• **Article** •

*** 20** Vol.** No.*: ***–***
doi: 10.1007/s11431-****

Hybrid method for accurate multi-gravity-assist trajectory design using pseudostate theory and deep neural networks

YANG Bin[1], FENG Jinglang[2], HUANG Xuxing[1], LI Shuang[1*], Omar Diab Pascual[1]

[1]*Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China;*
[2]*University of Strathclyde, Glasgow, Scotland G1 1XJ, United Kingdom*

Received ***, 20**; accepted ***, 20**; published online ***, 20**

This paper presents a novel hybrid method to design the continuous and accurate multi-gravity-assist trajectory for a high-fidelity dynamics. The gravitational perturbation of the primary body is considered during the gravity assistance. The pseudostate technique is applied to approximate the gravity-assisted trajectory, where the optimal sweepback duration is solved using a trained deep neural network. The major factors that affect the optimal sweepback duration of the approach and departure segments are investigated. The results show that the optimal sweepback duration of the approach segment only relies on the shape of the approach trajectory and is independent of the flight time. Then, a gravity-assisted trajectory patched strategy and a hybrid algorithm combining the particle swarm optimization and the sequential quadratic programming are developed to optimize the multi-gravity-assist trajectory. The proposed hybrid method is applied to the Europa orbiter mission. In comparison with the traditional patched conic method, this method demonstrates outstanding performance on accuracy and significantly reduces the computational time and complexity of the trajectory correction with the high-fidelity dynamics.

Pseudostate theory, multiple gravity assist, deep neural network, trajectory optimization, Jovian system

Citation: Du Y C, Huang X P, Zhang K Y, et al. Thermal stability of LiFePO$_4$/C-LiMn$_2$O$_4$ blended cathode materials. Sci China Tech Sci, 20**, ****, doi: 10.1007/s11431-****

1 Introduction

Gravity assist (GA) technology is a significantly effective approach to save fuel on interplanetary missions and has been widely applied in actual missions such as Mariner 10 [1], Galileo [2], and Cassini [3] et.al. The application of gravity assist, especially multiple gravity assistances (MGA), makes the mission trajectory complicated and diverse [4]. Meanwhile, it's also a significant challenge to design these trajectories because that there are many variables to be optimized.

Since gravity assist technology was first proposed in the 1950s [5], it has always been one of the research popularities in astrodynamics. Minovitch [6] developed the patched conic method (PCM) and provided a convenient and efficient approach to solve the gravity-assist trajectory. Then, many optimization methods based on the PCM and its improvements have been proposed for designing gravity-assist trajectories [7-9]. Prado [8] presented the powered swing-by method, which added an impulse maneuver during the GA and showed better performance than the pure GA. Zimmer and Ocampo [9] derived analytical derivatives of the cost and constraint functions with respect to the free parameters using the state transition matrix method, which was used to address gravity-assisted trajectory optimization. These methods improved the performance of GA trajectory design. However, the challenge is MGA trajectory optimization, which has no routine approach so far. Miller [10] and Strange [11] applied graphical methods based on the Tisserand's

[*]Corresponding author (email: lishuang@nuaa.edu.cn)

criterion to determine the MGA sequence. These graphical methods only match the energy but do not consider the actual phase of gravity-assisting celestial bodies. Pisarevsky et.al. [12] addresses the MGA trajectory by building and concatenating some blocks with different inclinations, which are a series of short sequences of Keplerian arcs. Some feasible solutions can be obtained quickly using this method, but it is difficult to find the optimal solution. Gad et.al. [13] developed a genetic algorithm with hidden genes to achieve the global mixed optimization of the sequence and local transfer trajectory for the MGA trajectory. It shows excellent global solution search capability if the number of GAs is limited. Wagner and Wei [14] proposed a hybrid algorithm for MGA trajectory optimization, which embedded a nonlinear programming solver in the genetic algorithm to solve the local segment.

The gravity-assisted trajectory parameter approximation is one of the essential problems for MGA trajectory optimization. In order to reduce the computational burden, the gravity-assisted process is commonly simplified as an instantaneous velocity maneuver, the value of which is approximated via PCM. However, there are two disadvantages of PCM: firstly, PCM only performs velocity matching and the solved trajectory is discontinuous in position. Secondly, it is assumed that the spacecraft's motion is only affected by the gravitation of the gravity-assisted celestial body during the gravity-assist process. This assumption will lead to a huge trajectory deviation when the perturbation of the central celestial body is significantly strong such as Jupiter. Due to these deficiencies, it's difficult to implement and correct those PCM-based solutions with the high-fidelity dynamical model [15-18].

The spacecraft's motion during gravity assist is mainly dominated by the gravitation of the gravity-assisted celestial body and the central primary celestial body, the dynamics of which is a typical restricted three-body problem. Prado [16] and Negri et.al. [17] investigated respectively the PCM error with the restricted three-body dynamics during gravity assist. Ferreira [18] analyzed the error of powered gravity assist in the restricted three-body problem. The results showed that the accuracy of the gravity-assisted trajectory in the restricted three-body dynamics is significantly improved than that of the PCM. Due to the heavy computational cost of the integration, the restricted three-body dynamics cannot directly be applied to the MGA trajectory optimization. Pseudostate theory proposed by Wilson [19] has outstanding performance on approximating trajectory in the restricted three-body dynamics, which was extensively applied to design the accurate gravity-assisted trajectory [20-22]. Byrnes [23] modeled the gravity-assisted trajectory as a three-body Lambert problem and transformed it into two conic Lambert problems using pseudostate theory. Its application to the Jovian system indicated that over 90% of the point-to-point conic error is eliminated. Parvathi and Ramanan [24] used the pseudostate technique to solve the departure and injection trajectory for direct interplanetary transfer considering a dynamical model with various perturbations. The accuracy of the pseudostate solution depends on the sweepback duration [25]. Sweetser [25] and Yang et.al. [26] generated the formulas for estimating the sweepback duration through empirical parameters and linearized approximation, respectively. Therefore, their generalization and accuracy are limited. Since machine learning (ML) and deep neural networks (DNN) have excellent capabilities of approximating complex nonlinear systems, they have been generally applied to address the low-thrust trajectory [27] and complex perturbation dynamics [28] in astrodynamics. Zhang et.al. [29] applied supervised machine learning to evaluate the reachable domain with gravity assist. Ampatzis and Izzo [30] used an artificial neural network to replace the traditional cost function in evolutionary algorithms, which significantly improves the computational efficiency without affecting optimization results.

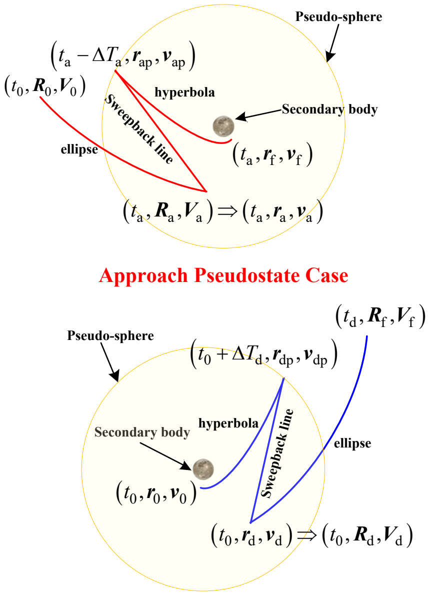
The motivation of this work is to achieve the fast preliminary design of GA trajectories in the high-fidelity dynamics. A patched strategy based on the pseudostate technique and a hybrid algorithm combined particle swarm optimization (PSO) and sequential quadratic programming (SQP) are proposed to address the continuous and accurate trajectory for MGA missions with high-fidelity dynamics. DNN is firstly applied to facilitate the pseudostate technique by identifying the optimal sweepback duration that has no universal and effective method of mathematical formula solution so far. The proposed method is applied to the preliminary design of MGA trajectory in the Jovian system, which shows significant advantages in computational efficiency and accuracy as compared with the traditional patched conic method.

2 Problem formulation

For the gravity-assisted interplanetary mission, the dynamical model of the transfer phase is different from that of the gravity-assist phase. The latter is generally simplified as a velocity breakpoint with gravity-assisted maneuver and is approximated using the patched conic method [13, 14]. Therefore, the MGA trajectory is transformed into the sequential connection of a series of gravity-assisted points. This rough splicing ignores the effect of the primary central celestial body on the spacecraft motion during the gravity-assist phase, which is not consistent with reality, especially for those systems with a large central celestial body (e.g. the Jovian system). To address this problem, a gravity assist model based on the pseudostate theory is built in this section to approximate the spacecraft motion in the domination of both the central primary body and gravity-assisted celestial body during the gravity-assist phase.

2.1 Pseudostate technique

Pseudostate theory is a simplification of the overlapped conic method and has outstanding performance on approximating the trajectory in the restricted three-body dynamics. The assumption of pseudostate theory is that the mass of the secondary is much smaller than that of the primary and the spacecraft move along a hyperbolic trajectory with respect to the secondary [25]. According to the direction (approach or departure) of the spacecraft's motion with respect to the secondary, there are two cases of the pseudostate technique, i.e. the approach pseudostate case (APC) and the departure pseudostate case (DPC).


Pseudo-sphere
$(t_a - \Delta T_a, \boldsymbol{r}_{ap}, \boldsymbol{v}_{ap})$
$(t_0, \boldsymbol{R}_0, \boldsymbol{V}_0)$
hyperbola
Secondary body
Sweepback line
ellipse
$(t_a, \boldsymbol{r}_f, \boldsymbol{v}_f)$
$(t_a, \boldsymbol{R}_a, \boldsymbol{V}_a) \Rightarrow (t_a, \boldsymbol{r}_a, \boldsymbol{v}_a)$
$(t_d, \boldsymbol{R}_f, \boldsymbol{V}_f)$
Pseudo-sphere
$(t_0 + \Delta T_d, \boldsymbol{r}_{dp}, \boldsymbol{v}_{dp})$
Secondary body
hyperbola
Sweepback line
ellipse
$(t_0, \boldsymbol{r}_0, \boldsymbol{v}_0)$
$(t_0, \boldsymbol{r}_d, \boldsymbol{v}_d) \Rightarrow (t_0, \boldsymbol{R}_d, \boldsymbol{V}_d)$

Approach Pseudostate Case

Departure Pseudostate Case

Figure 1 Diagram of the pseudostate technology. (Bold uppercase vectors, e.g. ($\boldsymbol{R}_0$, $\boldsymbol{V}_0$) and ($\boldsymbol{R}_f$, $\boldsymbol{V}_f$), are described in the inertial coordinate frame centered on the central primary celestial body. Bold lowercase vector, e.g. e.g. ($\boldsymbol{r}_d$, $\boldsymbol{v}_d$) , ($\boldsymbol{r}_{dp}$, $\boldsymbol{v}_{dp}$) and ($\boldsymbol{r}_a$, $\boldsymbol{v}_a$), are described in the inertial coordinate frame centered on the gravity-assisted celestial body.)

The diagram of the solutions of these two cases is illustrated in Figure 1. Taking the DPC as an example, the detailed solution steps are given as follows [25]:

(1) Propagate the initial state ($\boldsymbol{r}_0$, $\boldsymbol{v}_0$) in the gravitational field of the secondary body for the departure sweepback duration ΔT_d until reaching the pseudo-sphere ($\boldsymbol{r}_{dp}$, $\boldsymbol{v}_{dp}$).

(2) Propagate backward linearly along the opposite direction of the velocity $\boldsymbol{v}_{dp}$ for the sweepback duration.

(3) Transform the state with respect to the secondary body ($\boldsymbol{r}_d$, $\boldsymbol{v}_d$) into the state with respect to the primary body ($\boldsymbol{R}_d$, $\boldsymbol{V}_d$).

(4) Propagate forward to the desired time t_d in the gravitational field of the primary body.

The solution of the APC is the reverse of the above steps. For the pseudostate technique, the propagated trajectories are either the Keplerian conics (i.e. the hyperbola and the ellipse) or the straight line, which are solved analytically. Therefore, this technique has prominent computational efficiency.

In addition to the initial state and the required time, another essential parameter for the pseudostate technique is the sweepback duration, which is related to the size of the pseudosphere. The definition of pseudo-sphere is similar to the sphere of influence (SOI). Outside the pseudosphere, the spacecraft is only affected by the gravitation of the primary celestial body. Inside the pseudosphere, the motion of the spacecraft is dominated by the gravitational fields of both the primary and secondary celestial bodies [25]. This is different from the SOI model that does not consider the influence of the primary celestial body in the SOI. According to the above definition of the pseudo-sphere, the accuracy of the pseudostate technology completely depends on the radius of the pseudosphere that is usually represented by the sweepback duration. Therefore, the sweepback duration determines the approximation accuracy of the pseudostate technique. To ensure the accuracy of the pseudostate technique, a novel method to solve the sweepback duration will be proposed in section 3.

2.2 Gravity-assisted model using pseudostate technique

Due to the simple implementation and the high computational efficiency, the gravity-assist model based on the SOI model is popular in the preliminary design of interplanetary missions. However, its poor accuracy brings about trouble for subsequent trajectory implementation in the high-fidelity dynamical model. A gravity-assist model using the pseudostate technique is presented to fast solve the gravity-assisted trajectory in the high-fidelity dynamical model with the perturbation of the primary body.

The gravity-assisted model based on the pseudostate technique is explained in Figure 2, where a maneuver is performed at the specified position to patch the approach (red) and departure (blue) legs. The approach and departure legs are solved using the pseudostate technique according to the steps given in Section 2.1. Because the velocity of the spacecraft with respect to the gravity-assisted body has changed during gravity assistance, the sweepback durations of the approach and departure legs are not the same as well, which is different from that in reference [25] and can effectively improve the accuracy of such legs. In order to match the approach and departure legs at the designated position P (denoted as $\boldsymbol{r}_{GA}$), the necessary corrections are made to them and the details will be described in section

4.1. The velocity difference between the departure and approach legs at position P, $\Delta \boldsymbol{v} = \boldsymbol{v}_{\mathrm{GAd}} - \boldsymbol{v}_{\mathrm{GAa}}$ is matched by an impulse maneuver.

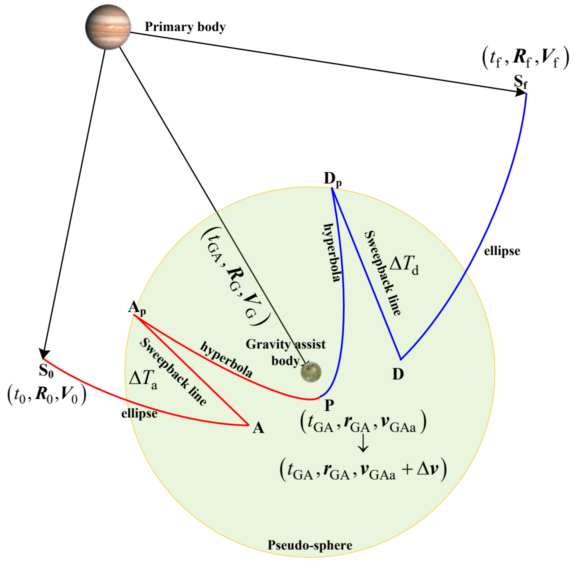

Primary body
$(t_\mathrm{f}, \boldsymbol{R}_\mathrm{f}, \boldsymbol{V}_\mathrm{f})$
$\mathrm{S_f}$
$\mathrm{D_p}$
hyperbola
Sweepback line
ΔT_d
ellipse
$(t_\mathrm{GA}, \boldsymbol{R}_\mathrm{G}, \boldsymbol{V}_\mathrm{G})$
$\mathrm{A_p}$
Sweepback line
hyperbola
Gravity assist body
ΔT_a
$\mathrm{S_0}$
$(t_0, \boldsymbol{R}_0, \boldsymbol{V}_0)$
ellipse
A
D
P
$(t_\mathrm{GA}, \boldsymbol{r}_\mathrm{GA}, \boldsymbol{v}_\mathrm{GAa})$
$\downarrow$
$(t_\mathrm{GA}, \boldsymbol{r}_\mathrm{GA}, \boldsymbol{v}_\mathrm{GAa} + \Delta\boldsymbol{v})$
Pseudo-sphere

Figure 2 The gravity-assisted model using the pseudostate technique

In summary, the application of the pseudostate technique achieves the fast and high-precision solution of the gravity-assisted trajectory with the dynamical model considering the perturbation of the primary body, which has the same computational complexity with that of the SOI model. Nevertheless, the proposed model has higher accuracy and matches both the position and the velocity simultaneously. In addition, the delta-V consumption is related to the matching position that needs to be optimized, the details of the optimization are introduced in Section 4.1.

2.3 MGA problem formulation

A multi-gravity-assisted model with deep space maneuvers is presented in this section, as shown in Figure 3. Two types of maneuvers, i.e. the deep-space maneuver and the gravity-assisted maneuver, are performed to patch the whole trajectory smoothly. The deep-space maneuver is scheduled in the middle of the transfer trajectory, which aims to correct the terminal position error of the approach segment of the next gravity assistance. The gravity-assisted maneuver is arranged at the matching point of the approach and departure legs of the gravity assist trajectory, where the closest position of the spacecraft to the gravity-assisted body is also located. Therefore, the total velocity increment of the MGA trajectory is

$$\Delta V_{\mathrm{tol}} = \left\|\Delta \boldsymbol{v}_0\right\| + \left\|\Delta \boldsymbol{v}_\mathrm{f}\right\| + \sum_{j=1}^{n}\left\|\Delta \boldsymbol{v}_j\right\| + \left\|\Delta \boldsymbol{v}_{\mathrm{GA}j}\right\| \tag{1}$$

where $\Delta \boldsymbol{v}_0$ and $\Delta \boldsymbol{v}_\mathrm{f}$ are the initial and terminal velocity change respectively, $\Delta \boldsymbol{v}_j$ and $\Delta \boldsymbol{v}_{\mathrm{GA}j}$ are the velocity change of the j-th deep-space maneuver and the j-th gravity-assisted maneuver, respectively. n is the number of gravity assistances.

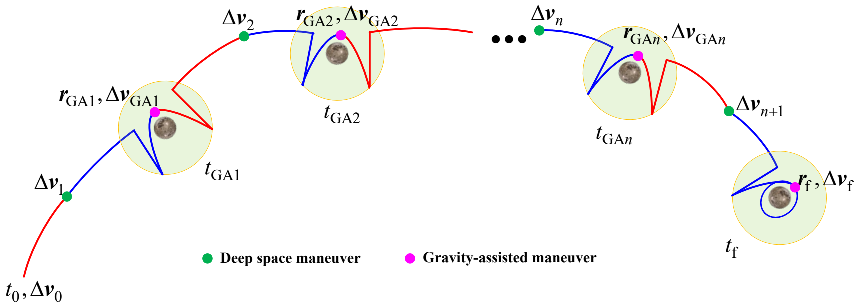

$\Delta \boldsymbol{v}_2$
$\boldsymbol{r}_{\mathrm{GA2}}, \Delta \boldsymbol{v}_{\mathrm{GA2}}$
$\Delta \boldsymbol{v}_n$
$\boldsymbol{r}_{\mathrm{GA}n}, \Delta \boldsymbol{v}_{\mathrm{GA}n}$
$\boldsymbol{r}_{\mathrm{GA1}}, \Delta \boldsymbol{v}_{\mathrm{GA1}}$
t_{GA2}
$t_{\mathrm{GA}n}$
$\Delta \boldsymbol{v}_{n+1}$
$\Delta \boldsymbol{v}_1$
t_{GA1}
$\boldsymbol{r}_\mathrm{f}, \Delta \boldsymbol{v}_\mathrm{f}$
● Deep space maneuver
● Gravity-assisted maneuver
t_f
$t_0, \Delta \boldsymbol{v}_0$

Figure 3 The MGA model with deep space maneuver using the pseudostate technique

Firstly, given the gravity-assist sequence and time, a series of Keplerian transfer segments are solved to connect the gravity-assisted celestial bodies in turn, which is given as follows.

$$\begin{cases} \left(\boldsymbol{V}_\mathrm{d}, \boldsymbol{V}_{\mathrm{GAa1}}\right) = L\left(\boldsymbol{R}_0, \boldsymbol{R}_{\mathrm{G1}}, t_{\mathrm{GA1}} - t_0\right) \\ \left(\boldsymbol{V}_{\mathrm{GAd}i}, \boldsymbol{V}_{\mathrm{GAa}i+1}\right) = L\left(\boldsymbol{R}_{\mathrm{G}i}, \boldsymbol{R}_{\mathrm{G}i+1}, t_{\mathrm{GA}i+1} - t_{\mathrm{GA}i}\right) \\ \left(\boldsymbol{V}_{\mathrm{GAd}n}, \boldsymbol{V}_\mathrm{a}\right) = L\left(\boldsymbol{R}_{\mathrm{G}n}, \boldsymbol{R}_\mathrm{f}, t_\mathrm{f} - t_{\mathrm{GA}n}\right) \end{cases} \tag{2}$$

where $\boldsymbol{L}(\cdot)$ denotes the solution function for the Lambert problem. $\boldsymbol{R}_0$ and $\boldsymbol{R}_\mathrm{f}$ are the initial and target positions of the

MGA transfer trajectory at t_0 and t_f, respectively. $\boldsymbol{R}_{\mathrm{G}i}$ is the position of the i-th gravity-assisted body at $t_{\mathrm{GA}i}$, which is solved from its ephemeris. $\boldsymbol{V}_{\mathrm{GAd}i}$ and $\boldsymbol{V}_{\mathrm{GAa}i}$ are the approach and departure velocities of the i-th gravity assist, respectively, which are obtained by solving the Lambert problem.

Then, the whole trajectory is divided into several approach legs (blue lines) and departure legs (red lines). The former starts at the middle time of each transfer segments $t_{\mathrm{m}i}$ and the latter starts at the gravity-assisted time $t_{\mathrm{GA}i}$.

$$\begin{cases} t_1 = 0.5\left(t_{\mathrm{GA1}} + t_0\right) \\ t_{\mathrm{m}i} = 0.5\left(t_{\mathrm{GA}i} + t_{\mathrm{GA}i-1}\right) \quad ,i = 2,3,\cdots,n \\ t_{n+1} = 0.5\left(t_\mathrm{f} + t_{\mathrm{GAn}}\right) \end{cases} \tag{3}$$

Deep-space maneuvers and gravity-assisted maneuvers are performed at $t_{\mathrm{m}i}$ and $t_{\mathrm{GA}i}$ to patch these segments, which will be described in section 4.1. These maneuvers are directly related to the matching position, which is shown as follows

$$\Delta \boldsymbol{v}_i^* = \boldsymbol{V}_{\mathrm{sa}i}^* - \boldsymbol{V}_{\mathrm{m}i} = F_{\mathrm{sa}}\left(\boldsymbol{R}_{\mathrm{m}i}, tof_{\mathrm{a}i}, \Delta T_{\mathrm{a}i}, \boldsymbol{r}_{GAi}^*\right) - \boldsymbol{V}_{\mathrm{m}i} \tag{4}$$

$$\begin{aligned} \Delta \boldsymbol{v}_{\mathrm{GA}i}^* &= \boldsymbol{v}_{\mathrm{GAd}i}^* - \boldsymbol{v}_{\mathrm{GAa}i}^* \\ &= F_{\mathrm{sd}}\left(\boldsymbol{r}_{\mathrm{GA}i}^*, \boldsymbol{R}_{\mathrm{m}i+1}, tof_{\mathrm{d}i}, \Delta T_{\mathrm{d}i}\right) \\ &\quad - P_\mathrm{a}\left(\boldsymbol{R}_{\mathrm{m}i}, \boldsymbol{V}_{\mathrm{sa}i}^*, tof_{\mathrm{a}i}, \Delta T_{\mathrm{a}i}\right) \end{aligned} \tag{5}$$

where $\boldsymbol{r}_{GAi}^*$ is the optimum matching position for the i-th gravity assistance. F_{sa} and F_{sd} are the processes of the approach and departure shooting correction, respectively. P_a is the propagation of the approach leg using the pseudostate technique. ($\boldsymbol{R}_{\mathrm{m}i}$, $\boldsymbol{V}_{\mathrm{m}i}$) is the spacecraft state at $t_{\mathrm{m}i}$. *tof* and ΔT are the time of flight and the sweepback duration, the subscripts "a" and "d" denote the approach and departure legs, respectively. $\boldsymbol{V}_{\mathrm{sa}i}^*$ and $\boldsymbol{v}_{\mathrm{GAd}i}^*$ are the spacecraft velocities at $t_{\mathrm{m}i}$ and $t_{\mathrm{GA}i}$ after the shooting correction. The method of solving the optimal matching position $\boldsymbol{r}_{GAi}^*$ for each gravity assistance and the shooting correction process will be introduced in detail in Section 4.

Therefore, the MGA trajectory optimization is transformed into a problem of searching the gravity assistance time windows. The performance index in Eq. (1) is denoted as the function of the initial time, the terminal time and the gravity-assist time and sequence, which is shown as follow,

$$J = \Delta V_{\mathrm{tol}} = f\left(t_0, t_\mathrm{f}, t_{\mathrm{GA1}}, N_{\mathrm{GA1}}, \cdots, t_{\mathrm{GA}n}, N_{\mathrm{GA}n}\right) \tag{6}$$

where t_0 and t_f are the initial and terminal time, $t_{\mathrm{GA}i}$ is the time of the i-th gravity assistance, $N_{\mathrm{GA}i}$ is the sign of the i-th gravity-assisted body and n is the number of gravity assistances.

In conclusion, a continuous MGA trajectory model is formulated base on the pseudostate technique in this section. Compared with the traditional PCM, the proposed model considers the gravity assistance process and the perturbation effect of the central celestial body on the gravity-assisted trajectory, which improves the accuracy of the solution. The application of the pseudostate technique replaces the integration of the perturbative trajectory and significantly reduces the calculation time.

3 Sweepback duration approximation using DNN

In the pseudostate technique, sweepback duration is defined as the time duration of the linear sweepback segment that connects the primary elliptical segment and the secondary hyperbolic segment. The sweepback duration is a crucial parameter and significantly affects the approximation accuracy of the pseudostate technique. Figure 4 shows a typical example of the effect of the sweepback duration on the terminal position error. When the sweepback duration approaches zero, the pseudostate method degenerates into the pure patched conic method. Then the sweepback duration increases to the optimum when the terminal position error reaches its minimum. After the optimum point, the error increases gradually with the increase of the sweepback duration.

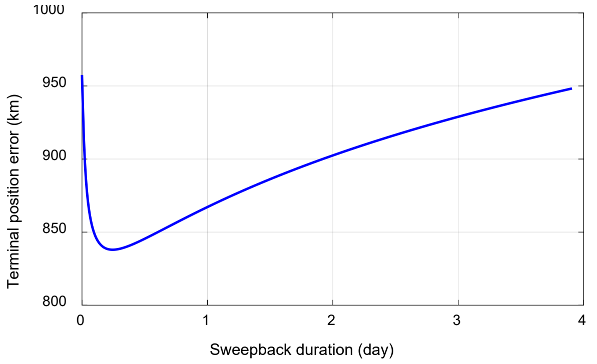


Figure 4 The effect of sweepback duration on the terminal position error

3.1 Sweepback duration analysis

For a specific pair of primary and secondary celestial bodies, the sweepback duration is a function of the spacecraft state ($\boldsymbol{r}$, $\boldsymbol{v}$) with respect to the secondary body, the position and velocity of the secondary body ($\boldsymbol{R}$, $\boldsymbol{V}$) with respect to the primary and time-of-flight *tof*. However, it is difficult to establish an explicit mathematical expression to describe this function. Therefore we devote to solve the mapping between the sweepback duration and its reliable variables using a deep neural network. First of all, it is necessary to identify the major factors from all dependent variables. The ones that have significant effects on the sweepback duration are reserved in the sample of the DNN. The others that have minor factors are neglected to reduce the dimensions of the sample. Without loss of generality, the subsequent simulations in this section take the Jupiter-Ganymede system as an example.

3.1.1 Departure case

The optimal sweepback duration depends on the departure trajectory. The prior-known parameters for solving a departure trajectory of the gravity assist are the initial position and velocity vectors of the spacecraft with respect to the secondary celestial body, the time-of-flight, and the position and velocity vectors of the secondary body. Therefore, they are the candidate parameters for calculating the optimal sweepback duration as well.

The magnitude of the initial position vector is the initial distance from the secondary body to the spacecraft, which is an important parameter to represent the effect of the gravitation of the secondary body on the spacecraft trajectory. Figure 5 shows that the relation between the optimal sweepback duration and magnitudes of initial positions. In particular, 13 cases with different initial time, time of flight, and initial velocities are randomly generated and tested. For each case, the optimal sweepback duration varies slightly with the magnitude of the initial position, and the gradient of the curve decreases with the increase of the altitude. This indicates that the influence of the initial position altitude on the optimal sweepback duration becomes less with the increasing initial altitude.

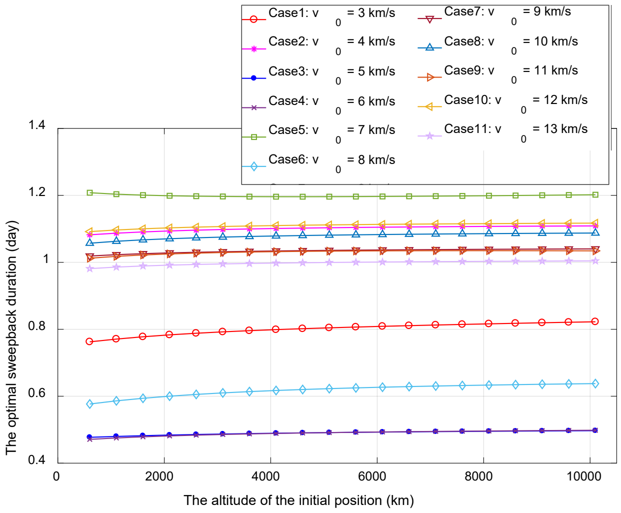


Figure 5 The optimal sweepback duration varies with different altitudes of the initial positions for departure case

The magnitude of the initial velocity vector indicates how fast the spacecraft moves away from the secondary body. Numerical simulations are performed to investigate the effect of the initial velocity magnitude on the optimal sweepback duration. In Figure 6, three test cases (Case A, B and C) are shown, and three examples with different initial altitudes (h0 = 100 km, 1000 km and 10000 km) are given for each case. The examples in the same case have the same initial time epoch, initial velocity direction and the time of flight. In each example, we sample different initial velocity magnitudes (markers on each line) to verify their effect on the optimal sweepback duration.

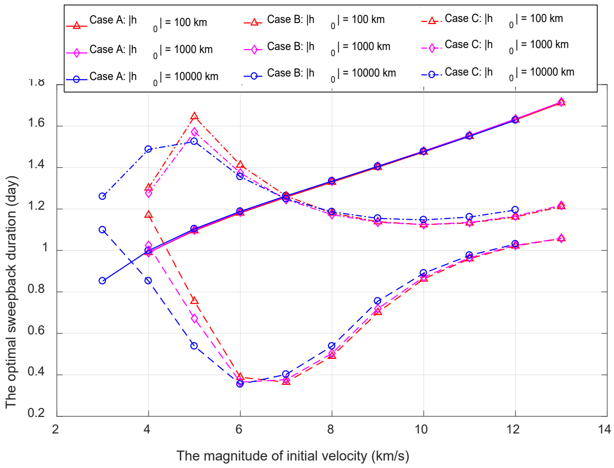


Figure 6 The effect of the magnitude of the initial velocity on the optimal sweepback duration for departure case

Some conclusions can be summarized from Figure 6. First, it is observed that the optimal sweepback duration varies with the initial velocity magnitude for all lines. This proves that the magnitude of the initial velocity is one of the major factors of the optimal sweepback duration. Second, the examples for the same case have the same variational tendencies of the optimal sweepback duration with the increasing initial velocity magnitude, but the value of the optimal sweepback duration is slightly different when the initial velocity magnitude is the same. This means the initial altitude has little effect on the optimal sweepback duration. Finally, the variational tendencies of the optimal sweepback duration with increasing initial velocity magnitude are obviously different for these lines with the same color in different cases, which have the same initial altitude but different initial velocity direction and the time of flight. This indicates that the optimal sweepback duration has strong correlations with the initial velocity direction and the time of flight.

As analyzed previously, the direction of the spacecraft's motion has a significant impact on the optimal sweepback duration and is mainly determined by the direction of the initial velocity. Since the departure trajectory is also relative to the position and velocity of the secondary body,

α and the angle between the initial velocity vector and the velocity vector of the secondary β to describe the direction of the spacecraft's motion. The variation of the optimal time ΔT^* with the direction of spacecraft's motion is given in Figure 7, where six cases with different initial time, time of flight, and initial velocity magnitudes are simulated.

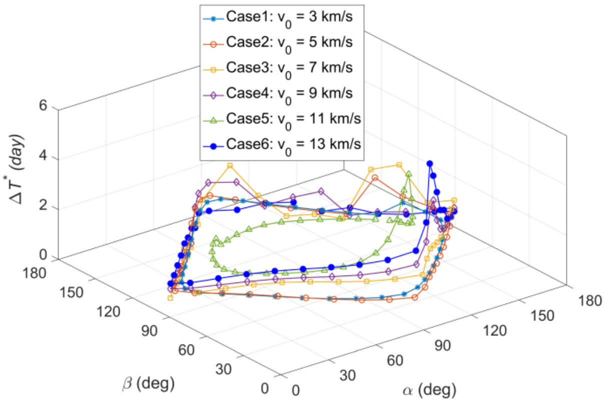


Figure 7 The variation of the optimal sweepback duration with the direction of initial velocity for departure case

In Figure 7, points on the same line have the same simulation parameters, except for the initial velocity direction. The optimal sweepback duration varies significantly with the directions of motion and its variational extent is related to the magnitude of the initial velocity.

Following on, 1000 cases are randomly generated and simulated to verify the effect of the flight time on the optimal sweepback duration. The results are given in Figure 8. The points on the same line have the same initial position and velocity. When the time of flight is generally short, the optimal sweepback duration varies significantly with the time of flight. However, the sweepback duration gradually converges to a fixed value with increasing time-of-flight, due to the fact that the range of the influence of the gravitational perturbation of the secondary is limited.

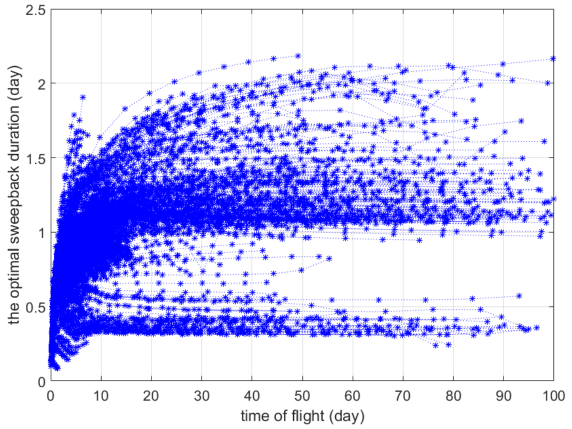


Figure 8 The effect of the time of flight on the optimal sweepback duration for departure case

In summary, the optimal sweepback duration of the departure case is mainly affected by the initial velocity and the time of flight and is slightly influenced by the magnitude of the initial position. Therefore, for the departure trajectory, the major factors considered for the optimal sweepback duration are the time of flight, the magnitude and direction of the initial velocity.

3.1.2 Approach case

The resolution of the approach trajectory is the reverse solving of the departure trajectory. The influence of the gravitation of the secondary on the spacecraft's motion gradually becomes stronger as it approaches the secondary. According to the steps of the solution described in section 2.1, the sweepback duration is only needed when solving the sweepback line arc and the hyperbolic arc. The elliptic arc can be calculated using the time of flight and the position and velocity of the spacecraft at the gravity-assisted time epoch t_{GA}. Therefore, the relevant factors of the optimal sweepback duration for the approach trajectory are the time of flight, the spacecraft velocity and the position and velocity of the secondary at the time epoch t_{GA}. Here we follow the analysis of the departure case to investigate the impact of the flight time, and the magnitude and direction of the spacecraft hyperbolic excess velocity on the optimal sweepback duration, respectively.

Numerical simulations are performed to verify the influence of the hyperbolic excess velocity magnitude v_{inf} on the sweepback duration. The results of five study cases are shown in Figure 9. The five different cases have different gravity-assist time epoch, the time of flight, and directions of the hyperbolic excess velocity.

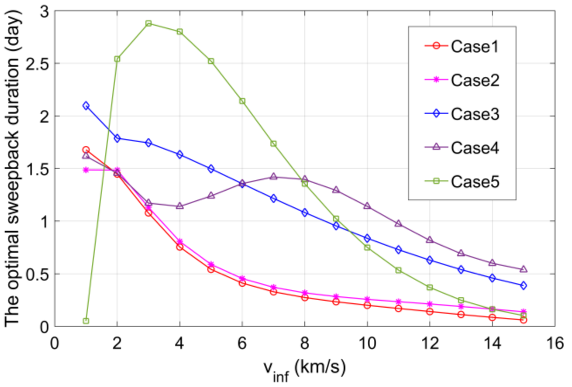


Figure 9 The effect of the magnitude of the hyperbolic excess velocity on the optimal sweepback duration for approach case

For each case, the value of the optimal sweepback duration changes significantly with the increasing magnitude of the hyperbolic excess velocity. In addition, the variational tendencies of different cases are different, indicating that other parameters such as the direction of v_{inf} also affect the optimal sweepback duration.

The direction of the hyperbolic excess velocity is represented by two angles between v_{inf} and the position and velocity of the secondary body at t_{GA}, which are denoted as θ and γ. To test the influence of this direction on the optimal sweepback duration, ten cases are simulated and shown in Figure 10. It is obvious that the optimal sweepback duration varies with the changes of the direction of the hyperbolic excess velocity for each case. The tendency of this variation is similar for the different cases, but magnitudes of these variations are different due to the different magnitudes of the hyperbolic excess velocity.

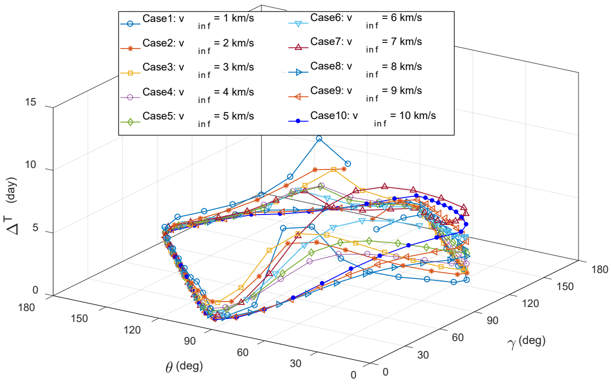


Figure 10 The effect of the direction of the hyperbolic excessive velocity on the optimal sweepback duration for approach case

Another potential factor that affects the optimal sweepback duration is the time-of-flight, which is investigated with numerical simulations and the result is shown in Figure 11. In order to eliminate the effects from other factors, each test case uses the same approach trajectory, which ensures the same hyperbolic excess velocities. For the simulation of each specific case, the initial state of the spacecraft is sampled at equal time interval along the approach trajectory. It is seen from Figure 11 that the optimal sweepback duration has a fixed value for each case, which proves that the time of flight does not affect the optimal sweepback duration for the approach trajectory.

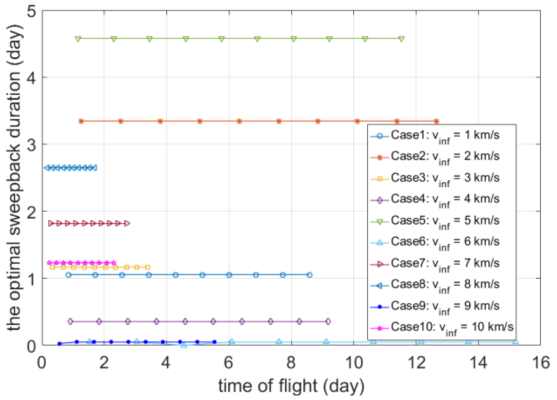


Figure 11 The effect of the time of flight on the optimal sweepback duration for approach case

Therefore, for the approach case, the optimal sweepback duration is mainly affected by the magnitude and direction of the hyperbolic excess velocity and is independent of the time of flight. This is different from the departure case.

3.2 Sweepback duration estimator via DNN

It is very challenging to estimate the optimal sweepback duration using the analytical mathematical formulas. A novel method based on DNN that does not require intuitive empirical parameters is proposed in this section to address this challenge. DNN trains and learns from a large sample database, and then establishes a mapping between the inputs and the outputs. DNN has demonstrated outstanding performance in approximating complex nonlinear systems.

A typical DNN structure is shown in Figure 12, consisting of an input layer, an output layer and several hidden layers. Each layer has a number of neurons and neurons in adjacent layers are connected to each other. These neurons receive signals from the previous layer and send the processed signals to the next layer. The signal processing includes weighted summation and activation operations that are given as follows [28].

$$\begin{cases} s = b + \sum_{k=1}^{N} w_k x_k \\ y = f(s) \end{cases} \tag{7}$$

where b is the bias coefficient and w_k is the weight coefficient of the input signal x_k. N is the number of neurons in the previous layer. $f(\cdot)$ is the activation function. The common activation functions include Sigmoid, tanh and relu are, whose mathematical expressions are given as follows [28]

$$\begin{cases} y = \mathrm{relu}(x) = \max(0, x) \\ y = \tanh(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \\ y = \mathrm{sigmoid}(x) = \dfrac{1}{1 + e^{-x}} \end{cases} \tag{8}$$

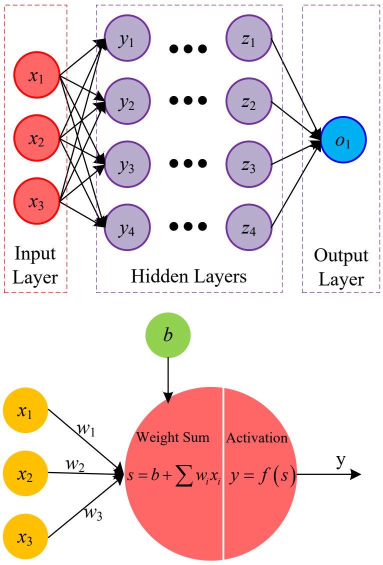


Figure 12 The illustration of the DNN structure and the neuron calculation

The goal of the training is to find the optimal weights and bias coefficients to minimize the output error. During the training process, the sample input is passed to DNN, and the predicted value is gathered from the output layer after processing by hidden layers. The error of the predicted value is then obtained as the difference between the real output and the prediction and is fed back to the input layer to adjust the weights and bias parameters of DNN.

However, the estimation error is unavoidable in approximation methods. If the estimated value is less than the optimum, the terminal position error quickly diverges as the estimated error increases. Nevertheless, this divergence trend is relatively slowly as the estimated value is over the optimum. This is a common phenomenon for both approach and departure cases in our simulations. Therefore, a confidence interval is introduced to reduce the influence of this asymmetric divergence error trend on both sides of the optimum, which is shown in Figure 13.

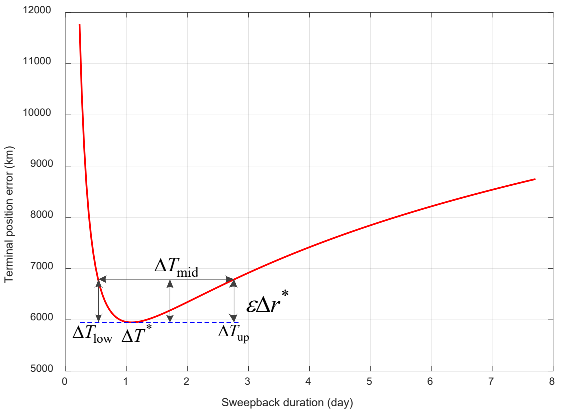


Figure 13 The definition of the confidence interval $[\Delta T_{\mathrm{low}}, \Delta T_{\mathrm{up}}]$ of the optimal sweepback duration

The steps to solve for the confidence interval are given as follows:

(1) Solve the optimal sweepback duration ΔT^*, and calculate the corresponding terminal position error Δr^*;

(2) Calculate the tolerance of the terminal position error of the confidence interval, i.e. $\Delta r_{\mathrm{tole}} = \varepsilon \Delta r^*$, where ε is the weight coefficient with the range of values [0, 1] and its default value is 0.01.

(3) Find the upper and lower bounds of the confidence interval, i.e. $[\Delta T_{\mathrm{low}}, \Delta T_{\mathrm{up}}]$, which are the sweepback duration corresponding to the terminal position error $(1+\varepsilon)\Delta r^*$.

(4) Calculate the median of the confidence interval, $\Delta T_{\mathrm{m}} = 0.5(\Delta T_{\mathrm{low}} + \Delta T_{\mathrm{up}})$.

In the subsequent simulations, the median of the confidence interval is used instead of the optimal sweepback duration. As the tolerance of the terminal position error of the confidence interval is small enough, the optimal sweepback duration is equivalent to the median of the confidence interval.

According to the analysis of the major factors affecting the optimal sweepback duration, the data structures of the training samples for the approach and departure optimal sweepback duration approximation are designed as follows

$$\begin{cases} S_{\mathrm{d}} = \{[r_0,\ v_0,\ \alpha,\ \beta,\ tof],\ \Delta T_{\mathrm{dm}}\} \\ S_{\mathrm{a}} = \{[v_{\mathrm{inf}},\ \theta,\ \gamma],\ \Delta T_{\mathrm{am}}\} \end{cases} \tag{9}$$

There are five and three inputs for departure and approach cases, respectively. Their outputs are both the median of the confidence interval of the optimal sweepback duration. The details of the training process are described in section 5.

4 MGA trajectory optimization algorithm

According to the model in Section 2, the variables to be optimized in the MGA trajectory optimization for a fixed gravity-assisted sequence are the time and the matching position of gravity assist. The optimal matching position can be solved from the given time of the gravity-assist. Therefore, a hybrid MGA trajectory optimization algorithm is proposed, including a global optimizer and a local trajectory solver. The global optimizer is applied to find the optimal time windows. The local solver is embedded in the global optimizer to obtain the optimal matching position and calculate the delta-V consumption for a specific time window.

4.1 The patching strategy of gravity-assisted trajectory

For a specific scenario with the given time and sequence of the gravity assistance, the transfer segments are patched using the pseudostate gravity-assisted model given in section 2. Therefore, it's important to find an optimal gravity-assisted matching position to reduce the delta-V for matching the trajectory. A gravity-assisted trajectory patched strategy is developed to smooth the gravity-assisted trajectory with the least delta-V required.

In the pseudostate gravity-assisted model, both the approach and departure legs consist of a primary elliptic arc, a sweepback straight line and a secondary hyperbolic arc, as shown in Figure 2. If the initial state (t_0, $\boldsymbol{R}_0$, $\boldsymbol{V}_0$) and the gravity-assisted time t_{GA} are fixed, the approach leg is solved with the terminal position of the primary elliptic arc of the approach leg $\boldsymbol{r}_{\mathrm{ae}}$. Firstly, the velocity of the primary elliptical arc is solved from the Lambert problem as follows,

$$\left(\boldsymbol{V}_{\mathrm{a0}},\boldsymbol{V}_{\mathrm{ae}}\right)=L(\boldsymbol{R}_0,\boldsymbol{R}_{\mathrm{GA}}+\boldsymbol{r}_{\mathrm{ae}},t_{\mathrm{GA}}-t_0) \tag{10}$$

where $\boldsymbol{R}_{\mathrm{GA}}$ is the position of the gravity-assist body that is obtained from its ephemeris after a given time t_{GA}. $\boldsymbol{r}_{\mathrm{ae}}$ is the terminal position of the approach elliptical arc, which is described in the inertial coordinate frame centered on the gravity-assisted body.

The terminal state of the approach legs ($\boldsymbol{r}_{\mathrm{GAa}}$, $\boldsymbol{v}_{\mathrm{GAa}}$) is then calculated as follows according to the pseudostate technique.

$$\begin{cases}\boldsymbol{v}_{\mathrm{ae}}=\boldsymbol{V}_{\mathrm{ae}}-\boldsymbol{V}_{GA}\\ \boldsymbol{r}_{\mathrm{ap}}=\boldsymbol{r}_{\mathrm{ae}}-\boldsymbol{v}_{\mathrm{ae}}\Delta T_{\mathrm{a}}\\ \boldsymbol{v}_{\mathrm{ap}}=\boldsymbol{v}_{\mathrm{ae}}\\ \left(\boldsymbol{r}_{\mathrm{GAa}},\boldsymbol{v}_{\mathrm{GAa}}\right)=\Psi_{\mathrm{ha}}\left(\boldsymbol{r}_{\mathrm{ap}},\boldsymbol{v}_{\mathrm{ap}},\Delta T_{\mathrm{a}}\right)\end{cases} \tag{11}$$

where Ψ_{ha} denotes the state transition matrix dominated by the gravitation of the gravity-assisted body. ΔT_{a} is the optimal sweepback duration of the approach leg, which is obtained using the trained DNN introduced in Section 3.

$$\Delta T_{\mathrm{a}}=DNN_{\mathrm{a}}\left(v_{\mathrm{ae}},\theta_{\mathrm{ae}},\gamma_{\mathrm{ae}}\right) \tag{12}$$

where v_{ae} is the magnitude of the velocity $\boldsymbol{v}_{\mathrm{ae}}$, θ_{ae} and γ_{ae} are the angles $\boldsymbol{v}_{\mathrm{ae}}$ makes with $\boldsymbol{R}_{\mathrm{GA}}$ and $\boldsymbol{V}_{\mathrm{GA}}$, respectively.

Therefore, the terminal position of the approach leg $\boldsymbol{r}_{\mathrm{GAa}}$ is a mapping of the position $\boldsymbol{r}_{\mathrm{ae}}$, donated as $\boldsymbol{r}_{\mathrm{GAa}}= F(\boldsymbol{r}_{\mathrm{ae}})$. When the matching position $\boldsymbol{r}_{\mathrm{GA}}$ is given, the position $\boldsymbol{r}_{\mathrm{ae}}$ can be obtained by solving the equation $\boldsymbol{r}_{\mathrm{GA}}= F(\boldsymbol{r}_{\mathrm{ae}})$ using the command *fsolve* in Maltab. The terminal velocity of the approach leg $\boldsymbol{v}_{\mathrm{GAa}}$ is then obtained from Eq. (11).

The matching position is used as the initial position of the departure leg and the initial velocity must be corrected to ensure that the departure leg reaches the desired position $\boldsymbol{R}_{\mathrm{f}}$. The initial position and velocity of the departure primary elliptical arc ($\boldsymbol{r}_{\mathrm{de}}$, $\boldsymbol{v}_{\mathrm{de}}$) can be solved from the following equation when the initial position and velocity ($\boldsymbol{r}_{\mathrm{GAd}}$, $\boldsymbol{v}_{\mathrm{GAd}}$) are known.

$$\begin{cases}\Delta T_{\mathrm{d}}=DNN_{\mathrm{d}}\left(r_{\mathrm{GAd}},v_{\mathrm{GAd}},\alpha,\beta,t_{\mathrm{f}}-t_{\mathrm{GA}}\right)\\ \left(\boldsymbol{r}_{\mathrm{dp}},\boldsymbol{v}_{\mathrm{dp}}\right)=\Psi_{\mathrm{hd}}\left(\boldsymbol{r}_{\mathrm{GAd}},\boldsymbol{v}_{\mathrm{GAd}},\Delta T_{\mathrm{d}}\right)\\ \boldsymbol{r}_{\mathrm{de}}=\boldsymbol{r}_{\mathrm{dp}}-\boldsymbol{v}_{\mathrm{dp}}\Delta T_{\mathrm{d}}\\ \boldsymbol{v}_{\mathrm{de}}=\boldsymbol{v}_{\mathrm{dp}}\end{cases} \tag{13}$$

where r_{GAd} and v_{GAd} are the magnitudes of the initial position and velocity of the departure leg. Ψ_{hd} denotes the state transition matrix of the hyperbolic trajectory dominated by the gravitation of the gravity-assisted body.

The desired velocity $\boldsymbol{V}_{\mathrm{de}}^*$ that brings the spacecraft to the final position is then solved from the Lambert problem.

$$\left(\boldsymbol{V}_{\mathrm{de}}^*,\boldsymbol{V}_{\mathrm{df}}\right)=L(\boldsymbol{R}_{\mathrm{GA}}+\boldsymbol{r}_{\mathrm{de}},\boldsymbol{R}_{\mathrm{f}},t_{\mathrm{f}}-t_{\mathrm{GA}}) \tag{14}$$

The patched departure leg is obtained by solving the following equation using *fsolve* in Maltab.

$$\Delta\boldsymbol{v}_{\mathrm{de}}=\boldsymbol{v}_{\mathrm{de}}-\left(\boldsymbol{V}_{\mathrm{de}}^*-\boldsymbol{V}_{\mathrm{GA}}\right)=G\left(\boldsymbol{v}_{\mathrm{GAd}}\right)=\boldsymbol{0} \tag{15}$$

The gravity-assisted trajectory patching aims to match the position of the departure and approach legs with the least delta-V consumption. Therefore, the objective function is defined as

$$
\begin{aligned}
J &= \left\|\Delta \boldsymbol{v}_{\mathrm{GA}}\right\| + \left\|\Delta \boldsymbol{V}_{0}\right\| + \left\|\Delta \boldsymbol{V}_{\mathrm{f}}\right\| \\
&= \left\|\boldsymbol{v}_{\mathrm{GAd}} - \boldsymbol{v}_{\mathrm{GAa}}\right\| + \left\|\boldsymbol{V}_{\mathrm{a0}} - \boldsymbol{V}_{0}\right\| + \left\|\boldsymbol{V}_{\mathrm{f}} - \boldsymbol{V}_{\mathrm{df}}\right\| \\
&= f\left(r_{\mathrm{GA}}, \alpha_{\mathrm{GA}}, \beta_{\mathrm{GA}}\right)
\end{aligned}
\tag{16}
$$

where $\left[r_{\mathrm{GA}}, \alpha_{\mathrm{GA}}, \beta_{\mathrm{GA}}\right]$ is the spherical coordinate of $\boldsymbol{r}_{\mathrm{GA}}$ in the inertial frame centered on the gravity-assisted body.

Several constraints are introduced to ensure the matching accuracy and security of the gravity-assisted trajectory. Firstly, the terminal state of the approach and the initial state of the departure must take place before and after the periapsis passage respectively, to ensure the accuracy of pseudostate technology. Secondly, the radius of the matching position must be larger than the safety altitude r_{GAmin}. The formulations of these constraints are given as follows.

$$
\begin{cases}
\boldsymbol{r}_{\mathrm{GAa}} \cdot \boldsymbol{v}_{\mathrm{GAa}} \geq 0 \\
\boldsymbol{r}_{\mathrm{GAd}} \cdot \boldsymbol{v}_{\mathrm{GAd}} \leq 0 \\
\left\|\boldsymbol{r}_{\mathrm{GA}}\right\| \geq r_{\mathrm{GAmin}}
\end{cases}
\tag{17}
$$

Here the default tool *fmincon* in MATLAB, whose build-in optimizer is set to the sequential quadratic programming, is used to solve the optimal matching position of the approach and departure legs. Because the robustness and efficiency of the method mainly depend on the initial guess, an adaptive initial guess algorithm is proposed to obtain a better guess. According to the pseudostate theory, the desired optimal matching position is close to the periapsis of approach and departure legs, where the angle between the approach and departure velocity vectors is the smallest or even equals zero. Therefore, the initial guess for SQP is obtained by the following equation,

$$
\boldsymbol{r}_{\mathrm{GAguess}} = r_{\mathrm{GAmin}} \frac{\left(\boldsymbol{r}_{\mathrm{ap}} \times \boldsymbol{v}_{\mathrm{ap}}\right) \times \left(\boldsymbol{r}_{\mathrm{dp}} \times \boldsymbol{v}_{\mathrm{dp}}\right)}{\left\|\left(\boldsymbol{r}_{\mathrm{ap}} \times \boldsymbol{v}_{\mathrm{ap}}\right) \times \left(\boldsymbol{r}_{\mathrm{dp}} \times \boldsymbol{v}_{\mathrm{dp}}\right)\right\|}
\tag{18}
$$

where ($\boldsymbol{r}_{\mathrm{ap}}$, $\boldsymbol{v}_{\mathrm{ap}}$) and ($\boldsymbol{r}_{\mathrm{dp}}$, $\boldsymbol{v}_{\mathrm{dp}}$) are the pseudo-states of the approach and departure legs, and r_{GAmin} is the minimum gravity-assist radius.

4.2 Hybrid optimization algorithm of MGA trajectory

A hybrid optimization algorithm combining the particle swarm optimization (PSO) and the sequential quadratic programming is proposed to optimize the multi-gravity-assist trajectory. The framework of the algorithm is illustrated in Figure 14. PSO is used to find the optimal time windows, while SQP is embedded in PSO as a local trajectory solver to calculate the fitness value that is the total delta-V defined in Eq. (1) and to optimize the gravity-assisted matching position. The gravity-assisted matching position optimization model has been introduced in Sections 2.3 and 4.1.

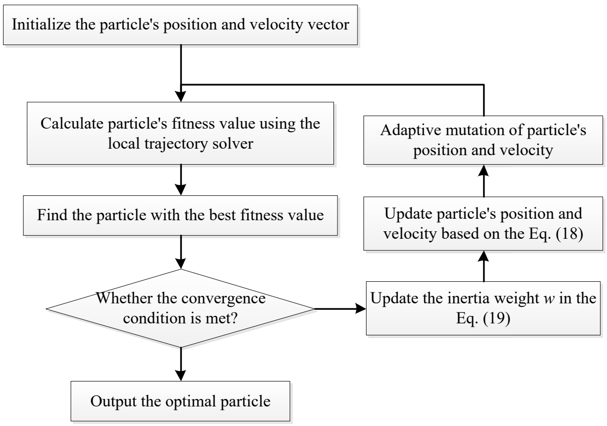

Initialize the particle's position and velocity vector
Calculate particle's fitness value using the local trajectory solver
Find the particle with the best fitness value
Whether the convergence condition is met?
Output the optimal particle
Update the inertia weight *w* in the Eq. (19)
Update particle's position and velocity based on the Eq. (18)
Adaptive mutation of particle's position and velocity

Figure 14 Schematic of the hybrid optimization algorithm combining PSO and SQP.

The principle of PSO is to find the optimal solution by making all the particles in the swarm move in a multi-dimensional hyper-volume [31, 32]. The motion of the particles is guided by the best-known positions of themselves and also the entire swarm in the search space. The particle motion is modeled as [31]

$$
\begin{cases}
\boldsymbol{V}_{i}^{k+1} = w_{k} \boldsymbol{V}_{i}^{k} + c_{1} r_{1}\left(\boldsymbol{X}_{ib}^{k} - \boldsymbol{X}_{i}^{k}\right) + c_{2} r_{2}\left(\boldsymbol{X}_{g}^{k} - \boldsymbol{X}_{i}^{k}\right) \\
\boldsymbol{X}_{i}^{k+1} = \boldsymbol{X}_{i}^{k} + \boldsymbol{V}_{i}^{k+1}
\end{cases}
\tag{19}
$$

where the subscript i and superscript k denote the i-th particle in the swarm and the k-th iteration, respectively. $\boldsymbol{X}$ and $\boldsymbol{V}$ are the position and velocity of the particle, respectively. $\boldsymbol{X}_{ib}^{k}$ is the best-known position of the i-th particle up to the k-th iteration. $\boldsymbol{X}_{g}^{k}$ is the swarm's best-known position up to the k-th iteration. w, c1 and c2 are, respectively, the inertia weight, the self-cognition coefficient and the global-cognitive coefficient, which are constants in the standard PSO algorithm.

To improve the performance of the standard PSO, two adjustments are made. Since the inertia weight determines the global and local search capabilities of the algorithm, a larger weight value enhances the global capability and weakens the local search capability. Actually, the algorithm is expected to have a stronger global and local capability in the early and later stages of execution, to avoid the local optimum and to improve computational efficiency. Therefore, an adaptive function is introduced to replace the constant inertia weight [32].

$$
w(k) = w_{0} - \left(w_{\mathrm{f}} - w_{0}\right)\left(\frac{k}{k_{\max}}\right)^{2}
\tag{20}
$$

where w_0 and w_f are the initial and final inertia weights, and $k_{\max}$ is the maximum number of iterations.

In addition, to improve the global optimization capability, an adaptive mutation is performed after the particle's position and velocity are updated. Particles are randomly selected and their position and velocity are randomly initialized.

5 Numerical simulations

This section uses the proposed method to perform the preliminary design of the MGA trajectory for the exploration of the Jovian system. Firstly, DNNs are built and trained to estimate the optimal sweepback duration for the gravity-assisted trajectories with the gravity-assisted central bodies of Europa, Ganymede and Calisto, respectively. Then, the trajectory of a Europa orbiter mission with multiple gravity assists of Galilean moons is designed using the proposed hybrid algorithm. The design is compared with the traditional patched conic method to validate the performance of the proposed algorithm in terms of accuracy and efficiency.

5.1 Optimal sweepback duration approximation using DNN

As mentioned in Section 3, since the major factors affecting the optimal sweepback duration of the approach and departure cases are different, two types of DNNs are built and trained to approximate the optimal approach and departure sweepback duration for each gravity-assist celestial body. The structures of these two DNNs are listed in Table 1, which are summarized from a series of simulation experiments. Both approach and departure DNNs are fully connected networks and use tanh and relu as the activation functions of the hidden and output layers, respectively. The departure DNN has more hidden layers because of more inputs. The training part runs using Python and the rest of the simulations, e.g. the hybrid optimization algorithm in Section 4, run using Matlab. All simulations are performed on the personal computer with Intel Core-i7 4.2 GHz CPU and 16 GB of RAM.

Table 1 The structures of the approach and departure DNNs.

Type of DNN	Approach DNN	Departure DNN
Neurons of input layer	3	5
Number of hidden layers	3	4
Neurons per hidden layer	30	30

The process of generating the approach sample is given as follows:

```
Approach Sample Generation Algorithm:
step 1:  Randomly generate the initial time $t_0$, the initial position $\boldsymbol{R}_0$ and
         the time of flight $tof$.
step 2:  Calculate the position $\boldsymbol{R}_{GA}$ and the velocity $\boldsymbol{V}_{GA}$ of the
         gravity-assist body at $t_0+tof$ using its ephemeris.
step 3:  Solve for the initial velocity $\boldsymbol{V}_0$ form the Lambert problem ($\boldsymbol{V}_0$,
         $\boldsymbol{V}_a$) = $L(\boldsymbol{R}_0, \boldsymbol{R}_{GA}, tof)$.
step 4:  Propagate the initial state ($\boldsymbol{R}_0$,$\boldsymbol{V}_0$) to the terminal state ($\boldsymbol{R}_f$,$\boldsymbol{V}_f$)
         under the gravitational influence of the Jupiter and the
         gravity-assisted body.
step 5:  Solve for the optimal sweepback duration $\Delta T^*$ to minimize the
         terminal position error $\Delta r^*$ using fmincon in Matlab.
step 6:  Calculate the median $\Delta T_{am}$ of the confidence interval of the
         approach sweepback duration via the method mentioned in
         Section 3.2.
step 7:  Calculate the angles between $\boldsymbol{v}_{inf} = \boldsymbol{V}_a-\boldsymbol{V}_{GA}$ and $\boldsymbol{R}_{GA}$ and $\boldsymbol{V}_{GA}$,
         which are denoted as $\theta$ and $\gamma$.
step 8:  Output the sample $S_a = \{[v_{inf}, \theta, \gamma], \Delta T_{am}\}$.
```

Similarly, the process of generating the departure sample is also given as follows:

```
Departure Sample Generation Algorithm:
step 1:  Randomly generate the initial time $t_0$, the hyperbolic excess
         velocity $\boldsymbol{v}_{inf}$ and the time of flight coefficient $\varepsilon_{tof}$ .
step 2:  Calculate the position $\boldsymbol{R}_{GA}$ and the velocity $\boldsymbol{V}_{GA}$ of the
         gravity-assisted body at $t_0$ using its ephemeris.
step 3:  Calculate the departure velocity $\boldsymbol{V}_d = \boldsymbol{V}_{GA} + \boldsymbol{v}_{inf}$, and the angle
         $\delta$ between $\boldsymbol{V}_d$ and $\boldsymbol{V}_{GA}$.
step 4:  If $\delta$ is less than 0.5 $\pi$ , continue to step 5, otherwise go back
         to step 1.
step 5:  Calculate the time of flight $tof = \varepsilon_{tof}T$, where $T$ is the orbital
         period of departure trajectory ($\boldsymbol{R}_{GA}$, $\boldsymbol{V}_d$).
step 6:  Calculate the initial position $\boldsymbol{r}_0$ and velocity $\boldsymbol{v}_0$ with respect to
         the gravity-assisted body using the method in Reference [24].
step 7:  Propagate the initial state ($\boldsymbol{r}_0$, $\boldsymbol{v}_0$) to the terminal state ($\boldsymbol{R}_f$, $\boldsymbol{V}_f$)
         under the gravitational influence of the Jupiter and the
         gravity-assisted body.
step 8:  Solve for the optimal sweepback duration $\Delta T^*$ to minimize
         the terminal position error $\Delta r^*$ using fmincon in Matlab.
step 9:  Calculate the median $\Delta T_{dm}$ of the confidence interval of the
         approach sweepback duration via the method mentioned in
         section 3.2.
step 10: Calculate the angles formed by $\boldsymbol{v}_0$ with $\boldsymbol{R}_{GA}$ and $\boldsymbol{V}_{GA}$, denoted
         as $\alpha$ and $\beta$ .
step 11: Output the sample $\boldsymbol{S}_d = \{[r_0, v_0, \alpha, \beta, tof], \Delta T_{dm} \}$.
```

The sample database is obtained by repeating the above process. The ranges of the randomly generated parameters are given in Table 2. The initial time t_0 is expressed in the modified Julian day. [R0, α_0, β_0] is the representation of $\boldsymbol{R}_0$ in a spherical coordinate inertial frame centered on Jupiter. [v_{inf}, α_{inf}, β_{inf}] is the representation of $\boldsymbol{v}_{inf}$ in the spherical coordinate inertial frame centered on the gravity-assisted body. R_J = 71492 km is the mean equatorial radius of the Jupiter.

Table 2 The ranges of the randomly generated initialization parameters for sample generation

Initialization parameter	Value ranges	Unit
t_0	[65941, 66151]	MJD2000
R_0	[5R_J, 100R_J]	km
α_0	[0, 360]	deg
β_0	[-10, 10]	deg
tof	[1, 100]	day

vinf	[1, 15] km/s	km/s
α_v	[0, 360]	deg
β_v	[-90, 90]	deg
ε_{tof}	[0, 1)	-

For the approach and departure cases of the Europa, Ganymede and Calisto, 50000 samples are generated using the proposed algorithm for each case. The Adaptive moment estimation (Adam) is used as the optimizer. The maximum epoch is set at 10000 and the initial learning rate is selected as 0.001. The buildup and the training of the DNN are based on the TensorFlow and Python. The mean square error (MSE) between the predication of DNN and the output of the sample are defined as the cost with the mathematical expression as [27]

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{21}$$

where n is the number of samples, and $\hat{y}_i$ and y$_i$ denote the DNN prediction and the actual sample output, respectively.

Finally, the variations of MSE during the training process are shown in Figure 15. The final MSEs of the DNN's prediction are less than 10^{-4} for all cases, which transforms into the mean absolute error (MAE) of the DNN's output in 0.01 per day. MAE is the average value of the error between the predicted value and the actual output.

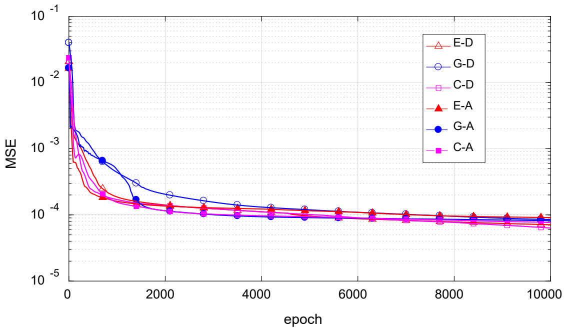


Figure 15 The variations of MSE during the training process (E: Europa, G: Ganymede, C: Calisto, A and D denote the approach and departure cases)

Then 1000 new samples are generated for each case to verify the performance of these trained DNNs. The terminal position errors of these samples are calculated using the sweepback duration estimated by these DNNs, and are given in Figure 16. For all cases, the terminal relative position error is greater than 0.25%, because the output of DNN is the median of the confidence interval instead of the optimal sweepback duration. Moreover, the means of the terminal relative position errors are all no larger than 0.5%, as reflected by the blue points in Figure 16. Their standard deviations are less than 0.3%. This indicates that the approximation accuracy of the DNNs can meet the requirements, since the error of the terminal relative position does not exceed 1% of the optimal value.

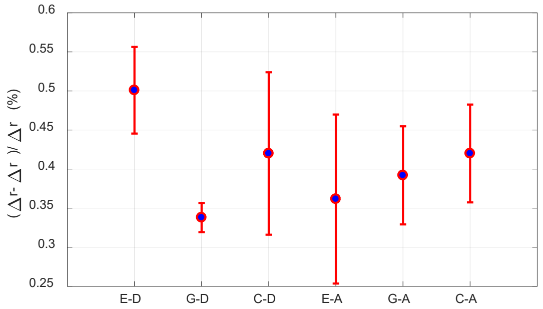


Figure 16 The terminal position errors of the 1000 new samples using the sweepback duration estimated by DNN (E: Europa, G: Ganymede, C: Calisto, A and D denote the approach and departure cases)

In this work, three measures are performed to simplify the problem for improving DNN's accuracy. First, the sample trajectory is limited to the low-inclination orbit (i.e. -10°≤β0≤10°). Because all inclinations of the Galilean moons' orbits are close to zero, the Europa orbiter mission trajectory also has a low inclination to obtain better gravitational assistance effects. Second, the sample dimension is reduced by identifying the major factors of the sweepback duration. Finally, for the same gravity-assist celestial body, two DNNs are trained separately for approach and departure cases. Different DNNs are trained for different gravity-assist celestial bodies.

5.2 Trajectory design of the Europa orbiter mission

The four Galilean moons are the focus of the Jovian system exploration, which is significant for understanding the solar system evolution and the origin of life. In this section, the proposed method is applied to perform the preliminary design of the Europa orbiter mission trajectory, which includes multiple gravity assists of Europa, Ganymede and Calisto.

It is assumed that the spacecraft parks on an elliptical orbit after being captured by Jupiter on June 12, 2039. The orbital elements of the parking orbit are listed in Table 3.

Table 3 The orbital elements of the initial parking orbit

Initial state	Value
MJD (day)	65951
a (km)	6878410.350
e	0.8982
inc (deg	0.206
RAAN (deg)	43.977
w (deg)	84.129
u (deg)	180

The Tisserand graph is a graphical technique for quick gravity-assist sequence search. It parameterizes the Tisserand invariant with respect to the infinity velocity of the spacecraft relative to the gravity assist body [33]. A feasible gravity-assist sequence that includes 9 gravity assistances is found via the Tisserand graph and shows in Figure 17. This sequence is represented by G-G-G-C-C-G-C-G-G-E, where E, G and C denote Europa, Ganymede and Calisto, respectively. First, three Ganymede gravity assistances are performed to lower the apoapsis of mission trajectory. Next, its periapsis is raised by applying two gravity assistances of Calisto. Finally, a G-C-G-G-E gravity-assisted sequence is employed for the rendezvous with Europa.

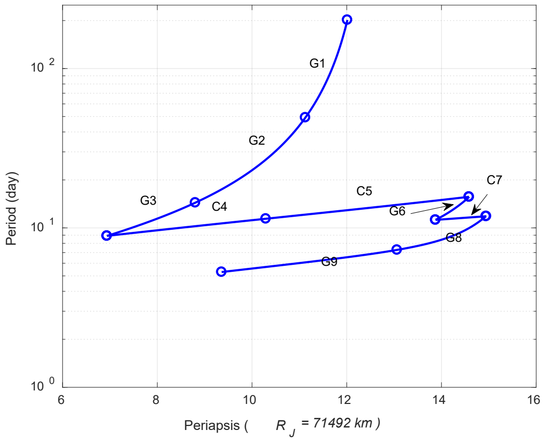


Figure 17 The gravity-assisted sequence of Galilean moons for Europa orbiter mission

The hybrid algorithm described in Section 4 is applied to solve the smooth mission trajectory in the restricted three-body dynamics. The swarm size and maximum generation of PSO are 100 and 200, respectively. The position accuracy tolerance of SQP is 1 km. The flight time constraints for each gravity-assisted leg are set according to the results of the Tisserand plot and are given in Table 4. In addition, the minimum safe altitude of the gravity-assisted legs of the Galilean moons is 100 km.

Table 4 The constraints of the time of flight for each gravity-assisted leg

Variables	Range (day)	Leg
tof_1	[80, 110]	Inject
tof_2	[47, 52]	G1
tof_3	[12, 17]	G2
tof_4	[7, 11]	G3
tof_5	[16,32]	C4
tof_6	[14,18]	C5
tof_7	[10,13]	G6
tof_8	[10,13]	C7
tof_9	[6, 9]	G8
tof_{10}	[4,6]	G9

The optimal results of the proposed method and the PCM are listed in Table 5. For comparison, the PCM model is also used to solve the same problem and its results are shown in Table 5. For the solution of the proposed method, the spacecraft takes 206.781 days and a delta-V of 1916.322 m/s to correct its trajectory and rendezvous Europa. The trajectory is illustrated in Figure 18. For the solution of the PCM, the flight time and delta-V are 247.962 days and 1858.455 m/s, respectively. In addition, the CPU computational time of the proposed method and PCM are 352.359 s and 95.246 s, respectively. Because the PCM is based on two-body dynamics, its computational efficiency is higher than that of the proposed method. However, its poor accuracy will cost more time and delta-V for the correction to the high-fidelity dynamical model.

A high-fidelity dynamical model that considers the gravitation of Jupiter and its four Galileo moons is used to verify the accuracy of the preliminary design. The formula of the high-fidelity dynamics is defined as

$$\ddot{\boldsymbol{R}} = -\mu_{\mathrm{J}} \frac{\boldsymbol{R}}{R^3} - \sum_{g=1}^{4} \mu_g \left(\frac{\boldsymbol{R}_{gs}}{R_{gs}^3} + \frac{\boldsymbol{R}_g}{R_g^3} \right) \qquad (22)$$

where μ_{J} is the gravitational constant of Jupiter. μ_{g} (g = 1,2,3, and 4) are the gravitational constants of Galilean moons. $\boldsymbol{R}$ and $\boldsymbol{R}_{\mathrm{g}}$ denote the position vectors of the spacecraft and the Galilean moon g. $\boldsymbol{R}_{\mathrm{gs}}$ is the relative position vector for the Galilean moon g to the spacecraft. All position vectors are described in the inertial coordinate frame centered on Jupiter.

The terminal position error of each leg of the preliminary design solutions is shown in Table 5, which is denoted as ΔR_{f}. The terminal position error of the preliminary design solution of the proposed method is reduced by an order of magnitude with the comparison of the PCM's solution.

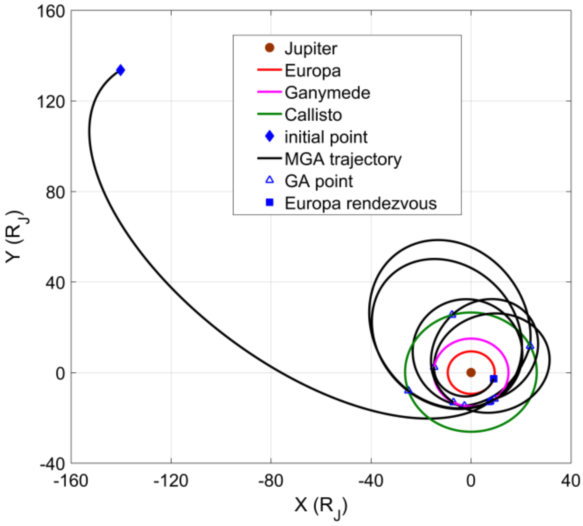


Figure 18 The MGA trajectory of the solution of the proposed method

Table 5 The preliminary design results of the proposed method and PCM for the Europa orbiter mission.

Leg	Proposed Method				PCM			
	time (MJD)	r_{pe} (km)	Δv (m/s)	ΔR_{f} (km)	time (MJD)	r_{pe} (km)	Δv (m/s)	ΔR_{f} (km)

Inject	65951	--	19.923	7848	65951		16.612	524231
G1	66036.393	2734	17.578	858	66064.047	2734	0.903	44126
G2	66086.448	3047.771	6.603	551	66112.940	2734	2.274	3699
G3	66099.992	10876.652	36.840	385	66127.009	4818.408	0.291	6547
C4	66102.306	6431.124	28.299	2436	66134.164	2508	24.278	10365
C5	66127.022	14817.494	2.741	154	66165.077	2508	8.457	35652
G6	66129.595	2734	2.342	142	66168.815	6232.847	3.014	78837
C7	66139.463	3631.935	52.287	112	66181.892	2508.	8.823	15162
G8	66142.583	9320.329	3.354	678	66185.794	2734.	2.756	113476
G9	66155.482	2734	2.019	268	66197.384	2734.	6.518	5930
E	66157.781	--	1744.336	151	66198.962	--	1784.529	3920
Total	--	--	1916.322	--	--	--	1858.455	--

To obtain the high-fidelity solution, the Newton's method is used to correct the preliminary solution in the high-fidelity dynamics. The terminal position tolerance of the correction is set to 1 km. The corrected results are given in Table 6. Δv_c is the velocity change after the correction.

Table 6 The correction results based on the high-fidelity dynamics using the proposed method and PCM solutions as the initial trajectories

Event	Proposed Method			PCM		
	Δv_c (m/s)	iterations	CPU time	Δv_c (m/s)	iterations	CPU time
Inject	21.939	312	31.034	17.612	1325	121.151
G1	16.257	145	13.559	38.245	973	91.305
G2	10.631	129	12.385	62.704	652	62.860
G3	29.465	86	8.249	16.931	1313	129.943
C4	35.129	69	6.524	62.265	1297	123.306
C5	9.411	36	3.316	38.957	1179	110.294
G6	5.132	48	4.559	62.204	655	62.362
C7	46.728	39	3.700	21.356	255	24.778
G8	9.264	44	4.195	12.756	657	65.109
G9	7.391	19	1.776	56.518	148	13.561
E	1714.336	24	2.326	1794.529	137	12.866
Total	1905.683	951	91.623	2184.077	8591	817.535

For each transfer leg, the terminal position error of the PCM solution is much larger than that of the proposed method solution, resulting in more iterations and more CPU computational time for the correction. The total computational time is defined as the summation of the computational time of the preliminary design and the correction. For the proposed method, the design and correction computational time are respectively 91.623 s and 352.359 s, and the total computational time is 443.982 s. For the PCM, the design and correction computational time are 95.246s and 817.535 s, respectively. Therefore, its total computational time is 912.781 s. After the correction, the total delta-V of the proposed method is 1905.683 m/s, which is less than the PCM solution 2184.077 m/s. In addition, Comparing the total delta-V of the preliminary design solution in Table 5 with the total delta-V after accurate correction in Table 6, the delta-V errors of preliminary design solutions of the PCM and the proposed method are 325.622 m/s and 10.936 m/s, respectively. Therefore, when considering both the preliminary design of MGA trajectory and the subsequent correction for the high-precision dynamics, the proposed method demonstrates significant advantages in both computational efficiency and accuracy.

6 Conclusions

A preliminary design method for generating accurate and continuous trajectories using the pseudostate technique and a hybrid algorithm has been proposed to solve the MGA trajectory. The major contribution of this novel method is to develop a gravity-assisted patched strategy based on the pseudostate technique, whose sweepback duration is firstly approximated using DNN. It is demonstrated that the DNN is capable to provide an accurate sweepback duration for the pseudostate gravity-assisted model, which significantly improves the accuracy and efficiency of solving the gravity-assisted trajectory. In the simulations of the Galilean moons gravity assistances, the terminal position error caused by the estimated sweepback duration using the well-trained DNN, does not exceed 1% of the optimal value. Additionally, both the proposed method and the PCM are applied to solve the MGA trajectory for the Europa orbiter mission. The simulation results show that the solution of the proposed method is more accurate and easier to refine in the high-fidelity dynamics. Therefore, the proposed method demonstrates significant advantages in the computational efficiency and accuracy for the high-fidelity dynamics.

The dataset and accuracy of DNN is a major concern of the application. In fact, we found it is very difficult to generate a satisfactory training dataset using one general DNN for all case and all gravity-assist bodies. This challenge is the focus of our future work. Additionally, the Jovian system is a nonlinear multi-body dynamics, which is the expansion goal of our future work as well.

The work was supported by the National Natural Science Foundation of China (Grant No. 61273051), sponsored by Qing Lan Project, Funding for Outstanding Doctoral Dissertation in NUAA (Grant No. BCXJ19-12), State Scholarship from China Scholarship Council (Grant No. 201906830066). The authors fully appreciate the financial support.

1 Murray B C. The Mariner 10 pictures of Mercury: an overview. J. Geophys Res, 1975, 80(17): 2342-2344.

2 O'Neil W J, Ausman N E, Gleason J A, et al. Project Galileo at Jupiter. Acta Astronaut, 1997, 40(2-8): 477-509.

3 Lebreton J P, Matson D L. An overview of the Cassini mission. Nuovo Cimento C, 1992, 15(6): 1137-1147.

4 Shen H X, Zhou J P, Peng Q B, et al. Multi-objective interplanetary trajectory optimization combining low-thrust propulsion and gravity-assist maneuvers. Sci China Tech Sci, 2012, 55(3): 841-847.

5 Battin R H. The determination of round-trip planetary reconnaissance trajectories. J. Aerosp Sci, 1959, 26(9): 545-567.

6 Minovitch M A. The invention that opened the solar system to exploration. Planet Space Sci, 2010, 58(6): 885-892.

7 Broucke R. The celestial mechanics of gravity assist, Astrodynamics Conference, 1988: 4220.

8 Prado A F B. Powered swingby. J. Guid. Control. Dyn., 1996, 19(5): 1142-1147.

9 Zimmer S, Ocampo C. Use of analytical gradients to calculate optimal gravity-assist trajectories. J. Guid. Control. Dyn., 2005, 28(2): 324-332.

10 Miller J, Weeks C. Application of Tisserand's criterion to the design of gravity assist trajectories, AIAA/AAS Astrodynamics Specialist Conference and Exhibit. 2002: 4717.

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

12 Pisarevsky D M, Kogan A, Guelman M. Building interplanetary trajectories with multiple gravity-assisted maneuvers. J. Spacecraft Rockets, 2007, 44(4): 985-992.

13 Gad A, Abdelkhalik O. Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization. J. Spacecraft Rockets, 2011, 48(4): 629-641.

14 Wagner S, Wie B. Hybrid algorithm for multiple gravity-assist and impulsive delta-V maneuvers. J. Guid. Control. Dyn., 2015, 38(11): 2096-2107.

15 Lu Y, Li H N, Li J S, et al. Design and optimization of low-energy transfer orbit to Mars with multi-body environment. Sci China Tech Sci, 2015, 58(10): 1660-1671.

16 Prado A. A comparison of the “patched-conics approach” and the restricted problem for swing-bys. Adv Space Res, 2007, 40(1): 113-117.

17 Negri R B, de Almeida Prado A F B, Sukhanov A. Studying the errors in the estimation of the variation of energy by the “patched-conics” model in the three-dimensional swing-by. Celest. Mech. Dyn. Astr., 2017, 129(3): 269-284.

18 Ferreira A F S, Moraes R V, Prado A F B A, et al. Errors of Powered Swing-By in the Restricted Three-Body Problem. J. Guid. Control. Dyn., 2019, 42(10): 2246-2257.

19 Wilson J.S. A pseudostate theory for the approximation of three-body trajectories, Astrodynamics Conference, 1969: 1061.

20 Ramanan R V. Integrated algorithm for lunar transfer trajectories using a pseudostate technique. J. Guid. Control. Dyn., 2002, 25(5): 946-952.

21 Ramanan R V, Adimurthy V. Nonimpact lunar transfer trajectories using the pseudostate technique. J. Guid. Control. Dyn., 2005, 28(2): 217-225.

22 Bao C, Li J, Baoyin H. Two-segment lunar free-return trajectories design using the pseudostate theory. Adv Space Res, 2018, 61(1): 97-110.

23 Byrnes D V. Application of the pseudostate theory to the three-body Lambert problem. J. Astronaut Sci, 1989, 37: 221-232.

24 Parvathi S P, Ramanan R V. Iterative pseudostate method for transfer trajectory design of interplanetary orbiter missions. J. Guid. Control. Dyn., 2016, 39(12): 2799-2809.

25 Sweetser III T H. Some notes on applying the one-step multiconic method of trajectory propagation. Proceedings of the AAS/AIAA Astrodynamics Conference, Kalispell, MT, Aug. 10-13, 1987. Part 1 (A89-12626 02-12). San Diego, CA, Univelt, Inc., 1988: 731-752.

26 Yang B, Yang H, Li S. Pseudostate theory based iterative preliminary design method for powered gravity-assist interplanetary trajectories. Acta Astronaut, 2019, 165: 139-149.

27 Li H, Chen S, Izzo D, et al. Deep networks as approximators of optimal low-thrust and multi-impulse cost in multitarget missions. Acta Astronaut, 2020, 166: 469-481.

28 Izzo D, Märtens M, Pan B. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. Astrodynamics, 2019, 3(4): 287-299.

29 Zhang K, Shang H, Chen Q, et al. Evaluating Gravity-Assist Range Set Based on Supervised Machine Learning, IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2018, 449(1): 012021.

30 Ampatzis C, Izzo D. Machine learning techniques for approximation of objective functions in trajectory optimization, Proceedings of the ijcai-09 workshop on artificial intelligence in space. 2009: 1-6.

31 Xia N, Han D, Zhang G F, et al. Study on attitude determination based on discrete particle swarm optimization. Sci China Tech Sci, 2010, 53(12): 3397-3403.

32 Ge X S, Sun K. Optimal control of a spacecraft with deployable solar arrays using particle swarm optimization algorithm. Sci China Tech Sci, 2011, 54(5): 1107-1112.

33 Heaton A F, Strange N J, Longuski J K, et al. Automated design of the Europa Orbiter tour. J. Spacecraft Rockets, 2002, 39(1): 17-22.