

Article

Multivariate Attention-Based Orbit Uncertainty Propagation and Orbit Determination Method for Earth–Jupiter Transfer

Zhe Zhang ^{1,2,3}, Yishuai Shi ¹ and Hongwei Han ^{1,4,*}¹ School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China² Peng Cheng Laboratory, Department of Mathematics and Theories, Shenzhen 518000, China³ Deep Space Exploration Laboratory, Beijing 100089, China⁴ Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China

* Correspondence: hanhongwei@bit.edu.cn

Abstract: Current orbit uncertainty propagation (OUP) and orbit determination (OD) methods suffer from drawbacks related to high computational burden, limiting their applications in deep space missions. To this end, this paper proposes a multivariate attention-based method for efficient OUP and OD of Earth–Jupiter transfer. First, a neural network-based OD framework is utilized, in which the orbit propagation process in a traditional unscented transform (UT) and unscented Kalman filter (UKF) is replaced by the neural network. Then, the sample structure of training the neural network for the Earth–Jupiter transfer is discussed and designed. In addition, a method for efficiently generating a large number of samples for the Earth–Jupiter transfer is presented. Next, a multivariate attention-based neural network (MANN) is designed for orbit propagation, which shows better capacity in terms of accuracy and generalization than the deep neural network. Finally, the proposed method is successfully applied to solve the OD problem in an Earth–Jupiter transfer. Simulations show that the proposed method can obtain a similar estimation to the UKF while saving more than 90% of the computational cost.

Keywords: orbit uncertainty propagation; orbit determination; deep space; multivariate attention; Earth–Jupiter transfer; UKF



Citation: Zhang, Z.; Shi, Y.; Han, H. Multivariate Attention-Based Orbit Uncertainty Propagation and Orbit Determination Method for Earth–Jupiter Transfer. *Appl. Sci.* **2024**, *14*, 4263. <https://doi.org/10.3390/app14104263>

Academic Editors: Rosario Pecora and Giovanni Bernardini

Received: 27 March 2024

Revised: 6 May 2024

Accepted: 9 May 2024

Published: 17 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Orbit uncertainty propagation (OUP) and orbit determination (OD) are essential in space missions such as near-Earth missions [1–3], asteroid exploration missions [4–6], lunar exploration missions [7], Venus exploration missions [8], and Jupiter exploration missions [9,10]. Among these space missions, deep space exploration missions rely on extremely accurate OUP and OD techniques. The OUP involves predicting the orbital states (and the associated uncertainties), whereas the OD involves determining the precise position and velocity of a space target, such as a spacecraft or an asteroid, with respect to a reference frame [11–13].

Accurate OUP and OD are crucial for many deep space missions, including spacecraft navigation, design of trajectory correction maneuvers, planetary exploration, and more [14,15]. High accuracy helps to optimize and design robust trajectories that consider uncertainties, which is particularly important for mission planning and execution [16,17]. Additionally, it enables the prediction of future spacecraft positions and velocities [18,19], which is useful for coordinating operations and avoiding collisions with other objects in deep space [20–22].

Classic OUP methods include Monte Carlo simulation [23], unscented transform (UT) [24–26], cubature rule [27], state transition matrix [28,29], state transition tensor (STT) [30–33], and Gaussian mixture model (GMM) [34]. Park and Scheeres investigated the OUP problem of a spacecraft on a Sun–Earth L1 orbit [32,33]. Kim and Park derived an STM for OUP of a fly-by spacecraft [35]. Boone and McMahan analyzed the stability of the Europa Lander orbit using STT [36]. Qiao et al proposed a semi-analytical uncertainty propagation method for a heliocentric constellation based on STT techniques [31].

In general, OD can be performed for a deep space probe using optical [37], pulsar-based [10], and doppler-Range radio [35] measurements. Qiao et al. investigated the possibility of optical navigation during the approach of an asteroid [5]. They performed observability analysis for the optical navigation and proposed an observability improvement method based on maneuvers. Mortari and Conway proposed a weighted least squares method for state estimation of interplanetary orbits using multi-angle measurements [38]. Christian investigated the possibility of deep-space optical navigation using a planet's centroid and apparent diameter in images [39]. Compared with angle-only measurements, the use of the apparent diameter can improve the observability of optical navigation. Christian further investigated the pose estimation and optical navigation methods using the features of the rotating celestial bodies (e.g., asteroids) and spheroidal planets with banded atmospheres (e.g., Jupiter) [9]. Andreis et al. analyzed the performance of optical navigation of a CubeSat in an Earth–Mars transfer [11]. Ma et al. reviewed deep space navigation methods and concluded that the conventional doppler-rRange radio navigation method can hardly satisfy the deep space mission requirement of high performance in real-time accuracy and coverage [10]. In addition, they found that the higher frequency of pulsars leads to a better navigation solutions compared to optical navigation. Ely et al. compared deep-space navigation methods using optical, pulsar-based, and radiometric measurements and concluded that significant pulsar time-of-arrival measurement noise limits its utility for approach and entry navigation [40].

The least squares (LS) and Kalman filter are two basic methods for OD [41,42]. In LS-based OD, a set of measurements, such as range and angle measurements, is used to estimate the initial state vector of a spacecraft. The basic idea of LS is to minimize the difference between the observed and predicted measurements by adjusting the initial state vector [43]. This adjustment is done by solving a system of linear equations, where the solution corresponds to the state vector that best fits the observations in a least-squares sense. LS-based OD is widely used in space-related applications such as satellite navigation and OD of space debris and asteroids. However, LS has limitations for onboard applications as it is a batch method that requires high computational cost.

The Kalman filter and its variants belong to the category of sequential methods, requiring a lower computational cost than LS. Commonly used Kalman filters include (to name a few) the extended Kalman filter (EKF), unscented Kalman filter (UKF) [24,25], cubature Kalman filter (CKF) [27], and Gaussian mixture filter (GMF) [44]. In order to effectively filter data on onboard applications, two fundamental prerequisites must be fulfilled, namely, achieving high accuracy while maintaining low complexity. Nevertheless, these requirements can sometimes be in conflict with one another [45]. For instance, the UKF is recognized for its superior accuracy compared to the EKF; however, its computational cost is significantly higher due to the integration of over ten sigma points [30,46].

Several research studies have been carried out recently to improve the efficiency of the OUP and OD methods for onboard applications. Greco and Vasile developed a surrogate-based particle filter in which a surrogate model replaces the orbit integrations in the OUP process of a conventional particle filter [47]. Zhou et al. developed an adaptive order-switching Kalman filter (AOSKF) for cislunar OD [45]. In AOSKF, a deep neural network (DNN) is employed to detect highly nonlinear points and the OUP's orders are adaptively adjusted according to the nonlinear detection results. More recently, Zhou et al. proposed a DNN-based orbital uncertainty propagation and estimation framework for low-Earth orbit (LEO) spacecraft in which the computational cost is significantly reduced [46]. In this method, the orbit propagation process in the conventional UKF is replaced by a DNN-based prediction process. However, such an OUP and OD method can be further improved, as the prediction capacity of the DNN is limited.

Motivated by Zhou et al.'s work [46], this paper aims to improve the computational efficiency of the OD using a neural network. A deep space transfer case, the Earth–Jupiter transfer, is taken as an example to show the performance of such a method. Compared with Zhou et al.'s work, the innovations of this paper can be summarized as follows:

- A multivariate attention-based neural network (MANN), rather than a simple DNN, is designed and utilized to replace the orbit integration process in the UT and UKF. It will be shown that the designed MANN has a much more advanced performance than the DNN in the investigated problem.
- An Earth–Jupiter transfer case, rather than an LEO case, is considered in this work. The input structure of the sample is comprehensively discussed and carefully designed.
- A method for fast generating a large number of samples for the Earth–Jupiter transfer case is presented.

The remainder of this paper is organized as follows. Section 2 describes the neural network-based OD framework. Section 3 discusses the construction of the samples and introduces the structure and training process of the MANN. Numerical simulations are provided in Section 4 and conclusions in Section 5.

2. Neural Network-Based Orbit Uncertainty Propagation Orbit Determination Framework

2.1. Orbit Uncertainty Propagation

In this work, the dynamics of a spacecraft during the transfer from Earth to Jupiter considers the point-mass gravitational accelerations and the Solar Radiation Pressure (SRP) perturbations; thus, the high-fidelity dynamics model can be written as

$$\dot{x} = \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ -\frac{\mu_S}{\|r\|^3}r + \sum_{k \in \{E, M, J\}} \mu_k \left(\frac{r_k - r}{\|r_k - r\|^3} - \frac{r_k}{\|r_k\|^3} \right) + C_R \frac{A}{m} P_{\odot} \text{AU}^2 \frac{r}{\|r\|^3} \end{bmatrix}, \quad (1)$$

where $x \in [r; v] \in \mathbb{R}^6$ represents the state vector (with $r \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$ being the position and velocity vectors, respectively) of the spacecraft in the heliocentric inertial coordinate, μ_S , μ_E , and μ_J denote the gravitational constants of the Sun, Mars, Earth, and Jupiter, respectively, $r_E \in \mathbb{R}^3$, $r_M \in \mathbb{R}^3$, and $r_J \in \mathbb{R}^3$ are the position vectors of the Earth, Mars, and Jupiter in the heliocentric inertial coordinate, C_R is the reflection coefficient, A/m is the area-to-mass ratio, and $P_{\odot} = 4.56 \times 10^{-6} \text{ N/m}^2$ is the reference value of the SRP at a distance of 1 AU (1 AU = 149,597,870 km, and $\text{AU}^2 = 149,597,870 \text{ km} \times 149,597,870 \text{ km}$) from the Sun [8,11]. Note that in this work the ephemeris data of the Earth, Mars, and Jupiter come from the Jet Propulsion Laboratory’s Development Ephemerides (DE432). In this work, the J2 term of the Earth and the propulsion term are not considered in order to simplify the problem (as in [11]). In addition, we define

$$c = C_R \frac{A}{m} \quad (2)$$

and in the remainder of this paper, given the initial epoch t_0 and the corresponding initial orbital state $x_0 \in \mathbb{R}^6$, the solution flow of the ordinary differential equation in Equation (1) can be represented as $x = \phi(t; x_0, t_0)$.

Orbit deviation propagation is the premise of orbit determination [46]. This is because orbit deviation propagation, also called the orbit uncertainty propagation technique, can help predict the uncertainties, which is necessary to calculate the Kalman gain in a Kalman filter framework. In addition, it can help provide gradient information for the LSM. As shown in Figure 1, the black solid line and the black dashed line represent the nominal orbit and the neighboring orbit, respectively, under the high-fidelity dynamics model. Let $\tilde{x}_0 = x_0 + \delta x_0 \in \mathbb{R}^6$ be the initial orbital state of the neighboring orbit, with $\delta x_0 \in \mathbb{R}^6$ being the initial deviation between the nominal and neighboring orbits. By propagating the orbits under high-fidelity dynamics, the orbital states of the nominal and neighboring orbits at any given epoch t can be obtained as $x = \phi(t; x_0, t_0)$ and $\tilde{x} = \phi(t; \tilde{x}_0, t_0)$. The orbit deviation between the nominal and neighboring orbits at a given epoch t is defined as

$$\delta x = \tilde{x} - x \in \mathbb{R}^6 \quad (3)$$

The orbit deviation propagation is used to calculate or predict the deviation δx given the initial state of the nominal orbit x_0 and the initial deviation δx_0 . In general, there are two ways to calculate the deviation δx . The first is to propagate the neighboring orbit directly and calculate the deviation δx using Equation (3). The other is to approximate the deviation δx using techniques such as state transition matrix (STM), state transition tensor (STT), or polynomial chaos (PC). Both approaches require numerical integrations, making its computational cost unacceptable for certain onboard applications, especially CubeSat-like platforms with limited computational capacity.

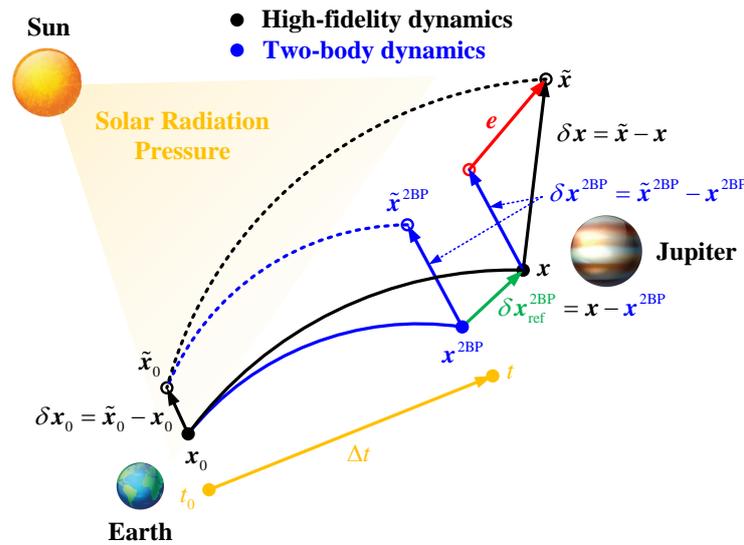


Figure 1. An illustration of the orbit deviation propagation framework.

Recently, a novel neural network-based method was developed in [46] to significantly improve the computational cost of the OD of an LEO spacecraft. This work extends such an idea to solve the OD problem of an Earth–Jupiter transfer spacecraft with the framework shown in Figure 1. In Figure 1, the orbits propagated using high-fidelity and two-body dynamics (considering only the Sun’s gravitational force) are represented by black and blue lines, respectively. In addition, the orbital state propagated by the two-body dynamics is distinguished using a superscript “2BP”. For example, $x^{2BP} \in \mathbb{R}^6$ and $\tilde{x}^{2BP} \in \mathbb{R}^6$ represent the orbital states of the nominal and neighboring orbits under the two-body dynamics. Under the two-body dynamics, the orbit deviation in Equation (3) has an analytical solution (i.e., $\delta x^{2BP} = \tilde{x}^{2BP} - x^{2BP}$ can be analytically computed). The detailed expressions for the orbit deviation propagation under the two-body dynamics are well presented in [46,48]. As shown in Figure 1, we define $e = \delta x - \delta x^{2BP}$, which represents the residual between the true (under the high-fidelity dynamics) and two-body orbit deviation propagation solutions. If the residual e can be predicted using a neural network, then the value of δx can be obtained without additional numerical integrations, which can significantly reduce the computational burden.

In this work, a multivariate attention-based neural network (MANN) is developed to predict the residual e . The designed structure and the training process of the MANN will be introduced in the next section. Let \hat{e}_{MANN} be the prediction of the MANN; then, the orbit deviation propagation solution can be rewritten as

$$\delta x \approx \hat{e}_{\text{MANN}} + \delta x^{2BP}. \tag{4}$$

In the following subsection, the solution in Equation (4) is combined with a UT method (denoted as MANN-UT) to improve the OD efficiency of the conventional UKF method.

2.2. Orbit Determination Method

The framework of the MANN-UT is illustrated in Figure 2. Given the mean of the initial state $x_0 \in \mathbb{R}^6$ and its associated covariance $P_0 \in \mathbb{R}^{6 \times 6}$, thirteen sigma points $\chi_i(t_0) \in \mathbb{R}^6$ ($i \in \{0, 1, \dots, 12\}$) can be obtained under the UT framework as follows:

$$\begin{aligned} \chi_i(t_0) &= x_0 & i &= 0 \\ \chi_i(t_0) &= x_0 + (\sqrt{(n + \lambda)P_0})_i & i &= 1, \dots, n \\ \chi_i(t_0) &= x_0 - (\sqrt{(n + \lambda)P_0})_i & i &= n + 1, \dots, 2n \end{aligned} \tag{5}$$

where $n = 6$ is the dimension of the orbit state, $\lambda = \alpha^2(n + \kappa) - n$ is an adjusted factor, $(\sqrt{(n + \lambda)P_0})_i$ is the i -th column of the matrix $\sqrt{(n + \lambda)P_0}$, $P_0 = \sqrt{P_0}(\sqrt{P_0})^T$, and $\sqrt{P_0}$ is computed using the Cholesky decomposition. In this work, $\kappa = 3 - n$ and $\alpha = 1$.

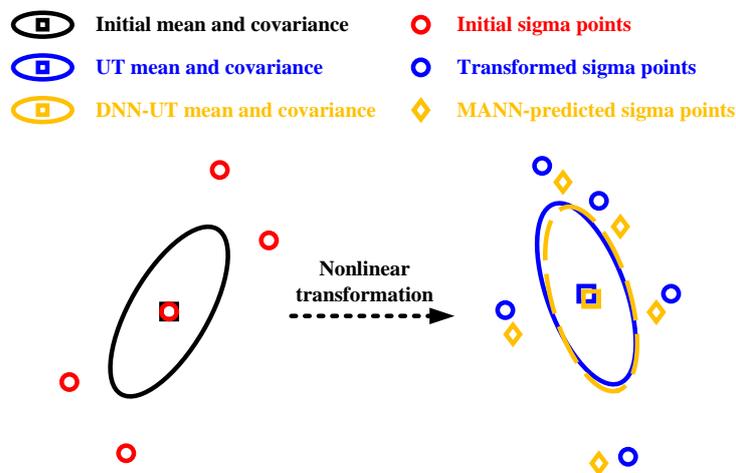


Figure 2. An illustration of the framework of MANN-UT.

In the conventional UT framework, the thirteen sigma points in Equation (5) are propagated from the initial epoch t_0 to a given future epoch t using high-fidelity dynamics; thus, we have $\chi_i(t) = \phi[t; \chi_i(t_0), t_0]$ ($i \in \{0, 1, \dots, 12\}$). After weighing these propagated sigma points, the mean and covariance can be computed as

$$\begin{aligned} \bar{x}(t) &= \sum_{i=0}^{2n} w_i^{(m)} \chi_i(t) \\ P(t) &= \sum_{i=0}^{2n} w_i^{(c)} [\chi_i(t) - \bar{x}(t)][\chi_i(t) - \bar{x}(t)]^T \end{aligned} \tag{6}$$

where $\bar{x}(t) \in \mathbb{R}^6$ and $P(t) \in \mathbb{R}^{6 \times 6}$ represent the mean and covariance at epoch t and $w_i^{(m)}$ and $w_i^{(c)}$ are the weights of the mean $\bar{x}(t)$ and covariance $P(t)$, with their expressions provided by

$$\begin{cases} w_i^{(m)} = \frac{\lambda_{UT}}{\lambda_{UT} + n}, w_i^{(c)} = w_i^{(m)} + (1 - \alpha^2 + \beta) & i = 0 \\ w_i^{(m)} = w_i^{(c)} = \frac{1}{2(\lambda_{UT} + n)} & \text{elsewise} \end{cases} \tag{7}$$

where $\beta = 2$ is used in this work.

Using Equation (4), the propagated sigma points can be written as

$$\begin{aligned} \chi_i(t) &= x(t) + \delta \chi_i^{2BP} + \hat{e}_{MANN,i} \\ &= \phi(t; x_0, t_0) + \delta \chi_i^{2BP} + \hat{e}_{MANN,i} \end{aligned} \tag{8}$$

where $\delta\chi_i^{2BP}$ denotes the two-body orbital deviation of the i -th sigma point.

Substituting Equation (8) into Equation (6), the mean can be approximated as

$$\begin{aligned} \bar{x}(t) &\approx \sum_{i=0}^{2n} w_i^{(m)} \{ \phi[t; \chi_i(t_0), t_0] + \delta\chi_i^{2BP} + \hat{e}_{MANN,i} \} \\ &= \phi(t; \chi_i(t_0), t_0) + \sum_{i=0}^{2n} w_i^{(m)} (\delta\chi_i^{2BP} + \hat{e}_{MANN,i}). \end{aligned} \tag{9}$$

In addition, the covariance $P(t)$ can be rewritten as

$$\begin{aligned} P(t) &\approx \sum_{i=0}^{2n} w_i^{(c)} \left[\delta\chi_i^{2BP} + \hat{e}_{MANN,i} - \sum_{j=0}^{2n} w_j^{(m)} (\delta\chi_j^{2BP} + \hat{e}_{MANN,j}) \right] \\ &\times \left[\delta\chi_i^{2BP} + \hat{e}_{MANN,i} - \sum_{j=0}^{2n} w_j^{(m)} (\delta\chi_j^{2BP} + \hat{e}_{MANN,j}) \right]^T. \end{aligned} \tag{10}$$

Motivated by [46], by applying such a MANN-UT method into a UKF framework, a MANN-assisted UKF (denoted as MANN-UKF) is presented herein.

First, the OD system of the Earth–Jupiter transfer orbit can be written as

$$\begin{cases} x_{k+1} = \phi(t_{k+1}; x_k, t_k) + \omega_{k+1} \\ z_{k+1} = h(x_{k+1}) + \varepsilon_{k+1} \end{cases} \tag{11}$$

where $x_k = x(t_k)$ and $x_{k+1} = x(t_{k+1})$ represent the orbital state at epochs t_k and t_{k+1} , respectively, z_{k+1} is the measurement vector at epoch t_{k+1} , $h(\cdot)$ denotes the measurement model, ω_{k+1} is the process noise, and ε_{k+1} is the measurement noise. We assume that the process and measurement noises are zero-mean Gaussian errors, with their covariance being Q_{k+1} and R_{k+1} , respectively, i.e., $\mathbb{E}(\omega_{k+1}\omega_{k+1}^T) = Q_{k+1}$ and $\mathbb{E}(\varepsilon_{k+1}\varepsilon_{k+1}^T) = R_{k+1}$. Note that Equation (11) is a discrete format of the OD system, which should be distinguished from Equations (1)–(10).

Based on the MANN-UT presented in Equations (9) and (10), the process of the MANN-UKF is provided as follows:

A. Time Update

- Given the estimated state \hat{x}_k and the associated covariance P_k at epoch t_k , calculate the predicted state $\hat{x}_{k+1|k}$ and the associated covariance $P_{k+1|k}$:

$$\hat{x}_{k+1|k} \approx \phi(t_{k+1}; \hat{x}_k, t_k) + \sum_{i=0}^{2n} w_i^{(m)} (\delta\chi_i^{2BP} + \hat{e}_{MANN,i}) \tag{12}$$

$$\begin{aligned} P_{k+1|k} &= \sum_{i=0}^{2n} w_i^{(c)} \left[\delta\chi_i^{2BP} + \hat{e}_{MANN,i} - \sum_{j=0}^{2n} w_j^{(m)} (\delta\chi_j^{2BP} + \hat{e}_{MANN,j}) \right] \\ &\times \left[\delta\chi_i^{2BP} + \hat{e}_{MANN,i} - \sum_{j=0}^{2n} w_j^{(m)} (\delta\chi_j^{2BP} + \hat{e}_{MANN,j}) \right]^T + Q_{k+1}. \end{aligned} \tag{13}$$

B. Measurement update

- Calculate the Cholesky decomposition of $P_{k+1|k}$ and generate sigma points $\chi_{i,k+1|k}$.
- Calculate the predicted measurement $\hat{z}_{k+1|k}$ and the associated covariance $P_{xz,k+1|k}$ and $P_{zz,k+1|k}$:

$$z_{i,k+1|k}^* = h(\chi_{i,k+1|k}) \tag{14}$$

$$\hat{z}_{k+1|k} = \sum_{i=0}^{2n} w_i^{(m)} z_{i,k+1|k}^* \tag{15}$$

$$P_{xz,k+1|k} = \sum_{i=0}^{2n} w_i^{(c)} (\chi_{i,k+1|k}^* - \hat{x}_{k+1|k})(z_{i,k+1|k}^* - \hat{z}_{k+1|k})^T \tag{16}$$

$$P_{zz,k+1|k} = \sum_{i=0}^{2n} w_i^{(c)} (z_{i,k+1|k}^* - \hat{z}_{k+1|k}) (z_{i,k+1|k}^* - \hat{z}_{k+1|k})^T + R_{k+1}. \tag{17}$$

- When the measurement z_{k+1} has been collected, calculate the estimated state \hat{x}_{k+1} and the associated covariance P_{k+1} at epoch t_k :

$$K_{k+1} = P_{xz,k+1|k} P_{zz,k+1|k}^{-1} \tag{18}$$

$$\hat{x}_{k+1} = x_{k+1|k} + K_{k+1} (z_{k+1} - \hat{z}_{k+1|k}) \tag{19}$$

$$P_{k+1} = P_{k+1|k} - K_{k+1} P_{zz,k+1|k} K_{k+1}^T. \tag{20}$$

Compared with traditional UKF, the advantage of using the proposed MANN-UKF is that it actually only needs to integrate one sigma sample point to calculate the prediction state $\hat{x}_{k+1|k}$ and covariance matrix P_{k+1} instead of integrating all thirteen sigma sample points, as the remaining twelve sigma sample points can be predicted by MANN. The estimation accuracy of MANN-UKF may be worse than that of UKF because of the inevitable prediction error when using MANN. This section introduces a neural network-assisted Earth–Jupiter transfer OD system, wherein MANN construction and training are shown in the next section.

3. Construction and Training of the Multivariate Attention Neural Network

3.1. Sample Construction

As shown in Figure 1, the goal of MANN is to predict e . The variables that are related to e are provided as follows:

- The initial state of the nominal orbit: $x_0 \in \mathbb{R}^6$.
- The initial deviation: $\delta x_0 \in \mathbb{R}^6$.
- The orbital state of the nominal orbit at the epoch t : $x \in \mathbb{R}^6$.
- The orbit deviation propagated using the two-body dynamics: $\delta x^{2BP} \in \mathbb{R}^6$.

Apart from the four variables, there are some other variables that are available and can be potentially helpful in predicting e :

- The deviation between the nominal orbits propagated using the high-fidelity and two-body dynamics: $\delta x_{ref}^{2BP} = x - x^{2BP} \in \mathbb{R}^6$. This variable is expected to carry the information on the errors between the high-fidelity and two-body dynamics.
- The state vectors of the Earth, Mars, and Jupiter in the heliocentric inertial coordinate at the initial epoch t_0 : $x_{E,0} \in \mathbb{R}^6$, $x_{M,0} \in \mathbb{R}^6$, and $x_{J,0} \in \mathbb{R}^6$. These variables contain information about the gravitational perturbation of the third body. Here, both the position of the perturbed object and the velocity of the perturbed bodies are considered, as the positions of the perturbed bodies in $[t_0, t]$ have an effect on the orbital propagation and adding the velocities of the perturbed bodies to the sample helps to provide information about the position of the perturbed bodies at future times.
- The propagated interval: $\Delta t = t - t_0 \in \mathbb{R}^1$. This is an important variable, as a longer propagated interval usually indicates more significant orbital deviations.
- The variable related to the SRP: $c \in \mathbb{R}^1$.

Therefore, the input structure of the sample is designed as follows:

$$s = [x_0; x; \delta x_0; \delta x^{2BP}; \delta x_{ref}^{2BP}; x_{E,0}; x_{M,0}; x_{J,0}; \Delta t; c] \in \mathbb{R}^{50} \tag{21}$$

and the structure of the sample is designed as

$$S = \{s \mapsto e : \mathbb{R}^{50} \mapsto \mathbb{R}^6\}. \tag{22}$$

Based on the above discussion of the input and output of the sample, a method is developed to quickly generate a large number of samples.

Step 1 : Generate ten Earth–Jupiter transfer orbits between 1 January 2024 and 31 December 2033.

Step 1.1: Divide the above periods (1 January 2024–31 December 2033.) with an interval of 1 year. For each time interval, optimize the launch window. The corresponding steps are as presented in Step 1.2–Step 1.3;

Step 1.2: Two-impulse transfer is employed. The launch time JD0 and transfer time ToF (Time-of-flight, unit: day) are taken as the optimization variables. At the launch epoch, the C3 is constrained to be no more than 100 km²/s², and at the arrival epoch the spacecraft is required to enter an orbit with an eccentricity of 0.98 and a height of 10,000 km at the perigee of the Jupiter. Let r_p and r_J be the magnitudes of the position vectors of spacecraft and Jupiter at the arrival epoch heliocentric inertial coordinate, respectively; then, the velocity increment Δv that is required for the spacecraft to enter the Jupiter orbit can be expressed as shown below.

$$\begin{aligned}
 \Delta v &= v_p^{in} - v_p \\
 v_p^{in} &= \sqrt{\|v_2 - v_J\|^2 + 2\mu_J/r_p} \\
 r_p &= r_J + 10^4 \text{ km} \\
 v_p &= \sqrt{\mu_J \left(\frac{2}{r_p} - \frac{1}{a_{Target}} \right)} \\
 a_{Target} &= \frac{r_p}{1 - e_{Target}}
 \end{aligned}
 \tag{23}$$

Taking the magnitude of the velocity increment as the objective, the optimization problem can be provided as follows:

$$\begin{aligned}
 \min_{\text{JD0, ToF}} \quad & J = \Delta v \\
 \text{s.t.} \quad & [v_1, v_2] = \text{Lambert}(r_{E,0}, r_{J,f}, \text{ToF}) \\
 & C_3 = \|v_1 - v_E\| \leq 100 \\
 & r_{E,0} = r_E(\text{JD0}), r_{J,f} = r_J(\text{JD0} + \text{ToF}).
 \end{aligned}
 \tag{24}$$

In this work, the Lambert transfer is solved using the Python toolkit `lamberthub` (<https://pypi.com.cn/project/lamberthub/>, accessed on 8 May 2024).

Step 1.3: The differential evolution (DE) algorithm is used to optimize the problem in Equation (24). The range of launch time JD0 is determined by Step 1.1 and the range of ToF is set as [10 days, 5 years]. The population is set to 50 and the number of maximal iterations is set to 100. In this study, the DE algorithm is realized through the Python toolkit `geatpy` (<https://pypi.com.cn/project/geatpy/>, accessed on 8 May 2024). The ten obtained Earth–Jupiter transfer orbits are shown in Figure 3.

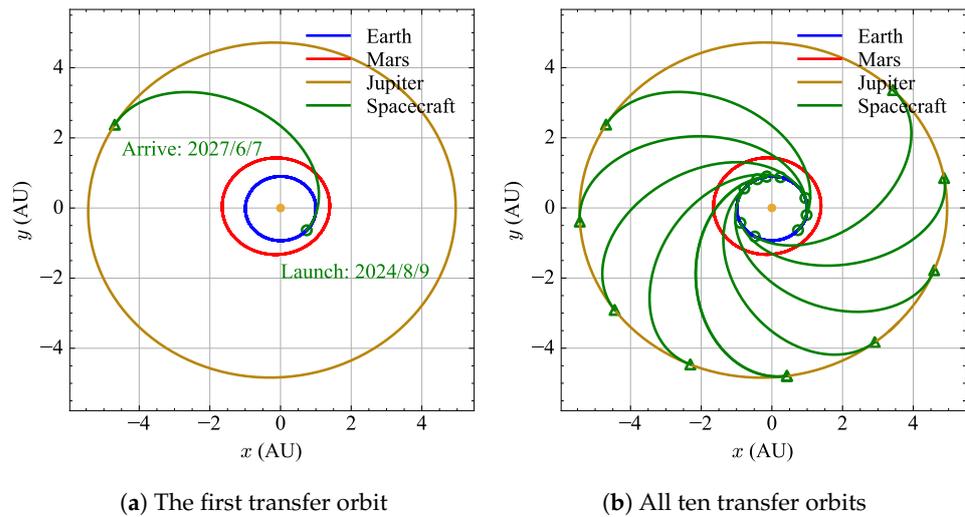


Figure 3. Optimized Earth–Jupiter transfer orbits.

Step 2: For the ten Earth–Jupiter transfer orbits obtained in Step 1, high-fidelity dynamics (i.e., Equation (1)) are used for orbit propagation. In order to avoid singularities at the Earth’s or Jupiter’s center, we remove the 60-day orbits at the beginning and end of the transfer orbits under two-body dynamics in Step 1. Taking the orbit state at the 60th day as the initial value and randomly generating c (variable of the SRP) between 0 and 0.1, orbit propagation is performed using high-fidelity dynamics.

Step 3: Each transfer orbit obtained in Step 2 under the high-precision dynamics is divided into ten orbits according to the time. For each split orbit, 100 initial deviations are randomly generated, where the position deviation is no more than 104 km per axis and the velocity deviation is no more than 0.1 km/s per axis. Additionally, the propagated time is randomly divided into ten epochs and a sample is constructed for each epoch according to Equation (22). Through the above methods, a total of 105 samples are obtained.

Note that one of the drawbacks of such neural network-based methods is that they cannot be employed in cases for which they are not trained. The performance of the neural network will degrade if it is trained under the dynamics in (1) and then employed under another dynamics. Therefore, if we want to consider additional accelerations, such as the J_2 term of the Earth and the propulsion term, the samples must be regenerated and the neural network should be retrained.

3.2. Structure of the Neural Network

In recent years, due to the rise of Chat-GPT, transformers (including improved versions such as iTransformer, Informer, Autoformer, Crossformer, etc.) have received significant attention in the fields of natural language processing (NLP) and computer vision (CV). In transformer-based architectures, the most crucial portion is the multi-head attention mechanism. The multi-head attention mechanism captures the dependencies between different variables by applying different weights. In recent years, some scholars have found that in NLP and CV contexts, network architectures based on a multi-head attention mechanisms exhibit better capacity than conventional neural networks such as RNN, DNN, and CNN. Inspired by the relevant research work, in this paper we construct a neural network model based on a multi-head attention mechanism, which is applied to solve the orbit deviation propagation problem in the Earth–Jupiter transfer process.

The structure of the Multivariate Attention-based Neural Network (MANN) constructed in this paper is shown in Figure 4.

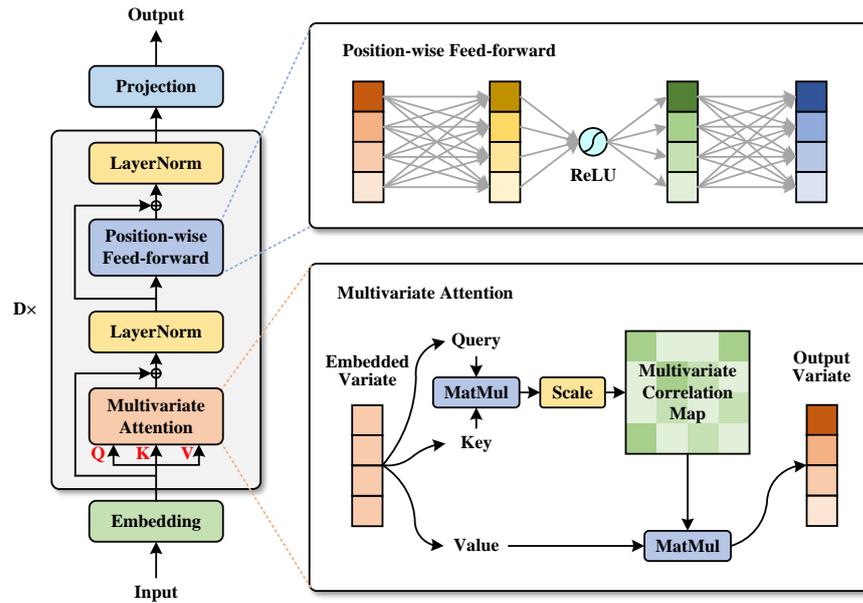


Figure 4. Structure of the designed MANN.

In Figure 4, the MANN developed in this paper is similar to the encoder module of the standard Transformer. As shown on the left side of Figure 4, the MANN consists of an embedding module, D (Depth) processing modules e (shown by gray in Figure 4) connected sequentially, and a projection module. Each processing module contains a multivariate attention layer, two normalization layers, and a feed-forward layer.

The main role of the embedding module is to expand the sample input $s^{50} \in \mathbb{R}^{50}$ into a dim -dimensional vector $s^{embedding} \in \mathbb{R}^{dim}$, with the expression provided as follows:

$$s^{embedding} = \text{Embedding}(s) \tag{25}$$

where dim is a user-defined parameter.

At the multi-head attention mechanism layer, for the i -th attention mechanism, the calculation process is as follows:

Step 1: Calculate the three vectors Q_i , K_i , and V_i using linear transformation:

$$Q_i = W_Q^i s^{embedding} \tag{26}$$

$$K_i = W_K^i s^{embedding} \tag{27}$$

$$V_i = W_V^i s^{embedding} \tag{28}$$

where $Q_i, K_i, V_i \in \mathbb{R}^{dim}$ and $W_Q^i, W_K^i, W_V^i \in \mathbb{R}^{dim \times dim}$.

Step 2: Calculate the scores by the sum dot product of Q_i and K_i :

$$S_i^0 = Q_i K_i^T. \tag{29}$$

Step 3: Scale the scores calculated in Step 2. In this paper, the scores are divided by 8, which allows the MANN to have a more stable gradient. Then, the result is normalized by softmax:

$$S_i^1 = \text{Softmax}(S_i^0 / 8) = \text{Softmax}(Q_i K_i^T / 8). \tag{30}$$

Step 4: Multiply the output from Step 3, S_i^1 , with the vector V_i to obtain the output of the i -th head:

$$z_i = S_i^1 V_i = \text{Softmax}(Q_i K_i^T / 8) V_i. \tag{31}$$

Step 5: Combine the outputs of all the heads and multiply them by the weight matrix W_O :

$$\mathbf{Z} = \text{concat}(\mathbf{z}_i)W_O. \quad (32)$$

Step 6: Map the output of Step 5 linearly to obtain the output with the dimension same as that of $s^{embedding}$:

$$\tilde{\mathbf{Z}} = W_Z\mathbf{Z}. \quad (33)$$

The above multi-head attention mechanism layer can be written as follows:

$$\tilde{\mathbf{Z}} = \text{MultiHead}(s^{embedding}). \quad (34)$$

The normalization layer is mainly used to improve training convergence, and can be expressed as

$$\mathbf{Z}_1 = \text{LayerNorm}_1(\tilde{\mathbf{Z}} + s^{embedding}). \quad (35)$$

The function of the feed-forward layer is to extract deeper features of the input data through linear transformation:

$$\mathbf{Z}_2 = \text{FeedForward}(\mathbf{Z}_1). \quad (36)$$

The second normalization layer can be written as

$$\mathbf{Z}_3 = \text{LayerNorm}_2(\mathbf{Z}_1 + \mathbf{Z}_2). \quad (37)$$

Finally, the projection module maps the data to required dimensions:

$$\hat{e}_{\text{MANN}} = \text{Projection}(\mathbf{Z}_3). \quad (38)$$

The above process can be summarized as follows:

$$\begin{aligned} s^{embedding} &= \text{Embedding}(s) \\ \tilde{\mathbf{Z}} &= \text{MultiHead}(s^{embedding}) \\ \mathbf{Z}_1 &= \text{LayerNorm}_1(\tilde{\mathbf{Z}} + s^{embedding}) \\ \mathbf{Z}_2 &= \text{FeedForward}(\mathbf{Z}_1) \\ \mathbf{Z}_3 &= \text{LayerNorm}_2(\mathbf{Z}_1 + \mathbf{Z}_2) \\ \hat{e}_{\text{MANN}} &= \text{Projection}(\mathbf{Z}_3) \end{aligned} \quad (39)$$

For the MANN architecture described above, the user-defined parameters include the depth (D) of the processing module (in gray), the output dimension (dim) of the embedding module, the number of multivariate attention heads in the module, and the dimensions of the heads in the embedding module.

3.3. Training of the Neural Network

For the MANN described in the previous subsection, the neural network model is constructed in the Pytorch environment in this work. During the training process, the generated 100,000 samples are divided into two groups: a training dataset including 80% of the samples and a validation dataset including the remaining 20% of the samples. The MANN is trained for 1000 iterations, and in each iteration, the training dataset is randomly shuffled into ten batches (each batch contains 8000 samples). The mean square error (MSE) is used as the loss function of neural network optimization, and Adaptive Moment Estimation (Adam) is used to optimize the neural network. We set each multivariate attention layer in the MANN to contain four heads, each with a dimension of 64. We first considered two sizes of MANN: one with depth 2 and dim 16, and a second with depth 8 and dim 256. The training results for the two sizes of MANN are shown in Figure 5a,b. As can be seen from Figure 5a,b, there is no apparent overfitting phenomenon when the

MANN is used. In addition, there is no obvious deviation between the MSE loss value of the training dataset and the validation dataset. Meanwhile, the minimum value of the MSE loss in the validation dataset of the first small-size MANN is about 10^{-7} , while that in the validation dataset of large-size MANN is about 10^{-8} , indicating that increasing the depth and dimension of the MANN can help to improve the prediction results of neural networks.

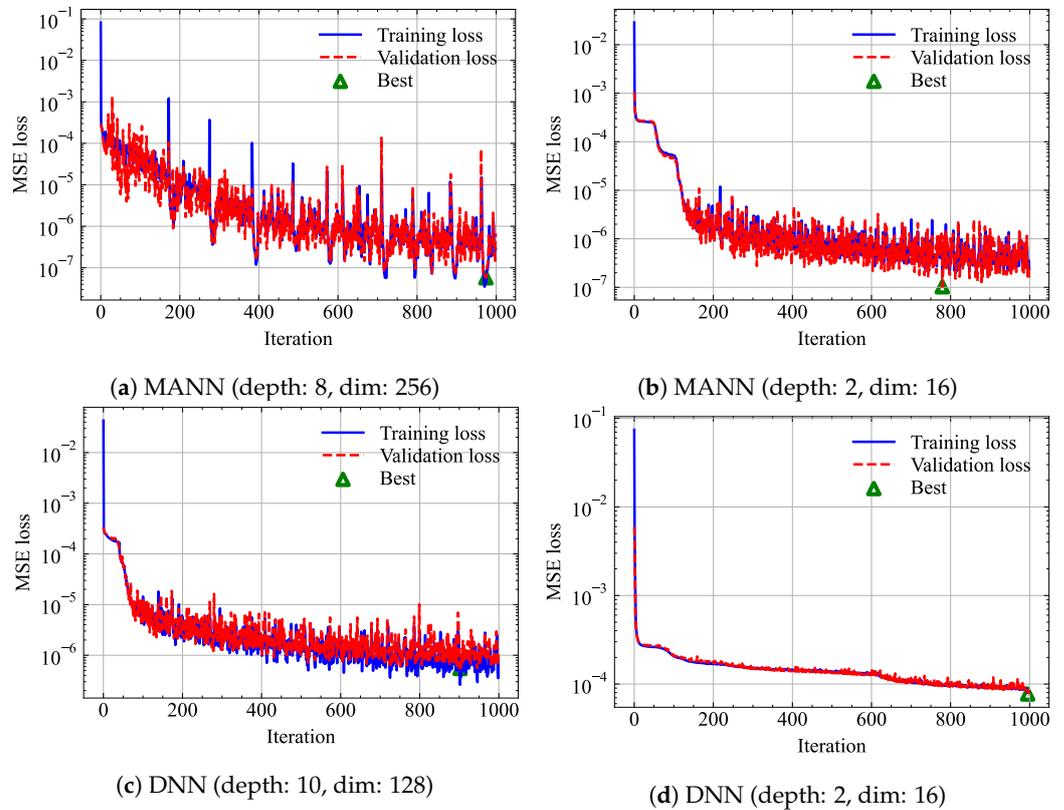


Figure 5. Time histories of the MSE losses of the MANN and DNN.

Figure 5c,d shows the training process of a traditional DNN to show the advantages of the MANN neural network. As shown in Figure 5c, the DNN shows obvious overfitting in the training process, and the MSE loss in the training dataset is significantly smaller than the MSE loss in the validation dataset. At the same time, the MSE loss values of the DNN are at least one order of magnitude larger than those of the MANN. This shows that the developed MANN has advantages in prediction accuracy and generalization compared with the traditional DNNs.

The training performances of MANNs and DNNs with different sizes are provided in Tables 1 and 2. It can be seen from Tables 1 and 2 that the MANN has significant advantages over the DNNs when the structure size is small. For example, with depth 2 and dim 16, the minimal MSE loss of the validation dataset of MANN is 1.0263×10^{-7} while that of DNN is 7.7148×10^{-5} . The minimal MSE loss of the validation dataset of the MANN is less than the order of 10^{-7} , while that of the DNN is larger than 10^{-6} in most cases. In the following simulation, the MANN with the smallest MSE loss (shown in bold in Table 1) is adopted as the neural network to predict the e .

Table 1. Training performance of MANNs with different sizes.

Depth	Dim	Minimal Training MSE Loss	Minimal Validation MSE Loss
2	16	1.7585×10^{-7}	1.0263×10^{-7}
	32	2.2242×10^{-7}	1.3180×10^{-7}
	64	1.8228×10^{-7}	1.8267×10^{-7}
	128	1.0864×10^{-7}	1.4407×10^{-7}
	256	7.7139×10^{-8}	1.2502×10^{-7}
4	16	1.1686×10^{-7}	1.0519×10^{-7}
	32	2.5985×10^{-7}	1.0549×10^{-7}
	64	1.2736×10^{-7}	1.0895×10^{-7}
	128	2.0587×10^{-7}	1.1150×10^{-7}
	256	6.4876×10^{-8}	8.1694×10^{-8}
6	16	1.4862×10^{-7}	1.2556×10^{-7}
	32	3.3665×10^{-7}	1.8520×10^{-7}
	64	1.3504×10^{-7}	1.1473×10^{-7}
	128	6.6006×10^{-8}	8.7543×10^{-8}
	256	4.2020×10^{-8}	6.2024×10^{-8}
8	16	1.6521×10^{-7}	1.2702×10^{-7}
	32	1.5892×10^{-7}	1.1527×10^{-7}
	64	1.0456×10^{-7}	1.0343×10^{-7}
	128	9.5767×10^{-8}	1.1350×10^{-7}
	256	3.4753×10^{-8}	5.7216×10^{-8}

Table 2. Training performance of DNNs with different sizes.

Depth	Dim	Minimal Training MSE Loss	Minimal Validation MSE Loss
2	16	7.9980×10^{-5}	7.7148×10^{-5}
	32	1.9595×10^{-5}	1.9039×10^{-5}
	64	3.4492×10^{-6}	3.1238×10^{-6}
	128	7.5148×10^{-7}	7.8275×10^{-7}
4	16	3.1147×10^{-5}	2.8625×10^{-5}
	32	4.4568×10^{-6}	4.7672×10^{-6}
	64	1.1522×10^{-6}	1.3511×10^{-6}
	128	2.9337×10^{-7}	5.6742×10^{-7}
6	16	1.0570×10^{-5}	1.0608×10^{-5}
	32	4.5231×10^{-6}	4.5524×10^{-6}
	64	7.6149×10^{-7}	1.0962×10^{-6}
	128	2.6376×10^{-7}	6.0340×10^{-7}
8	16	9.0265×10^{-6}	9.6495×10^{-6}
	32	1.1945×10^{-6}	1.5034×10^{-6}
	64	3.8197×10^{-7}	6.5156×10^{-7}
	128	1.9960×10^{-7}	5.6136×10^{-7}
10	16	1.6846×10^{-5}	1.6793×10^{-5}
	32	1.1377×10^{-6}	1.5126×10^{-6}
	64	3.9176×10^{-7}	7.9620×10^{-7}
	128	2.6055×10^{-7}	5.4780×10^{-7}

4. Simulation Results

We take the first orbit (i.e., the orbit departing on 9 August 2024 in Figure 3a) as an example for simulation. Note that as the initial deviations are different, the samples in this simulation are not in the training or validation datasets to show the performance of the MANN in a case for which it is not trained. After deleting 60 days at the beginning and

ending epochs, the total flight time is about 911.7 days. The parameters for the simulations are listed in Table 3. We assume that a total of ten measurements can be obtained during the transfer process, i.e., the measurement interval is about 91 days (no measurement is available at the initial epoch). Assume that the position vectors can be measured. In general, the position vectors can be measured by observing planets or beacons. For example, by observing two different planets and obtaining the corresponding line-of-sight vectors, we can compute its own-position vector. Therefore, the measurement model is written as follows:

$$z_{k+1} = h(x_{k+1}) + \varepsilon_{k+1} = r_{k+1} + \varepsilon_{k+1}. \tag{40}$$

Table 3. Parameters for the simulations.

Parameters		Values
Initial position vector (km)	x	1.61654772×10^8
	y	7.73129404×10^7
	z	2.00240488×10^7
Initial velocity vector (km/s)	x	-2.64982864×10^0
	y	3.22553899×10^1
	z	1.21074585×10^1
Initial epoch		8 October 2024 10:02:47
Ending epoch		8 April 2027 02:19:21

First, a single simulation was performed, with the initial position errors set to 10,000 km per axis and the initial velocity error set to 0.1 km/s per axis. The standard deviation of the measurements was set as 1000 km per axis. The initial covariance was set as

$$P_0 = \begin{bmatrix} 10^8 I_3 \text{ km}^2 & 0_{3 \times 3} \\ 0_{3 \times 3} & 10^{-4} I_3 \text{ km}^2/\text{s}^2 \end{bmatrix}. \tag{41}$$

The single simulation results are shown in Figures 6 and 7. In Figures 6 and 7, the blue curve is the OD error of the single simulation and the red curve is the standard deviation forecast by the MANN-UKF and UKF. As can be seen from Figures 6 and 7, the OD errors of both methods are within the $\pm 3\sigma$ boundaries. In Figures 6 and 7, the symbol σ denotes the STD of estimated errors, which is calculated analytically using Equation (20). The values of the estimated STDs converge in the first 100 days. The position and velocity STDs at the initial epoch are 10,000 km and 0.01 km/s, respectively. The STDs of the position estimated errors converge to approximately 300 km at the ending epoch.

Figure 8 shows the Mahalanobis distance (MD) results of the two methods in the single simulation, with the MD defined as follows:

$$MD = \frac{\sqrt{(\hat{x} - x)P^{-1}(\hat{x} - x)^T}}{\sqrt{6}} \tag{42}$$

where x and \hat{x} represent the true orbit and the estimated orbit, respectively, and P is the estimated covariance matrix. The Mahalanobis distance is a statistical measure used to compare a set of observations or variables. It takes into account the covariance between variables, unlike other distance measures such as Euclidean distance. Essentially, it measures the distance between a point and a distribution or a set of points, taking into account the correlation between the variables. In Equation (42), a value of MD closer to 1 indicates better uncertainty estimation performance [46]. As can be seen from Figure 8, at the ending epoch, the estimation performance of MANN-UKF is slightly worse than that of UKF. This is because the MANN method has lower prediction accuracy when the deviation is slight [46].

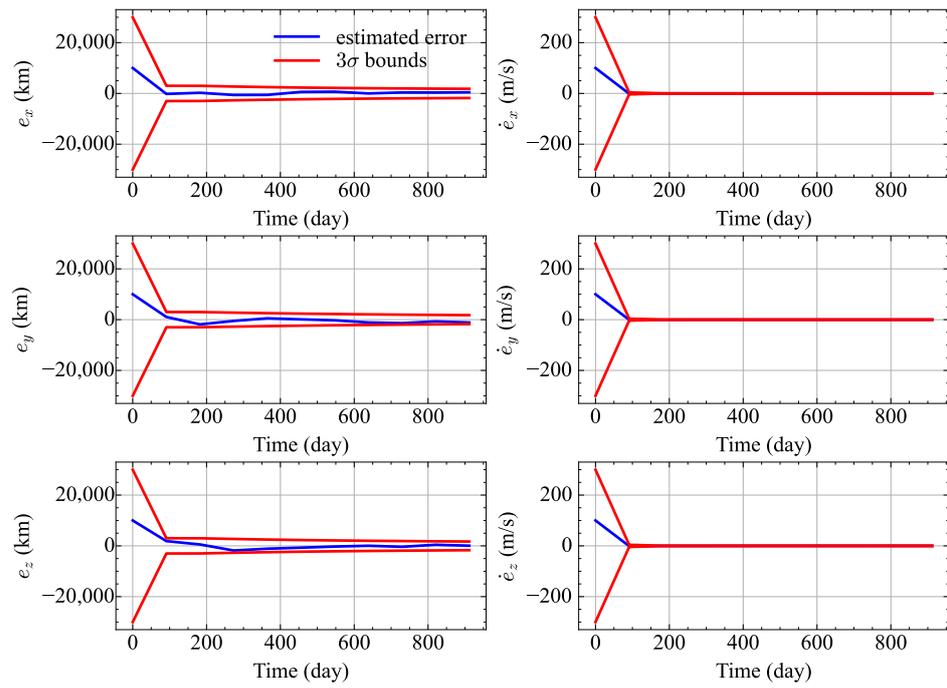


Figure 6. Orbit determination errors of the single simulation of MANN-UKF.

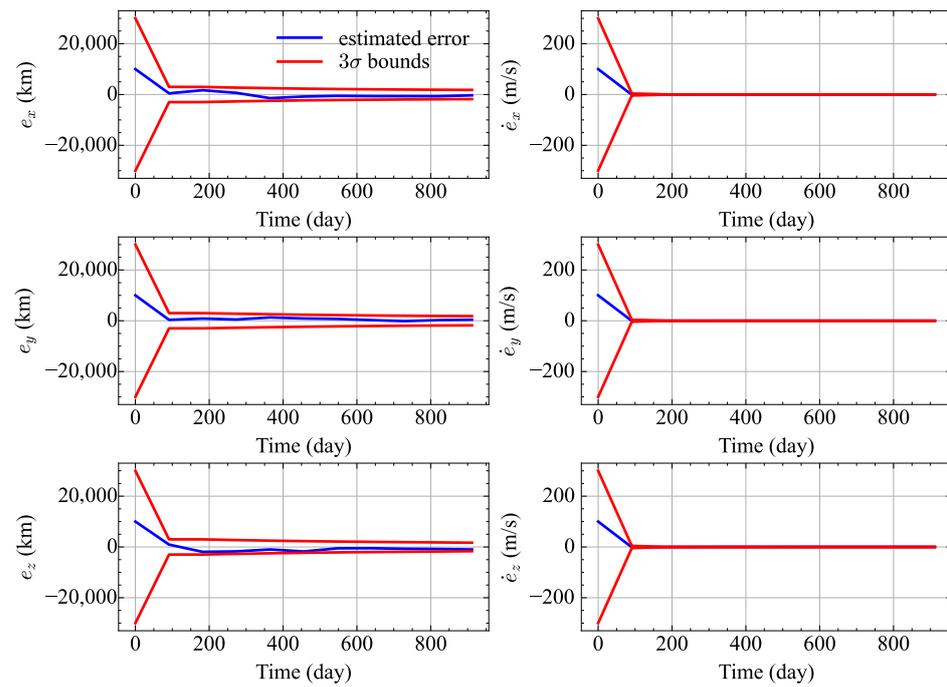


Figure 7. Orbit determination errors of the single simulation of UKF.

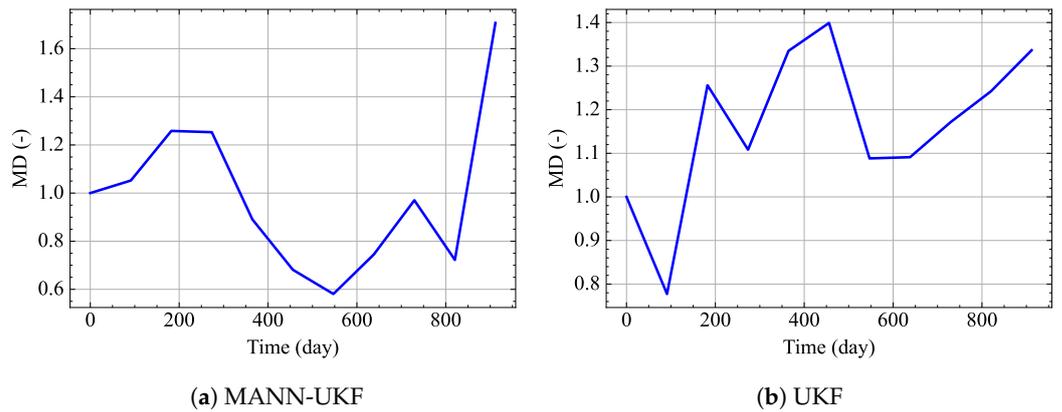


Figure 8. MD results of the single simulation.

A Monte Carlo (MC) simulation was further carried out, with the results shown below. For each method, 100 MC simulations were performed; for each MC simulation, the initial guess was obtained by the following equation:

$$\hat{x} = \mathcal{N}(x; x_0, P_0). \tag{43}$$

The MC simulation results are shown in Figures 9 and 10. In Figures 9 and 10, the 3σ boundaries were statistically obtained based on the results of 100 MC simulations. The curve of one standard deviation for both methods is provided in Figure 11. As shown in Figure 11, the convergence tendency of estimation accuracy of MANN-UKF and UKF is very close. For example, during the epochs from 200 days to 600 days, the MANN-UKF is slightly more accurate than the UKF in terms of the estimated errors along the x-axis. Slight differences exist as initial estimations are randomly generated in the MC simulation. However, around the ending epoch the performance of MANN-UKF is slightly worse than that of UKF. This is because around the ending epoch the estimation error is smaller than that in the early stage, and the prediction accuracy of MANN decreases, resulting in the decline of the prediction accuracy. As a result, the performance of MANN-UKF decreases.

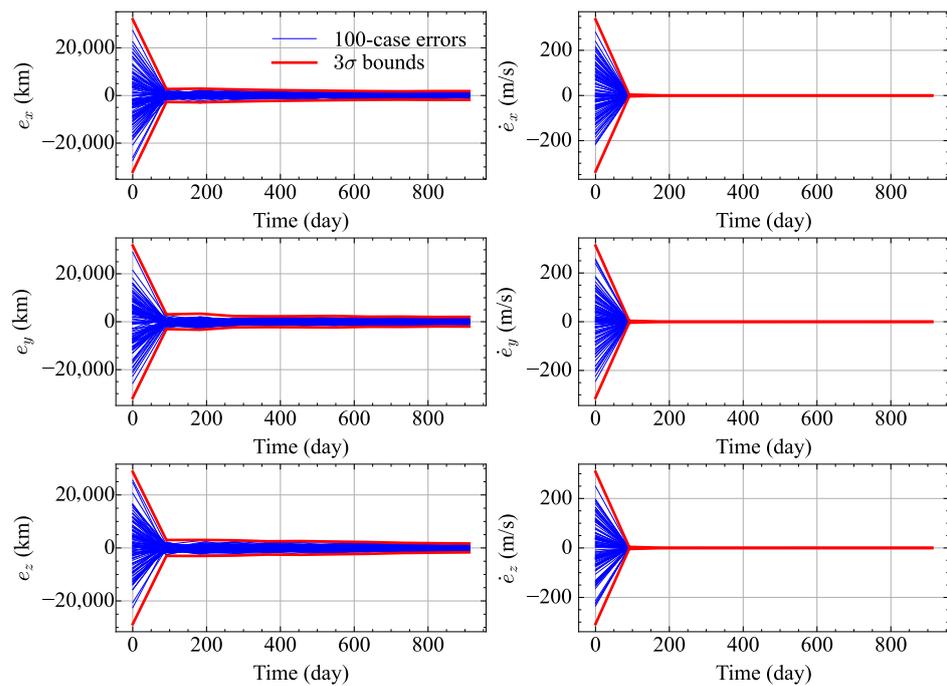


Figure 9. MC results of MANN-UKF.

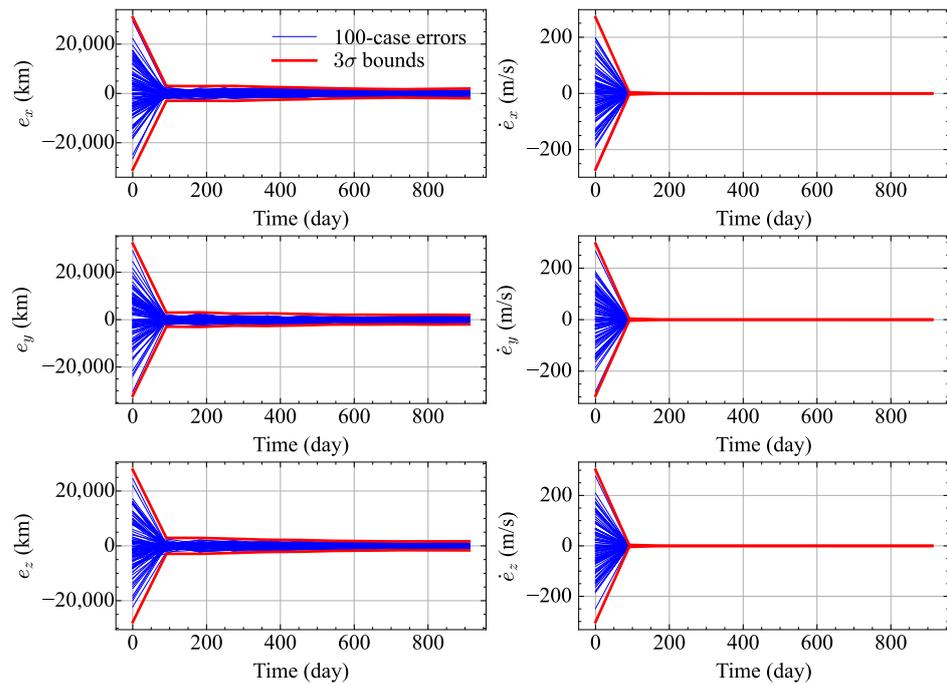


Figure 10. MC results of UKF.

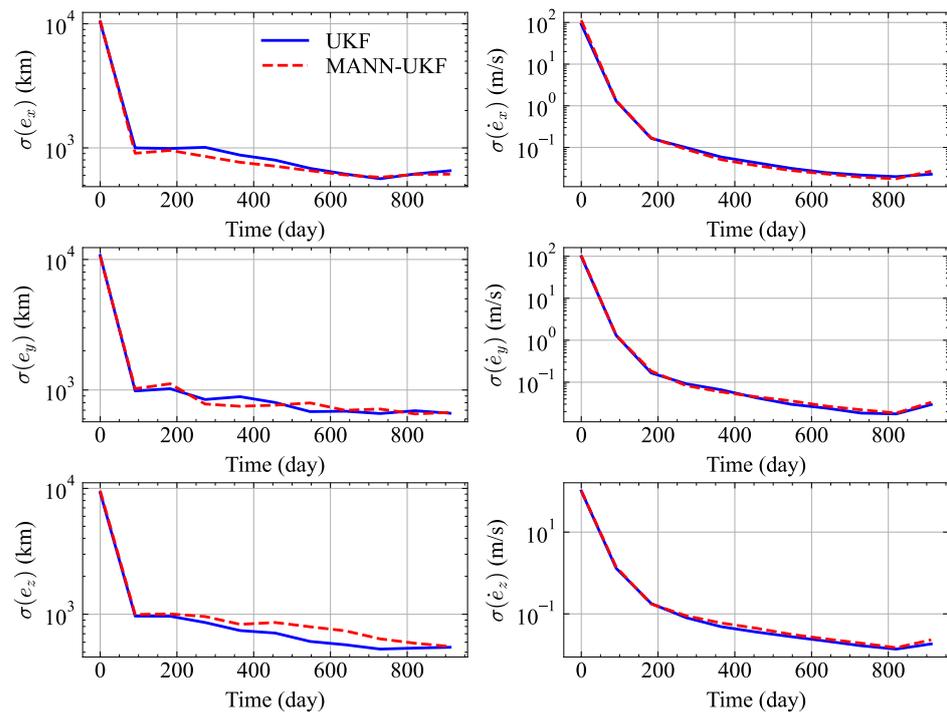


Figure 11. MC results comparing MANN-UKF and UKF.

Figures 12 and 13 are Box and Whisker Plots. The line splitting the box in two represents the median value. The upper edge and lower edge of the box represent the upper quartile and lower quartile respectively. The values at which the vertical lines stop are the values of the upper and lower values of the data. The single points on the diagram show the outliers.

As shown in Figure 12, the MDs of UKF are around 1, while the MC of MANN-UKF are between 1 and 5. This is again due to the decline in the prediction accuracy of the

MANN when the deviation is small, resulting in inaccurate prediction of the covariance matrix.

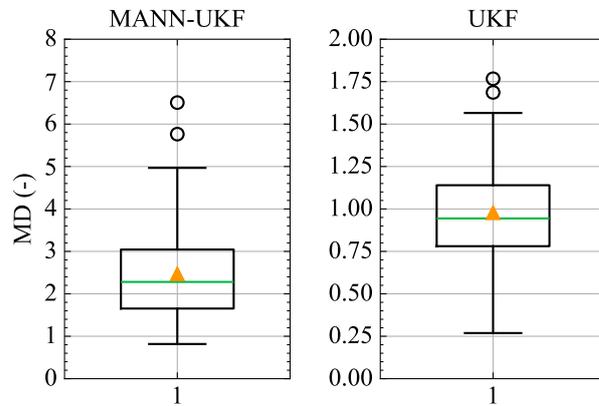


Figure 12. Comparison of MD between MANN-UKF and UKF in the MC simulations.

The primary purpose of the proposed MANN-UKF method is to reduce the computational cost in OD. Figure 13 and Table 4 compare the CPU times of the two methods. The CPU time of the UKF method is about 6.6 s, while the proposed MANN-UKF method only needs about 0.54 s. The proposed method saves on CPU time by more than 90%.

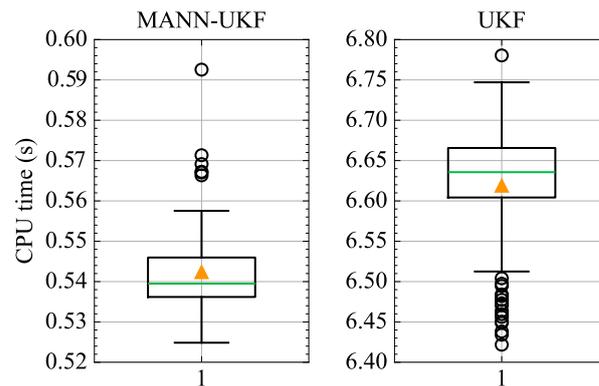


Figure 13. Comparison of CPU times between MANN-UKF and UKF.

Table 4. CPU times of MANN-UKF and UKF.

	Mean (s)	Maximum (s)	Minimum (s)
MANN-UKF	0.5423	0.5925	0.5248
UKF	6.6188	6.7804	6.4218

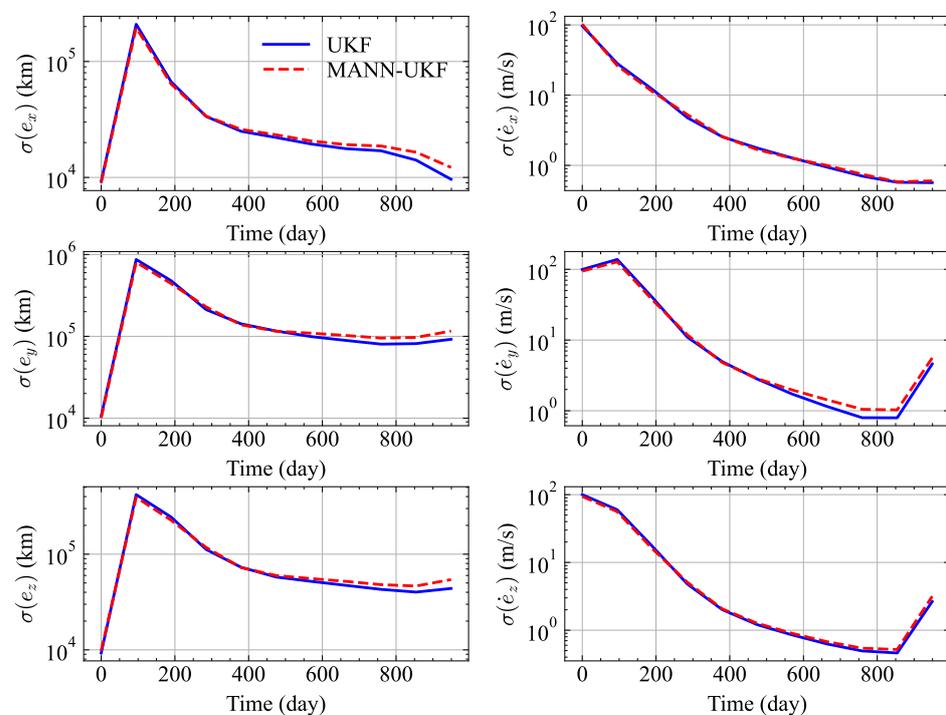
Apart from the above case, a second case is presented to investigate the performance of the proposed method. The parameters of the second case are shown in Table 5. In this case, line-of-sight (LOS) measurements are employed instead of the measurement model in Equation (40). We assume that the spacecraft can obtain the LOS measurements (from the spacecraft to the Jupiter) using optical sensors. The LOS measurement model is represented as follows [42,49,50]:

$$z_{k+1} = h(x_{k+1}) + \varepsilon_{k+1} = \frac{r_J - r_{k+1}}{\|r_J - r_{k+1}\|} + \varepsilon_{k+1}. \tag{44}$$

Table 5. Parameters for the simulations (second case).

Parameters		Values
Initial position vector (km)	x	9.45604828×10^7
	y	1.43195513×10^8
	z	4.76187910×10^7
Initial velocity vector (km/s)	x	-2.05939633×10^1
	y	2.64682048×10^1
	z	9.49888317×10^0
Initial epoch		9 November 2025 03:39:46
Ending epoch		15 June 2028 06:27:40

In this simulation, the STDs of the LOS measurements are set as 10^{-4} per axis (similar to [42]). One hundred MC runs were performed using two algorithms, with the estimated STDs shown in Figure 14. It can be seen from Figure 14 that the proposed MANN-UKF can output OD solutions with accuracy and convergence similar to those of the UKF.

**Figure 14.** MC results comparisons between MANN-UKF and UKF (second case).

5. Conclusions

This paper proposes a multivariate attention neural network (MANN)-based orbit uncertainty propagation (OUP) and orbit determination (OD) method for Earth–Jupiter transfer. The proposed method relies on a carefully designed MANN for orbit deviation propagation, which is further employed to predict uncertainties for Earth–Jupiter transfer OD. Training results show that the MANN outperforms conventional deep neural networks (DNNs) regarding accuracy and generalization. The minimal loss of the MANN is lower than 10^{-7} , which is at least two orders of magnitude smaller than that of a conventional DNN. Using the proposed MANN-based method, the orbit estimation errors quickly converge. Orbit estimation results reveal that the MANN-based method has accuracy close to that of an unscented Kalman filter (UKF) while saving more than 90% of the computational time of UKF. Moreover, the Mahalanobis distance results show that the proposed method has worse uncertainty estimation performance than the UKF method.

In the future, more complex dynamics should be considered to test the robustness of the proposed method. In addition, combining the MANN with other estimation methods, such as a cubature Kalman filter or Gaussian sum filter, is suggested.

Author Contributions: Methodology, Z.Z.; Writing—review & editing, Y.S.; Supervision, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 62394353).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bechini, M.; Gu, G.; Lunghi, P.; Lavagna, M. Robust spacecraft relative pose estimation via CNN-aided line segments detection in monocular images. *Acta Astronaut.* **2024**, *215*, 20–43. [[CrossRef](#)]
2. Zhou, X.; Wang, S.; Qin, T. Multi-Spacecraft Tracking and Data Association Based on Uncertainty Propagation. *Appl. Sci.* **2022**, *12*, 7660–7671. [[CrossRef](#)]
3. Christian, J.A. Optical Navigation Using Iterative Horizon Reprojection. *J. Guid. Control Dyn.* **2016**, *39*, 1092–1103. [[CrossRef](#)]
4. Zhao, Y.; Yang, H.; Li, S.; Zhou, Y. On-board modeling of gravity fields of elongated asteroids using Hopfield neural networks. *Astrodynamics* **2023**, *7*, 101–114. [[CrossRef](#)]
5. Qiao, D.; Zhou, X.; Zhao, Z.; Qin, T. Asteroid Approaching Orbit Optimization Considering Optical Navigation Observability. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 5165–5179. [[CrossRef](#)]
6. Ogawa, N.; Terui, F.; Mimasu, Y.; Yoshikawa, K.; Ono, G.; Yasuda, S.; Matsushima, K.; Masuda, T.; Hihara, H.; Sano, J.; et al. Image-based autonomous navigation of Hayabusa2 using artificial landmarks: The design and brief in-flight results of the first landing on asteroid Ryugu. *Astrodynamics* **2020**, *4*, 89–103. [[CrossRef](#)]
7. Qi, Y.; De Ruiter, A. Study of correction maneuver for lunar flyby transfers in the real ephemeris. *J. Guid. Control Dyn.* **2018**, *41*, 2112–2132. [[CrossRef](#)]
8. Zhou, X.; Li, X.; Huo, Z.; Meng, L.; Huang, J. Near-Earth Asteroid Surveillance Constellation in the Sun-Venus Three-Body System. *Space Sci. Technol.* **2022**, *2022*, 9864937. [[CrossRef](#)]
9. Christian, J.A. Pole Estimation and Optical Navigation Using Circle of Latitude Projections. *J. Guid. Control Dyn.* **2023**, *47*, 407–416. [[CrossRef](#)]
10. Ma, X.; Fang, J.; Ning, X. An overview of the autonomous navigation for a gravity-assist interplanetary spacecraft. *Prog. Aerosp. Sci.* **2013**, *63*, 56–66. [[CrossRef](#)]
11. Andreis, E.; Franzese, V.; Topputo, F. Onboard Orbit Determination for Deep-Space CubeSats. *J. Guid. Control Dyn.* **2022**, *45*, 1466–1480. [[CrossRef](#)]
12. Zhou, X.; Qin, T.; Meng, L. Maneuvering Spacecraft Orbit Determination Using Polynomial Representation. *Aerospace* **2022**, *9*, 257. [[CrossRef](#)]
13. Beţco, D.; Pârnu, P.V.; Ciudin, S. Autonomous navigation for the Martian Inspection Drone. *Acta Astronaut.* **2024**, *216*, 55–63. [[CrossRef](#)]
14. Fossà, A.; Losacco, M.; Armellin, R. Perturbed initial orbit determination. *Astrodynamics* **2024**, 1–6. [[CrossRef](#)]
15. Yin, W.; Shi, Y.; Shu, L.; Gao, Y. Autonomous navigation of an asteroid orbiter enhanced by a beacon satellite in a high-altitude orbit. *Astrodynamics* **2024**. [[CrossRef](#)]
16. Capuano, V.; Harvard, A.; Chung, S.J. On-board cooperative spacecraft relative navigation fusing GNSS with vision. *Prog. Aerosp. Sci.* **2022**, *128*, 100761. [[CrossRef](#)]
17. Peng, H.; Bai, X. Comparative evaluation of three machine learning algorithms on improving orbit prediction accuracy. *Astrodynamics* **2019**, *3*, 325–343. [[CrossRef](#)]
18. Sun, Q.; Gutiérrez, J.L.R.; Yu, X. Deep Neural Network-Based 4-Quadrant Analog Sun Sensor Calibration. *Space Sci. Technol.* **2023**, *3*, 24. [[CrossRef](#)]
19. Izzo, D.; Märtens, M.; Pan, B. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics* **2019**, *3*, 287–299. [[CrossRef](#)]
20. Yu, Z.; Cui, P.; Crassidis, J.L. Design and optimization of navigation and guidance techniques for Mars pinpoint landing: Review and prospect. *Prog. Aerosp. Sci.* **2017**, *94*, 82–94. [[CrossRef](#)]

21. Ge, D.; Cui, P.; Zhu, S. Recent development of autonomous GNC technologies for small celestial body descent and landing. *Prog. Aerosp. Sci.* **2019**, *110*, 100551. [[CrossRef](#)]
22. Li, R.; Sun, C.; Yu, X.; Zhang, L.; Wei, J.; Fang, Q. Space Noncooperative Target Trajectory Tracking Based on Maneuvering Parameter Estimation. *Space Sci. Technol.* **2024**, *3*, 78. [[CrossRef](#)]
23. Luo, Y.; Yang, Z. A review of uncertainty propagation in orbital mechanics. *Prog. Aerosp. Sci.* **2017**, *89*, 23–39. [[CrossRef](#)]
24. Garcia, R.V.; Pardal, P.; Kuga, H.K.; Zanardi, M.C. Nonlinear filtering for sequential spacecraft attitude estimation with real data: Cubature Kalman Filter, Unscented Kalman Filter and Extended Kalman Filter. *Adv. Space Res.* **2019**, *63*, 1038–1050. [[CrossRef](#)]
25. Yan, H.; Liu, H.; Sun, X.; Xu, M. Accuracy Evaluation of Marginalized Unscented Kalman Filter. *Space Sci. Technol.* **2024**, *3*, 85. [[CrossRef](#)]
26. Zhou, X.; Jia, F.; Li, X. Configuration Stability Analysis for Geocentric Space Gravitational-Wave Observatories. *Aerospace* **2022**, *9*, 519. [[CrossRef](#)]
27. Santos-León, J.C.; Orive, R.; Acosta, D.; Acosta, L. The Cubature Kalman Filter revisited. *Automatica* **2021**, *127*, 109541. [[CrossRef](#)]
28. Qiao, D.; Zhou, X.; Li, X. Analytical configuration uncertainty propagation of geocentric interferometric detection constellation. *Astrodynamics* **2023**, *7*, 271–284. [[CrossRef](#)]
29. Zhou, X.; Cheng, Y.; Qiao, D.; Huo, Z. An adaptive surrogate model-based fast planning for swarm safe migration along halo orbit. *Acta Astronaut.* **2022**, *194*, 309–322. [[CrossRef](#)]
30. Roa, J.; Park, R.S. Reduced Nonlinear Model for Orbit Uncertainty Propagation and Estimation. *J. Guid. Control Dyn.* **2021**, *44*, 1578–1592. [[CrossRef](#)]
31. Qiao, D.; Zhou, X.; Li, X. Feasible domain analysis of heliocentric gravitational-wave detection configuration using semi-analytical uncertainty propagation. *Adv. Space Res.* **2023**, *72*, 4115–4131. [[CrossRef](#)]
32. Park, R.S.; Scheeres, D.J. Nonlinear mapping of Gaussian statistics: Theory and applications to spacecraft trajectory design. *J. Guid. Control Dyn.* **2006**, *29*, 1367–1375. [[CrossRef](#)]
33. Park, R.S.; Scheeres, D.J. Nonlinear semi-analytic methods for trajectory estimation. *J. Guid. Control Dyn.* **2007**, *30*, 1668–1676. [[CrossRef](#)]
34. Sun, P.; Colombo, C.; Trisolini, M.; Li, S. Hybrid Gaussian Mixture Splitting Techniques for Uncertainty Propagation in Nonlinear Dynamics. *J. Guid. Control Dyn.* **2022**, *46*, 770–780. [[CrossRef](#)]
35. Kim, Y.; Park, S.Y. Perturbative analysis on orbital kinematics of flybys and applications to Doppler observation. *J. Guid. Control Dyn.* **2015**, *38*, 1690–1698. [[CrossRef](#)]
36. Boone, S.; McMahon, J. Directional State Transition Tensors for Capturing Dominant Nonlinear Effects in Orbital Dynamics. *J. Guid. Control Dyn.* **2022**, *46*, 431–442. [[CrossRef](#)]
37. Hu, R.; Huang, X.; Xu, C. Integrated visual navigation based on angles-only measurements for asteroid final landing phase. *Astrodynamics* **2023**, *7*, 69–82. [[CrossRef](#)]
38. Mortari, D.; Conway, D. Single-point position estimation in interplanetary trajectories using star trackers. *Celest. Mech. Dyn. Astron.* **2017**, *128*, 115–130. [[CrossRef](#)]
39. Christian, J.A. Optical Navigation Using Planet’s Centroid and Apparent Diameter in Image. *J. Guid. Control Dyn.* **2014**, *38*, 192–204. [[CrossRef](#)]
40. Ely, T.; Bhaskaran, S.; Bradley, N.; Lazio, T.J.W.; Martin-Mur, T. Comparison of Deep Space Navigation Using Optical Imaging, Pulsar Time-of-Arrival Tracking, and/or Radiometric Tracking. *J. Astronaut. Sci.* **2022**, *69*, 385–472. [[CrossRef](#)]
41. Wang, Y.; Bai, X.; Peng, H.; Chen, G.; Shen, D.; Blasch, E.; Sheaff, C.B. Gaussian-Binary classification for resident space object maneuver detection. *Acta Astronaut.* **2021**, *187*, 438–446. [[CrossRef](#)]
42. Zhou, X.; Qin, T.; Macdonald, M.; Qiao, D. Observability analysis of cooperative orbit determination using inertial inter-spacecraft angle measurements. *Acta Astronaut.* **2023**, *210*, 289–302. [[CrossRef](#)]
43. Prabhu, K.; Majji, M.; Alfriend, K.T. Least Sum of Absolute Residuals Orbit Determination. *J. Guid. Control Dyn.* **2021**, *45*, 468–480. [[CrossRef](#)]
44. Schiemenz, F.; Utmann, J.; Kayal, H. Adaptive Gaussian Mixture based orbit determination with combined atmospheric density uncertainty consideration. *Adv. Space Res.* **2020**, *66*, 1609–1634. [[CrossRef](#)]
45. Zhou, X.; Qiao, D.; Li, X. Adaptive Order-Switching Kalman Filter for Orbit Determination Using Deep-Neural-Network-Based Nonlinearity Detection. *J. Spacecr. Rocket.* **2023**, *60*, 1724–1741. [[CrossRef](#)]
46. Zhou, X.; Qiao, D.; Li, X. Neural Network-Based Method for Orbit Uncertainty Propagation and Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 1176–1193. [[CrossRef](#)]
47. Greco, C.; Vasile, M. Robust Bayesian Particle Filter for Space Object Tracking Under Severe Uncertainty. *J. Guid. Control Dyn.* **2021**, *45*, 481–498. [[CrossRef](#)]
48. Khatri, Y.; Scheeres, D.J. Nonlinear semi-analytical uncertainty propagation for conjunction analysis. *Acta Astronaut.* **2023**, *203*, 568–576. [[CrossRef](#)]

49. Casini, S.; Cervone, A.; Monna, B.; Gill, E. On line-of-sight navigation for deep-space applications: A performance analysis. *Adv. Space Res.* **2023**, *72*, 2994–3008. [[CrossRef](#)]
50. Hu, Y.; Sharf, I.; Chen, L. Three-spacecraft autonomous orbit determination and observability analysis with inertial angles-only measurements. *Acta Astronaut.* **2020**, *170*, 106–121. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.