Delft University of Technology

A Gravity Assist Mapping Based on Gaussian Process Regression

Liu, Yuxin; Noomen, Ron; Visser, Pieter

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

A Gravity Assist Mapping Based on Gaussian Process Regression

Yuxin Liu[1] · Ron Noomen[1] · Pieter Visser[1]

Abstract

We develop a Gravity Assist Mapping to quantify the effects of a flyby in a two-dimensional circular restricted three-body situation based on Gaussian Process Regression (GPR). This work is inspired by the Keplerian Map and Flyby Map. The flyby is allowed to occur anywhere above 300 km altitude at the Earth in the system of Sun-(Earth+Moon)-spacecraft, whereas the Keplerian map is typically restricted to the cases outside the Hill sphere only. The performance of the GPR model and the influence of training samples (number and distribution) on the quality of the prediction of post-flyby orbital states are investigated. The information provided by this training set is used to optimize the hyper-parameters in the GPR model. The trained model can make predictions of the post-flyby state of an object with an arbitrary initial condition and is demonstrated to be efficient and accurate when evaluated against the results of numerical integration. The method can be attractive for space mission design.

Keywords Gravity assist · Circular restricted three-body problem · Gaussian process regression

✉ Yuxin Liu
Yuxin.Liu@tudelft.nl

Ron Noomen
R.Noomen@tudelft.nl

Pieter Visser
P.N.A.M.Visser@tudelft.nl

[1] Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, The Netherlands

Introduction

In the past decades, valuable insight about celestial bodies in our Solar System has been gained using spacecraft, such as Jupiter visited by Galileo [1]. Provided the payload capacity of launch vehicles is sufficient, it is possible to fly a direct transfer to the above destination. However, it is not necessarily the first choice due to the required propellant mass. In order to reduce the propellant consumption, typically a gravity assist (GA) technique is used. For instance, a combination of flybys at Venus and Earth was performed by Galileo before arriving at the final target Jupiter. In the preliminary design, the patched conics approach is commonly used to investigate flybys, which splits a multi-body problem into a succession of two-body problems [2–5]. This model has a fundamental assumption: the trajectory of the spacecraft is driven by one celestial body only. In addition to traditional GA's, distant encounters can be used to construct low-energy trajectories between the moons of giant planets [6–8]. However, the patched conics model cannot be used to study low-energy trajectories or describe the phenomena such as the invariant manifolds. The Circular Restricted Three-Body Problem (CR3BP) framework is typically used to conduct these studies [9–11].

Unlike the patched-conics model, the secondary keeps affecting the spacecraft in the CR3BP model. Distant encounters can be modelled in the CR3BP. However, there is no clear boundary condition for such a flyby because the motion of the spacecraft is always affected by the secondary. The Keplerian map is a useful tool to study the flyby effects in the CR3BP. This concept was first introduced by Petrosky and Broucke to study the chaotic behavior of comets near Jupiter [12]. They studied perturbation theory in the CR3BP framework to investigate the energy change of a particle with negligible mass after each flyby. The Keplerian map was further improved by Shevchenko with quite elementary methods [13]. Ross and Scheeres exploited the concept of the Keplerian map to investigate semi-analytically high-altitude flybys in the Planar Circular Restricted Three-Body Problem (PCR3BP) [6]. It can predict well the variation of Keplerian energy during a flyby by calculating quadratures of energy-kick functions. However, the assumption of constant orbital elements using Picard's method fails if the spacecraft is very close to the secondary. It focuses only on the cases of a Jacobi constant around 3.0 and has poor accuracy for high three-body energies. Later, the Keplerian map for the fully spatial case was developed for the initial-value problem, which allows for applications to low-energy spacecraft mission design [14]. In order to model the variation of the orbital elements of a massless particle in the CR3BP, Alessi and Sánchez developed a type of Keplerian map, called Kick map, based on semi-analytical formulations in the spatial CR3BP [15]. It is worth noting that the integration of this formulation is done numerically, which requires a computational effort similar to that of the propagation in the CR3BP. Moreover, this estimation is prone to higher error when the massless particle reaches a distance to the secondary smaller than the Hill radius (typically, about 0.01 AU in the Sun-(Earth+Moon)-spacecraft system). Motivated by the Keplerian map and the Tisserand-Poincaré graph, Campagnola et al. developed a flyby map to study flybys in the PCR3BP [16]. The flyby map has a wider domain of applicability than the Keplerian map. It found two types of flybys and illustrated that the direct type

exists at all energies and is more efficient than the retrograde type. The GPR-based Gravity Assist Mapping aims at developing an accurate and efficient method of quantifying the flyby effects. This method holds the advantage of a prominent accuracy at the cost of a low computational efficiency, because the state of the particle must not be integrated during the encounter.

Inspired by the Keplerian map and the flyby map, we develop a Gravity Assist Mapping (GAM) based on Gaussian Process Regression (GPR), a machine-learning technique. The method is used for quantifying flyby effects in the PCR3BP. It works for a broad range of energy levels, i.e. beyond the domain of applicability the Keplerian map, and requires smaller computational cost.

In this research, the dynamical model of a PCR3BP is presented first in Section Planar Circular Restricted Three-Body Problem. In Section Gravity Assist Mapping, it is described how the GPR model is to learn the features of flybys from training samples generated by PCR3BP equations of motion. The covariance function, the key module of the GPR model, is improved based on the analysis of flyby dynamics. The number and distribution of training samples are selected by taking the Root-Mean-Square Error (RMSE) and the Mean-Absolute-Percentage Error (MAPE) of test samples as criterion. In Section Performance of GPR-based Gravity Assist Mapping, the performance of the mapping technique is assessed. By training the model with the information provided by hundreds of training samples, the GPR-based approach can accurately capture the features of flybys. Finally, the conclusions are drawn in Section Conclusions.

Planar Circular Restricted Three-Body Problem

In the preliminary design of a space mission, simplified dynamical models are commonly used to reduce the computational effort. In this research, the PCR3BP model is used for constructing GA trajectories.

The PCR3BP assumes a system with three bodies: a massless particle and two massive primaries P_1 and P_2 with mass M_1 and M_2, respectively. Generally, we refer to body P_2 with the smaller mass as the secondary. The motion of the particle is driven by the simultaneous gravitational pull of both primaries. The primaries are assumed to rotate around the barycenter of the system in circular orbits. The orbital plane of these two massive bodies is taken as the reference plane. Characteristic quantities are commonly defined such that a normalized model can be applied to any three-body system.

In the rotating reference frame, which has the barycenter as origin, the frame rotates at the same angular velocity as the primaries do around this barycenter. The equations of motion in non-dimensional units for the particle (spacecraft) are given as [17]

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

where $\mu = M_2/(M_1 + M_2)$ represents the mass parameter, and r_1 and r_2 the normalized distances between the particle and the primary and secondary respectively.

In this paper, the Sun-(Earth+Moon)-spacecraft system is used with a mass ratio of 3.036×10^{-6}. Obviously, Equation 1 is analytically unsolvable. Moreover, the trajectory solutions are very sensitive to the initial conditions. This leads to the difficulty of assessing flyby effects accurately. Given an initial condition, the flyby effects can be calculated using the numerical propagation of the CR3BP equations of motion.

Gravity Assist Mapping

GPR is a regression method to estimate the relation between inputs and outputs in terms of a specific model. Krige first developed GPR for the estimation of the distribution of gold [18]. For astrodynamics purposes, GPR was adopted to efficiently assess the accessibility of main-belt asteroids and found thousands of potential mission targets [19]. It has also been used for modelling the gravity field of small bodies or the optimization of low-thrust trajectories [20, 21]. GPR is able to learn effectively the correlation between inputs and outputs of a process. A GPR-based method will be developed here to evaluate the post-flyby state of the spacecraft in a PCR3BP system.

GPR Fundamentals

As a supervised learning tool, GPR is used to predict the output at arbitrary input based on empirical information. Such information is obtained by observing the output of a group of independent input data in a specific input space. In this section, we will discuss the elementary aspects of GPR; the interested reader is referred to [22]. In this work, both input and output are the Keplerian elements of the spacecraft state in an inertial reference frame. The origin is centered at the largest primary and all the bodies are contained in the X-Y plane. The connection between the state of the spacecraft before flyby and after flyby can be described by a mapping function:

$$F : [a_A, e_A, \omega_A] \mapsto [a_B, e_B, \omega_B] \tag{2}$$

where a is the semi-major axis, e the eccentricity and ω the argument of periapsis. Subscripts A and B denote before- and post-flyby, respectively. Each of the orbital elements represents a feature of the state vector. One set of input vector and corresponding output vector comprises a so-called sample. In the inertial frame, Fig. 1 shows the osculating orbital elements of the spacecraft at the initial condition. The frame is centred at the primary and the X-axis points to the vernal equinox. Two assumptions are made for these initial conditions: the spacecraft starts from the apoapsis (i.e. $\theta = -\pi$); the initial true anomaly of the secondary θ_{P2} is selected such that it arrives at the negative X-axis when the spacecraft (in terms of initial osculating orbit) reaches periapsis. The time span before the secondary reaches the negative X-axis is set to be half of the orbital period of the initial osculating orbit of the spacecraft. Therefore, the relationship of phasing between the spacecraft and the secondary is described by the argument of periapsis ω only. When $\omega = 180°$, the minimum distance between the particle and the secondary happens at the periapsis passage and a flyby effect can be expected (See Appendix A for more details).
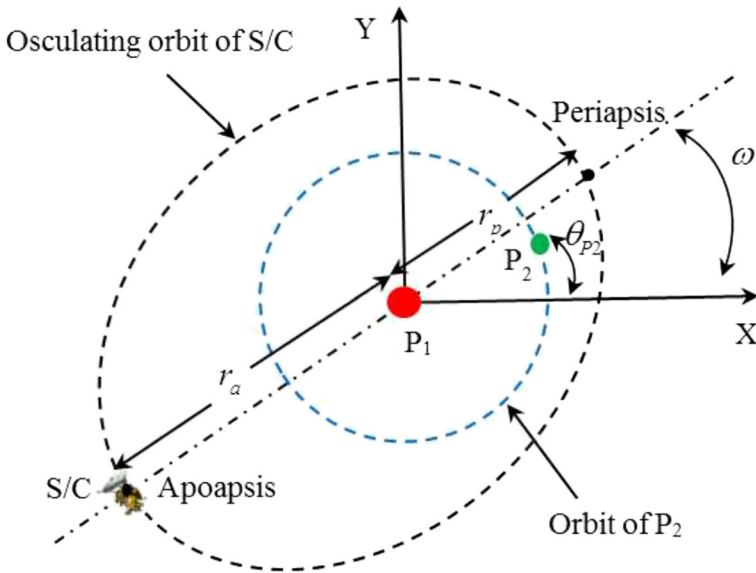
Fig. 1 Osculating orbit of the spacecraft in the inertial frame centered at the primary

The output will be predicted by GPR using the general function f

$$f : x \mapsto y \tag{3}$$

where $x = [a_A, e_A, \omega_A]$ is a vector, and y represents one of the variables of the post-flyby state $[a_B, e_B, \omega_B]$. Obviously, the dimension of the output vector determines the number of GPR models. In our case, the output vector consists of three elements, so three GPR models are required to build the Gravity Assist Mapping.

The method of Bayesian inference is used to predict the value of the output y with respect to the input x. This procedure works on a Gaussian process which is basically an infinite collection of random variables. Selecting an arbitrary finite number of this collection can form a joint Gaussian distribution [22]. For the GPR method, the random variable is taken as the value of the function $f(x)$. The Gaussian process based on the function $f(x)$ is specified by [23]

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{4}$$

where x' is an input vector in the input space. $k(x, x')$ represents the covariance function if $x \neq x'$ and the variance of x if $x = x'$. $m(x)$ is the mean function.

Commonly, the prior value for $m(x)$ is taken to be zero before obtaining more information about the mapping. After training the model, the posterior value will be inferred, which is not confined to be zero [22]. Therefore, the module $k(x, x')$ becomes the only determinant of the process. The information about the Gaussian process in Equation 4 can be gained by observing flyby effects for known inputs. The observation of an output (one feature of the post-flyby state) for a given input (before-flyby state) is actually generated using the propagation in the PCR3BP model. The

input and corresponding output compose a so-called training sample. Assuming we have N training samples, the training dataset D is defined as

$$D = \{(X, Y)|X = [x_1, x_2, \ldots, x_N], Y = [y_1, y_2, \ldots, y_N]\} \tag{5}$$

where X and Y are the training input matrix (the states before flyby) and the output vector (the selected feature of the post-flyby states), respectively. The output predicted by the GPR is assumed to differ from the ground truth by a value that follows a Gaussian distribution. Given the training inputs, the joint Gaussian distribution of outputs obtained from the GPR is:

$$Y \sim \mathcal{N}(0, K(X, X)) \tag{6}$$

where K is the covariance matrix consisting of N^2 covariance functions, written as

$$K(X, X) = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \cdots & k(x_N, x_N) \end{pmatrix} \tag{7}$$

Assuming that y^* is the prediction for a new input vector x^*, then the training outputs Y and the predicted output y^* form a joint Gaussian distribution:

$$\begin{pmatrix} Y \\ y^* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K(X, X) & K(X, x^*) \\ K(x^*, X) & k(x^*, x^*) \end{pmatrix}\right) \tag{8}$$

where $K(X, x^*)$ represents the vector of N covariance functions calculated at each entry of training inputs X and new input x^*. Clearly, because of symmetry, $K(x^*, X) = K(X, x^*)^T$.

By conditioning the joint Gaussian distribution in Equation 8 on the observations, the predicted distribution of output y^* has a Gaussian form and is completely specified by the mean $\mu(y^*)$ and the covariance $cov(y^*)$.

$$y^*|x^*, X, Y \sim \mathcal{N}(\mu(y^*), cov(y^*)) \tag{9}$$

The values of $\mu(y^*)$ and $cov(y^*)$ can be evaluated by

$$\mu(y^*) = K(X, x^*)^T K(X, X)^{-1} Y \tag{10}$$

and

$$cov(y^*) = k(x^*, x^*) - K(X, x^*)^T K(X, X)^{-1} K(X, x^*) \tag{11}$$

Generally, the expected value $\mu(y^*)$ can be taken as the predicted output y^*. Once a GPR model has been built, the prediction can be obtained by matrix multiplication of the $N \times 1$ vector $K(X, x^*)$, the $N \times N$ matrix $K(X, X)^{-1}$ and the $1 \times N$ vector Y. Equation 10 can be further simplified into:

$$y^* = K(X, x^*)^T Q_{train} \tag{12}$$

where $Q_{train} = K(X, X)^{-1}Y$ is precalculated. The prediction efficiency improves about N times after this simplification. When performing the evaluations for N^* new inputs, the outputs are obtained by

$$Y^* = K(X, X^*)^T Q_{train} \tag{13}$$

where $Y^* = [y_1^*, y_2^*, ..., y_{N^*}^*]$ is a vector of predicted outputs and $K(X, X^*)$ is an $N^* \times N$ matrix.

GPR Covariance Function

The covariance function describes the relation between samples by evaluating a function in terms of inputs. It is the most basic and essential module for any GPR model. The covariance functions can have various mathematical expressions and parameters which are called hyper-parameters. A perfect covariance function contributes to obtaining an exact representation of mapping between input and output. In addition, the generalization of a GPR model depends heavily on the choice of the covariance function. Unfortunately, a universal rule is not available for such a choice [24]. In our research, it needs to capture the dynamics of flybys in the PCR3BP. Two criteria are considered to evaluate the performance of the covariance function. One criterion is to minimize the cost function:

$$f_{cost} = -\log p(y|x) \tag{14}$$

where the marginal likelihood $p(y|x)$ describes the probability of obtaining the true training output given a training input. This process is equivalent to maximize the following function [22]:

$$\log p(y|x) = -\frac{1}{2}y^T K^{-1} y - \frac{1}{2}\log |K| - \frac{N}{2}\log 2\pi \tag{15}$$

Equation 15 is also the objective function to optimize the hyper-parameters using training samples. The value of $\log p(y|x)$ is always negative. A $\log p(y|x)$ closer to zero means that the possibility of observing true outputs is higher. Another criterion is to minimize the difference between the numerically propagated outputs of the PCR3BP and the predictions by the GPR. The differences are evaluated on test samples with inputs generated randomly in the input space.

The covariance function selected here initially is the Rational Quadratic Automatic Relevance Determination (RQARD) function [25]. The classical RQ covariance function is stationary, i.e. it depends on the distances between sample points in Euclidean space only. The different influence of orbital elements on the flyby effect cannot be reflected by the RQ covariance function. To distinguish these effects, the Automatic Relevance Determination (ARD) distance measure is integrated into the RQ covariance function. Three specific weights l_a, l_e and l_ω (Equation 17 below) will be optimized based on the sensitivity of the flyby effects to the corresponding orbital elements. The optimized value of the weight is a reflection of the influence of that particular feature. The RQARD function is defined as:

$$k_{\text{RQARD}}(x, x') = \sigma \, exp \left(1 + \frac{(x - x')^T Q(x - x')}{2\alpha}\right)^{-\alpha} \tag{16}$$

Here σ and α are the magnitude parameter and shape parameter of the output distribution, respectively. Q is a symmetric matrix containing weights

$$Q = \text{diag}\left(\frac{1}{l_a^2}, \frac{1}{l_e^2}, \frac{1}{l_\omega^2}\right) \tag{17}$$

where l_a, l_e and l_ω are weights for the elements a, e and ω respectively. Clearly, the RQARD formulation emphasizes the influence of near-by training samples, whereas samples further away have less influence.

Some covariance functions can interpret specific model characteristics, such as a cosine or periodicity. For this research, a new covariance function is built by taking the influence of the initial argument of periapsis ω_A on the variation of the semi-major axis a during a flyby into consideration. ω_A turns out to be the most influential parameter because it specifies the phasing between the massless particle and the secondary [6]. Under the predefined boundary, Fig. 2 shows a typical flyby effect of a_B for initial ω_A values between 174° and 186° at $[a_A, e_A] = [1.2591\,\text{AU}, 0.2]$ in the Sun+(Earth+Moon)-spacecraft system. The post-flyby a_B tends to have a sinusoidal curve in terms of ω_A. This is caused by the phase change of the flyby: the spacecraft can gain Keplerian energy if the secondary is ahead of it and vice versa. The Keplerian energy is computed for the osculating orbit with respect to the primary when the spacecraft moves far away from the secondary. For this near-Keplerian orbit, its Keplerian energy of the spacecraft with respect to the primary is proportional to the inverse of the semi-major axis [6]. The three-body energy is invariant during the flyby. In order to model this flyby effect, we incorporate a cosine function of ω_A into the GPR function. The combined covariance function (SUM) is expressed as

$$k_{\text{SUM}}(x, x') = \sigma\, exp\left(1 + \frac{(x - x')^T Q(x - x')}{2\alpha}\right)^{-\alpha} + p^2 cos\left(\frac{\pi(\omega_A - \omega_A')}{180h}\right) \tag{18}$$

where σ and α are hyper-parameters defined in Equation 16, and p and h are hyper-parameters denoting magnitude and phase transition, respectively. When ω_A' has a value close to ω_A, the last term in Equation 18 is approximately equal to its maximum p^2. This leads to a sample with ω_A' gaining more information from a sample with ω_A than from other samples.

Training Samples

As mentioned earlier, the GPR model obtains the empirical information by optimizing hyper-parameters using training samples. The training set, in particular their number and distribution, determines the performance, accuracy and efficiency of the GPR model. In principle, the accuracy of prediction improves with the information provided by an increasing number of training samples. However, a smaller training size will reduce the computational effort with respect to Equation 12. In order to balance accuracy and efficiency, the number of training samples is an essential variable which will be analysed in Section Performance of GPR-based Gravity Assist Mapping. Taking the Sun-(Earth+Moon)-spacecraft system as an example, the training input
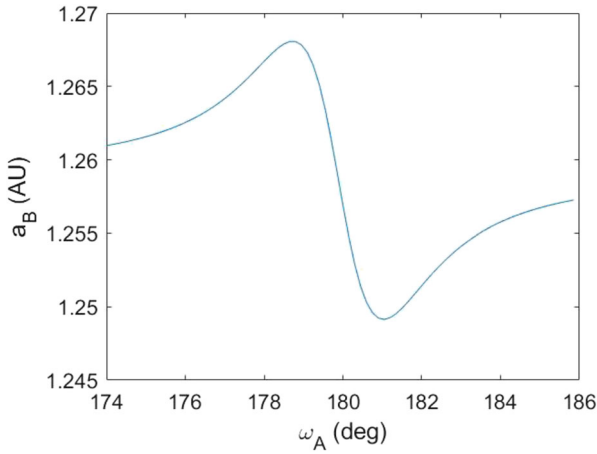
Fig. 2 Variation of semi-major axis after a flyby. This result is obtained by numerical integration in the system of Sun-(Earth+Moon)-spacecraft

vectors a_A, e_A, ω_A are randomly generated in the input space such that the initial radius of periapsis r_p, apoapsis r_a and ω_A meet the conditions

$$\begin{cases} r_p \in [1.00004464, 1.02], \text{ AU} \\ r_a \in [1.01, 2.02], \text{ AU} \\ \omega_A \in [170, 190], \text{ deg} \end{cases} \tag{19}$$

The boundaries are selected to meet a number of requirements. First, the osculating orbit of the spacecraft has a Minimum Orbit Intersection Distance (MOID) with the secondary which allows a minimum of 300 km altitude at Earth taking the Galileo mission as reference [26]. Second, the spacecraft starts outside the Hill sphere such that the status of the spacecraft with respect to the primary can be defined using the Keplerian elements. Third, ω_A is defined in order to observe an obvious flyby effect; beyond these boundaries effectively nothing happens (cf. Fig. 2). After generating sets of $[r_p, r_a, \omega]$, the training inputs can be obtained by

$$\begin{cases} a_A = \dfrac{r_p + r_a}{2}, \text{ AU} \\ e_A = \dfrac{r_a - r_p}{r_a + r_p} \\ \omega_A \in [170, 190], \text{ deg} \end{cases} \tag{20}$$

Once the training inputs have been generated, the perfect post-flyby states are acquired by numerical propagation using Equation 1. The numerical propagation adopts adaptive step-size Runge-Kutta method with a maximum stepsize of 1×10^{-6} and relative and absolute tolerances of 1×10^{-12}, such that the quality of training samples is ensured. It is of importance to define a termination condition for the propagation: in principle one orbital period of the initial osculating orbit is defined as end epoch. The GPR-based Keplerian map is to assess the effect of a single
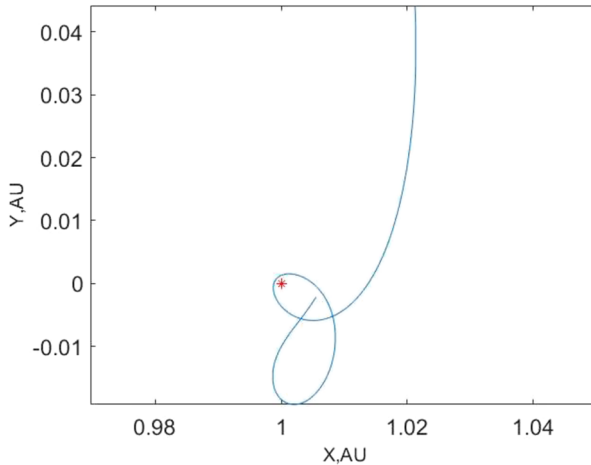
Fig. 3 Example of a trajectory in synodical frame. The red star represents Earth. The Sun at $(-\mu, 0)$ is not shown in this figure

flyby. However, the selected boundaries for input space do not guarantee a single fly by as shown in Fig. 3. The initial condition of this particular trajectory is [1.02 AU, 0.007, 177.84°]. Apparently, the spacecraft has a second close encounter with Earth at the final stage of the propagation. The second flyby results in an inconsistency of the training outputs: most are based on one flyby, some are based on two such events. There is no clear boundary for a flyby in the PCR3BP, which makes it difficult to remove second flybys by monitoring the position or velocity of the spacecraft. To overcome this difficulty, the Tisserand parameter is used as the index for the termination condition [27]:

$$T = \frac{1 - \mu}{a} + 2\sqrt{a(1 - e^2)} \tag{21}$$

Using this parameter as termination index is inspired by the value of the Tisserand parameter before and after a flyby when the spacecraft is far away from the secondary [28]. Figure 4 illustrates the evolution of the Tisserand parameter for the trajectory in Fig. 3. The Tisserand parameter drops dramatically when the first flyby happens, then it increases rapidly and begins to decrease again when the second close encounter is coming. To prevent the second flyby, the propagation is terminated when the Tisserand parameter is maximised after the first drop.

at the top of the curve after the first flyby. In addition, the cases when an Earth impact takes place are removed for obvious reasons.

Performance of GPR-based Gravity Assist Mapping

In order to evaluate the accuracy of the GPR-based Gravity Assist Mapping, a set of test samples has been generated whose inputs are randomly distributed over the input space. To quantify the quality of any combination of GPR model and
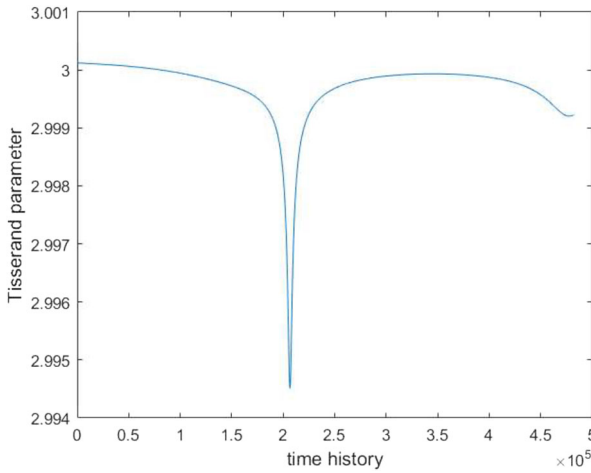
Fig. 4 The evolution history of the Tisserand parameter for the trajectory of Fig. 3. The X-axis shows the steps of the numerical integration for one orbital period

number and distribution of training samples, the Root-Mean-Square Error and the Mean-Absolute-Percentage Error are adopted:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_{\text{PCR3BP}}^{i} - y_{\text{GPR}}^{i})^2}{n}} \tag{22}$$

and

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_{\text{PCR3BP}}^{i} - y_{\text{GPR}}^{i}}{y_{\text{PCR3BP}}^{i} - x_{\text{PCR3BP}}^{i}} \right| \tag{23}$$

In principle, an increment of the number of training samples can be expected to contribute to the reduction of errors because the GPR model is offered more empirical information. So the acceptable error level serves for choosing the size of the training dataset.

The procedure of building a GPR-based Gravity Assist Mapping is summarized in the following steps:

(1) Define the mass ratio for a specific three-body system. The Sun-(Earth+ Moon)-spacecraft system is taken as an example with a mass ratio of 3.036×10^{-6}. The mass ratio is the only key parameter in the PCR3BP, so it is easy to implement the GPR-based Gravity Assist Mapping for different three-body systems;

(2) Construct the covariance function. RQARD and SUM (Equations 16 and 18) are used and analyzed here;

(3) Generate training inputs. The input space defined in this work selects a broader boundary for flyby passage, instead of considering only the distant encounters

outside the Hill sphere. Training set sampling uses four different distributions as shown in Fig. 5: Systematic Sampling (SS), Random Sampling (RS), Latin Hypercube Sampling (LHS), and Stratified Random Sampling (SRS), further described in Section Accuracy;

(4) Calculate training outputs using the numerical PCR3BP propagation. The termination conditions on time and the Tisserand parameter are employed to guarantee the consistency of the training outputs;

(5) Optimize the hyper-parameters based on the training samples, i.e. maximize the outcome of Equation 15. Conjugate gradient ascent is used to find optimal values of the hyper-parameters. This is used in a multi-start approach to avoid being trapped in local minima (more precisely: the gradient procedure is initiated at 10 different initial combinations of hyper-parameter values);

(6) Determine the optimal size of the set of training data. This is done by adding training samples gradually until a stable error, i.e. the difference between the outputs of the numerical PCR3BP and the GPR models on the test dataset, is observed.

(7) Predict the output given a new input. Based on the training samples, the selected covariance function and the corresponding optimized hyper-parameters, the prediction can be obtained.

Accuracy

We generate a test set meeting the conditions of Equation 19 to assess the accuracy of the method. The test set consists of 500 (N^*) input vectors randomly distributed over the input space, and is kept fixed for all evaluations in a certain application. Their corresponding outputs are obtained by numerical propagation in the PCR3BP model and by the GPR model. To evaluate the performance for each model, both the MAPE and RMSE will be calculated using Equations 22 and 23. The Sun-(Earth+Moon)-spacecraft system is considered for all applications.
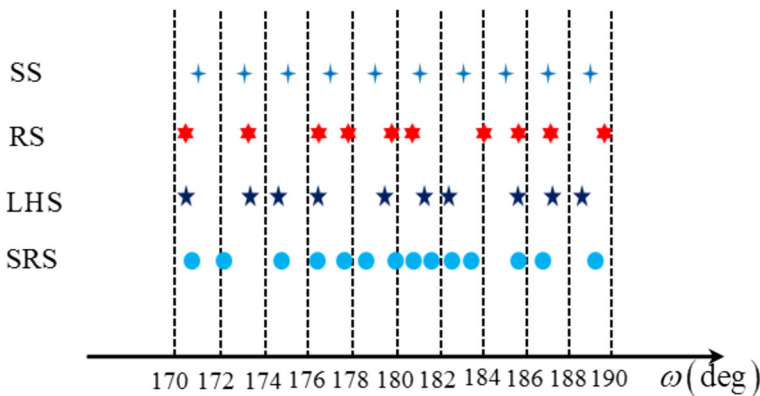

Fig. 5 Illustration of training samples using different distributions

Systematic Sampling

By assuming that the training data can represent the empirical information of the entire input space, we create a dataset by systematic sampling:

$$\begin{cases} r_p \in [1.00004464 : (1.02 - 1.00004464)/N_{division} : 1.02], \text{ AU} \\ r_a \in [1.01 : (2.02 - 1.01)/N_{division} : 2.02], \text{ AU} \\ \omega_A \in [170 : (190 - 170)/N_{division} : 190], \text{ deg} \end{cases} \tag{24}$$

$N_{division} = 6, 7, 8, 9, 10, 11$ indicates to what extent each input feature is partitioned.

With increasing training size, Fig. 6 shows MAPE and RMSE of the test set on the semi-major axis using the RQARD and SUM covariance functions, respectively. In this case, the RQARD function performs better than SUM, but MAPE is larger than 1000%. Obviously, the systematic sampling fails to offer useful information for the GPR model. This is because the GPR model uses Bayesian inference, a type of statistical inference, which strongly relies on the assumption of random selection [29]. If this assumption is not satisfied, as is the case with regular sampling, the basis for any GPR model drops out and the output is to be considered as unrealistic.

Random Sampling

In this test, we distribute the training inputs randomly over the entire input space. MAPE and RMSE results are shown in Fig. 7. The SUM function has a better performance than the RQARD function. For the case of 780 training samples, SUM obtains the lowest MAPE value of 19.7% and an RMSE of 5.8×10^{-3} AU. We generate another two sets of training samples using different random seed numbers and perform the predictions using the SUM function. The trends of MAPE and RMSE with respect to three training data sets show that the random seed number is not an influential parameter for the level of convergence.
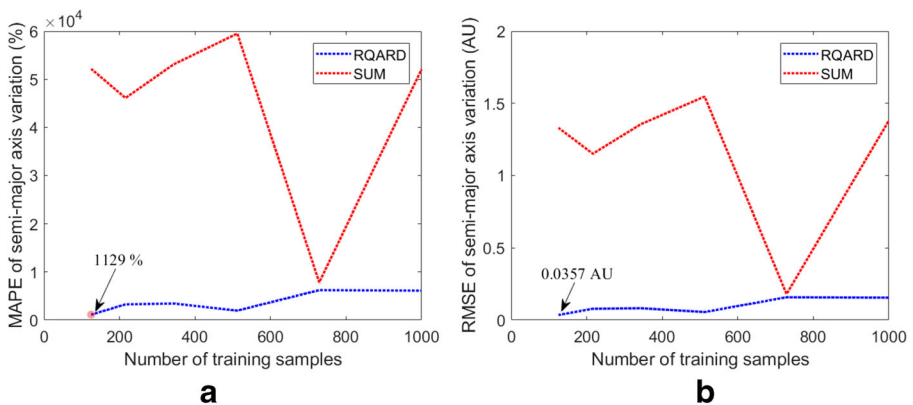

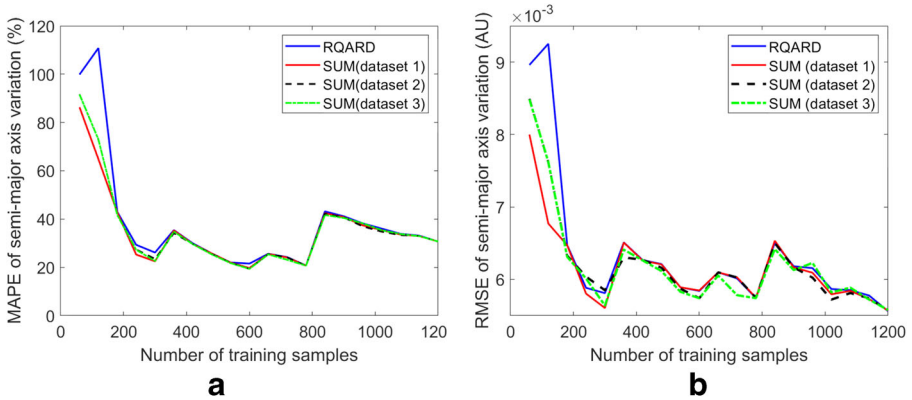**Fig. 6** The error of test samples when the training dataset uses systematic sampling

Fig. 7 The error of test samples when the training dataset has a random distribution

Latin Hypercube Sampling

LHS is a statistical method to generate random samples especially for multi-dimensional problems [30]. This method is generally used to avoid clustering of random samples in a specific subspace. Shown in Fig. 8, the SUM function captures the model mechanics more quickly than RQARD. In addition, the SUM function has a lowest MAPE value of 24.1% and an RMSE of 4.9×10^{-3} AU at a training data size of 900. The LHS has a poorer accuracy than random sampling with respect to MAPE.

Stratified Random Sampling

Figure 9 shows the number of MAPE values larger than 10% for the subspaces of ω_A using random sampling. Due to the stronger flyby effects around $\omega_A = 180°$ according to Fig. 2, the error for these test points is worse. More training points can
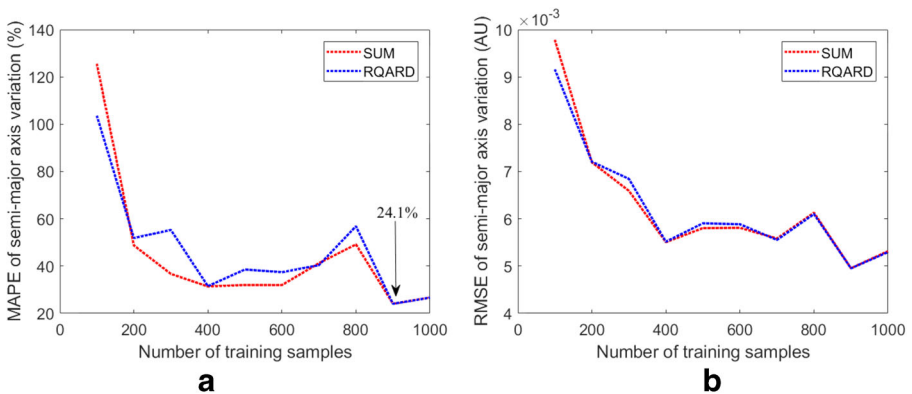

Fig. 8 The error of test samples when the training dataset has an LHS distribution
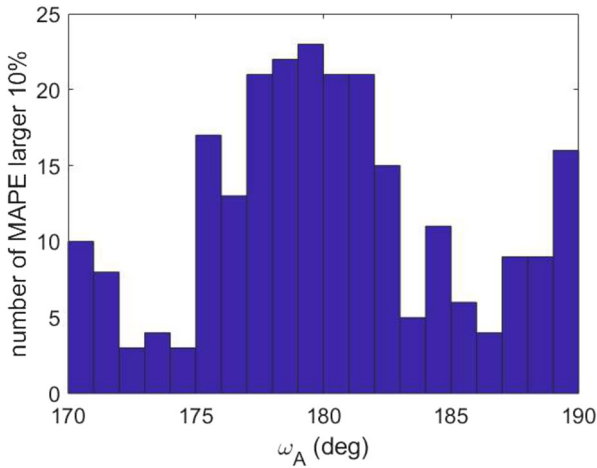
Fig. 9 Number of MAPE larger than 10% based on random sampling

be added for the subspace of $\omega_A \in [175°, 185°]$ to gain more information over this subspace.

In order to improve the performance of prediction for the subspace of $\omega_A \in [175°, 185°]$, the sampling density of this range can be taken as twice as that of the remaining area: an SRS method. The training size is increased by 60 every time. For each 60 training samples, there are 10 samples within $[170°, 175°]$, 40 samples within $[175°, 185°]$ and 10 samples within $[185°, 190°]$. Figure 10 shows the corresponding RMSE results. The performance becomes stable when using an increasing number of training samples. In Figs. 10a and b, the RMSE of these training datasets converges to the same level with a training data size larger than 1500. The prediction of ω needs more training samples than that of a and e. This indicates that the GPR function needs more information to learn the change of ω. We generate another two groups of training samples using the same SRS method with different random seed numbers. For the prediction of a, the standard deviation among these three groups is about 1.0×10^{-4} AU when a number of more than 1500 training samples are used. The random seed number for generating training dataset is a negligible influence factor for the performance of GPR models.

Summary of Sampling Methods

We used four types of sampling methods for the generation of the training samples for comparison. The prediction using RS performs better than that of SS and LHS. Based on RS, we developed an SRS specifically for this case ($h > 300$ km) due to the larger errors for the test samples in the subspace of $\omega_A \in [175°, 185°]$. The RMSE using SRS is reduced to less than 1/4 of that of RS. By using different training datasets, the training process of the GPR model is illustrated to be insensitive to the random seed numbers.
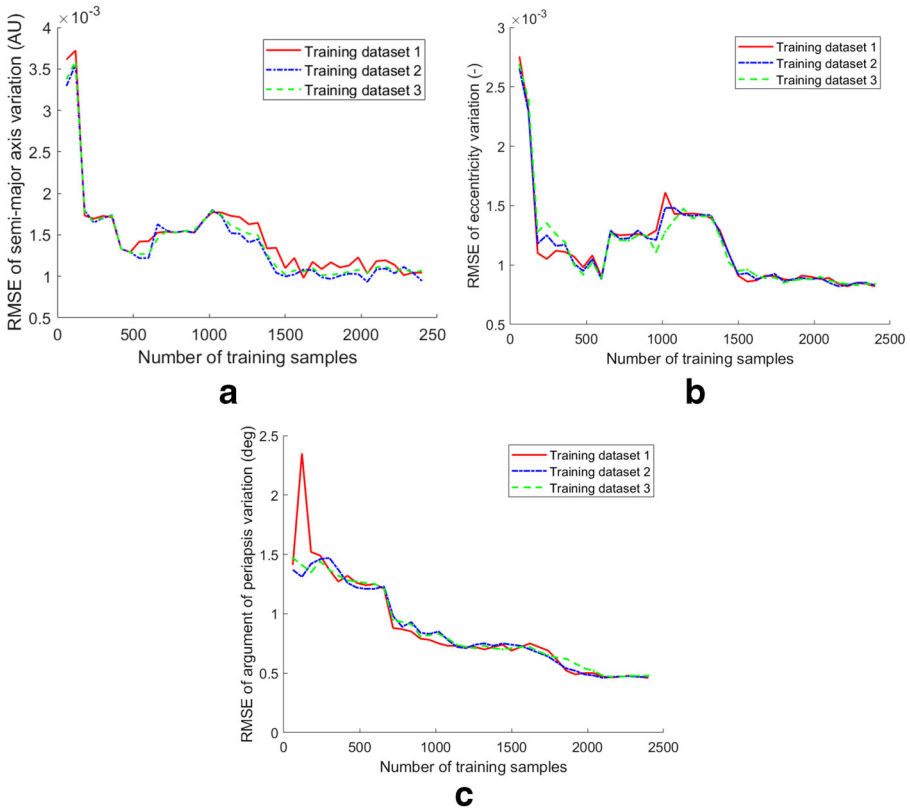
Fig. 10 The RMSE of test samples when the training dataset adopts a stratified random sampling. The SUM covariance function is used

Flybys Outside the Sphere of Influence of the Secondary

In this application, the periapsis of the initial osculating orbits of the spacecraft are chosen such that they are above the sphere of influence of the secondary (SoI): $r_p > 1.0062$ AU (i.e. $r_p > r_{SoI} + r_2$) in the system of Sun-(Earth+Moon)-spacecraft. Figure 11 shows the errors of the semi-major axis predictions after the flybys. After some fluctuations for training data sizes between 0 and 1000, the error becomes gradually smaller and stable until the data size reaches 2400. The RMSE is selected as the main criterion for optimizing hyper-parameters. Using the same set of hyper-parameters, MAPE could have a slightly different trend compared to that of RMSE. The performance of predicting eccentricity is presented in Fig. 12. The RMSE of e decreases to a value of 1.0×10^{-3} using 2000 training samples. The maximum eccentricity e_A among the test cases is 0.33. In Fig. 13, the RMSE of ω becomes lower than 0.4° with more than 1800 training samples. The convergence of optimizing hyper-parameters for the prediction of the output state is illustrated in Fig. 14: the hyper-parameters l_a, l_e and l_ω are selected. When the training data size becomes
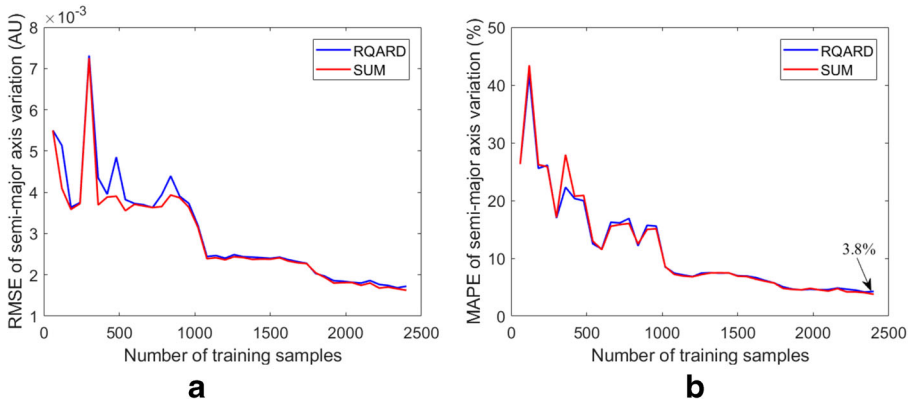
Fig. 11 The semi-major axis error of test samples when the training dataset has a random sampling. The periapsis of initial osculating orbit of the spacecraft is outside the sphere of influence of the secondary

larger than 1000, the GPR model converges gradually to the optimum. Although the optimization process converged at negative values for the hyper-parameters eventually, this has no consequences since the quadratic value of them is used (Equation 17). RS is used in this case because a significant difference of error between different subspaces is not observed.

Flybys Outside the Hill Sphere of the Secondary

For the prediction of distant flyby effects, this application investigates the condition of the radius of periapsis being larger than 1.01 AU (i.e. $r_p > r_2 + r_{Hill}$). Figure 15 shows the accuracy of prediction with an increasing number of training samples. For three orbital elements, the curves flatten out gradually after using 1300 training samples. The accuracy is better than the case of SoI. MAPE of δa and δe is no more
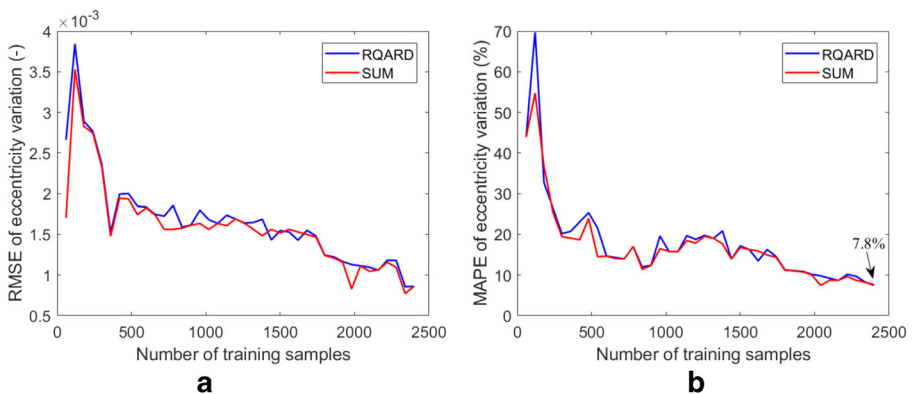

Fig. 12 The eccentricity error of test samples when the training dataset has a random sampling. The periapsis of initial osculating orbit of the spacecraft is outside the sphere of influence of the secondary

Fig. 13 The ω error of test samples when the training dataset has a random sampling. The periapis of initial osculating orbit of the spacecraft is outside the sphere of influence of the secondary. The best MAPE obtained is 5.7%
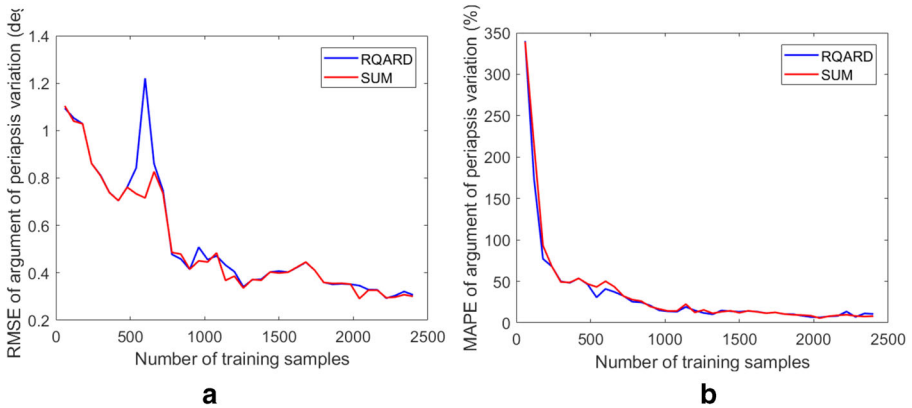
than 1%. The MAPE of $\delta\omega$ has been improved by a factor of two. It shows that the low-energy case is easier for GPR to learn. This is mainly because the flyby effects at low energies are less complex than the previous applications, e.g. the retrograde flybys disappear at such energies [16]. Therefore, less training samples are required to obtain a reliable model.

The comparison between the performance of our GPR-GAM and the Keplerian Map (KM) model of Ross et al. [6] (fully reproduced here) is shown in Table 1. 1500 training samples are considered in our GPR model and the SUM covariance function is selected. The test set has 500 samples with inputs randomly distributed in the input space and outputs numerically calculated by the PCR3BP propagation. Both tests are performed in MATLAB 2017b using a laptop of Core i7 CPU and 8.00
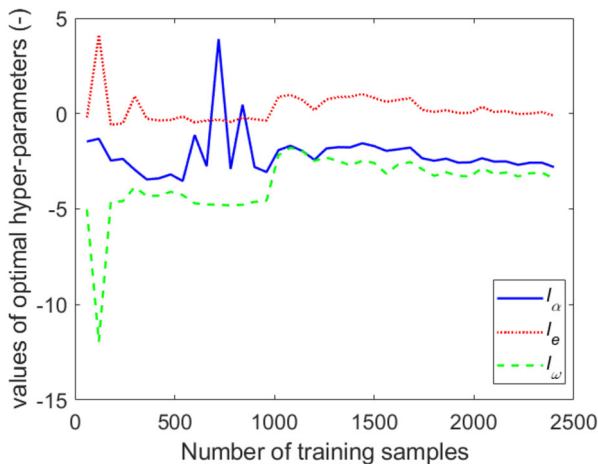

Fig. 14 Evolution of optimal hyper-parameters. The flybys are outside the SoI of the secondary
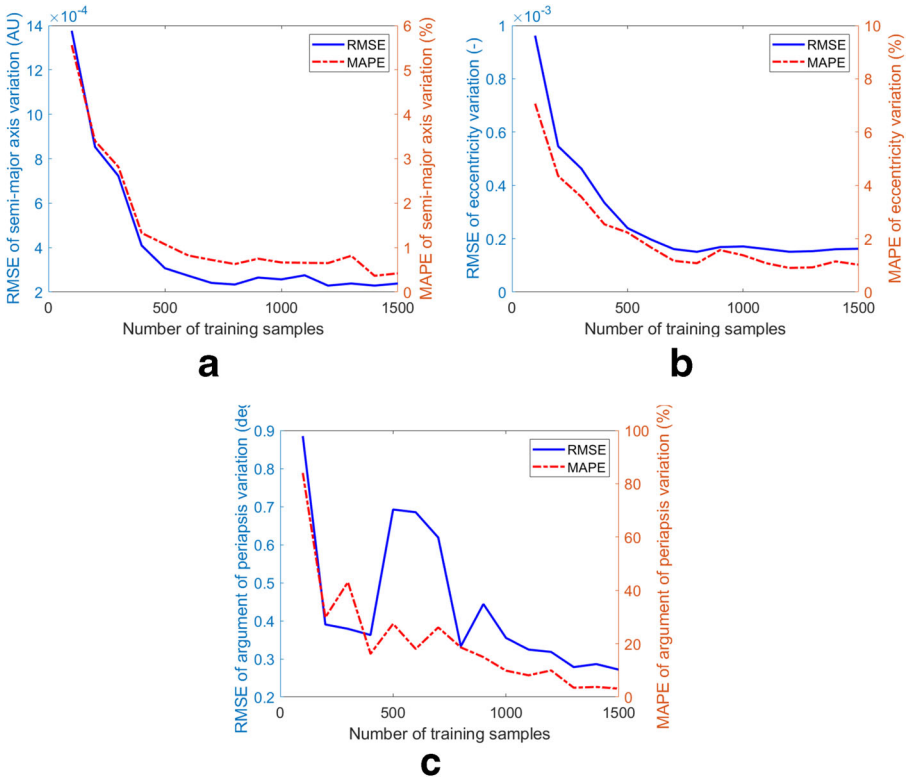
Fig. 15 The RMSE and MAPE of test samples when the training dataset adopts a stratified random sampling. The SUM covariance function is used

GB RAM. The RMSE of predicting a using GPR-GAM is only 1/6 of that of the KM. In addition, the GPR-GAM improves the accuracy of predicting ω to a level of 4.72×10^{-3} rad. The KM fails to quantify the variation of ω accurately due to simply evaluating ω in terms of orbital period change. The CPU time of prediction spent on a single test sample is the same for both a and ω. KM is much slower in predicting a because of using the numerical integration but is faster in the prediction of ω.

Table 1 The performance of GPR-GAM compared to that of Ross et al. [6] (fully reproduced here)

	RMSE		Prediction CPU Time* [s]	
	a [AU]	ω [rad]	a	ω
GPR	2.38×10^{-4}	4.72×10^{-3}	1.16×10^{-6}	1.16×10^{-6}
Ross et al.	1.42×10^{-3}	1.57	0.064	1.04×10^{-6}

*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
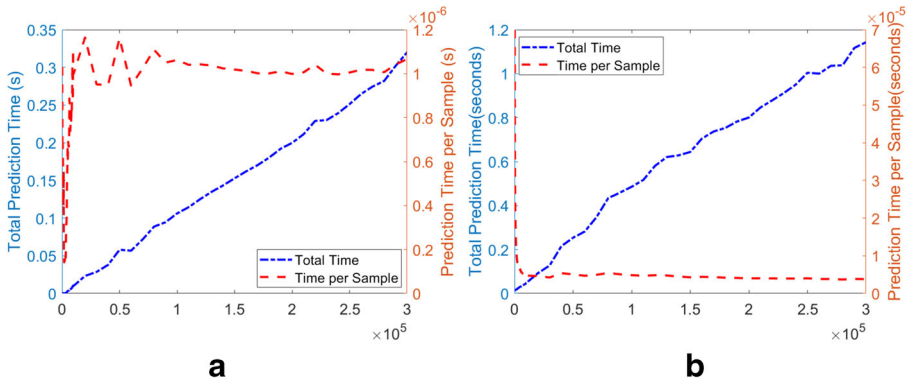
Fig. 16 Time efficiency of the GPR-GAM for one orbital element. Equation 13 is used for the prediction of multiple inputs. The cost of generating training samples is not included

The time efficiency of the GPR method is demonstrated for up to 300,000 test samples in Figure 16. When using 1500 training samples for the case outside the Hill sphere, the total prediction time increases linearly with the number of test samples. It costs 0.32 s for the evaluation of 300,000 samples. The prediction time per sample is always lower than 1.2×10^{-6} s. For the case outside the SoI in Figure 16b, the computational effort is bigger because of using 2400 training samples. It spends only 1.14 s for the same number of test samples and the CPU time per sample is 3.8×10^{-6} s. When applying GPR to the case outside the Hill sphere, the generation of the training samples (0.28 s per sample) and training the GPR model (1530 s for 10 multi-starts) cost 1950 s in total. However, the GPR model is off-the-shelf once the training process is completed, and the values in Table 1 give an accurate estimation of the computation time needed to predict the flyby effect of a single input.

Table 2 summarizes the performance of GPR for various applications. The accuracy of the GPR model becomes better when the closest passage of the spacecraft moves further from the secondary. The radius of the Hill sphere is close to that of the SoI but the accuracy improves especially for a and e. The prediction of ω is the worst which can be further investigated.

Table 2 Summary of the GPR performance for all applications (SUM covariance function)

Flyby distance	Distribution	MAPE [%]			RMSE		
		a	e	ω	a [AU]	e [-]	ω [rad]
> 300 km altitude	RS	19.7	85.6	97.5	0.0056	0.0062	0.0511
	SRS	5.6	22.3	24.5	0.0009	0.0008	0.0082
> SoI	RS	3.8	7.8	5.7	0.0003	0.0007	0.0051
> Hill sphere	RS	0.4	1.0	3.1	0.0002	0.0002	0.0047

The number of training samples N for the three cases of flyby distance are 2400, 2400, 1500 respectively. The SRS is not applied to the second and third cases because the error in different parts of the input space is similar. The values of corresponding hyper-parameters are given in Appendix B

Conclusions

A model of the Gravity Assist Mapping was proposed based on a machine-learning method called Gaussian Process Regression. It was trained by a dataset consisting of thousands of samples generated using numerical integration in the planar CR3BP framework. The generation of training data takes into account elliptical orbits of low, moderate and high eccentricity. We determined the size of training samples by choosing a stable RMSE on the test dataset. A covariance function (SUM) was developed combining the cosine term and the rational quadratic term with automatic relevance determination (RQARD), which captures well the dynamics of flybys. The GPR-based model can assess the flyby effect more efficiently given the initial condition of a particle compared to methods based on numerical integration. The model for various planar CR3BP systems can be built by simply changing the mass ratio parameter. Using a machine-learning method, this work investigated and predicted the flyby effect instead of numerical integration of the equations of motion in previous contributions. The domain of applicability is beyond that of the Keplerian map. The GPR-MAP requires to be trained to model the flyby effect of trajectories belonging to different domain.

The results demonstrate that the GPR-based Gravity Assist Mapping has a good performance on accuracy. Compared to previous work in literature, significant improvements have been made, in particular when using a combined covariance function and stratified random sampling. This can contribute to two main studies in astrodynamics. On the one hand, the characteristic of the GA in the CR3BP can be investigated by a great deal of accurate data produced by the GPR model, providing a deep understanding of the third-body effect. On the other hand, the GPR model can be considered to design multi-flyby missions, which takes the advantage of a high efficiency to update the post-flyby status. The developed GPR-GAM is still a prototype. It has some limitations, such as being unable to classify collision orbits in test samples because they are removed in the training samples. The post-flyby status near the neighbourhood of the secondary might be neglected using the Tisserand termination condition. Further improvements include developing a classification model to categorise collision orbits given an arbitrary initial condition and using a distance termination condition.

Appendix A: Minimum Distance Between Spacecraft and Secondary

All variables are non-dimensionalized using $(M_1 + M_2)$ for the mass, the semi-major axis of the secondary a_{P2} for the length, and $\sqrt{a_{P2}^3/(G(M_1 + M_2))}$ for the time. For the circular orbit of the secondary, the true anomaly θ_{P2} is set to be 0 at the position of the positive X-axis. The initial true anomaly of the secondary $\theta_{P2}^{initial}$ is selected such that it arrives at the negative X-axis when the spacecraft (in terms of initial osculating orbit) reaches periapsis. This is specified by:

$$\theta_{P2}^{initial} = \pi(1 - (a_A)^{\frac{3}{2}}) \tag{25}$$

where a_A is the semi-major axis of the initial osculating orbit of the spacecraft. The distance d between the spacecraft and the secondary is defined using the law of cosines:

$$d = \sqrt{r_1^2 + r_{P2}^2 - 2r_1 r_{P2} cos(\gamma)} \tag{26}$$

where r_1 represents the distance between the spacecraft and the primary, r_{P2} the distance between the secondary and the primary, γ denotes the angle contained between r_1 and r_{P2}. Since r_{P2} has a constant value of 1, Equation 26 is simplified to:

$$d = \sqrt{1 + r_1^2 - 2r_1 cos(\gamma)} = \sqrt{f(r_1, \gamma)} \tag{27}$$

Assuming a fixed value of γ, the partial derivative $\frac{\partial f(r_1, \gamma)}{\partial r_1}$ is:

$$\frac{\partial f(r_1, \gamma)}{\partial r_1} = 2r_1 - 2cos(\gamma) \tag{28}$$

The value of the right-hand side is always larger than zero given the initial conditions of this paper. The function $f(r_1, \gamma)$ is monotonically increasing. When r_1 equals the radius of periapsis r_p and $\gamma = 0$, d has the minimum value $| r_p - 1 |$. This condition corresponds to $\omega = \pi$.

Appendix B: Values of Hyper-Parameters

The optimized values of the hyper-parameters of the models in Table 2 are given below.

Table 3 The optimized hyper-parameters of GPR-GAM for the first case in Table 2

Output	σ	α	l_a	l_e	l_ω	p	h
a	−1.2450	−1.2043	2.9608	0.7480	−4.2359	0.5279	−11.7555
e	−3.3324	−2.4650	4.8943	1.7674	−8.8055	−0.0180	−10.4190
ω	−2.1321	−2.1245	1.7252	1.2615	−3.2145	−0.2121	−7.2210

Table 4 The optimized hyper-parameters of GPR-GAM for the second case in Table 2

Output	σ	α	l_a	l_e	l_ω	p	h
a	−2.0214	−2.5534	0.2000	−0.8179	3.4195	3.4419	−0.2592
e	−1.7113	−2.0229	0.1768	−0.7881	3.6202	2.8485	−0.2926
ω	−2.7010	−2.9755	0.5394	2.1231	1.4008	−0.3179	−1.1486

Table 5 The optimized hyper-parameters of GPR-GAM for the third case in Table 2

Output	σ	α	l_a	l_e	l_ω	p	h
a	−1.9506	−3.3963	−3.3770	−0.3370	−3.4057	1.9674	−3.1555
e	−1.3524	−2.2365	0.2718	−4.2451	−1.9327	4.0959	−4.2971
ω	−1.8572	−2.6009	0.7222	3.4731	−1.1252	0.7433	0.8022

Table 6 The optimized hyper-parameters of GPR-GAM for the fourth case in Table 2

Output	σ	α	l_a	l_e	l_ω	p	h
a	−1.8168	−2.4690	−0.2082	−1.5223	3.2384	0.9166	0.5527
e	−1.3950	−2.3792	−0.0306	−1.3639	2.2226	2.9447	−0.3096
ω	−1.4090	−2.9224	0.9743	4.7129	0.0638	−0.7502	0.2507

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent This research does not involve any studies with human participants.

References

1. D'Amario, L.A., Bright, L.E., Wolf, A.A.: Galileo trajectory design. Space science reviews **60**(1-4), 23–78 (1992)
2. Broucke, R.A.: The celestial mechanics of gravity assist. AIAA/AAS Astrodynamics Conference, Minneapolis, Minnesota (1988)

3. Longuski, J.M., Williams, S.N.: Automated design of gravity-assist trajectories to mars and the outer planets. Celest. Mech. Dyn. Astron. **52**(3), 207–220 (1991)
4. Strange, N.J., Longuski, J.M.: Graphical method for gravity-assist trajectory design. J. Spacecr. Rocket. **39**(1), 9–16 (2002)
5. de Almeida Prado, A.F.B., de Felipe, G.: An analytical study of the powered swing-by to perform orbital maneuvers. Adv. Space Res. **40**(1), 102–112 (2007)
6. Ross, S.D., Scheeres, D.J.: Multiple gravity assists, capture, and escape in the restricted three-body problem. IAM J. App. Dynamic. Sys. **6**(3), 576–596 (2007)
7. Ross, S.D., Koon, W.S., Lo, M.W., Marsden, J.E.: Design of a multi-moon orbiter. American Astronautical Society. No. 114, pp. 669–683. (2003)
8. Whiffen, G.J.: An investigation of a Jupiter Galilean moon orbiter trajectory. In: AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MN, AAS. 03–544 (2003)
9. Koon, W.S., Lo, M.W., Marsden, J.E., Ross, S.D.: Heteroclinic connections between periodic orbits and resonance transitions in celestial mechanics. Chaos: An Interdisciplinary Journal of Nonlinear Science **10**, 427–469 (2000)
10. Koon, W.S., Lo, M.W., Marsden, J.E., Ross, S.D.: Low energy transfer to the moon. Celest. Mech. Dyn. Astron. **81**, 63–73 (2001)
11. Howell, K.C., Marchand, B.G., Lo, M.W.: Temporary satellite capture of short-period Jupiter family comets from the perspective of dynamical systems. J. Astro. Sci. **49**(4), 539–557 (2001)
12. Petrosky, T.Y., Broucke, R.: Area-preserving mappings and deterministic chaos for nearly parabolic motions. Celest. Mech. Dyn. Astron. **42**(1), 53–79 (1987)
13. Shevchenko, I.I.: The Kepler map in the three-body problem, New Astronomy. New Astron. **16**(2), 94–99 (2011)
14. Pe nagaricano, M.O., Scheeres, D.J.: A Perturbation Theory. Acta Astronaut. **67**(2), 27–37 (2010)
15. Alessi, E.M., Sánchez, J.P.: Semi-Analytical Approach for Distant Encounters in the Spatial Circular Restricted Three-Body Problem. Journal of Guidance, Control and Dynamics **39**(2), 351–359 (2016)
16. Campagnola, S., Skerritt, P., Russell, R.P.: Flybys in the planar, circular, restricted, three-body problem. Celest. Mech. Dyn. Astron. **113**(3), 343–368 (2012)
17. Szebehely, V.: Theory of Orbits. Academic Press, New York (1967)
18. Krige, D.G.: A statistical approach to some basic mine valuation problems on the Witwatersrand. J. South. Afr. Inst. Min. Metall. **52**(6), 119–139 (1951)
19. Shang, H., Liu, Y.: Assessing Accessibility of Main-Belt Asteroids Based on Gaussian Process Regression. J. Guid. Control. Dyn. **40**(5), 1144–1154 (2017)
20. Gao, A., Liao, W.: Efficient gravity field modeling method for small bodies based on Gaussian process regression. Acta Astronaut. **157**, 73–91 (2019)
21. Bouwman, L., Liu, Y., Cowan, K.: Gaussian Process models for preliminary low-thrust trajectory optimization. In: Astrodynamics Specialist Conference, Portland, ME. AAS 19–873, pp. 3687–3706. (2019)
22. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning, pp. 7–31, 79–128. Massachusetts Institute of Technology Press, Cambridge, Massachusetts London, England (2006)
23. Olds, A.D., Kluever, C.A., Cupples, M.L.: Interplanetary Mission Design Using Differential Evolution. J. Spacecr. Rocket. **44**(5), 1060–1070 (2007)
24. Sun, S., Zhang, G., Wang, C., Zeng, W., Li, J., Grosse, R.: Differentiable Compositional Kernel Learning for Gaussian Processes. Proceedings of the 35th International Conference on Machine Learning **80**, 4828–4837 (2018)
25. Duvenaud, D.: Automatic model construction with Gaussian processes. Doctoral dissertation, University of Cambridge (2014)
26. Anderson, J.D., Campbell, J.K., Ekelund, J.E., Ellis, J., Jordan, J.F.: Anomalous orbital-energy changes observed during spacecraft flybys of Earth. Phys. Rev. Lett. **100**(9), 091–102 (2008)
27. Tisserand, F.: Traité de mécanique céleste. Gauthier-Villars (1891)
28. Campagnola, S., Kawakatsu, Y.: Three-dimensional resonant hopping strategies and the Jupiter Magnetospheric Orbiter. J. Guid. Control. Dyn. **35**(1), 340–344 (2012)
29. Osborne, J.G.: Sampling errors of systematic and random surveys of cover-type areas. J. Am. Stat. Assoc. **37**(218), 256–264 (1942)

30. McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.