Gaussian process models for preliminary low-thrust trajectory optimization

Bouwman, Lieve; Liu, Yuxin; Cowan, Kevin

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

GAUSSIAN PROCESS MODELS FOR PRELIMINARY LOW-THRUST TRAJECTORY OPTIMIZATION

Lieve Bouwman,[*] Yuxin Liu[†] and Kevin Cowan[‡]

Low-thrust trajectories can benefit the search for propellant-optimal trajectories, but increases in modeling complexity and computational load remain a challenge for efficient mission design and optimization. In this paper, an approach for developing models utilizing Gaussian Process (GP) regression and classification is proposed to perform computationally efficient optimization while obtaining acceptable accuracies for trajectories based on exponential sinusoid shaping. The goal of this work is to predict a combination of values of input variables which corresponds to a shape-based trajectory with the smallest total velocity increment (ΔV) or propellant mass fraction (J_m). A GP classification model is constructed to assess whether a given combination of values of input variables corresponds to a feasible trajectory. GP regression models are developed to predict the total ΔV and J_m corresponding to a combination of shape parameters, which can replace the required integration along the shape. In addition, advanced regression models are developed to predict the target values while requiring only three input parameters, thereby replacing the entire shape computation. In order to develop a GP model that fits the problem at hand, the underlying functions and parameters should be selected rationally. In this work, a novel model development approach is proposed to ensure that the mean function, covariance function, likelihood function, inference method, and hyperparameters, which dominate the performance of the models, are chosen rationally in terms of mean absolute percentage error (MAPE) and prediction time. Using this approach, GP models are developed and tested on transfer trajectories from Earth to Mars and Ceres, and from Mars to Earth, and their performance, in terms of MAPE and prediction time, is compared to that of more common optimization techniques in combination with the exponential sinusoid and other shape-based methods. The results demonstrate that the computation time can significantly be reduced while achieving promising MAPE's, especially when the goal is to locate regions of feasible or near-optimal trajectories. The proposed model development procedure is tested for robustness, which provides confidence in the proposed approach. Furthermore, it is found that the models which map three input variables directly to a ΔV or J_m value perform better than the ones trained with shape information, which demonstrates the strength of GP models as applied to low-thrust trajectory optimization.

INTRODUCTION

Trajectories flown with low-thrust rocket engines, such as ion propulsion systems, are attractive due to their propellant efficiency and high reliability.[1] This becomes especially important for interplanetary missions where large velocity changes are generally required. To obtain ΔV values required for interplanetary flight, the engine has to operate for thousands of hours, which complicates the dynamics of the vehicle and results in challenging trajectory design and optimization. Most of the methods that are used to solve this optimization challenge for low-thrust trajectories are based on a non-linear programming problem. Although these methods are successful in the detailed design of a trajectory, they fail to efficiently explore a large number of candidate trajectories. For the preliminary design of trajectories, analytical methods have proven to be powerful tools. The shape-based method first introduced by Petropoulos[2] approximates the spacecraft's

---

[*] MSc Graduate, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands. E-mail: lievebouwman@gmail.com.

[†] PhD Candidate, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands. E-mail: yuxin.liu@tudelft.nl.

[‡] Education Fellow + Lecturer, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands. E-mail: k.j.cowan@tudelft.nl.

trajectory with an exponential sinusoid and is well-known due to its wide applicability. A generalized implementation of the exponential sinusoid using the multi-revolution Lambert's problem was proposed by Izzo.[3] The resulting algorithm efficiently locates near-optimal trajectories, but a time-consuming search process is required to locate these trajectories. Depending on the shape-based method and the computer used, most shapes take about 0.3-2 s computation (CPU) time per trajectory,[1] which may result in a rather slow process when a large search space needs to be investigated. Given the increasing interest in low-thrust trajectories, faster calculations are needed when exploring a search space for a specific mission.

Gaussian Process regression (GPR) and classification (GPC) are useful techniques to predict a large number of target values within limited CPU time. D.G. Krige[4] was the first to propose the GPR method with the goal of estimating the distribution of gold. Since then, GPR has been applied four times in the aerospace field: 1) to evaluate the aerodynamic coefficient of a spaceplane,[5] 2) to assess the accessibility of main-belt asteroids,[6] 3) to model the gravity field of small bodies[7] and 4) to evaluate the Mars entry terminal state.[8] GPC has not been used before to classify problems related to aerospace. The main advantage of a method based on a GP is that it is flexible and computationally efficient for handling the relation between inputs and outputs in both regression and classification problems.[6] A GP model is composed of functions, so in theory an unlimited number of different models could be developed. Generally, the GP model for a specific problem is selected by trial and error[6] or based on models used for comparable problems, leaving the functionality of the model as a black box. In this paper, a procedure for developing models using a GP is proposed to further speed up the preliminary optimization of low-thrust trajectories based on exponential sinusoid shaping. The robustness of the choices made in this development procedure is also assessed.

This paper begins by discussing the implementation of the exponential sinusoid according to the multi-revolution Lambert's problem, in order to generate training data for the GP models. Next, the basics of GPR and GPC are discussed, together with the proposed development procedure for models based on a GP. The performance of the models is explained, and the robustness of the model development procedure is assessed. Models are developed for three mission test cases. The models for the test case from Earth to Mars are presented and their robustness measured. The models developed for the other two test cases are presented in Reference 9. The performance reached for each of the test cases, in terms of accuracy, CPU time, and the ability to locate regions of near-optimal trajectories, is discussed, followed by the conclusions.

MODELING LOW-THRUST TRAJECTORIES USING THE EXPONENTIAL SINUSOID

Theory of the Exponential Sinusoid

Consider the equations of motion of a point mass subject to a Newtonian gravity field and some additional force (ie. thrust) acting on the particle:[10]

$$\begin{cases} \ddot{r} - r\dot{\theta}^2 + \mu/r^2 = F\sin\alpha \\ \ddot{\theta}r + 2\dot{\theta}\dot{r} = F\cos\alpha \end{cases} \tag{1}$$

where r is the radial distance to the particle*, θ is the polar angle, μ is the gravitational parameter of the central body, F is the magnitude of the thrust acceleration, and α is the angle of attack. The normalized acceleration a is obtained by normalizing F with respect to the local gravitational acceleration, Eq. (2).

$$a \equiv F/(\mu/r^2) \tag{2}$$

The general equation of a pure exponential sinusoid (ie. exposin) is defined by:[2]

$$r = k_0 \exp\left[k_1 \sin\left(k_2\theta + \phi\right)\right] \tag{3}$$

where k_0, k_1, k_2, and ϕ are constants. k_0 is a scaling factor. k_1 is the dynamic range parameter, which controls the ratio of apoapsis distance to periapsis distance. k_2 is called the winding parameter, since it determines the number of revolutions until reaching apocenter. The phase angle ϕ defines the orientation of the exposin in its

---

*Superscripts: $\dot{\Box}$ = derivative with respect to time t; $\ddot{\Box}$ = second derivative with respect to time t

plane. The values of the parameters should be chosen such that the resulting trajectory fulfills the boundary conditions for the initial and final position and the time of flight (TOF). The idea of Petropoulos' work[2] is to find a solution for the equations of motion using Eq. (3). A tangential thrust profile is assumed, such that $\alpha = \gamma$. The flight-path angle γ, which is equal to $dr/d\theta$, can be written as follows:

$$\tan\gamma = k_1 k_2 \cos\left(k_2\theta + \phi\right) \tag{4}$$

As thrust tangential to the velocity vector maximizes the energy gain and thus the velocity change,[10] this assumption is deemed reasonable. Using Eq. (1), Eq. (3), and Eq. (4), and the tangential thrust assumption, the normalized thrust acceleration can, after some manipulation,[9] be derived as:

$$a = \frac{(-1)^n \tan\gamma}{2\cos\gamma} \left[\frac{1}{\tan\gamma^2 + k_1 k_2^2 s + 1} - \frac{k_2^2(1 - 2k_1 s)}{(\tan\gamma^2 + k_1 k_2^2 s + 1)^2} \right] \tag{5}$$

where $s \equiv \sin k_2\theta + \phi$; when $n = 0$, the thrust is directed along the velocity vector and when $n = 1$ against the velocity vector.

The TOF can be computed by integrating the inverse of the angular velocity over the polar angle:[3]

$$
\begin{aligned}
\text{TOF} &= \int_{t_0}^{t_f} dt = \int_{\theta_0}^{\theta_f} \frac{dt}{d\theta} d\theta \\
&= \int_{\theta_0}^{\theta_f} \sqrt{r^3(\tan\gamma^2 + k_1 k_2^2 s + 1)/\mu} \;\; d\theta \\
&= \sum_{i=1}^{i_{\max}} \sqrt{r^3(\tan\gamma^2 + k_1 k_2^2 s + 1)/\mu} \;\; \Delta\theta_i
\end{aligned}
\tag{6}
$$

where $i_{\max}$ is the specified number of intervals. Eq. (6) can be solved using a numerical integration technique.

Implementation Using Lambert's Approach

Petropoulos' work[2] was quite successful from a numerical point of view, yet it lacked a solution to go from a generic point P_1 to another point P_2 in a specified amount of time. In the case of ballistic arcs, this problem, named *Lambert's problem*, permits a number of solutions. In line with this work, Izzo proposed the *multi-revolution Lambert's problem* for exponential sinusoids, which is a convenient approach for implementation of the exposin.[3] It starts with assuming k_2 fixed and known, which reduces the problem to all exposins that are defined by only three free parameters: k_0, k_1, and ϕ. After some simplification, it can be shown that with r_0, r_f, ψ, and the number of revolutions N required, and for an assumed value of k_2, a class $S_{k2}[r_0, r_f, \psi, N]$ of exposins exists passing through r_0 and r_f, with ψ the transfer angle. An exposin is classified as a valid trajectory whenever the condition $k_1 k_2^2 < 1$ is fulfilled. Within a defined class, the only free parameter is γ_0, which determines the differences within the family of exposins, namely the TOF. A numerical method can be used to find the intersection between a required TOF and the TOF curve corresponding to a class of exposins. With the value for γ_0 defined, the geometric parameters (k_0, k_1, and ϕ) can be obtained.[9]

Using this generic approach, the problem is to choose the shape parameter k_2 for each exposin. A direct interplanetary low-thrust transfer between two bodies will therefore have a three-dimensional input vector $\boldsymbol{x} = [t_0, k_2, \text{TOF}]$, where t_0 is the departure date. When this input vector is known, the ephemerides of the departure planet at departure and of the arrival planet at arrival should be evaluated. Finally, the unique exposin matching the TOF requirement must be found.

Total Velocity Increment Computation

The first method to assess the efficiency of trajectories is via the total velocity increment (ΔV) required to fly a specific exposin. This value consists of a low-thrust part along the arc (ΔV_{LT}) and two impulsive

shots: one at departure (ΔV_0) and one at arrival (ΔV_f). The ΔV_{LT} can be determined by integrating the thrust acceleration F over the flight time t, which in turn can be obtained as a function of the polar angle[*]:

$$\Delta V_{\mathrm{LT}} = \int_0^{t_f} F\,dt = \int_0^{\theta_f} \frac{F}{d\theta}\,d\theta \tag{7}$$

Impulsive shots are modeled to match the velocity of the departure (V_{dep}) and arrival (V_{arr}) planets to the velocity required at the initial ($V_{0,\exp}$) and final ($V_{f,\exp}$) position of the exposin:

$$\Delta V_0 = V_{0,\exp} - V_{\mathrm{dep}} \quad ; \quad \Delta V_f = V_{\mathrm{arr}} - V_{f,\exp}$$

The velocity at any point along the exposin can be computed using the radial and tangential components[†]:[10]

$$V_r = r\dot\theta \tan\gamma \quad ; \quad V_t = r\dot\theta$$

where, after some manipulation,[9] the derivative of the angular rate is found using Eq. (8).

$$\dot\theta^2 = \left(\frac{\mu}{r^3}\right) \frac{1}{\tan\gamma^2 + k_1 k_2^2 (\sin k_2\theta + \phi) + 1} \tag{8}$$

The velocity vectors along the exposin should be converted from polar coordinates to Cartesian coordinates to match the velocities of the planets.

Propellant Mass Fraction Computation

A second measure for the efficiency of trajectories is the propellant mass fraction J_m:[11]

$$J_m = 1 - \exp\left(-\frac{\Delta V_0 + \Delta V_f}{I_{\mathrm{sp,chem}} \cdot g_0} - \frac{\Delta V_{\mathrm{LT}}}{I_{\mathrm{sp,LT}} \cdot g_0}\right) \tag{9}$$

where $I_{\mathrm{sp,chem}}$ and $I_{\mathrm{sp,LT}}$ are the specific impulses for the chemical and low-thrust engine respectively, and g_0 is the standard gravitational acceleration. As payload mass increases when propellant mass fraction decreases, the most optimal trajectories are those with the lowest value of J_m. The efficiency of the trajectories depends not only on the total ΔV, but also on the relative magnitude of ΔV_{LT} and $\Delta V_0 + \Delta V_f$, so it could be argued that J_m is a better parameter to assess the optimality of trajectories.

GENERATION OF TRAJECTORIES

In order to train any GP model, training data must be generated. In this work, the exposin is implemented according to the multi-revolution Lambert's problem to model low-thrust trajectories. The problem is reduced by assuming a transfer using an exposin with one revolution ($N = 1$). The goal is to locate regions of near-optimal trajectories within a defined three-dimensional input space $[t_0, k_2, \mathrm{TOF}]$. It is assumed that the spacecraft has a mass of 1000 kg and is equipped with both an ion propulsion engine and a chemical rocket engine, with specific impulses of 3000 s and 350 s, respectively. The maximum achievable thrust acceleration ($F_{\max}$) of the ion propulsion engine is taken as $3 \cdot 10^{-4}$ km/s^2.[1]

Input vectors with uniform randomly distributed values for t_0, k_2, and TOF, within defined bounds, are fed into a program written by the authors, which computes the corresponding exposin for each input vector, using the approach outlined above. With the k_2 and t_0 values defined by the input vector, a class of exposins can be computed if any exist.[9] The Regula Falsi method[12] is then used to find the unique exposins that matches the required TOF within this class; the TOF is evaluated using the Midpoint method.[13] The data generation program is implemented using the ECLIPJ2000 reference frame with the Sun as center of the reference frame. The JPL DE405 ephemerides are used to evaluate the ephemerides of the departure and arrival planets at times

---

[*]Subscripts: 0 = initial value, f = final value, LT = low-thrust
[†]Subscripts: r = radial component, t = tangential component

t_0 and t_0+TOF, respectively.[*] As in the work of Izzo,[3] the centers of mass of the arrival and departure planets are taken, respectively, as the r_0 and r_f positions. One-way light time and stellar aberration corrections are applied to the states of the planets. With the unique exposin defined, the ΔV and J_m values corresponding to this trajectory can be computed. The Midpoint method[13] is used again for the numerical integration of ΔV_{LT}. Using the criterion, as defined by Gondelach,[1] of a 0.1% accuracy in ΔV required to rank trajectories, 1000 integration steps have been selected. As the Midpoint method is a one-step method, first the thrust acceleration required along the low-thrust arc is evaluated at 1000 points along the trajectory, using Eq. (2) and Eq. (5). The impulsive velocities can be computed by matching the velocities of the exposin to those of the planets,[9] after which J_m can be found. As the main purpose of the exposin is to provide an initial guess for a detailed numerical optimization, only the gravitational acceleration of the Sun is taken into account; all other perturbations are ignored, which is in accordance with the work of both Petropoulus[2] and Izzo.[3]

Finally, three criteria have been selected to classify whether a specific input vector specifies a feasible exposin:
(1) the condition $k_1 k_2^2 < 1$ should be fulfilled,[2]
(2) the difference between the required and actual TOF of a trajectory must be smaller than 1 second, and
(3) the thrust profile required to follow the low-thrust arc should not exceed the maximum available thrust.
The program used to generate training data, described above, is referred to as "the data generation program".

GAUSSIAN PROCESS MODELS FOR PREDICTION OF OPTIMAL TRAJECTORIES

Due to the constraints on feasible trajectories as discussed in the previous section, not all input vectors contain an exposin that can actually be flown. Before predicting near-optimal trajectories in terms of ΔV and J_m from a set of candidate input vectors, each of the input vectors is first checked for correspondence to a feasible exposin. This is where classification using a GPC model becomes useful. The mapping function, which associates the "feasible" or "not feasible" output with the input vector, is defined as the predicting target, where $\boldsymbol{x} = [t_0, k_2, \text{TOF}]$ and $Q = -1$ (not feasible) or $Q = 1$ (feasible):

$$f_{\text{GPC}} : \mathbb{R}^3 \to \mathbb{R},$$
$$\boldsymbol{x} \mapsto Q. \tag{10}$$

Once each input vector is classified as feasible or not feasible, a GPR model is used to predict the ΔV or J_m values corresponding to the feasible trajectories:

$$f_{\text{GPR},1} : \mathbb{R}^3 \to \mathbb{R},$$
$$\boldsymbol{x} \mapsto \Delta V \quad \text{or} \quad \boldsymbol{x} \mapsto J_m. \tag{11}$$

where $\boldsymbol{x} = [t_0, k_2, \text{TOF}]$. This model will be referred to as "GPR model 1"; it directly maps the input vector to a ΔV or J_m value, so it is only trained with shape parameter k_2. To investigate the applicability of the GP-based method, a second GPR model has been developed ("GPR model 2") that maps the entire shape information onto a ΔV or J_m value, thereby replacing only the integration step, where $\boldsymbol{x} = [r_0, \psi, k_0, k_1, k_2, \phi]$:

$$f_{\text{GPR},2} : \mathbb{R}^6 \to \mathbb{R},$$
$$\boldsymbol{x} \mapsto \Delta V \quad \text{or} \quad \boldsymbol{x} \mapsto J_m. \tag{12}$$

For a GP model, the prediction of an output value given an input can be achieved using Bayesian inference operating on a GP in function space. In a GP, every random collection of variables has a multivariate normal distribution.[14] In this work, the input vector $\boldsymbol{x}$ is considered to be the random variable following a Gaussian distribution. A GP over the function $f(\boldsymbol{x})$ is defined by:[14]

$$f(\boldsymbol{x}) \sim GP(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}^*)) \tag{13}$$

where $\mu(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}^*)$ are the mean function and covariance function, respectively, and $\boldsymbol{x}^*$ represents a point in the input domain[†]. The GP does not contain any prior information until some data is observed. It

---
[*]https://naif.jpl.nasa.gov/naif/toolkit.html
[†]Superscripts: □* = prediction data from Gaussian Process

is therefore necessary to train any GP model with information provided by training samples. For M training samples, a training set is constructed as:

$$\mathcal{D} = \{\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_M}]^T, \boldsymbol{Y} = [y_1, y_1, ..., y_M]^T\} \tag{14}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the training input matrix and output vector, respectively*. The training inputs $\boldsymbol{x}_i, 1 \leq i \leq M$ are generated with a random uniform distribution in the input domain and are normalized between 0 and 1. $\boldsymbol{Y}$ is considered to be free of noise as the training outputs are obtained by the data generation program.

It is assumed that the ΔV and J_m values of M* input vectors need to be assessed. The predicting input and output vectors are defined as $\boldsymbol{X^*} = [\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, ..., \boldsymbol{x}_{M^*}^*]$ and $\boldsymbol{Y^*} = [\Delta V_1^*, \Delta V_2^*, ..., \Delta V_{M^*}^*]$ or $\boldsymbol{Y^*} = [J_{m_1}^*, J_{m_2}^*, ..., J_{m_{M^*}}^*]$. The joint distribution of the training $\boldsymbol{Y}$ and predicting outputs $\boldsymbol{Y^*}$ is then defined as:

$$\begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{Y^*} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\boldsymbol{X}) \\ \mu(\boldsymbol{X^*}) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{X}, \boldsymbol{X}) & K(\boldsymbol{X}, \boldsymbol{X^*}) \\ K(\boldsymbol{X^*}, \boldsymbol{X}) & K(\boldsymbol{X^*}, \boldsymbol{X^*}) \end{bmatrix} \right) \tag{15}$$

where the covariance matrix $K(\boldsymbol{X}, \boldsymbol{X})$ is given by:

$$K(\boldsymbol{X}, \boldsymbol{X}) = \begin{bmatrix} k(\boldsymbol{x_1}, \boldsymbol{x_1}) & k(\boldsymbol{x_1}, \boldsymbol{x_2}) & ... & k(\boldsymbol{x_1}, \boldsymbol{x_M}) \\ k(\boldsymbol{x_2}, \boldsymbol{x_1}) & k(\boldsymbol{x_2}, \boldsymbol{x_2}) & ... & k(\boldsymbol{x_2}, \boldsymbol{x_M}) \\ ... & ... & ... & ... \\ k(\boldsymbol{x_M}, \boldsymbol{x_1}) & k(\boldsymbol{x_M}, \boldsymbol{x_2}) & ... & k(\boldsymbol{x_M}, \boldsymbol{x_M}) \end{bmatrix} \tag{16}$$

The likelihood function defines the probability density of the observations given the parameters, and inference allows the prediction of new targets given a dataset and the associated GP model.[14]

Model Development Procedure

Two sufficient conditions are met to develop a GP model: (1) the probability of obtaining correct training outputs given the model is maximal, (2) the errors in predicting outputs are limited to an acceptable level.[6]

A GP is defined by its mean function, covariance function, likelihood function, and inference method, and a combination of them should be found where (1) and (2) are satisfied. The selected mean function, covariance function, and likelihood function will contain parameters, called the "hyperparameters", which should be optimized in order to improve the performance. Thus "model development" in this paper refers to (a) the selection of the mean, covariance, and likelihood functions and the inference method, and (b) the optimization of the hyperparameters.

To assess the performance of the developed GPR models, three error measures ε were used: the mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean square error (RMSE), given by Equations 17–19.

$$\varepsilon_{\text{MAE}}(M) = \frac{1}{M_T} \sum_{i=1}^{M_T} |T_{\text{DG},i} - T_{\text{GPR},i}| \tag{17}$$

$$\varepsilon_{\text{MAPE}}(M) = \frac{100\%}{M_T} \sum_{i=1}^{M_T} |\frac{T_{\text{DG},i} - T_{\text{GPR},i}}{T_{\text{DG},i}}| \tag{18}$$

$$\varepsilon_{\text{RMSE}}(M) = \sqrt{\frac{1}{M_T} \sum_{i=1}^{M_T} (T_{\text{DG},i} - T_{\text{GPR},i})^2} \tag{19}$$

---

*$\square^T$ = transpose of a vector

In the above equations, M is the training set size, M_T the test set size, and $T_{\text{DG},i}$ and $T_{\text{GPR},i}$ the target values (ΔV or J_m) obtained from the data generation program and the GPR model, respectively. For the GPC model only the MAPE value is of interest, Eq. (20).

$$\varepsilon_{\text{MAPE,GPC}}(M) = \frac{100\%}{M_T} \sum_{i=1}^{M_T} \frac{|Q_{\text{DG},i} - Q_{\text{GPR},i}|}{2} \tag{20}$$

In theory, one could define infinitely many mean functions and covariance functions, and thus infinitely many GP models. As we cannot theoretically define the optimal GP model for a specific application, and using a trial and error procedure is generally highly time consuming, a model development procedure is proposed which consists of three phases: (i) a Preliminary model selection phase, (ii) a Combined model selection phase, and (iii) a Hyperparameter optimization phase. During the first two phases, the same method is used, as visualized in Figure 1. The details of each selection phase are discussed in the following subsections.
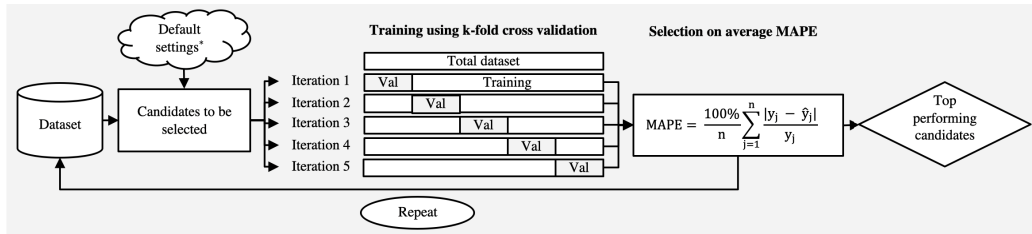

Figure 1: Model selection and training method

Preliminary Model Selection Phase During the preliminary phase, candidates which are promising in terms of MAPE are separately sought for the mean function, the covariance function, the likelihood function, and the inference method. Each of them is defined as a "candidate to be selected", as referred to in Figure 1. While this procedure is performed for one candidate (e.g. the mean function), default settings are used for the other candidates. The selected default settings are: no mean function, a squared exponential covariance function with automatic relevance determination, a Gaussian likelihood function, and Gaussian inference for both training and prediction. The dataset contains 250 samples on which k-fold cross validation is applied[14] such that $k - 1$ subsets are selected for training and the remaining subset for validation, i.e. defining the accuracy of the models. This process is repeated in total k times; in this work, $k = 5$. For each individual candidate, all possible functions that were provided by the gpml toolbox in MATLAB are evaluated where, for the composite functions, up to two base functions are combined.* The procedure is repeated four times with different data sets to increase the robustness of validation, after which the average MAPE for each function is computed. For the mean function, covariance function, and inference method, the five functions resulting in the lowest MAPE values are selected. The single best likelihood function is selected.

Combined Model Selection Phase In the combined model selection phase, the top five functions for the mean function, covariance function, and inference method, and the top likelihood function, as selected in the preliminary phase, are combined into a GP model. In total, 45 different GP models have to be evaluated according to the procedure depicted in Figure 1, where the GP model is the "candidate to be selected". A dataset containing 1250 samples is used and the procedure is repeated once, after which the optimal GP model based on MAPE value is selected. One could optionally take the CPU time for prediction into account as a second selection criterion.

Hyperparameter Optimization During the third phase model training is performed, in which the values of the hyperparameters are selected. The most common way to optimize the hyperparameters (components of the vector $\boldsymbol{\theta}$) is by using the maximum likelihood method, where the optimal values of the hyperparameters

---

*http://www.gaussianprocess.org/gpml/code/matlab/doc/

are found by minimizing the following objective function:[14]

$$J(\boldsymbol{\theta}) = -\log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{Y}^T K^{-1}(\boldsymbol{X}, \boldsymbol{X})\boldsymbol{Y} + \frac{1}{2}\log|K(\boldsymbol{X}, \boldsymbol{X})| + \frac{M}{2}\log 2\pi$$

with p the likelihood function. To keep the procedure consistent, all hyperparameters are initialized with zeros, after which the conjugate gradient method is used to find optimal values.[14] 1000 training samples are used, and the validation is performed by 1000 test samples. It is expected that multiple local optima exist for the values of the hyperparemeters. Although a gradient-based marginal likelihood optimization method might converge to a local optimum instead of the global one, it is chosen due to its fast execution. Furthermore, as the optima found for the values of the hyperparameters result in satisfactory behavior, it is decided that using a more powerful optimization technique is not required. In other work,[15] random searches have been advised for the optimization of hyperparameters. However, due to the large number of hyperparameters of the models under consideration, this approach is unattractive for preliminary optimization as it would require long CPU times to find optimal combinations of hyperparameters.

Training Set Size

With the GP models developed, the number of training samples must be determined. Training any GP model with a larger number of samples will generally improve performance. As a consequence, however, the CPU time needed for prediction will also increase, so it's a trade-off given that one is unable to define the optimal training set size.[16] Therefore, the MAE is taken as the evaluation criterion for determining the training set size of the regression models.[6] The training set size is initialized to 50 samples and is increased until a balance between prediction time and accuracy is reached using the following incremental update equation $M_{s+1} = M_s + m$, where m is the specified increment of training samples, taken as 50. For every M_s training samples, $n + 1$ MAEs are evaluated using:

$$E(M_s, m, n) = [\varepsilon_{\text{MAE}}(M_s), \varepsilon_{\text{MAE}}(M_s + m), ..., \varepsilon_{\text{MAE}}(M_s + nm)]^T$$

where n is set to 10. Next, the difference between the maximum and minimum values in vector E is computed:

$$\Delta\varepsilon_{\text{MAE}} = \max(E(M_s, m, n) - \min(E(M_s, m, n) \tag{21}$$

For each update step, the procedure is repeated five times with different training samples, and the average values are taken, with the goal of increasing the robustness of validation. When $\Delta\varepsilon_{\text{MAE}}$ becomes smaller than a specified threshold, the increment update process is stopped. For both the regression models, this threshold is defined as 0.01 km/s when predicting ΔV and 0.001 when predicting J_m. For the classification model, the same approach is used, but as only the MAPE is of interest, $\Delta\varepsilon_{\text{MAPE,GPC}} \leq 0.1\%$ is taken as the threshold. Using this approach, it can be assured that the GP models are stable and a balance between accuracy and CPU time for prediction is achieved.

Robustness of Model Development Procedure

In this work a novel model development procedure is proposed, so a robustness analysis with respect to the choices made during this procedure (e.g. the size of the dataset) is conducted. All choices are made with the goal of optimizing the trade-off between CPU time of the development process and robustness of the validation. It should be verified whether these choices are made rationally and whether choosing a more elaborate model development procedure would result in the development of models performing better in terms of MAPE. The procedure as previously proposed will be referred to as "the baseline model development procedure" and the models developed using this procedure as the "baseline models".

Preliminary Model Selection Phase The size of the dataset used in the preliminary phase is set to 250 samples. It is expected that the larger the dataset, the more robust the validation of the models. To test whether 250 samples is enough during the preliminary phase, a comparison with the performance of the final developed models is performed when using a five times larger dataset. The baseline model development procedure is used, but with a dataset of 1250 samples during the preliminary phase. The models resulting from this modified procedure are compared to the baseline models in terms of MAPE.

Combined Model Selection Phase In the baseline model development procedure, only the top performing GP model is selected during the combined phase. However, the best performing model in the combined phase will not necessarily perform best after hyperparameter optimization is applied. Therefore, the development procedure is modified by selecting the top three performing models during the combined phase to apply the hyperparameter optimization on all three of them. All other settings are kept the same as in the baseline model development procedure. The MAPE of the models resulting from this modified development procedure are compared to that of the baseline models.

Hyperparameter Optimization As it is likely that the conjugate gradient method converges to a local optimum, it should be verified how much the final performance deviates when the hyperparameters belonging to another (local or global) optimum are selected. Therefore, while keeping the preliminary and combined stage unchanged, differential evolution is used to initialize the values of the hyperparameters, after which the conjugate gradient method is applied to determine the actual values of the hyperparameters. Combining these two methods, it is more likely to converge to the global optimum instead of a local one. The settings for the differential evolution are taken as $F_p = 0.75$, $C_r = 0.8$, and $I = 20D$, with I the number of individuals and D the dimension of the problem (i.e. the number of hyperparameters to be optimized).[17] Again, the performance of the models resulting from this modified procedure is compared to that of the baseline models.

The robustness of the choices made in each phase of the development procedure is assessed with Eq. (22):

$$\%_{\text{robustness}} = (1 - \frac{\text{MAPE}_{\text{model},x} - \text{MAPE}_{\text{model},0}}{\text{MAPE}_{\text{model},0}}) \cdot 100\% \tag{22}$$

where x and 0 indicate models resulting from the modified and baseline model development procedure.

TEST CASES

The performance of the GP-based method for the preliminary optimization of low-thrust trajectories, in terms of accuracy and CPU time, was assessed for three mission test cases. Using the proposed model development procedure, GP models were developed for these test cases: rendezvous missions from Earth to Mars, from Mars to Earth, and from Earth to Ceres. Characteristics of each target body are shown in Table 1[*].

Table 1: Characteristics of target bodies for test cases

	a (AU)	e (-)	i (deg)	Orbital period (days)	Synodic period (days)
Mars	1.5237	0.0934	1.8506	687	780
Earth	1.0000	0.0167	0.0000	365	-
Ceres	2.7671	0.0758	10.593	1682	467

Rendezvous Mission from Earth to Mars

The Earth-Mars transfer has been used before to test the performance of other shape-based methods,[1,18,19] so it is selected as a test case in this work to compare the results obtained with the developed GP models to external results computed by others. The results reported in Table 5 are used, which cover shaping techniques and DITAN. DITAN is a trajectory optimization tool and can be considered state of the art for low-thrust interplanetary trajectory design.[19] The bounds for the input space are selected as t_0: [58848, 61769] MJD and TOF: [500, 2000] days, in accordance with the work of Novak and Vasile.[18] k_2 is selected to lie within the bounds [0.01, 1], allowing for up to 50 revolutions around the Sun, a number deemed unlikely to be exceeded in practice. The launch window between 1 Jan 2020 and 31 December 2027 is large enough to contain almost four synodic periods of the Sun-Earth-Mars system, which is equal to 780 days on average.

---

[*]semi-major axis a, eccentricity e, and inclination i (with respect to the mean ecliptic and equinox of J2000), orbital period, and synodic period (with respect to Earth)

Rendezvous Mission from Mars to Earth

The developed GP models should ideally be tested against as many test cases as possible. The performance was thus also tested for a mission from an outbound to an inbound target, namely a rendezvous mission from Mars to Earth. To the authors' knowledge, no other relevant results are available for direct comparison, so the performance is compared to that of the Earth-Mars test case; the same input space is therefore chosen.

Rendezvous Mission from Earth to Ceres

The third mission test case is selected as a rendezvous mission from Earth to the asteroid Ceres. Due to its much larger inclination than Mars, it is expected to be a challenging target in terms of predicting feasible and near-optimal trajectories. As this test case has not been used before in low-thrust trajectory optimization, the same input space as for the Earth-Mars test case is chosen to allow for comparison between both test cases.

SELECTED GP MODELS

Using the baseline model development procedure as proposed above, five GP models are selected for each of the test cases: a GPC model, GPR model 1 with target value ΔV, GPR model 1 with target value J_m, GPR model 2 with target value ΔV, and GPR model 2 with target value J_m. The models developed for the Earth-Mars test case, the equations of the underlying functions, and the values of the hyperparameters found, are presented in this section. A method to assess the robustness of the developed models is also discussed.

Selected Models for a Rendezvous Mission from Earth to Mars

Classification Model   For the classification model, the mean function is selected as the sum of a constant and a polynomial mean function, with two polynomials:

$$\mu(\boldsymbol{x}) = \mu_{\text{polynomial}}(\boldsymbol{x}) + \mu_{\text{constant}}(\boldsymbol{x}) = \sum_{i=1}^{D} \sum_{j=1}^{d} a_{ij} \cdot \boldsymbol{x}_i^j + c \tag{23}$$

D is the input space dimension, d the number of polynomials, and a_{ij} and c hyperparameters to be selected.

The covariance function is selected as the sum of a piecewise polynomial covariance function with automatic relevance determination (PPard) and a squared exponential covariance function with automatic relevance determination (SEard):

$$k(\boldsymbol{x}, \boldsymbol{x}^*) = k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{PPard}} + k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{SEard}} \tag{24}$$

$$k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{PPard}} = s_{f,\text{PPard}}^2 \cdot \max(1 - r, 0)^{j+d} \cdot f(r, j) \tag{25}$$

$$k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{SEard}} = s_{f,\text{SEard}}^2 \cdot e^{-(\boldsymbol{x} - \boldsymbol{x}^*)' \cdot \text{inv}(P) \cdot \frac{\boldsymbol{x} - \boldsymbol{x}^*}{2}} \tag{26}$$

$$\text{where the distance } r \text{ is defined as:} \quad r = \sqrt{(\boldsymbol{x} - \boldsymbol{x}^*)' \text{inv}(P)(\boldsymbol{x} - \boldsymbol{x}^*)} \tag{27}$$

where the P matrix is diagonal with automatic relevance determination parameters $\ell_1^2, ..., \ell_D^2$. Furthermore, d is the number of polynomials (in this case 1), $j = \text{floor}(D/2) + d + 1$, and s_f^2 is the signal variance. The function $f(r, j)$ is equal to $1 + r(j + 1)$ for $d = 1$.

Finally, Gaussian likelihood is selected, given by Eq. (28), using Leave-One-Out (LOO) inference for training and Gaussian noise inference for prediction:

$$p(y_i | f_i) = \frac{e^{\frac{-(y_i - f_i)^2}{2s_n^2}}}{\sqrt{2\pi s_n^2}} \tag{28}$$

where f is a scalar latent function value, and s_n is the standard deviation of the noise.

The classification model has a three dimensional input vector, so the following hyperparameters must be set, where M, C, and L indicate the hyperparameters of the mean, covariance, and likelihood function, and subscripts P and S stand for "PPard" and "SEard", respectively:

$$
\begin{aligned}
\boldsymbol{\theta}_C^M &= \quad [a_{11}, a_{12}, a_{21}, a_{22}, a_{31}, a_{32}, c]^T \\
\boldsymbol{\theta}_C^C &= \quad [\ell_{1,P}, \ell_{2,P}, \ell_{3,P}, s_{f,P}, \ell_{1,S}, \ell_{2,S}, \ell_{3,S}, s_{f,S}]^T \\
\boldsymbol{\theta}_C^L &= \quad \log(s_n)
\end{aligned}
$$

Regression Model 1 for Prediction of ΔV The mean function is selected as the product of a polynomial mean function ($d = 3$) and a constant mean function:

$$
\mu(\boldsymbol{x}) = \mu_{\text{polynomial}}(\boldsymbol{x}) \cdot \mu_{\text{constant}}(\boldsymbol{x}) \tag{29}
$$

The covariance function is selected as the product of PPard ($d = 2$) and a rational quadratic covariance function with automatic relevance determination (RQard):

$$
k(\boldsymbol{x}, \boldsymbol{x}^*) = k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{PPard}} \cdot k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{RQard}} \tag{30}
$$

$$
k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{RQard}} = s_{f,\text{RQard}}^2 [1 + \frac{(\boldsymbol{x} - \boldsymbol{x}^*)' \text{inv}(P)(\boldsymbol{x} - \boldsymbol{x}^*)}{2\alpha}]^{-\alpha} \tag{31}
$$

with α a shape parameter and $f(r, j)$ in Eq. (25) equal to $1 + r(j + 2) + (j^2 + 4j + 3)/3r^2$ for $d = 2$. The likelihood function and inference method for training and prediction are selected as for the GPC model. The following hyperparameters must be set for this GPR model with a three dimensional input vector, where subscript R indicates the "RQard" covariance function:

$$
\begin{aligned}
\boldsymbol{\theta}_{R1_{\Delta V}}^M &= \quad [a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}, c]^T \\
\boldsymbol{\theta}_{R1_{\Delta V}}^C &= \quad [\ell_{1,P}, \ell_{2,P}, \ell_{3,P}, s_{f,P}, \ell_{1,R}, \ell_{2,R}, \ell_{3,R}, s_{f,R}, \alpha]^T \\
\boldsymbol{\theta}_{R1_{\Delta V}}^L &= \quad \log(s_n)
\end{aligned}
$$

Regression Model 1 for Prediction of J_m When the goal is to predict the propellant mass fraction J_m instead of ΔV, the continuous target values for the regression models differ significantly. Therefore, the model development procedure is repeated.
The mean function is selected as the second power of a constant mean function, with P the power (in this case 2):

$$
\mu(\boldsymbol{x}) = \mu_{\text{constant}}(\boldsymbol{x})^P = c^P \tag{32}
$$

The same covariance function is selected as for the prediction of ΔV, but with $d = 1$ for PPard. Instead of LOO inference for training, as selected for the model when predicting ΔV, Laplace inference is selected. The following hyperparameters must be set:

$$
\begin{aligned}
\boldsymbol{\theta}_{R1_{J_m}}^M &= \quad c \\
\boldsymbol{\theta}_{R1_{J_m}}^C &= \quad [\ell_{1,P}, \ell_{2,P}, \ell_{3,P}, s_{f,P}, \ell_{1,R}, \ell_{2,R}, \ell_{3,R}, s_{f,R}, \alpha]^T \\
\boldsymbol{\theta}_{R1_{J_m}}^L &= \quad \log(s_n)
\end{aligned}
$$

Regression Model 2 for Prediction of ΔV Selecting the mean function as the sum of a polynomial mean function ($d = 4$) and a linear mean function results in the smallest MAPE:

$$
\mu(\boldsymbol{x}) = \mu_{\text{polynomial}}(\boldsymbol{x}) + \mu_{\text{linear}}(\boldsymbol{x}) \tag{33}
$$

$$
\mu_{\text{linear}}(\boldsymbol{x}) = \sum_{i=1}^{D} c_i \cdot \boldsymbol{x}_i \tag{34}
$$

with c_i hyperparameters to be selected.

The covariance function is selected as the product of the PPard covariance function ($d = 1$) and a squared exponential covariance function with isotropic length scale (SEiso):

$$k(\boldsymbol{x}, \boldsymbol{x}^*) = k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{PPard}} \cdot k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{SEiso}} \tag{35}$$

$$k(\boldsymbol{x}, \boldsymbol{x}^*)_{\text{SEiso}} = s_{f,\text{SEiso}}^2 \cdot e^{-(\boldsymbol{x} - \boldsymbol{x}^*)' \cdot \text{inv}(P) \cdot \frac{\boldsymbol{x} - \boldsymbol{x}^*}{2}} \tag{36}$$

For the SEiso covariance function, P is ℓ^2 times the unit matrix, with ℓ an isotropic lengthscale.

In accordance with GPR model 1, Gaussian likelihood and Gaussian inference for prediction are selected; LOO likelihood is selected for training. The input space has six-dimensions, so the hyperparameters are set:

$\boldsymbol{\theta}_{R2_{\Delta V}}^M$	$[a_{11}, a_{12}, a_{13}, a_{14}, a_{21}, a_{22}, a_{23}, a_{24}, a_{31}, a_{32}, a_{33}, a_{34}, a_{41}, a_{42}, a_{43}, a_{44},$ $a_{51}, a_{52}, a_{53}, a_{54}, a_{61}, a_{62}, a_{63}, a_{64}, c_1, c_2, c_3, c_4, c_5, c_6]^T$
$\boldsymbol{\theta}_{R2_{\Delta V}}^C$	$[\ell_{1,P}, \ell_{2,P}, \ell_{3,P}, \ell_{4,P}, \ell_{5,P}, \ell_{6,P}, s_{f,P}, \ell_S, s_{f,S}]^T$
$\boldsymbol{\theta}_{R2_{\Delta V}}^L$	$\log(s_n)$

Regression Model 2 for Prediction of J_m   Only one difference exist for GPR model 2 when predicting J_m instead of ΔV: in accordance with GPR model 1, the mean function is selected as the second power of a constant mean function. The following hyperparameters must be set:

$\boldsymbol{\theta}_{R2_{J_m}}^M$	c
$\boldsymbol{\theta}_{R2_{J_m}}^C$	$[\ell_{1,P}, \ell_{2,P}, \ell_{3,P}, \ell_{4,P}, \ell_{5,P}, \ell_{6,P}, s_{f,P}, \ell_S, s_{f,S}]^T$
$\boldsymbol{\theta}_{R2_{J_m}}^L$	$\log(s_n)$

Selected Values of the Hyperparameters

The optimal values selected for the hyperparameters of the five models are given in Reference 9.

Assessing the Robustness of the Developed Models

When one wants to apply the provided GP models, referred to as "the initial models", on a slightly different mission scenario than the Earth-Mars test case as selected in this work ("the initial mission scenario"), one could either use the models provided or repeat the model development procedure. In this subsection, the robustness of the developed models (i.e. their dependency on the specified mission scenario) is assessed by testing their performance in terms of MAPE and prediction time when applied to slightly different mission scenarios. The goal is to provide some confidence in the application of these models to mission scenarios that differ from the one discussed in this paper. To assess this robustness, the initial models are tested on 15 different mission scenarios by applying three deviations along five mission design dimensions (Table 2).

While one of these mission design parameters is varied, all other settings are kept equal to the initial mission scenario. For each of the 15 new mission scenarios, training and test samples are generated. The initial models are applied on all 15 mission scenarios, with the number of training samples equal to those presented in Table 3. The difference in performance with respect to the initial mission scenario is computed, where x and 0 indicate the deviated and initial mission scenarios, respectively:

$$\%_{\text{difference}} = \frac{\text{MAPE}_{\text{scenario,x}} - \text{MAPE}_{\text{scenario,0}}}{\text{MAPE}_{\text{scenario,0}}} \cdot 100\% \tag{37}$$

NUMERICAL RESULTS AND DISCUSSION

In order to evaluate model performance, test samples with a uniform random distribution in the input space are used for all GP models. In previous work,[6,7] the ratio between training and test samples varied between

Table 2: Mission scenarios

	Initial	Deviation 1	Deviation 2	Deviation 3
i) A different range for t_0	Jan 1 2020 – Dec 31 2027	Jan 1 2028 – Dec 31 2035	Jan 1 2036 – Dec 31 2043	Jan 1 2044 – Dec 31 2051
ii) A smaller range for t_0 …	Jan 1 2020 – Dec 31 2027	Jan 1 2020 – Dec 31 2023	Jan 1 2020 – Dec 31 2027	Jan 1 2020 – Dec 31 2021
…and/or TOF	500–2000 days	500–1250 days	500–875 days	500–2000 days
iii) A larger range for t_0 …	Jan 1 2020 – Dec 31 2027	Jan 1 2020 – Dec 31 2035	Jan 1 2020 – Dec 31 2027	Jan 1 2020 – Dec 31 2051
…and/or TOF	500–2000 days	500–3500 days	500–6500 days	500–2000 days
iv) A different target planet	Mars	Ceres	Pallas	Vesta
v) A different number of revolutions (N) for the exposin	1	2	3	4
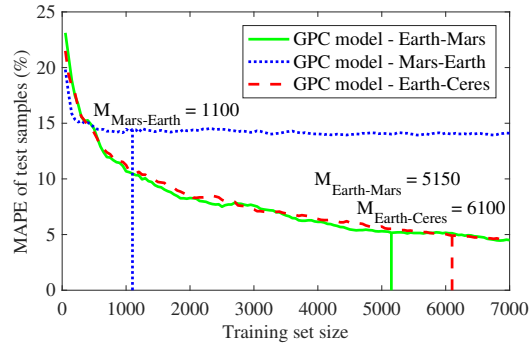
Figure 2: Relationship between MAPE of test samples and number of training samples for GPC models for all three test cases

1:1 and 1:5, and the number of required training samples is expected to lie between 1000 and 5000, so 1000 test samples are selected. The number of training samples for which a balance between accuracy and prediction time is obtained is determined using the approach outlined above. This is demonstrated in Figure 2, which shows the resulting number of training samples for the GPC models for all test cases. Note that the performance of the GPC models barely improves when more training samples are used than the number selected by the iterative procedure. The prediction time for each of the models is computed as the CPU time for the prediction of 1000 samples and is averaged over 1000 runs. Furthermore, the results are discussed for the robustness analysis of the model development procedure and for the robustness of the developed models, as outlined above. All algorithms are implemented in MATLAB 2017a and are executed on a desktop computer with an Intel 2 GHz processor and 8.0 GB memory operating on a 64-bit macOS platform. In the following, the performance of the developed GP models is discussed in terms of accuracy, CPU time, and their ability to locate regions of near-optimal trajectories, for each of the three test cases defined above.

Rendezvous Mission from Earth to Mars

Numerical Results of GP Models The determined number of training samples, the corresponding MAE, MAPE, and RMSE values, and the prediction times, are provided in Table 3. The prediction time is dependent on both the number of training samples and the complexity of the model.

Classification Model Out of the 1000 test samples that are predicted, on average 948 are predicted correctly. Out of the 52 that are predicted incorrectly, 29 infeasible trajectories are predicted as feasible and 23 feasible trajectories are predicted as infeasible. GP models have not been applied before to classification problems in the aerospace field. In several other classification applications, unrelated to this work, accuracies between 1.5% and 33% have been found.[20–23] Compared to these numbers, the MAPE of 5.16% achieved in this work is located near the lower bound of this range. However, it should be noted that these MAPE values are not the best indicator for the performance achieved in this work, but are provided here as reference for the reader regarding the range of values that could be obtained for classification problems.

Regression Models When evaluating the performance of each of the GPR models, it becomes clear that 1) the performance is better when predicting J_m instead of ΔV and 2) the performance of GPR model 1 is better than that of GPR model 2. It is expected that GPR models are, in general, better in predicting targets with a smaller ratio between the minimum and maximum target value. As the values for J_m vary between 0.48 and 1.00, and for ΔV between 6.06 and about 130 km/s, this could explain the first observation, which is referred to as "hypothesis 1". The second observation could be explained by the fact that GPR model 2 is primarily trained with shape information, while GPR model 1 is trained with parameters containing information on the input space. Therefore, GPR model 2 should be able to accurately predict ΔV_{LT}, but is expected to contain less information on the impulsive velocities required at departure and arrival. This hypothesis is referred to as "hypothesis 2".

GPR has been applied once before for the prediction of ΔV values. In the work of Shang and Liu[6] it was used to predict the ΔV values required to access main-belt asteroids using (high-thrust) transfer trajectories, and MAEs of 0.06-0.12 km/s were found. It should be noted that the values to be predicted ranged within 6-12 km/s, which is a much smaller ratio between the minimum and maximum value than the one used in this test case, from which it is expected that high accuracies could be obtained more easily. As a result, the MAEs obtained for the GPR models in this work when predicting ΔV are worse, starting from 7.49 km/s for GPR model 1. The MAPEs found in the work of Shang and Liu range between 0.58% and 1.33%, which comes close to the value obtained in this work for GPR model 1 when predicting J_m. Another way to assess the performance is in comparison with other machine learning algorithms. In the work of Li,[15] the MAPEs of several machine learning algorithms applied on low-thrust fuel optimization problems have been listed. For eight machine learning algorithms, the corresponding MAPEs varied between 0.33% and 3.66%. Similar MAPEs are found for both GPR models when predicting J_m.

Table 3: Numerical results for a rendezvous mission from Earth to Mars

Model	# training samples	MAE (km/s) or (-)	MAPE (%)	RMSE (km/s) or (-)	CPU time prediction (s)
GPC	5150	-	5.16	-	9.63
GPR 1 - ΔV	4100	7.49	23.16	13.07	6.43
GPR 1 - J_m	2350	0.01	1.38	0.02	1.45
GPR 2 - ΔV	2700	33.26	119.08	40.01	2.16
GPR 2 - J_m	2350	0.03	3.74	0.06	1.60

Optimal Trajectories The best trajectories found previously with other shape-based methods ([1,18,24]), to-gether with the optimization techniques used, are presented in Table 5. To find the overall best trajectory belonging to the exposin shape, three optimization techniques are selected: differential evolution (DE),[25] a grid search (GS), and a grid search with mesh refinements. The total ΔV values and maximum accelerations corresponding to the optimal trajectories in terms of ΔV modeled by the exposin shape are shown in Table 5. The optimal value for J_m is found using differential evolution and is given in Table 9. In this work, the globally optimal trajectory is defined as the overall best trajectory found by either the differential evolution, the grid search, or the adaptive grid search.

3700

Differential Evolution   The classical version of differential evolution (DE) is chosen, as developed by Storn and Price.[25] To determine the optimal values for the mutation probability F_p and crossover probability C_r used in the DE, F_p is varied between 0.5 and 0.8, and C_r between 0.85 and 0.95, as optimal values for comparable problems were found to lie within these ranges.[1,26] The number of individuals I turns out to be optimal between 50 and 100 based on trial and error. Using a grid of 0.05 for F_p and C_r, and 10 for I, the optimal settings are found to be 0.5, 0.85, and 100 for F_p, C_r, and I, respectively. The input vectors corresponding to the optimal trajectories found in terms of ΔV and J_m are given by $\boldsymbol{x} =$[61175.64, 0.4007, 618.04] and $\boldsymbol{x} =$ [61358.04, 0.4359, 949.05], respectively.

Grid Search   A grid search (GS) is implemented, where the grid is specified as: $t_0 = [58848 : 15 : 61769]$, $k_2 = [0.01 : 0.025 : 1]$, and TOF = $[500 : 20 : 2000]$, in line with the work of Novak and Vasile.[18] At each grid point, the ΔV and J_m values corresponding to this input vector are computed, and the optimal trajectory in terms of ΔV is found to belong to input vector $\boldsymbol{x} =$ [61173.00, 0.3850, 620.00].

Adaptive Grid Search   An adaptive grid search (AGS) is implemented, where a coarse grid is used that is refined around the obtained interim solution. The initial grid (G_0) contains 75 grid points, where five points are spaced equally for all three dimensions. After evaluation of all 75 grid points, the grid is refined around the interim solution by halving the length of the search interval (G_1). This process of refining the grid is repeated i times, until the solution at G_i is the same as G_{i-10} and therefore convergence is reached. The input vector for the optimal trajectory obtained with the AGS in terms of ΔV is given by $\boldsymbol{x} = $ [62701.02, 0.1970, 562.74].

From Table 5, it is evident that for all three optimization techniques, less optimal trajectories in terms of total ΔV are found for the exposin shape than for the other shape-based methods. As the GP models are trained with transfer trajectories based on the exposin shape, they won't be able to find trajectories with ΔV values smaller than 6.06 km/s, which is therefore used as the global optimum for comparison of GP model performance. As given in Table 9, $J_m = 0.48$ is used as the global optimum for the propellant mass fraction.

Table 4: Numerical results for the prediction of ΔV_{LT} for a rendezvous mission from Earth to Mars

Model	# training samples	MAE (km/s) or (-)	MAPE (%)	RMSE (km/s) or (-)	CPU time prediction (s)
GPR 1 - ΔV	5050	0.20	2.18	0.33	3.98
GPR 2 - ΔV	1100	0.04	0.54	0.12	0.15

When the GPC model and GPR model 1 are placed in series, it is possible to predict ΔV and J_m belonging to feasible trajectories given a set of input vectors, thereby replacing the functionality of the data generation program. The more samples that are fed to the GP models, the higher the probability that trajectories with ΔV and J_m values close to the global optima are found. On the other hand, the corresponding prediction time also increases. Figure 3 shows a trade-off between the optima found and the corresponding prediction time, at different numbers of samples to be predicted. These plots are produced as follows. The specified number of samples (e.g. 10,000) is fed to the GPC model. The GPC model is trained with the number of training samples as provided in Table 3. The trajectories classified by the GPC model as feasible are fed into GPR model 1. The predicted values for each of the input vectors are ranked, and the top 50 input vectors are selected. The actual target values are computed for these 50 input vectors, making use of the data generation program. Averaged over 50 runs, the target value of the best trajectory is given as a dot in Figure 3. The error-bar shown spans the range between the worst and best values found for the best trajectory within these 50 runs. This procedure is performed for both the prediction of ΔV and J_m. For the prediction of J_m, a decreasing average target value and increasing prediction time, at an increasing number of prediction samples, is clearly visible. Furthermore, it can be observed that the error bars get smaller for a larger number

Table 5: Required ΔV and maximum thrust accelerations F_{max} corresponding to minimum-ΔV trajectories found by different methods for a rendezvous mission to Mars

Method	Optimization technique	ΔV [km/s]	F_{max} $[10^{-4}$ m/s$^2]$
Hohmann[19]	-	5.50	-
Hodographic - time[1]	Nelder-Mead	5.77	1.5
Hodographic - polar angle[1]	Nelder-Mead	5.81	1.6
Spherical[18]	Grid Search	5.74	2.2
Pseudo-equinoctial[19]	Evolutionary Branching	5.83	1.6
DITAN[19]	Direct Finite Element Transcription + Sparse optimizer	5.66	1.5
Exponential sinusoid	Differential Evolution	6.24	1.2
Exponential sinusoid	Grid Search	6.25	1.2
Exponential sinusoid	Adaptive Grid Search	6.06	1.2

of samples to be predicted, indicating that the reliability of the average J_m value increases. Such a clear trend is not observed for the prediction of ΔV, and the target values barely get better with increasing samples to be predicted, although the average prediction time does increase. This observation could be explained by the fact that the accuracies achieved for the prediction of J_m are much better than those for the prediction of ΔV, and the resulting optimal trajectories are therefore more reliable. Additionally, it becomes clear from Figure 3a that the optimal trajectories found deviate somewhat from the global optima, which could be explained as follows. When producing 100,000 randomly generated input vectors, the smallest ΔV values found generally lie between 6.25 and 6.30 km/s and the smallest J_m values between 0.49 and 0.52. Furthermore, as the training set contains only 4100 or 2350 training samples for the prediction of ΔV and J_m respectively, the models are likely not trained with (near-)globally optimal trajectories. If the goal is to obtain globally optimal trajectories, it should be investigated how the performance of the GPR models could be improved by adding the globally optimal trajectories to the training set.
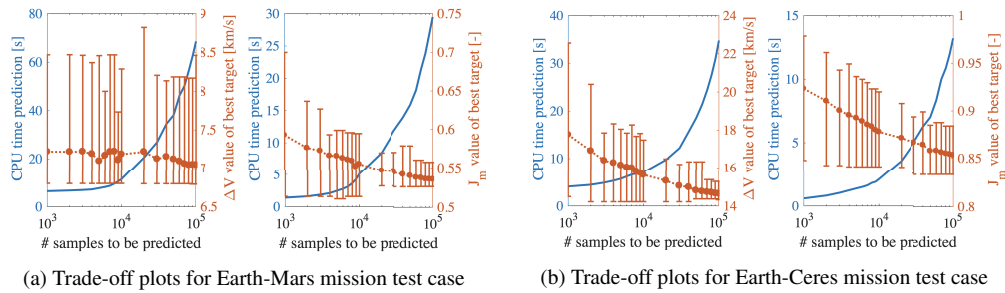

(a) Trade-off plots for Earth-Mars mission test case   (b) Trade-off plots for Earth-Ceres mission test case

Figure 3: Trade-off plots

Although the GP models are not most suitable for finding globally optimal trajectories, they are a powerful tool for the prediction of a large input space with the goal of locating the regions of feasible or near-optimal trajectories. Classifying the feasibility of 1,000,000 candidate input vectors, and predicting the ΔV or J_m values corresponding to the feasible trajectories, is achieved within a total computation time for prediction

of less than 1400 seconds. If assessing these values with the data generation program, this would take more than one week. Training the models takes about 30 s CPU time per model, but as the hyperparameters are available (see Reference 9), training the model is unnecessary and only prediction has to be repeated. The CPU time required to generate the training samples takes 0.7 s per trajectory times the amount of training samples required, in this case 5150. For these purposes, the CPU time can be reduced approximately 150 times. It should be noted that for an increasing number of candidates to be assessed, the reduction in CPU time is even larger as prediction time does not increase linearly with prediction samples, which is evident in Figure 3a.

Predicting ΔV Along the Low-Thrust Arc The total ΔV required for an interplanetary transfer is composed of a part delivered by the chemical engine and a part along the the low-thrust arc, delivered by the ion propulsion engine. To prove hypothesis 2, used to explain why GPR models 1 are performing better than GPR models 2, the prediction only of ΔV_{LT} for the Earth-Mars mission test case is discussed in this subsection. As only ΔV_{LT} has to be predicted, J_m is linearly related to this value and is therefore not of importance. The model development procedure is repeated for GPR model 1 and GPR model 2, with ΔV_{LT} as target value. The same input space as for the Earth-Mars mission test case is used.

The determined number of training samples and the numerical results achieved for both models are presented in Table 4. It can be observed that the performance of GPR model 2 is significantly better than that of GPR model 1, which is contrary to the performance of the GPR models for target value ΔV. This observation could be explained by the fact that GPR model 2 is trained with full shape information (input vector defined as $\boldsymbol{x} = [r_0, \psi, k_0, k_1, k_2, \phi]$), and is therefore able to accurately predict the value for ΔV_{LT}. As for most transfer trajectories the largest portion of the total ΔV is composed of the impulsive ΔVs, the performance of GPR model 2 is much worse when predicting ΔV. The values found for ΔV_{LT} for the Earth-Mars mission test case range within about 3.20 km/s and 16.60 km/s. As the ratio between minimum and maximum target values is much smaller than for the target value ΔV, and better performance is reached for both GPR models when predicting ΔV_{LT}, the results obtained for this test case support hypothesis 1 as well.

Rendezvous Mission from Mars to Earth

Numerical Results of GP Models The numerical results are given in Table 6. It is observed that for all models slightly worse performance is achieved than for the Earth-Mars test case. Better performance is obtained for GPR models 1 than GPR models 2, and the models perform significantly better when predicting J_m instead of ΔV, which is in line with the Earth-Mars test case. Hypotheses 1 and 2 are supported by both observations.

Optimal Trajectories The optimal trajectories in terms of ΔV and J_m are provided in Table 8 and Table 9, respectively, together with the corresponding input vectors at which these value are achieved. The DE, GS, and AGS are applied on this optimization problem, and the optima are found using the DE. If one would produce the trade-off plots for the Mars-Earth test case, results comparable to those shown in Figure 3 are observed. In line with the Earth-Mars test case, it can be concluded that the GP models are not perfectly suited for the computation of globally optimal trajectories, but they significantly outperform other optimization techniques in efficient exploration of a large search space for feasible and near-optimal trajectories.

Rendezvous Mission from Earth to Ceres

Numerical Results of GP models The numerical results achieved for the Earth-Ceres mission test case are presented in Table 7. The performance achieved for the GPC model is slightly better than the one reached for the Earth-Mars test case. The GPR models show significantly better performance than for the other two test cases. Using hypothesis 1, this could be explained by the fact that for the Earth-Ceres test case the ratios between minimum and maximum values of ΔV and J_m are smaller. Consistent with the other two test cases and in accordance with hypothesis 2, GPR models 1 show better performance than GPR models 2.

Optimal Trajectories The optimal trajectories in terms of ΔV and J_m values are provided in Table 8 and Table 9, respectively. The DE, GS, and AGS are applied to this optimization problem, and the optima are found using the DE. The trade-off between the number of prediction samples and target values found is

Table 6: Numerical results for a rendezvous mission from Mars to Earth

Model	# training samples	MAE (km/s) or (-)	MAPE (%)	RMSE (km/s) or (-)	CPU time prediction (s)
GPC	1100	-	13.55	-	0.31
GPR 1 - ΔV	5000	9.35	31.24	14.75	11.22
GPR 1 - J_m	2350	0.02	2.14	0.03	1.41
GPR 2 - ΔV	1100	47.82	187.06	49.10	0.28
GPR 2 - J_m	1000	0.06	7.12	0.09	0.17

Table 7: Numerical results for a rendezvous mission from Earth to Ceres

Model	# training samples	MAE (km/s) or (-)	MAPE (%)	RMSE (km/s) or (-)	CPU time prediction (s)
GPC	6100	-	4.90	-	12.22
GPR 1 - ΔV	3900	3.96	9.03	7.36	4.02
GPR 1 - J_m	2100	0.01	0.55	0.01	0.57
GPR 2 - ΔV	2100	37.07	96.07	38.78	1.21
GPR 2 - J_m	2250	0.02	1.63	0.02	0.43

Table 8: Globally optimal trajectories in terms of ΔV for all test-cases using the traditional optimization techniques (DE, AGS, and GS)

Mission testcase	ΔV (km/s)	t_0 (MJD)	k_2 (-)	TOF (days)
Earth-Mars	6.06	62701.02	0.1970	562.74
Earth-Mars (low-thrust arc)	3.15	61263.21	0.5837	507.33
Mars-Earth	6.34	61157.18	0.4341	580.45
Earth-Ceres	13.85	60290.56	0.5506	1007.22

Table 9: Globally optimal trajectories in terms of J_m for all test-cases using the traditional optimization techniques (DE, AGS, and GS)

Mission testcase	J_m (-)	t_0 (MJD)	k_2 (-)	TOF (days)
Earth-Mars	0.48	61358.04	0.4359	949.05
Mars-Earth	0.39	59599.01	0.3728	651.99
Earth-Ceres	0.76	59197.22	0.2639	1178.24

shown in Figure 3b. It is evident that both the ΔV and J_m values steadily decrease for an increasing number of samples to be predicted, and as the error-bars are getting smaller, the reliability increases. The difference between Figures 3a and 3b for target value ΔV could be explained by the higher accuracies reached for the Earth-Ceres than for the Earth-Mars test case. Although the results for finding globally optimal trajectories are better for the Earth-Ceres mission test case than for the other two, the main strength of the GP models is still found in the efficient evaluation of a large input space.

Robustness of the Model Development Procedure

The robustness analysis as detailed above is applied to both the GPC model and GPR model 1 with target value ΔV, for the rendezvous mission from Earth to Mars. The development procedure for the GPR model turned out to be highly robust, especially in the preliminary and combined phase, where (according to Eq. (22)) a robustness of more than 99% is reached. A robustness of 96.1% is obtained for the hyperparameter optimization phase after performing 50 iterations, which takes more than 100 hours. For the application of the proposed procedure on the development of a GPC model, a robustness of more than 99% is again achieved in the preliminary phase. The robustnesses achieved in phases 2 and 3, of respectively 92% and 86%, are lower than the ones achieved for the GPR model. It should be taken into account that the deviated procedure requires significantly more CPU time than the baseline model development procedure.

Robustness of the Developed Models

The numerical results achieved with the initial GP models are assessed for all 15 mission scenarios of the Earth-Mars test case. Along the five mission design dimensions as specified above, the results averaged over the three deviations are taken. The MAPE values obtained for the averaged deviated mission scenarios, together with those for the initial mission scenario, are provided for all GP models in Figure 4 (left). The right plot in Figure 4 shows the relative difference in MAPE with respect to the initial mission scenario, which is computed using Eq. (37). Positive percentages indicate a larger MAPE and therefore worse accuracies. It can

be observed that for all models, better performance is achieved when they are applied on a mission scenario with a smaller range for t_0 and/or TOF. This result is expected, as fewer training samples are generally required for a smaller range between the input bounds.[14] Higher accuracies could be obtained because the same numbers of training samples as for the initial mission scenario are used. When the models are applied on a test case with a different range for t_0, but of the same size, the performance is comparable to that of the initial mission scenario. Worse performance is obtained when the models are applied on mission scenarios with a larger range for t_0 and/or TOF, as generally more training samples are required. When the models are applied on mission scenarios with a different target planet, better performance is achieved for all regression models. As the ratio between minimum and maximum ΔV and J_m values for missions to Ceres, Pallas, and Vesta is smaller than for a mission to Mars, the higher accuracies obtained for the GPR models support hypothesis 1. For mission scenarios with a different number of revolutions, significantly worse performance is observed.
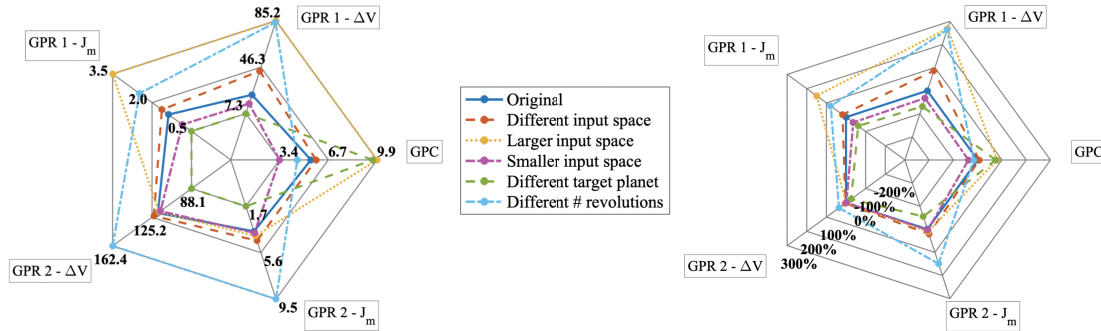

Figure 4: Spider plots indicating the robustness of the developed models for the Earth-Mars test case. Left: MAPE values, Right: relative difference in MAPE with respect to initial mission scenario (taken as 0%)

CONCLUSIONS

In this paper, a novel procedure is proposed for the development of models utilizing GP regression and classification to perform computationally efficient, preliminary optimization for direct, low-thrust trajectories. The low-thrust trajectories are modeled using the exponential sinusoid shape. The analysis performed in this work demonstrates that the proposed procedure is (highly) robust, especially where the development of GPR models is concerned. GP models have been developed for test cases from Earth to Mars and Ceres, and from Mars to Earth. In all test cases, the main advantage of the GP-based method is its efficiency in locating regions of feasible and near-optimal trajectories, especially when a large input space is explored. For the evaluation of 1,000,000 candidate trajectories, the GP-based prediction method is approximately 150 times faster than numerical trajectory computation. This speed advantage increases for an increasing number of candidates.

GPR models trained with only three input space parameters showed higher accuracies than the ones trained with full shape information. This observation is caused by the fact that the shape-trained GPR models can accurately predict ΔV_{LT}, but are not informative about the impulsive ΔVs. Additionally, significantly higher accuracies were achieved when predicting propellant mass fraction values (J_m) instead of ΔV. As one of the main objectives of space missions is to decrease J_m, models able to accurately predict J_m could be of significant importance in the preliminary design of low-thrust missions. From the results presented in this paper, a hypothesis is developed which states that GPR models can reach higher accuracies when the ratio between the minimum and maximum target values gets smaller. This hypothesis is supported by the results of all three test cases, but further work is necessary to confirm this.

MAPE values obtained for the regression models with target value J_m ranged between 0.55% and 2.14%, which is in line with other work.[6] For the classification of feasible exposins, MAPE values between 4.90%

and 13.55% were achieved. When GPC and GPR models are placed in series, it is possible to predict (regions of) near-optimal trajectories while requiring only three input parameters. Trajectories located in these regions may serve as initial guesses for more refined optimization techniques. The models presented have been tested for dependency on the Earth-Mars test case presented in this paper. Comparable results, in terms of MAPE and prediction time, were achieved when the models were applied on a mission scenario with a different or smaller range for t_0 and/or TOF or a different target planet than the Earth-Mars mission scenario.

REFERENCES

[1] D. J. Gondelach and R. Noomen, "Hodographic-Shaping Method for Low-Thrust Interplanetary Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 52, No. 3, 2015, pp. 728–738.

[2] A. E. Petropoulos and J. M. Longuski, "A shape-based algorithm for the automated design of low-thrust, gravity-assist trajectories," *Advances in the Astronautical Sciences*, Vol. 109 III, No. 5, 2002, pp. 2321–2336.

[3] D. Izzo, "Lambert's problem for exponential sinusoids," Tech. Rep. April, ESA/ESTEC, 2005.

[4] D. G. Krige, "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand," *Journal of the Chemical Metallurgical & Mining Society of South Africa*, Vol. 52, No. 6, 1951.

[5] R. Dufour, J. d. Muelenaere, and A. Elham, "Trajectory driven multidisciplinary design optimization of a sub-orbital spaceplane using non-stationary Gaussian process," *Structural and Multidisciplinary Optimization*, Vol. 52, No. 4, 2015, pp. 755–771.

[6] H. Shang and Y. Liu, "Assessing Accessibility of Main-Belt Asteroids Based on Gaussian Process Regression," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 5, 2017, pp. 1144–1154.

[7] A. Gao and W. Liao, "Efficient Gravity Field Modeling Method for Small Bodies Based on Gaussian Process Regression," *Acta Astronautica*, Vol. 157, 2019, pp. 73–91.

[8] A. Gao, G. Y. Wang, S. S. Wu, and T. Song, "Efficient Evaluation of Mars Entry Terminal State Based on Gaussian Process Regression Efficient," *IOP Conference Series: Materials Science and Engineering*, 2018.

[9] L. Bouwman, "Gaussian Process Models for Preliminary Low-thrust Trajectory Optimization," Master's thesis, Delft University of Technology, 2019.

[10] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. Space Technology Library, 2nd ed., 2013.

[11] M. D. Vasile, O. Schütze, O. Junge, G. Radice, "Spiral Trajectories in Global Optimisation of Interplanetary and Orbital Transfers Table of Contents," Tech. Rep. 5, 2006.

[12] M. Dowell and I. Jarratt, "A Modified Regula Falsi Method for Computing the Root of an Equation," *BIT Numerical Mathematics*, Vol. 11, No. 2, 1971, pp. 168–174.

[13] P. C. Hammer, "The Midpoint Method of Numerical Integration," *Mathematics Magazine*, Vol. 31, No. 4, 1958, pp. 193–195.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, Vol. 14. Massachusetts Institute of Technology, 2004.

[15] H. Li, S. Chen, D. Izzo, and H. Baoyin, "Deep Networks as Approximators of Optimal Transfers Solutions in Multitarget Missions," *arXiv*, 2019, arXiv:1902.00250.

[16] J. Pericchi, O. Berger, and R. Luis, "Training Samples in Objective Bayesian Model Selection," *The Annals of Statistics*, Vol. 32, No. 3, 2004, pp. 841–869.

[17] M. Vasile and E. Minisci, "Analysis of Some Global Optimization Algorithms for Space Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 47, No. 2, 2010.

[18] D. M. Novak and M. Vasile, "Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, 2011, pp. 128–147.

[19] P. d. Pascale and M. Vasile, "Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, 2006, pp. 1065–1076.

[20] K. Kapoor, A.and Grauman, R. Urtasun, and T. Darrell

[21] M. Kuss and C. E. Rasmussen, "Assessing Approximate Inference for Binary Gaussian Process Classification," *Journal of Machine Learning Research*, Vol. 6, 2005, pp. 1679–1704.

[22] R. M. Neal *Bayesian Statistics*, p. 475.

[23] C. K. I. Williams and D. Barber, "Bayesian Classification With Gaussian Processes," Vol. 20, No. 12, 1998, pp. 1342–1351.

[24] M. Vasile, O. Schütze, and O. Junge, "Spiral trajectories in global optimisation of interplanetary and orbital transfers," *ESA Technical Report*, Vol. 31, 2006, pp. 0–45.

[25] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, Vol. 11, No. 4, 1997, pp. 341–359.

[26] J. Englander, "Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions," *University of Illinois at Urbana-Champaign*, 2013.