of corresponding hyper-parameters. The values of these hyper-parameters are initialized randomly. Then the learning procedure starts, which is actually a process of optimizing the hyper-parameters using the empirical information. This information comes from the training dataset D_{train}, defined by:

$$D_{\text{train}} = \{(\boldsymbol{X},\boldsymbol{Y})|\boldsymbol{X} = [\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N],\boldsymbol{Y} = [y_1,y_2,\ldots,y_N]\} \tag{9}$$

where $(\boldsymbol{x}_i,y_i)$ represents one training sample $(i=1,2,\ldots,N)$, and N is the number of training samples. According to the properties of GP, the collection of training outputs $\boldsymbol{Y}$ can be written as a joint Gaussian distribution: (Liu et al., 2021):

$$\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{m}(\boldsymbol{X}),\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})) \tag{10}$$

where $\boldsymbol{m}(\boldsymbol{X})$ represents the mean function and will be elaborated in Section 4.1.

$$\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) = \begin{pmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & k(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1,\boldsymbol{x}_N) \\ k(\boldsymbol{x}_2,\boldsymbol{x}_1) & k(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2,\boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N,\boldsymbol{x}_1) & k(\boldsymbol{x}_N,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_N,\boldsymbol{x}_N) \end{pmatrix} \tag{11}$$

The objective of the training process is to optimize the hyper-parameters such that the GPR-GAM model has the highest possibility to reproduce the ground-truth outputs. This is done by maximizing the marginal likelihood $\log p(\boldsymbol{Y}\mid\boldsymbol{X})$ using the following equation (Rasmussen and Williams, 2006):

$$\log p(\boldsymbol{Y}\mid\boldsymbol{X}) = -\frac{1}{2}\boldsymbol{Y}^T\boldsymbol{K}^{-1}\boldsymbol{Y} - \frac{1}{2}\log\mid\boldsymbol{K}\mid - \frac{N}{2}\log 2\pi \tag{12}$$

In principle, the marginal likelihood represents the possibility of obtaining true outputs given the inputs and hyper-parameters. The hyper-parameters that yield a high marginal likelihood would give rise to good predictions. In order to predict the output y^* for a new input $\boldsymbol{x}^*$, which are called test output and test input respectively, we construct the joint distribution of the training outputs and y^* as

$$\begin{pmatrix} \boldsymbol{Y} \\ y^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{m}(\boldsymbol{X}) \\ m(\boldsymbol{x}^*) \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X},\boldsymbol{x}^*) \\ \boldsymbol{K}(\boldsymbol{X},\boldsymbol{x}^*)^T & k(\boldsymbol{x}^*,\boldsymbol{x}^*) \end{pmatrix}\right) \tag{13}$$

where $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x}^*)$ represents the vector of the covariance functions evaluated at all pairs of training inputs $\boldsymbol{X}$ and new input $\boldsymbol{x}^*$.

Based on Bayesian inference, the predicted y^* given $\boldsymbol{x}^*$ has a Gaussian distribution

$$y^*|\boldsymbol{x}^*,\boldsymbol{X},\boldsymbol{Y} \sim \mathcal{N}(\mu(y^*),cov(y^*)) \tag{14}$$

where the value of $\mu(y^*)$is obtained from

$$\mu(y^*) = m(\boldsymbol{x}^*) - K(\boldsymbol{X},\boldsymbol{x}^*)^T\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})^{-1}(\boldsymbol{Y}-\boldsymbol{m}(\boldsymbol{X})) \tag{15}$$

Typically, we take the value of $\mu(y^*)$as the predicted output (Rasmussen and Williams, 2006). In order to improve the efficiency of the prediction effort, we simplify Eq. 15 by merging matrices (Liu et al., 2021):

$$\mu(y^*) = m(\boldsymbol{x}^*) - K(\boldsymbol{X},\boldsymbol{x}^*)^T\boldsymbol{Q}_{\text{train}} \tag{16}$$

where $\boldsymbol{Q}_{\text{train}} = \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})^{-1}(\boldsymbol{Y}-\boldsymbol{m}(\boldsymbol{X}))$can be obtained immediately after the training process has been completed, with the obvious advantage that it needs to be calculated only once.

4.1. Mean function

When constructing GPR models for the planar CR3BP, a zero-mean function is sufficient for this simpler scenario (Liu et al., 2021). However, due to the increased complexity of fully spatial CR3BP dynamics, compared to planar dynamics, we add a non-zero mean function in this work. The option of having a non-zero function offers the possibility to interpret the model and express the empirical information more correctly. The function employed here is a straightforward constant:

$$m(\boldsymbol{x}) = c \tag{17}$$

where c is one of the hyper-parameters that will be optimized while training.

4.2. Covariance function

The supervised machine learning generates predictions using the relationship between samples. The test output is predicted based on the correlation between its corresponding input and all the training inputs. In a GPR method, the covariance function is the most crucial module because it describes this similarity between different samples and plays a key role in interpolation. What remains is the specific formulation of this covariance function, and the number and distribution of training samples. In principle, there is no universal rule for choosing an appropriate covariance function. The choice is typically based on experience, the information about the characteristics of the relation between inputs and outputs provided by training samples, and the error over test samples. For the planar CR3BP, the authors selected a sum covariance function combining a Rational Quadratic function with Automatic Relevance Determination (RQARD) and a cosine function to learn the flyby effects (Liu et al., 2021). In this new spatial framework, we test nine commonly-used covariance functions in addition to this sum function and elaborate on the three best options with the smallest error (Rasmussen and

Williams, 2006). Only the best three (based on first evaluations) are detailed here; the others will be mentioned briefly at the end of this section.

The first one is the Neural Network (NN) covariance function:

$$k_{\mathrm{NN}}(\boldsymbol{x},\boldsymbol{x}') = s^2_{f,\mathrm{NN}} \cdot \sin^{-1}\left(\frac{\boldsymbol{x}\boldsymbol{P}\boldsymbol{x}'^{\mathrm{T}}}{\sqrt{(1+\boldsymbol{x}\boldsymbol{P}\boldsymbol{x}'^{\mathrm{T}})(1+\boldsymbol{x}'\boldsymbol{P}\boldsymbol{x}^{\mathrm{T}})}}\right) \quad (18)$$

where $s^2_{f,\mathrm{NN}}$ is the signal variance. The matrix $\boldsymbol{P} = \eta \times I_5$ is a matrix of η times I_5, which represents an identity matrix of size five. Both $s^2_{f,\mathrm{NN}}$ and η are hyper-parameters to be optimized.

The second one is the Squared Exponential covariance function with Automatic Relevance Determination (SEARD):

$$k_{\mathrm{SEARD}}(\boldsymbol{x},\boldsymbol{x}') = s^2_{f,\mathrm{SEARD}} \cdot \exp\left(\frac{-(\boldsymbol{x}-\boldsymbol{x}')\cdot\boldsymbol{Q}\cdot(\boldsymbol{x}-\boldsymbol{x}')^{\mathrm{T}}}{2}\right) \quad (19)$$

where $s^2_{f,\mathrm{SEARD}}$ is the signal variance for this function. The matrix $\boldsymbol{Q}$ is explained below.

The third one is the aforementioned RQARD:

$$k_{\mathrm{RQARD}}(\boldsymbol{x},\boldsymbol{x}') = s^2_{f,\mathrm{RQARD}}\left(1+\frac{(\boldsymbol{x}-\boldsymbol{x}')\boldsymbol{Q}(\boldsymbol{x}-\boldsymbol{x}')^T}{2\alpha}\right)^{-\alpha} \quad (20)$$

Here $s^2_{f,\mathrm{RQARD}}$ and α are the signal variance and shape parameter, respectively.

The latter two covariance functions include an ARD term $\boldsymbol{Q}$, which is a symmetric matrix containing weights

$$\boldsymbol{Q} = \mathrm{diag}\left(\frac{1}{l_a^2},\frac{1}{l_e^2},\frac{1}{l_i^2},\frac{1}{l_\omega^2},\frac{1}{l_\phi^2}\right) \quad (21)$$

where l_a, l_e, l_i, l_ω and l_ϕ, which are called characteristic length-scale hyper-parameters, are weights for the elements a, e, i, ω and ϕ, respectively. This ARD term controls the similarity between samples. Therefore, these five input elements have a different influence on predicting the output of a particular test sample.

In order to show the improvement of using ARD, the performances of another two covariance functions Squared Exponential (SE) and Rational Quadratic (RQ) are also presented in Section 5.1. For these two functions without the ARD term, the terms $\boldsymbol{Q}$ in Eqs. 19 and 20 have a simplified form $\frac{1}{l^2}\times I_5$.

The other four covariance functions that were initially considered are Piecewise Polynomial, Matérn, and these two formulations using ARD (Rasmussen and Williams, 2006). As mentioned earlier, their performance was such that they were excluded from further analysis. It is particularly noteworthy that the sum covariance function is also excluded, which has a good performance for the planar CR3BP studied by the authors previously (Liu et al., 2021). It indicates that the characteristics of the mapping function have changed considerably from the planar CR3BP to the spatial one. A different covariance function is required for the new mapping function, i.e. Eq. 7.

4.3. Training samples

Training samples are necessary for the supervised learning method to provide empirical information about a system. In this paper, a training sample consists of a training input of five orbital elements and an output of the variation of a specific element due to a flyby (Eq. 7). For the GPR-GAM model, the inputs are generated randomly using the uniform distribution in the input space. The range of values of each dimension is different (Table 1) which would let the large-value input element dominate the results. To avoid this, the inputs are normalized to $[0,1]$.

Given a training input, the final state of the spacecraft is obtained by numerically propagating the motion as described by Eq. 1 over one unperturbed orbital period of the initial osculating orbit. The propagation employs an RK4 integration in MATLAB® 2018b (The Mathworks Inc., 2018) using a relative error tolerance of 1.0×10^{-9} and an absolute error tolerance of 1.0×10^{-12}, leading to uncertainties of 0.15 km (i.e. 1.0×10^{-7} % on the scale of the problem) after one orbital period of integration. Starting from the exterior Hill's region, it is possible for the spacecraft to be temporally trapped by the secondary. The training output, i.e. the variation of Keplerian orbital elements, is calculated by subtracting that input from the final osculating orbital elements of the spacecraft.

The training process then uses a conjugate gradient method to look for the optimal values of hyper-parameters maximizing the outcome of Eq. 12 (Rasmussen and Williams, 2006). Based on the theory of Bayesian inference, the GPR-GAM model has a high possibility of producing results close to the training outputs with optimal hyper-parameters. Training samples are also used by the GPR-GAM model when predicting the output for a new input $\boldsymbol{x}^*$, as in the matrix $\boldsymbol{X}$ denoted in Eq. 15.

4.4. Accuracy evaluation and training data size

To evaluate the performance of a GPR-GAM model, a set of test samples is generated. Following the same way of generating training samples, a group of 500 test samples T are generated and used. This set is kept constant for all further evaluations. Exact test outputs y_{CR3BP}'s (still, the effects δa etc. of the flyby) obtained by the numerical CR3BP propagation are employed as benchmark and compared with results $y_{\mathrm{GPR-GAM}}$'s obtained by GPR-GAM. The criterion to quantify their quality, we use the Mean Absolute Error (MAE), which is defined as:

$$\mathrm{MAE}(N) = \frac{1}{N^*}\sum_{i=1}^{N^*}\left|y_{\mathrm{CR3BP},i}(N) - y_{\mathrm{GPR-GAM},i}(N)\right| \quad (22)$$

This equation shows the MAE of N^* test samples when using N training samples.

Generally, a larger training dataset can be expected to decrease the MAE since more information is provided for the GPR-GAM model to capture the characteristics of the problem at hand. However, as shown in Eq. 15, more training samples will increase the burden of computation. It remains a problem to identify the optimal size of the training dataset due to the trade-off between accuracy and efficiency. Bouwman et al. (2019) defines a threshold ϵ_{MAE} for the fluctuation of the MAE. Increasing the number of training samples is stopped when ΔMAE becomes smaller than that particular threshold. For GPR-GAM, we improve this method so that it works for different types of output elements. When estimating the MAE of test data T with an increasing number of training samples, the training data size N starts from 100 and is increased with 100 each time.

A set of MAE's over a certain range of training data size is defined as:

$$E(N) = [\mathrm{MAE}(N), \mathrm{MAE}(N+n), \ldots, \mathrm{MAE}(N+9n)] \tag{23}$$

where $n = 100$. The fluctuation of MAE over this range is given as:

$$\Delta\mathrm{MAE}(N) = \max(E(N)) - \min(E(N)) \tag{24}$$

When the number of training samples for the GPR-GAM model meets the condition, i.e. ΔMAE is smaller than 5 % of the lowest MAE over the last 10 iterations:

$$\frac{\Delta\mathrm{MAE}(N)}{\min(E(N))} < 0.05, \tag{25}$$

The prediction converges at the point k_s:

$$k_s = \underset{k=1,\ldots,9}{\operatorname{argmin}}\, E(N) \tag{26}$$

The value N_s is selected as the final training data size:

$$N_s = N + n \times k_s \tag{27}$$

This criterion can be applied to every output element and ensures that an accurate GPR-GAM model with minimum training samples is obtained.

5. Performance of GPR-based gravity assist mapping

5.1. Different covariance functions

The performance of GPR-GAM is shown in Fig. 4 as a function of an increasing number of training samples, from 100 to 5000. Fig. 4 presents the MAE of test samples using five different covariance functions. The covariance functions of SE and RQ with the ARD strategy are shown in addition to those without ARD. For every output element, RQARD performs better than RQ. Also, SEARD obtains more accurate results than SE except for predicting $\delta\omega$. The results clearly demonstrate the positive effect of ARD. For predicting an output, the five input features play different roles in Eq. 7. The accuracy is improved by emphasizing the effects of some particular inputs using ARD. The RQARD function has the best accuracy with regards to all outputs. The NN function comes second except for quantifying $\delta\Omega$.

Using RQARD, the MAE tends to become stable when the training data size is larger than 3000. To better quantify this by applying the criterion of Section 4.4, the minimum stable training data size for the five outputs to converge are

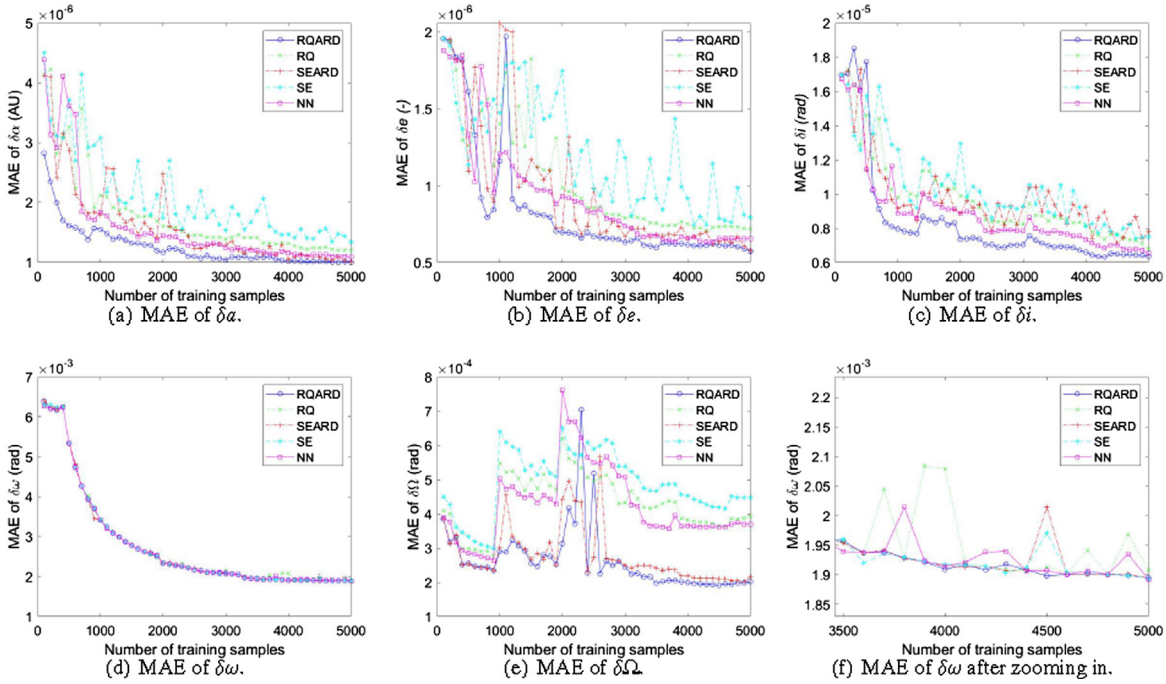

Fig. 4. The MAE of test samples when using five different covariance functions.

Table 2
The values of length-scales, signal variance and shape parameters using the RQARD covariance function. The subscript of the length-scale hyper-parameters stands for the order of influence of an input element on predicting a particular output using the GPR model.

	δa	δe	δi	$\delta\omega$	$\delta\Omega$
l_a^2	14.54_5	17.56_5	8.79_5	1.14_4	13.39_4
l_e^2	0.17_1	0.28_2	1.01_3	0.02_1	0.07_1
l_i^2	0.21_2	0.03_1	0.38_2	0.57_2	7.17_3
l_ω^2	0.66_3	4.06_4	0.04_1	2.25_3	26.15_5
l_ϕ^2	1.07_4	0.48_3	1.28_4	6.90_5	3.79_2
$s^2_{f,\mathrm{RQARD}}$	8.29	15.76	19.98	0.068	0.518
α	0.15	3.22	1.23	3.37	1.21
c	0.002	−0.001	0.001	0.03	0.0004

[4700,3500,4300,4200,4500]. Due to the complexity of the spatial CR3BP, significantly more training samples are required than for the planar case (Liu et al., 2021), which should not come as a surprise. Predicting eccentricity is the fastest to become stable. When increasing the number of training samples from 100 to 1000, the MAE of δa and δi decrease sharply. However the results tend to be really stable using more than 4000 training samples.

Inspecting Figs. 4(c)–(f), in particular the zoomed-in plot in Fig. 4(f), the conclusion can be drawn that $\delta\omega$ is the most difficult element to predict, in terms of level of convergence. Predicting $\delta\Omega$ is the second-most difficult thing to do (at a level of 1.0×10^{-5} rad), whereas δi is the most accurate one.

The full set of hyper-parameters of the mean and covariance functions is presented in Table 2. According to Eq. 21, the influence of an input element on predicting a particular output is inversely proportional to the absolute value of the corresponding length-scale. For a set of five length-scales in predicting a specific output, a lower value of a length-scale represents a higher influence of the corresponding input feature because all the inputs are normalized before training. For example, the eccentricity plays an important role in predicting $\delta a, \delta\omega$ and $\delta\Omega$. The predictions of δe and δi most rely on i and ω, respectively. The second-most influential input element is i ranking first in the prediction of δe, and second in that of $\delta a, \delta i$ and $\delta\omega$. This illustrates the importance of inclination in the flyby effects in a spatial CR3BP. The signal variance parameter $s^2_{f,\mathrm{RQARD}}$ represents the amplitude of the RQARD covariance function. The value of $s^2_{f,\mathrm{RQARD}}$ for ω is small, which means that the similarity between two samples is small even though their inputs are close to each other. This is supportive of the large values in the MAE results shown in Fig. 4. The shape parameter α controls how the similarity changes when a sample moves away from the nearest sample, together with the length-scale of course. The similarity is highest when two samples coincide. A larger α indicates that the similarity decreases fast when two samples are further separating. The prediction of $\delta\omega$ has a small $s^2_{f,\mathrm{RQARD}}$ and large α, which means an accurate prediction of $\delta\omega$ relies on the availability of training samples very close to the test samples. ω is a difficult parameter to determine and predict in view of some small values of eccentricity produced in the domain in Table 1, in particular when $e < 0.1$.

5.2. Comparison with semi-analytical method

For the same group of test samples, we compare the GPR-GAM model to the Semi-analytical Keplerian Map (SKM), which is called the Kick Map, developed by Alessi and Sánchez (2016). Since the Kick Map is known for working well outside the Hill sphere of the perturbing body, an additional comparison is performed for the case of $r_2 > 0.01$ AU. The performance of a Planar-GPR-GAM (P-GPR-GAM) model developed by the authors for the planar CR3BP is also presented (Liu et al., 2021). The P-GPR-GAM studies flyby effects in a three-parameter input space of a, e and ω.

In Table 3, the MAE and CPU time of these methods are presented. Note that the SKM statistics are based on an implementation of this technique on the first author's PC (Core i7 CPU and 8.00 GB RAM), and not on the numbers reported in Alessi and Sánchez (2016) itself; in this way platform-specific effects are eliminated. In the case of $r_2 > 6678$ km, the GPR-GAM has a better accuracy than the SKM for all outputs. The advantage is clear with regards to $\delta a, \delta e$ and δi. In particular, the MAE of δa by

Table 3
The performance of GPR-GAM compared to that of Alessi and Sánchez (2016) (fully reproduced here). The computational time is obtained by taking the mean value of 1000 repeated experiments. The value is the CPU time spent on one test sample. The CPU time of numerical integration of the equations of motion of CR3BP is listed in the last row.

MAE		δa [AU]	δe [–]	δi [rad]	$\delta\omega$ [rad]	$\delta\Omega$ [rad]
$r_2 > 6678$ km	GPR-GAM	5.7×10^{-5}	3.4×10^{-5}	6.3×10^{-6}	1.9×10^{-3}	1.9×10^{-4}
	SKM	1.9×10^{-4}	7.6×10^{-5}	1.1×10^{-5}	2.2×10^{-3}	2.1×10^{-4}
$r_2 > 0.01$ AU	GPR-GAM	3.3×10^{-5}	2.1×10^{-5}	3.8×10^{-6}	1.3×10^{-3}	0.8×10^{-4}
	SKM	7.3×10^{-5}	2.1×10^{-5}	5.9×10^{-6}	1.2×10^{-3}	1.0×10^{-4}
Time*[s]	GPR-GAM	5.9×10^{-6}	1.5×10^{-6}	3.8×10^{-6}	3.6×10^{-6}	5.2×10^{-6}
	SKM	7.5×10^{-2}	7.5×10^{-2}	7.5×10^{-2}	7.5×10^{-2}	7.5×10^{-2}
	P-GRP-GAM	1.2×10^{-6}	1.2×10^{-6}	-	1.2×10^{-6}	-
	CR3BP	5.9×10^{-1}	5.9×10^{-1}	5.9×10^{-1}	5.9×10^{-1}	5.9×10^{-1}

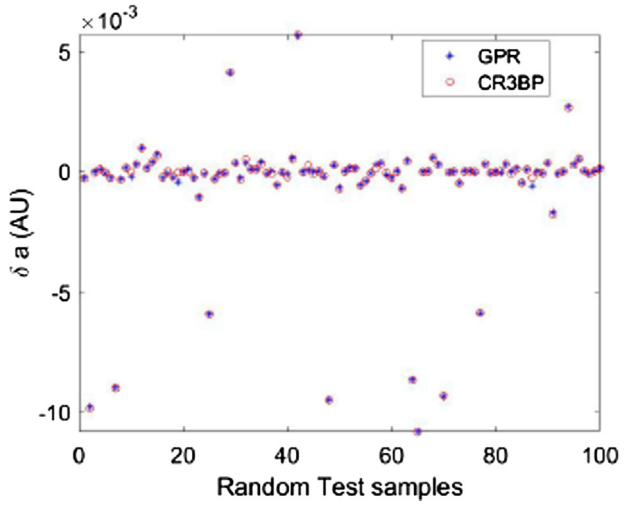

Fig. 5. The difference of predictions between the GPR-GAM and the numerical integration for 100 random test samples.

SKM is about 3.3 times that of using GPR-GAM. The efficiency of predicting δa is improved by a factor of 1.27×10^4. 7.5×10^{-2}s is the total time of SKM predicting five outputs of a single test sample. However, predicting these five outputs using GPR-GAM, which spends a total of 2.0×10^{-5}s, is still much faster than SKM. GPR-GAM has seen an improvement on the accuracy and efficiency over SKM. Note that the time in this table shows the prediction component only. The training time needed for a GPR-GAM model using 4000 training samples is 32 min on the first author's PC. The advantage of GPR-GAM is that it has to learn only once and can be easily applied to a next prediction.

In the case of $r_2 > 0.01$ AU, the accuracy of both SKM and GRP-GAM is improved due to simpler dynamical properties. The overall accuracy of SKM approximations increases by about a factor of two. With regards to $\delta a, \delta i$ and $\delta\Omega$, GPR-GAM still has better performance over SKM. SKM is more accurate for the prediction of $\delta\omega$. When the minimum distance between the spacecraft and the Earth is larger than the radius of the Hill sphere, the difference of MAE between these two methods decreases. In the planar case, P-GPR-GAM uses only 1500 training samples to reach a stable prediction. In addition to a lower dimension of the input space, using less training samples benefits the efficiency drastically.

Fig. 5 shows the difference of predictions between the GPR-GAM and the numerical integration method for the first 100 test samples. For most of the predictions, the GPR-GAM outputs almost overlap with those of the numerical integration. The maximum error is 3.3×10^{-4} AU (49,369 km) for the No.87 test sample, where an obvious separation is observed. However, this magnitude of error does not occur often. The minimum error within these 100 samples is only 4.7×10^{-8} AU (7 km). By converting the Keplerian orbital elements into the Cartesian coordinates in the ecliptic coordinate system centered at the Sun, the MAE of Cartesian distances between the GPR-GAM outputs and those of the numerical integration is calculated by

$$\mathrm{MAE}_{\mathrm{Car}} = \frac{1}{100}\sum_{i=1}^{100}|\mathrm{orb2Car}(\boldsymbol{y}_{\mathrm{CR3BP},i}) - \mathrm{orb2Car}(\boldsymbol{y}_{\mathrm{GPR-GAM},i})| \quad (28)$$

where $\boldsymbol{y}$ consists of all the Keplerian orbital elements of the spacecraft. orb2Car is a function converting $\boldsymbol{y}$ into Cartesian coordinates, and only the position components are used. When taking into account the error of predicting every Keplerian orbital element, $\mathrm{MAE}_{\mathrm{Car}}$ for these 100 test samples is 37,063 km.

5.3. Robustness when changing training dataset

To study the influence of the training samples on the accuracy, we generate another two groups of training samples in the same input space. These groups use different seed numbers to generate different (quasi) random samples. For all cases, the RQARD covariance function is selected to build the GPR-GAM model. For the same set of test samples, Fig. 6 shows the comparison of the GPR-GAM models trained by these three groups of training samples. The MAE drops sharply when going from 100 to 1000 samples and there is an obvious difference between datasets when using less than 3000 training samples. A steady decline begins at 3000 until the three curves converge using more than 4000 training samples. An eventual close agreement between three groups ensures the robustness of prediction using GPR-GAM.

The prediction of δe converges and stabilizes fastest, which supports the observation of using the least number of training samples in Section 5.1. In Fig. 6(c), for inclination, the results of the three datasets converge slowly after going through a gap of 0.5×10^{-5}rad. A minimum dataset of 4000 training samples is required to obtain a stable performance of δi. In Fig. 6(d), the behavior of $\delta\omega$ using three datasets has an insignificant difference compared to the other output elements, but the zoomed-in plot (Fig. 4(f)) shows a more clear difference. Predicting $\delta\Omega$ is also difficult for GPR-GAM in view of the significant fluctuation in Fig. 6(e). Using different groups of training samples, strong fluctuations can be observed between 1000 and 3000 training samples. $\delta\Omega$ has relatively larger values due to a flyby than those of δi and $\delta\omega$, which is a cause of this fluctuation. Increasing the number of training samples is necessary to stabilize the MAE.

The results are summarised in Table 4. To quantify the consistency of the MAE outcome, the Percentage of Deviation (PD) is calculated by subtracting the lowest value from the highest value, and then dividing the result by the highest value. Quantifying $\delta\omega$ with different training datasets has the lowest PD. The accuracy of evaluating $\delta\Omega$ is affected more seriously than the other orbital ele-

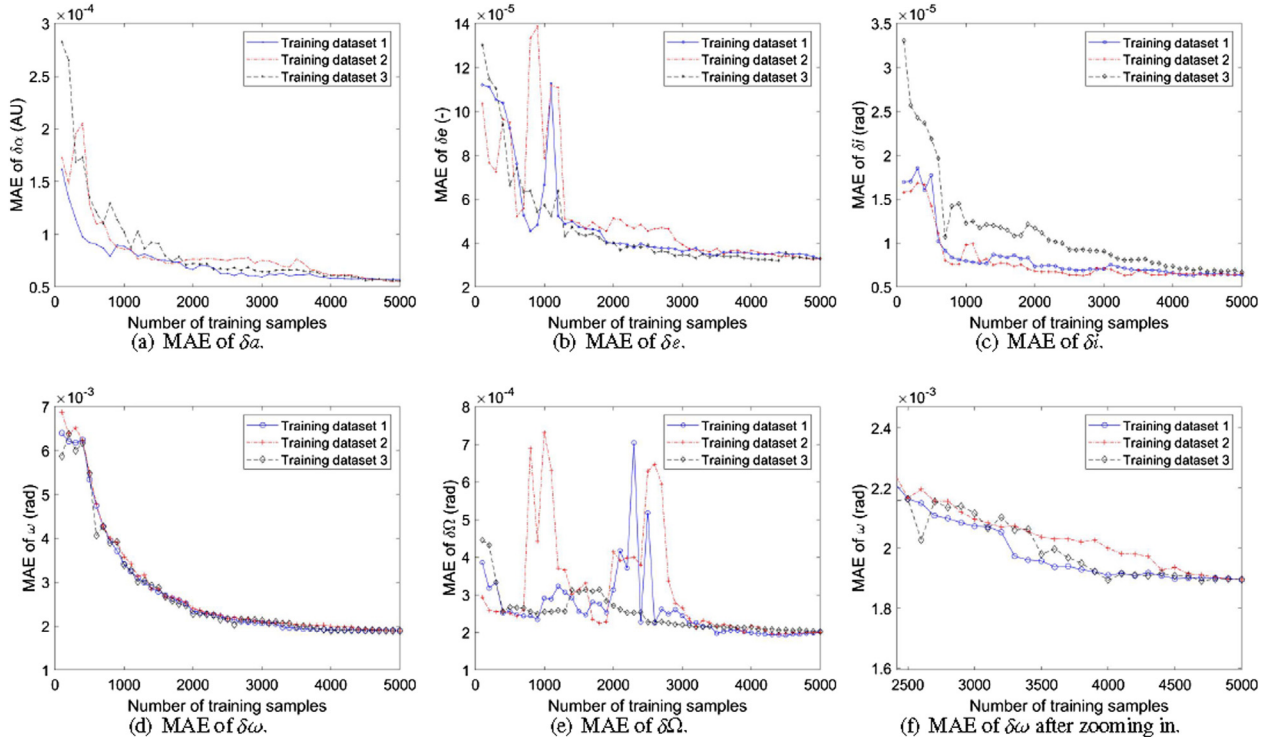

Fig. 6. The MAE of test samples when using three different training datasets.

Table 4
The performance of GPR-GAM over three different training datasets using the same test dataset. The values of MAE and PD are shown.

	δa [AU]	δe [–]	δi [rad]	$\delta\omega$ [rad]	$\delta\Omega$ [rad]
Group 1	5.74×10^{-5}	3.42×10^{-5}	6.32×10^{-6}	1.89×10^{-3}	1.91×10^{-4}
Group 2	5.56×10^{-5}	3.49×10^{-5}	6.34×10^{-6}	1.90×10^{-3}	1.96×10^{-4}
Group 3	5.64×10^{-5}	3.56×10^{-5}	6.68×10^{-6}	1.89×10^{-3}	2.02×10^{-4}
PD	3.14%	3.93%	5.38%	0.53%	5.44%

Table 5
The performance of GPR-GAM over three different test datasets using the same training samples. The values of MAE and PD are shown.

	δa [AU]	δe [–]	δi [rad]	$\delta\omega$ [rad]	$\delta\Omega$ [rad]
Group 1	5.74×10^{-5}	3.42×10^{-5}	6.32×10^{-6}	1.89×10^{-3}	1.91×10^{-4}
Group 2	5.78×10^{-5}	3.40×10^{-5}	6.39×10^{-6}	1.84×10^{-3}	1.89×10^{-4}
Group 3	5.75×10^{-5}	3.38×10^{-5}	6.24×10^{-6}	1.85×10^{-3}	1.93×10^{-4}
PD	0.69%	1.17%	2.35%	2.65%	2.07%

Table 6
The performance of GPR-GAM over the various energy domains. The first line represents the low-energy domain ($C_J > C_{L3}$), whereas the second line represents the full domain.

	δa [AU]	δe [–]	δi [rad]	$\delta\omega$ [rad]	$\delta\Omega$ [rad]
low	3.71×10^{-5}	4.28×10^{-5}	8.08×10^{-6}	2.56×10^{-3}	2.31×10^{-4}
full	5.7×10^{-5}	3.4×10^{-5}	6.3×10^{-6}	1.9×10^{-3}	1.9×10^{-4}

ments. However, the absolute difference between different groups is small and the convergence is good as shown in Fig. 6(e).

5.4. Robustness when changing test dataset

To evaluate the robustness of GPR-GAM when predicting different test datasets, we generate another two groups of test samples using different random seed numbers. Akin to the group used before, each group has 500 samples. The original training data set and the RQARD covariance function are used for building the GPR-GAM model. By applying the criterion of Section 4.4 to the results in Fig. 4, the training data size is set at [4700, 3500, 4300, 4200, 4500] for each orbital element. The performance of these three groups is presented in Table 5, where the first line obviously refers to results already presented in Tables 3 and 4. The PD is calculated for these parameters. These PD's are smaller than those in Table 4, except for $\delta\omega$. The GPR-GAM model has a better robustness when changing test datasets than that when changing training datasets. The results for δa have the smallest PD which is only 0.69%. This suggests that predicting δa using GPR-GAM has the best generalization (Rasmussen and Williams, 2006). For every element, the overall PD's between groups is lower than 3.0%. The results demonstrate that the GPR-GAM model has a stable performance when predicting the flyby effects for a new given initial condition.

5.5. Low-energy cases

The computations so far sample the input space in a uniform way, irrespective of the energy level. One could argue that the high-energy part of the domain does not really need a three-body formulation, and that the low-energy part of the domain is the more challenging. In order to analyze the specific application of the GPR-GAM model to low-energy transfers, we generate a group of 500 test samples with $C_J > C_{L3}$. According to the initial condition defined by the input space, the spacecraft starts from somewhere in the exterior Hill's region. This means that the following situation is not considered when the Jacobi constant is smaller than that of C_{L1}: the spacecraft is located initially in the vicinity of either the Sun or the Earth and travels to the neighbourhood of the other. However, it does allow the cases of the spacecraft starting from the exterior realm and moving into the neighbourhood of the Earth when $C_{L2} > C_J > C_{L3}$. The predictions are obtained by using the training dataset 1 and the corresponding hyperparameters shown in Table 2, so for the model that is based on the entire energy range. The MAE for the low-energy part of the domain are shown in Table 6.

The MAE's remain at the same level of magnitude as those of the entire Jacobi energy cases in Table 4, with a slight degradation of about $20-30\%$ (except for δa). This is mainly because of the more complex variation of orbital elements due to the higher chaotic dynamics, particularly when $C_{L2} > C_J > C_{L3}$.

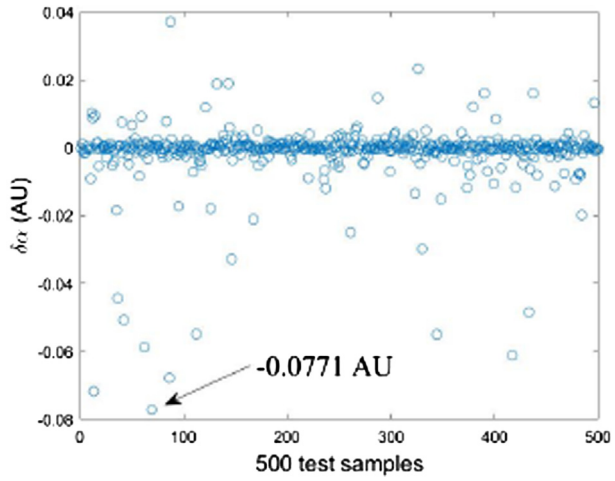

Fig. 7. The variation of the semi-major axis induced by a flyby for 500 test samples in the low-energy regime.

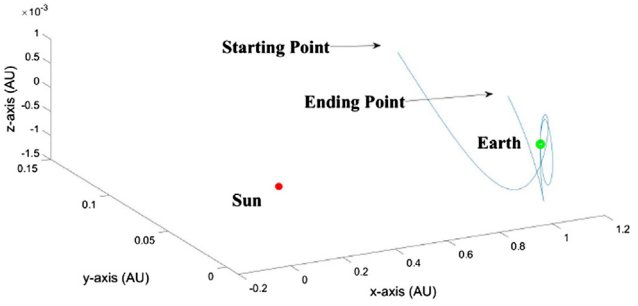

Fig. 8. Trajectory of a test sample ($C_J = 3.00057$). In order to show the trajectory properly in three dimensions, the scales of three axes are not equal.

Using the numerical CR3BP propagation, the variation of the semi-major axis induced by a flyby for 500 test samples in the low-energy regime is shown in Fig. 7. Although the majority of δa results have an absolute value of about 1.0×10^{-3} AU, a significant percentage of them are larger than 0.01 AU in absolute terms. The initial conditions with a smaller phasing angle would generally result in a larger variation of the semi-major axis. The largest change among these is -0.0771 AU. In the rotating reference frame, the trajectory of this case ($C_J = 3.00057$) is shown in Fig. 8. The osculating Keplerian elements of its initial condition $[a, e, i, \omega, \phi, \theta]$ are $[1.028\text{AU}, 0.008, 0.086°, 0.932°, -0.524°, -180°]$.

The value of C_J meets the condition $C_{L2} > C_J > C_{L3}$. The interior and exterior Hill's region are connected which allows a spacecraft motion between these two realms. Over one unperturbed orbital period computed from the initial condition, the spacecraft travels from the exterior Hill's region into the neighbourhood of the Earth temporarily before moving out. The closest distance between the space-

Table 7
The prediction of the test sample ($C_J = 3.00057$).

	δa [AU]	δe [–]	δi [deg]	$\delta\omega$ [deg]	$\delta\Omega$ [deg]
Numerical integration	−0.0771	0.0470	20.1637	−20.2493	9.0656
GPR-GAM	−0.0773	0.0464	20.1127	−20.4192	8.9913
PD	0.26 %	1.28 %	0.25 %	0.85 %	0.82 %

craft and the center of the Earth is 7223 km. The variation of the Keplerian elements and the predictions obtained by GPR-GAM are shown in Table 7. The predictions are well approximated especially for δa and δe. The prediction of $\delta i, \delta\omega$ and $\delta\Omega$ yield poorer accuracy compared to the MAE in Table 6, but the relative error is small compared to the large variation of these elements. It is worth noting that the value of the semi-major axis of the final condition is 0.951 AU, which is beyond the predefined input space in the current work. Extending the input space to the initial conditions of starting from the interior Hill's region is of interest for future work.

6. Conclusions

A Gravity Assist Mapping was developed based on Gaussian Process Regression for evaluating flyby effects in the fully spatial CR3BP. A new criterion was proposed to select the number of training samples for every orbital element. Due to the increasing complexity of a fully spatial CR3BP compared to that of the planar one, a larger training dataset is required. Compared to a previous semi-analytical method, GPR-GAM has achieved a better accuracy. The CPU time for prediction is a factor 10^3 faster than that of this semi-analytical method. The domain of applicability is extended beyond that of the Keplerian Map.

According to the MAE, the RQARD covariance function performs best. Based on the optimized values of the length-scale, the influence of different input elements on flyby effects was discussed. The stable robustness of GPR-GAM was illustrated by changing the training and test datasets. It indicates that quantifying δa has the best generalization property. The prediction of $\delta\omega$ is more difficult than that of the other output elements. Further attention needs to be paid to the various ranges of output. The results show the ability of the GPR-GAM model to predict the dynamics of the CR3BP system with relatively high accuracy compared to SKM. The GPR-GAM model has been shown to be a versatile tool that can be applied for a wide range of energy levels. The quality of predictions in the low-energy domain is almost comparable to that in the high-energy part.

The input space considers only the initial conditions of the spacecraft starting from the exterior Hill's region, and the time of flight is limited to one orbital period of the initial osculating orbit. The periapsis passage of the initial osculating orbit is located in the same region. The phenomenon of the spacecraft temporarily being captured by the secondary is observed for this input space, which is not amenable to a patched-conics technique. However, future work will consider extension of the input space to include more complex situations. The angular phasing and the longitude of the ascending node are used in the input and the output vectors, respectively. The flexibility of the GPR technique allows the mapping function to have different formulations. In order to explore the applications of GPR-GAM, four subjects are of interest for further investigation: a) using an alternative set of orbital elements; b) employing the orbital elements in the rotating reference frame; c) extending the time of flight to multiple orbital periods, or using a Poincaré section as boundary conditions, such that more complex dynamics is involved and more CPU time could be saved on predicting the status of the spacecraft; d) establishing GPR-GAM for different CR3BP systems considering the various goals of space missions, e.g. SMART-1 (Moon) or exploration missions to Jupiter or Saturn (Sánchez et al., 2015). The GPR technique is an extremely interesting modelling option, in view of the CPU time that can be gained over techniques that fully rely on numerical integration, satisfying the same accuracy requirements.

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.

Acknowledgement

The author Liu gratefully acknowledges the China Scholarship Council (CSC) for its financial support.

References

Alessi, E., Sánchez, J., 2016. Semi-Analytical Approach for Distant Encounters in the Spatial Circular Restricted Three-Body Problem. J. Guidance Control Dyn. 39 (2), 351–359. https://doi.org/10.2514/1.G001237.

Bate, R.R., Mueller, D.D., White, J.E., 1971. Fundamentals of astrodynamics. Courier Corp., 279–292

Beutler, G., 2004. Methods of Celestial Mechanics: Volume I: Physical, Mathematical, and Numerical Principles. Springer Science & Business Media, pp. 152–174.

Bouwman, L., Liu, Y., Cowan, K., 2019. Gaussian Process models for preliminary low-thrust trajectory optimization. In: Astrodynamics Specialist Conference, Portland, ME, vol. 171, no. 873, pp. 3687–3706.

Breen, P.G., Foley, C.N., Boekholt, T., Zwart, S.P. 2019. Newton vs the machine: solving the chaotic three-body problem using deep neural networks. arXiv, 000, 1–6.

Broucke, R., 1988. The Celestial Mechanics of Gravity Assist. In: AIAA/AAS Astrodynamics Conference, Minneapolis, Minnesota.

Campagnola, S., Boutonnet, A., Schoenmaekers, J., Grebow, D.J., Petropoulos, A.E., Russell, R.P., 2014. Tisserand-leveraging transfers. J. Guid. Control Dyn. 37 (2), 1205–1222.

Campagnola, S., Skerritt, P., Russell, R.P., 2012. Flybys in the planar, circular, restricted, three-body problem. Celestial Mech. Dyn. Astron. 113 (3), 343–368.

Chirikov, R.V., Vecheslavov, V.V., 1989. Chaotic dynamics of comet Halley. Astron. Astrophys. 221, 146–154.

Conley, C.C., 1968. Low energy transit orbits in the restricted three-body problems. SIAM J. Appl. Math. 16 (4), 732–746. https://doi.org/10.1137/0116060.

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

Krige, D., 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. J. South Afr. Inst. Min. Metall. 52 (6), 119–139.

Liu, Y., Noomen, R., Visser, P., 2021. A Gravity Assist Mapping Based on Gaussian Process Regression. J. Astronaut. Sci. 68, 248–272

Longuski, J.M., Williams, S.N., 1991. Automated design of gravity-assist trajectories to mars and the outer planets. Celestial Mech. Dyn. Astron. 52 (3), 207–220.

Malyshkin, L., Tremaine, S., 1999. The Keplerian map for the planar restricted three-body problem as a model of comet evolution. Icarus 141 (2), 341–353. https://doi.org/10.1006/icar.1999.6174.

Neves, R., Sánchez, J.P., Colombo, C., Alessi, E.M., 2018. Analytical and Semi-Analytical Approaches to the Third-Body Perturbation in Nearly Co-Orbital Regimes. In: 69th International Astronautical Congress, Bremen, Germany, pp. 1–10.

Peñagaricano, M.O., Scheeres, D.J., 2010. A Perturbation Theory. Acta Astronaut. 67 (2), 27–37. https://doi.org/10.1016/j.actaastro.2009.11.011.

Petrosky, T., Broucke, R., 1987. Area-preserving mappings and deterministic chaos for nearly parabolic motions. Celestial Mech. Dyn. Astron. 42 (1), 53–79. https://doi.org/10.1007/BF01232948.

Rasmussen, C., Williams, C., 2006. Gaussian processes for machine learning. Massachusetts Institute of Technology Press, pp. 7–31, 79–128.

Ross, S., Scheeres, D., 2007. Multiple gravity assists, capture, and escape in the restricted three-body problem. IAM J. Appl. Dyn. Syst. 6 (3), 576–596. https://doi.org/10.1137/060663374.

Sánchez, J.P., Colombo, C., Alessi, E., 2015. Semi-analytical perturbative approaches to third body resonant trajectories. In: Proceedings of the 66th International Astronautical Congress, International Astronautical Federation (IAC), Jerusalem, Israel, p. IAC-15.

Shang, H., Liu, Y., 2017. Assessing Accessibility of Main-Belt Asteroids Based on Gaussian Process Regression. J. Guid. Control Dyn. 40 (5), 1144–1154. https://doi.org/10.2514/1.G000576.

Strange, N., Longuski, J., 2002. Graphical method for gravity-assist trajectory design. J. Spacecraft Rockets 39 (1), 9–16.

Szebehely, V., 1967. Theory of Orbits. Academic Press, New York. https://doi.org/10.1016/B978-0-12-395732-0.X5001-6.

The Mathworks Inc., 2018. MATLAB version 9.5.0.944444.

Zhou, J.L., Sun, Y.S., Zheng, J.Q., Valtonen, M.J., 2000. The transfer of comets from near-parabolic to short-period orbits: map approach. Astron. Astrophys. 364, 887–893.