7th IAA Planetary Defense Conference – PDC 2021
26-30 April 2021, Vienna, Austria

IAA-PDC-21-07-29

HOVERING CONTROL FOR GRAVITY TRACTOR USING ASYNCHRONOUS METHODS FOR REINFORCEMENT LEARNING

Jucheng Lu, Bingwei Wei, Haibin Shang, Pingyuan Cui, Rui Xu, Shengying Zhu, Ai Gao
Beijing Institute of Technology, 100081 Beijing, People's Republic of China, 010-68913550, shanghb@bit.edu.cn

Keywords: *Gravity Tractor, Hovering Control, Binary Asteroid Systems, Reinforcement Learning*

ABSTRACT

Potentially Hazardous Object (PHO) refers to the near-Earth object which has a minimum orbital intersection distance with Earth of less than 0.05 AU. These objects include binary asteroid systems e.g. 1999kw4. Researchers has studied a series of deflection schemes, such as gravity tractor, kinetic impactors, laser beaming and low-thrust deflection via electric propulsion or solar sails. Gravity tractor is a long-term hovering project, which uses the mutual gravitational force between a hovering spacecraft and a target object as a towline. Comparing with hovering in a unitary asteroid, both the superposition of the gravity field and the evolution of the binary asteroid increase the uncertainty of this dynamic system. The traditional control theories may be failed when they face this highly uncertain system. This paper proposes a novel hovering control method based on reinforcement learning (RL) with asynchronous methods for achieving the aim of adapting the uncertain environment.

In this paper, the gravity field of the binary asteroid system is modeled as double ellipsoids model which the system's exterior potential can be superposed from both ellipsoids. The triaxial ellipsoid's gravitational potential energy is calculated both by an elliptical integral and by a second degree second order spherical harmonic series, which shows the discrepancy of the environments. In order to retain the general feature of the gravitational field, the ellipsoid is chosen as gravity-best-fit ellipsoid whose gravity potential is consistent with the irregular asteroid's gravity potential in a distance. The

spacecraft plays the role of the agent in RL. The control is determined by the policy which is composed of actor and critic, where artificial neural network (ANN) is employed as the parameter description. An asynchronous method is employed to train the parameter of the ANN in this paper. The model is trained during the interaction between the agent and the environment while RL algorithm makes the agent adapt different environment and evolve with the variation of the environment. To demonstrate that the controller can adapt the change of the dynamics and learn online, the training environment differs from the test environment in numerical experiments. Simulation shows that the spacecraft can achieve and maintain the hovering state in spite of the poor precision of the training environment. The position error can be reduced to 1m in a changing uncertain environment. Further more, the control can be improved using the data which is produced during this long-term mission.

Introduction

Potentially Hazardous Object (PHO) on a collision course could lead to a widespread damage. The Chelyabinsk meteor proved that the risk of impact upon the Earth by PHO is possible. The NASA Planetary Defense Coordination Office (PDCO) was established in 2016 to address and plan response to the asteroid impact hazard. To alter its trajectory, human has investigated a variety of schemes, including Gravity Tractor(GT)[1], laser beaming and kinetic-energy impactor. Compared with the others, GT is a reliable method because this deflection method is insensitive to the structure, surface properties, and rotation state of the asteroid.

GT is proposed as a low-energy long-term asteroid deflection concept using the mutual gravitational force between a hovering spacecraft and a target asteroid as a towline to alter its trajectory. The gravitational coupling/towing concept has been studied previously, this paper primarily focuses on the dynamics and hovering control of GT spacecraft. As a long-term mission, the control concept should be fuel-efficient and maintain a distinguish accuracy. Wie[2] demonstrates the practical hovering control feasibility of an SSGT spacecraft for towing NEAs. Wie[3] proposed a system of multiple gravity tractors in halo orbits, which produced larger velocity change and provided multispacecraft redundancy. Furfaro[4] employed two-sliding control to hover in

a body-fixed Cartesian coordinate frame of a uniformly rotating asteroid. A homogeneous controller is modified to trade off precision and propellant consumption. The time-varying environment in binary asteroid systems leads to many technically challenging astrodynamic control problem. Additional perturb produced by the rotation state of the binary asteroid and the uncertain gravitational field could invalidate the stability of the traditional control concept. Therefore, it is interesting to develop a novel control to solve these problems.

Reinforcement learning(RL) has achieved great progress in control area. It also has been employ to solve the astrodynamic control problem already. Guzzetti[5] used RL theory to design an orbit station keeping control algorithm within a chaotic environments. They noted that the RL can not only achieve the effect that are as accurate as current algorithm, but also adapt to uncertainties. The RL does not require an analytical description of the system dynamics, where some algorithm need to model the dynamics as function. They suggested that the RL could continue learning online to adapt to uncertainties. Gaudet[6] proposed to use RL to hover near a small body and demonstrate the robustness of the controller. Using Monte Carlo simulation, they demonstrate that the controller is fairly fuel-efficient and robust even in the situation where the external forces acting on the spacecraft are a significant fraction of the spacecraft's maximum thrust capabilities.

In this work we primarily focus on the GT hovering control problem, we propose to use an asynchronous RL to achieve and maintain the hovering state in a binary asteroid systems. The RL controller described in this paper will quickly drive the spacecraft to achieve and maintain the hovering state without establishing an analytical description of the system dynamics to function. We investigate the influence of the dynamics changing and demonstrate that RL controller could reduce the degeneration of the controller which is produced by the dynamics changing.

The rest of the paper is organized as follows. In Sec.II, the gravitational field of the binary asteroid system is formulated. In Sec.III, the principles behind the *Asynchronous Methods for Deep Reinforcement Learning* are introduced. In Sec.IV, the behavior of the proposed controller is illustrated. We investigated that the algorithm could continue learning online to adapt to the change of the environment. Some implementation details are described in Sec.V.

Conclusions are drawn in Sec. VI.

Hovering Problem Formulation

Full Two-Body Problem(F2BP) refers to considering the shape, mass distribution and orientation of the asteroids when we investigate the dynamics of the binary asteroid system. The double ellipsoid model is a common model to describe the F2BP. The study of the dynamics of a massless body in the F2BP is referred as the Restricted Full Three Body Problem (RF3BP).

The equations of motion of a particle in the inertial coordinate are established first. Because this paper focuses on the hovering control problem, we assume that the barycenter of the binary asteroid system is fixed in the inertial coordinate while it has orbit motion in reality. The exterior potential of the system can be superposed from both ellipsoids, and we can obtain the equations of motion of the gravity tractor

$$\begin{cases} \ddot{x} = \dfrac{\partial U_1}{\partial x} + \dfrac{\partial U_2}{\partial x} + p_x + u_x \\ \ddot{y} = \dfrac{\partial U_1}{\partial y} + \dfrac{\partial U_2}{\partial y} + p_y + u_y \\ \ddot{z} = \dfrac{\partial U_1}{\partial z} + \dfrac{\partial U_2}{\partial z} + p_z + u_z \end{cases} \tag{1}$$

where $U_i, i = 1,2$ is the potential of the asteroid, $\boldsymbol{u} = \left[u_x, u_y, u_z\right]^T$ is the control acceleration vector, $\boldsymbol{p} = \left[p_x \quad p_y \quad p_z\right]^T$ is the perturb acceleration vector. The perturb can be raised by solar radian pressure.

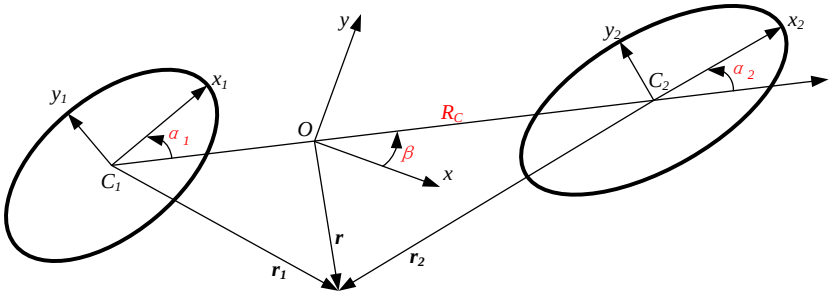

Fig.1 A conceptual illustration of a binary asteroid system gravity tractor

The binary asteroid is illustrated in Fig.1. The motion of binary asteroids can be modeled as the planar problem. Assume the barycenter of the system can be considered to be coincident with its center of orbit O.Woo[7] has suggested that four generalized coordinates $R_c, \beta, \alpha_1, \alpha_2$ fully define the rigid-body planar

motion of the two bodies around O. The origin of the body-fixed frame $C_i x_i y_i z_i$, for $i=1,2$ is located at the center of mass of the respective body. The x_i-axes of the body-fixed frame are assumed to be aligned along the minimum principal axes of inertia of the bodies. The z_i-axes are assumed to be aligned along the maximum principal axes, which is normal to the orbital plane as well. The distance between C_1 and C_2 is denoted by R_c. Vector $\boldsymbol{R}_c$, joining the two centers of mass, is oriented at an angle β with respect to the inertial frame. The relative attitude of the bodies with respect to $\boldsymbol{R}_c$ are defined by angles α_1 and α_2.The binary asteroid system includes two triaxial ellipsoid.

Triaxial ellipsoid model is a common asteroid gravity field model. Hu[8]has suggested that a uniformly rotating arbitrary second degree and order gravity field can be described through second degree and order gravity coefficients

$$U=\frac{\mu}{r}+\left[-\frac{\mu C_{20}\left(x^2+y^2-2z^2\right)}{2r^5}+\frac{3\mu C_{22}\left(x^2-y^2\right)}{r^5}\right] \tag{2}$$

where μ is the gravity constant of the asteroid, $r=\sqrt{x^2+y^2+z^2}$ is the distance between the gravity tractor and the barycenter of the asteroid, C_{20}, C_{22} are the gravity coefficients.

Although the second degree and order gravity field can be applied in arbitrary mass distribution, we assume that the asteroid is an ellipsoid with semi-major areas $\alpha>\beta>\gamma$ along its x, y, and z axes, which is same as Hu[8].

The moments of inertia of the ellipsoid is

$$\begin{cases} I_{xx}=\dfrac{\beta^2+\gamma^2}{5} \\ I_{yy}=\dfrac{\alpha^2+\gamma^2}{5} \\ I_{zz}=\dfrac{\alpha^2+\beta^2}{5} \end{cases} \tag{3}$$

The second degree and order gravity coefficients are directly related to the principal moments of inertia of the body

$$\begin{cases} C_{20}=-\dfrac{1}{2}\left(2I_{zz}-I_{xx}-I_{yy}\right) \\ C_{22}=\dfrac{1}{4}\left(I_{yy}-I_{xx}\right) \end{cases} \tag{4}$$

In this section, the gravitational field of the binary asteroid system is established. The binary asteroid system is considered as two triaxial ellipsoid.

In order to improve the computation efficiency, we employ the second degree and order gravity model rather than the elliptic integrals[9] to describe the gravitational field of the triaxial ellipsoid.

Hovering Control of Gravity Tractor based on Asynchronous Advantage Actor-Critic(A3C)

The RL describes a scenario where an agent interacts with an environment $\mathcal{E}$ over a number of discrete time steps[10]. At each time step t, the agent can acquire a state s_t from the environment and select an action a_t according to its policy π, where π is a mapping from state s_t to action a_t. In return, the environment delivers the next state s_{t+1} depend on the state transition and a scalar reward r_t according to the reward function. The process continues until reaching a certain time or a terminal state. The sequence of the state and action called episode τ.

The goal of the RL is acquiring an optimal policy which maximizes the return from each state s_t. The return of the episode which starts at time step t

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{5}$$

is the total accumulated reward from each time step t with discount factor $\gamma \in (0,1]$. Note $R(\tau)$ is the return of the episode τ.

Assume that both the state transition and the policy are random, the probability along the episode τ under policy π with initial state probability distribution function ρ_0 is

$$P(\tau|\pi) = \rho_0(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t)\pi(a_t|s_t) \tag{6}$$

where $P(s_{t+1}|s_t, a_t)$ is the probability of state transition which transferring the state s_t to the state s_{t+1} when taking action a_t, $\pi(a_t|s_t)$ is the probability which the policy π choose action a_t under the state s_t.

The expectation of the return under the policy π is

$$J(\pi) = \int_{\tau} P(\tau|\pi) R(\tau) = E_{\tau\sim\pi}\left[R(\tau)\right] \tag{7}$$

Consequently, the RL problem can be formulated to

$$\pi^* = \arg\max_{\pi} J(\pi) \tag{8}$$

where π^* is the optimal policy, $J(\pi)$ is the target function.

Value functions are important in RL theory. The action value

$$Q^{\pi}(s,a)=E\left[R_t \middle| s_t=s,a\right] \tag{9}$$

is the expected return for selecting action a in state s and following policy π.

Similarly, the value of state s under policy π is defined as

$$V^{\pi}(s)=E\left[R_t \middle| s_t=s\right] \tag{10}$$

and is simply the expected return for following policy π from state s. According to the role of the value function when making decision, the RL method can be divided into value-based methods and policy-based methods.

In value-based RL methods, the action value function is represented using a function approximator, such as a neural network. The action is chosen to acquire a maximum action value in current time step. In contrast to value-based methods, policy-based methods directly parameterize the policy $\pi_\theta(a|s)$ and update the parameters θ by performing, typically approximate, gradient ascent on $J(\pi)$

$$\theta_{t+1}=\theta_t+\alpha\nabla_\theta J(\pi_\theta)\big|_{\theta_t} \tag{11}$$

where $\nabla_\theta J(\pi_\theta)\big|_{\theta_t}$ is the policy gradient with parameter θ

$$\nabla_\theta J(\pi_\theta)=E_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T-1}\nabla_\theta \log\pi_\theta(a_t|s_t)\right)\left(\sum_{k=0}^{T-1}\gamma^k r_{t+k}\Big|_{t=0}\right)\right] \tag{12}$$

The expectation can be estimated by sampling during the interaction rather than calculating directly. The estimated value is

$$\nabla_\theta J(\pi_\theta)\approx\frac{1}{N}\sum_{i=1}^{N}\left[\left(\sum_{t=0}^{T-1}\nabla_\theta \log\pi_\theta(a_t|s_t)\right)\left(\sum_{k=0}^{T-1}\gamma^k r(s_{t+k},a_{t+k})\Big|_{t=0}\right)\right] \tag{13}$$

where N is the number of the episode.

An actor-critic(AC) architecture is established by using an action value function to replace the return

$$\nabla_\theta J(\pi_\theta)\approx\frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T-1}\left[\nabla_\theta \log\pi_\theta\left(a_t^n|s_t^n\right)Q^{\pi_\theta}\left(s_t^n|a_t^n\right)\right] \tag{14}$$

where the value function Q^{π_θ} is approximated by a critic network.

The critic network is updated according to the loss function, which is the Temporal-Difference Error(TD-Error)

$$loss=\frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T-1}\left[r_t^n+\max_{a_{t+1}^n}Q^{\pi_\theta}\left(s_{t+1}^n|a_{t+1}^n\right)-Q^{\pi_\theta}\left(s_t^n|a_t^n\right)\right]^2 \tag{15}$$

The training process of the neural network has been a problem for machine learning. The added network increases the difficulty of the training process. Baseline makes the training process more stable. The state value function is usually employed as the baseline. The policy gradients can be calculated in equation(16)

$$\nabla_\theta J(\pi_\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T-1}\left[\left(Q^{\pi_\theta}\left(s_t^n \middle| a_t^n\right) - V^{\pi_\theta}\left(s_t^n\right)\right)\nabla_\theta \log \pi_\theta\left(a_t^n \middle| s_t^n\right)\right] \tag{16}$$

where $Q^{\pi_\theta}\left(s_t^n \middle| a_t^n\right) - V^{\pi_\theta}\left(s_t^n\right)$ is called advantage function.

There are two value function in equation(16), which requires extra network. According to the relationship between the state value function and the action value function

$$Q^{\pi}(s_t, a_t) = E\left[r_t + V^{\pi}(s_{t+1})\right] \tag{17}$$

The action value function can be approximated by the state value function

$$Q^{\pi}(s_t, a_t) = r_t + V^{\pi}(s_{t+1}) \tag{18}$$

Substituting equation(18) into equation (16) can obtain the policy gradients of the Advantage Actor-Critic(A2C)method

$$\nabla_\theta J(\pi_\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T-1}\left[\left(r_t^n + V^{\pi_\theta}\left(s_{t+1}^n\right) - V^{\pi_\theta}\left(s_t^n\right)\right)\nabla_\theta \log \pi_\theta\left(a_t^n \middle| s_t^n\right)\right] \tag{19}$$

The loss function of the critic is

$$loss = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T-1}\left[r_t^n + V^{\pi_\theta}\left(s_{t+1}^n\right) - V^{\pi_\theta}\left(s_t^n\right)\right]^2 \tag{20}$$

Google has proposed a multi-threaded asynchronous method[11] for optimizing the neural network controller. They use asynchronous actor-learners running in parallel on multiple CPU threads on a single machine to explore different parts of the environment. This method no longer relies on experience replay for stabilizing learning process.

Figure.2 shows a conceptual illustration of Asynchronous Advantage Actor-Critic(A3C). After the agent in each thread interacts with the environment to obtain a certain amount of data, it calculates the gradient of the neural network loss function in its own thread. These gradients will not update the neural network in its own thread, but update the public network which is called global network. Hence, several threads will independently use the accumulated

gradient to update the neural network model parameters of the common part. In addition to stabilizing learning, using multiple parallel actor-learners helps to reduce the training time.

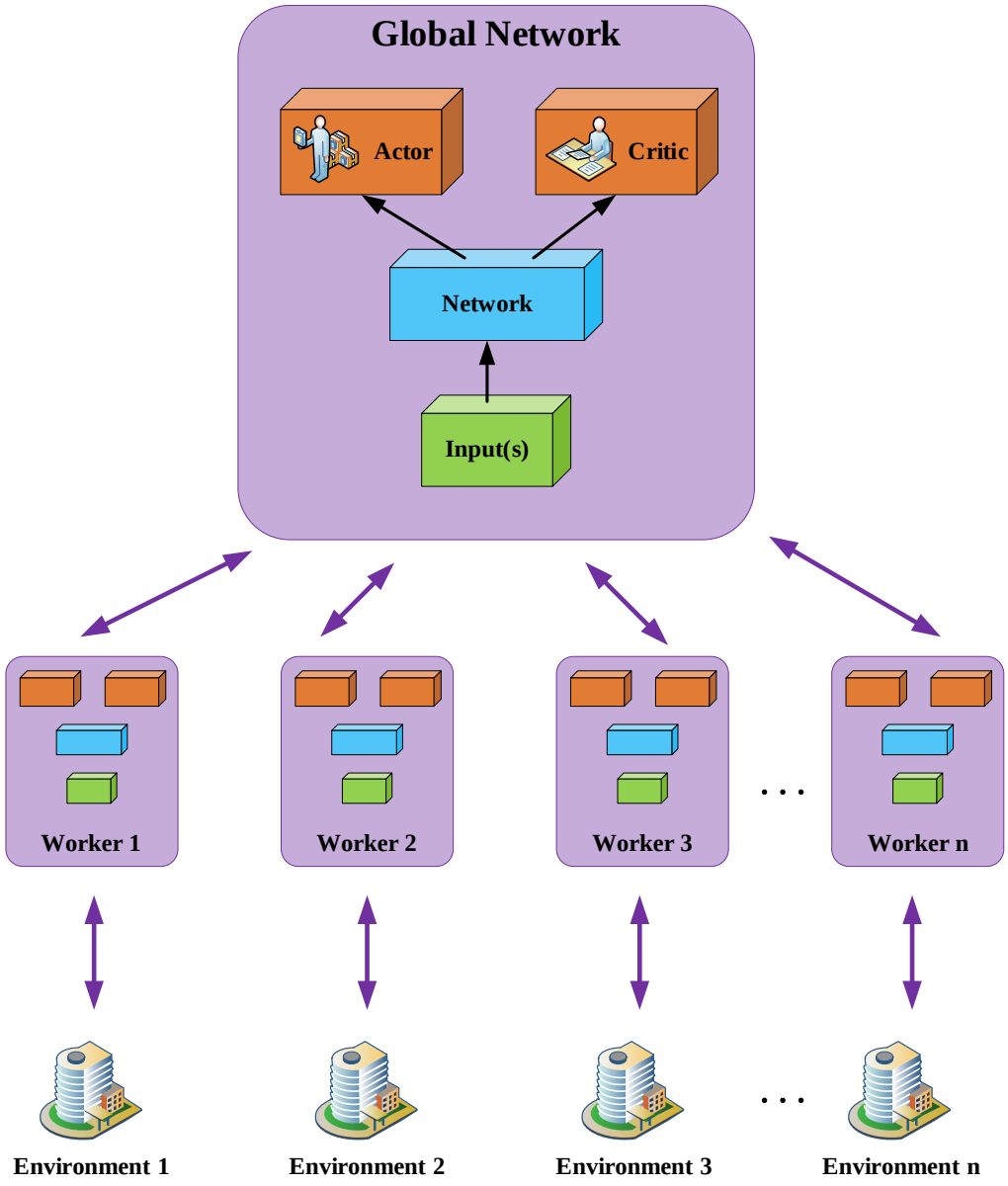

Fig.2 A conceptual illustration of A3C

This paper employs the asynchronous RL method to design the gravity tractor hovering control. The RL problem is typically formulated as a Markov Decision Process (MDP). An MDP includes states space, actions space, a reward function, state transition probabilities and an optional discount rate. In this paper, the states space

$$S = \begin{bmatrix} x - x_d & y - y_d & z - z_d & \dot{x} - \dot{x}_d & \dot{y} - \dot{y}_d & \dot{z} - \dot{z}_d \end{bmatrix} \tag{21}$$

consist of the deviation of the position and the velocity so that the controller

could drive the GT to the hovering state, where $\boldsymbol{r}=\begin{bmatrix} x & y & z \end{bmatrix}^T$ is the spacecraft's current location, $\boldsymbol{r}_d=\begin{bmatrix} x_d & y_d & z_d \end{bmatrix}^T$ is the spacecraft desired location. Because the gravitational attraction is determined by the position, the deviations are coupled. The actions space

$$A=\begin{bmatrix} u_x & u_y & u_z \end{bmatrix} \tag{22}$$

is chosen directly from the control accelerations. As a result, the policy has six-dimensional state space and a three-dimensional action space.

The reward function

$$R(s)=k_r\left\|\boldsymbol{r}-\boldsymbol{r}_d\right\|+k_v\left\|\dot{\boldsymbol{r}}-\dot{\boldsymbol{r}}_d\right\| \tag{23}$$

describes the deviation of the position primarily. The deviation of the velocity is added so that the GT could achieve the hovering state in a shorter time. k_r, k_v are the weight coefficients which balance the influence of the deviations.

In the hovering control problem, the state transition is determinate. The state transition probabilities is dominated by the dynamic introduced in Sec.II. The discount rate decides the influence of the reward in the subsequent steps.

Numerical Simulations

To demonstrate the adaptive capacity, three numerical simulations are done. The first simulation is a scenario that the dynamics in testing is the same as the one in training. In second simulation, the uncertainty of the dynamics is added and the result is illustrated. The third one is to shows that the model could learning online. This ability is demonstrated by initializing the parameter of the neural network with previous policy. The current policy through training could guarantee the control accuracy respect to a new environment. When working in a changed environment, the RL model could adapt to the difference according to the data and guarantee the accuracy of the control.

The simulator used to calculate a candidate policy's training process is Euler integration with a 0.1 second time-step and a control frequency of 10Hz. The simulation is considered with the parameter illustrated in Table 1.

Table.1 Physical parameters of the binary asteroid system and GT

Physical parameter	Magnitude	Unit
Thrust	[-1,1]	N
Mass of GT	10000	kg
Hovering Position	[6000,0,0]	m
Perturb	$[2,-3,4]\times10^{-5}$	m/s^2

Semi-axis of asteroid 1	[1.417,1.361,1.183]	km
Semi-axis of asteroid 2	[0.595,0.450,0.343]	km
Density of asteroid 1	1.97×10^{15}	kg/km^3
Density of asteroid 2	2.81×10^{15}	kg/km^3
Period of asteroid 1	2.7645	h
Period of asteroid 2	17.4223	h
Period of system	17.4223	h

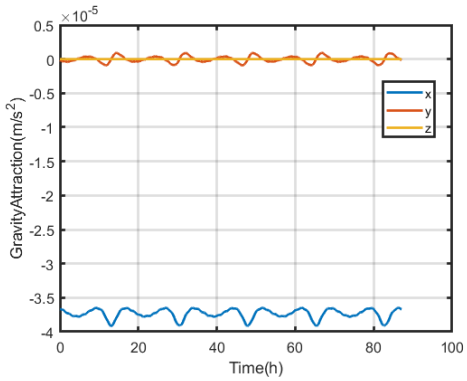

Fig.3 The gravity acceleration on the desire hovering position

Without loss of generality, the desire hovering position could be chosen on the *x*-axis. Fig.3 shows the gravity acceleration on the desire hovering position in five period of system. Because of the rotation and the revolution, the gravity acceleration on the desire hovering position is time-varying.

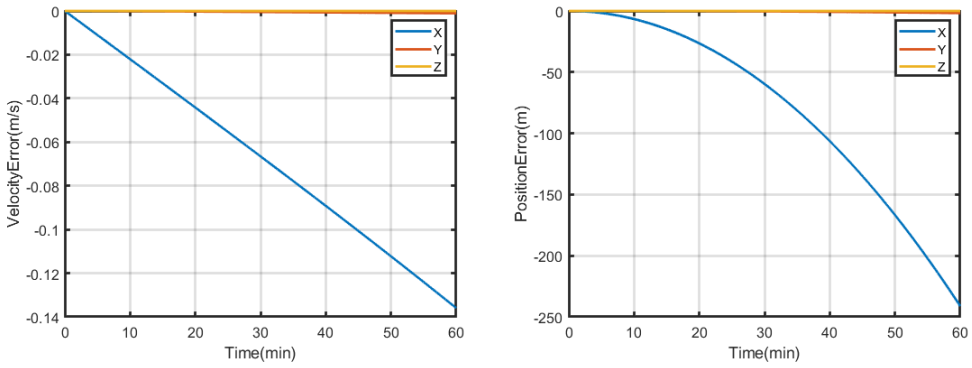

Fig.4 The deviation of the state without control

Fig.4 shows the deviation of the state without control. If the gravity tractor were to be uncontrolled, the gravity tractor should drift in a short time. Assume that the GT locates on the desire position at the begin, the GT will move 250m respect to the initial position in one hour.

Table 2 The NN architecture of the actor-critic frame

	Actor		Critic	
	units	activation	units	activation
Input Layer	6	/	6	/
Layer1	200	tanh	100	tanh
Layer2	200	tanh	100	tanh
Output Layer	3 3	tanh softplus	1	None

The network architecture is illustrated in Table 2. The actor network and the critic network have common part. The output layer of the actor network decides the mean and the standard deviation of the random policy.

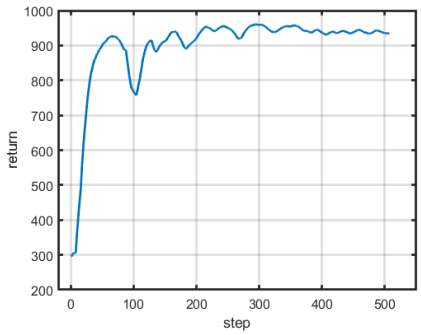

Fig.5 Policy optimization evolution

The return during the training steps is illustrated in Fig.5. The return increases rapidly at the beginning of the training process, which means the policy is optimized immediately. The policy may get optimal parameters after 400 steps, where the return arrives at a plain. The return curve in the last 100 steps has oscillation because the candidate policy belongs to random policy. Note that the policy in this stage is π_1 and the environment is env_1.

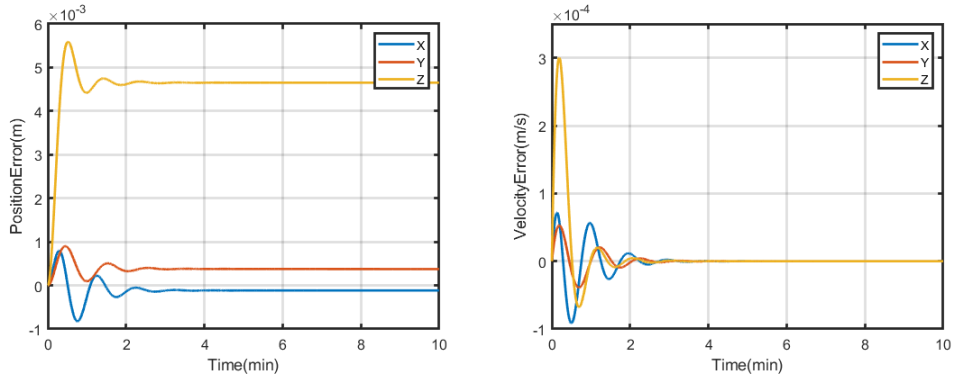

Fig.6 The deviation of the state with policy π_1 in env_1

Figure 6 shows that the policy π_1 is able to achieve and maintain the hovering state. The GT has achieved the desired hovering state in 10min. The steady-state error is 0.0046686m in position and 8.6245×10^{-9} m/s^2 in velocity.

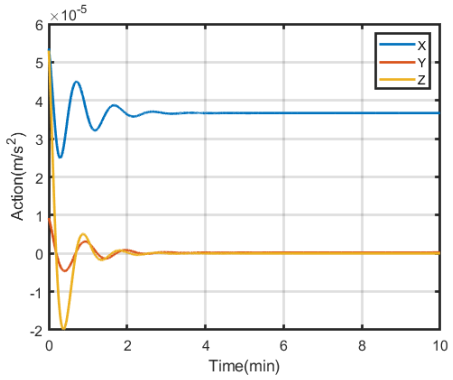

Fig.7 Acceleration command as a function of time in short-term

The acceleration command is illustrated in Fig.7. The command satisfies the maximum acceleration constraint of the thruster. After the GT achieves the hovering state, the acceleration command is nearly a constant.

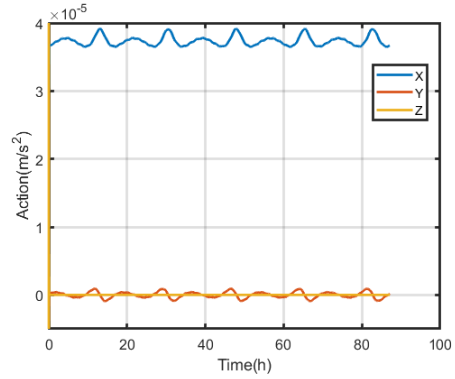

Fig.8 Acceleration command as a function of time in long-term

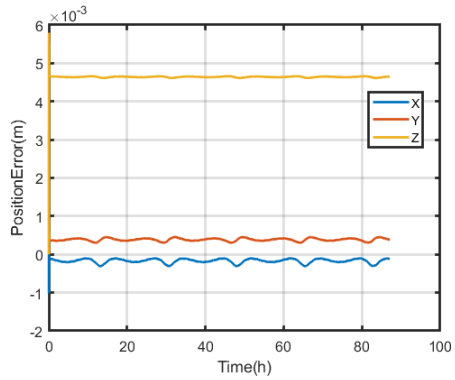

Fig.9 The deviation of the position as a function of time in long-term

Compared with Fig.3, after achieving the hovering state, the thruster could balance the gravitational attraction without knowledge of the dynamics(as

shown in Fig.8). There is some fluctuation in Fig.9 because the evaluated policy in the simulation is training in a short time, where its parameters can not track(adapt) the change of the gravity for such a long time. The adaptive capacity will be demonstrated later. During the mission, the parameters of the NN will be corrected while training online and the fluctuation could be suppressed. In another word, the policy is time-varying.

The mission of GT is long-term. Assume that the dynamics of the system changes slowly with time, and there is a conspicuous difference between the preliminary stage and the terminal stage. This difference could be raised by the solar radian pressure or the evolution of the system. In this paper, a constant acceleration is employed to stand for this change. Although the policy π_1 could accomplish the hovering control, the effect gets worse. Denote the environment which has a different dynamics is env_2. The deviation of the state in a new environment env_2 is illustrated in Fig.10.

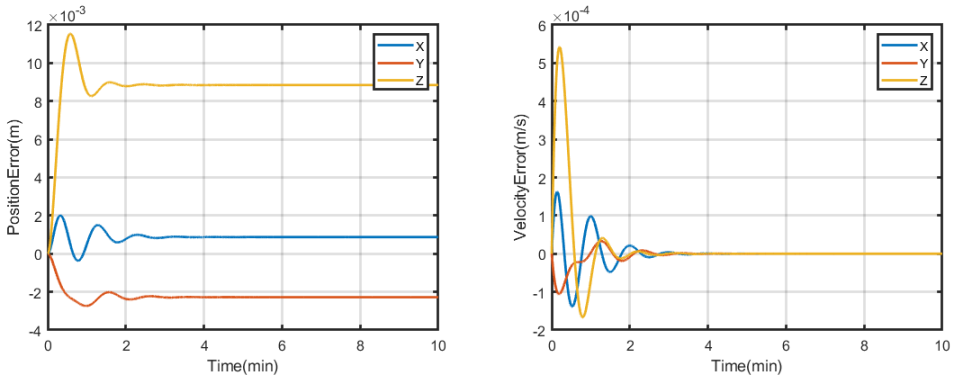

Fig.10 The deviation of the state with policy π_1 in environment env_2

The parameters of the policy are optimal respect to a certain environment. If the evaluated environment is different from the training environment, the ability of the policy is degraded. The steady-state error is 0.0091777m in position and 9.4973×10^{-9} m/s^2 in velocity.

A new policy should be raised to adapt the environment so that the controller could maintain its accuracy. Using the parameters of the NN in policy π_1 to initialize the NN of the policy π_2 and the sample produce during the mission to train the model, the result can be improved.

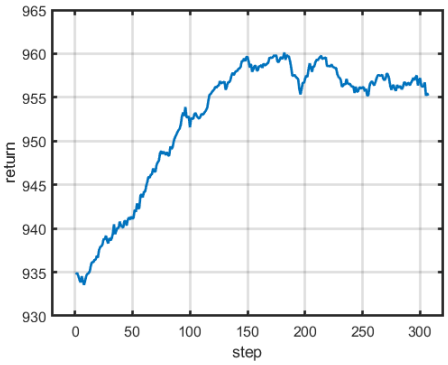

Fig.11 Policy optimization evolution

The return is illustrated in Fig.11. After learning for a number of steps, the return arrives at another plain. It means that the reinforcement learning model has been trained as an optimal policy again. The gravitational attractions on the hovering position change slowly, and the RL model will search for an optimal policy to the environment steadily as a result.

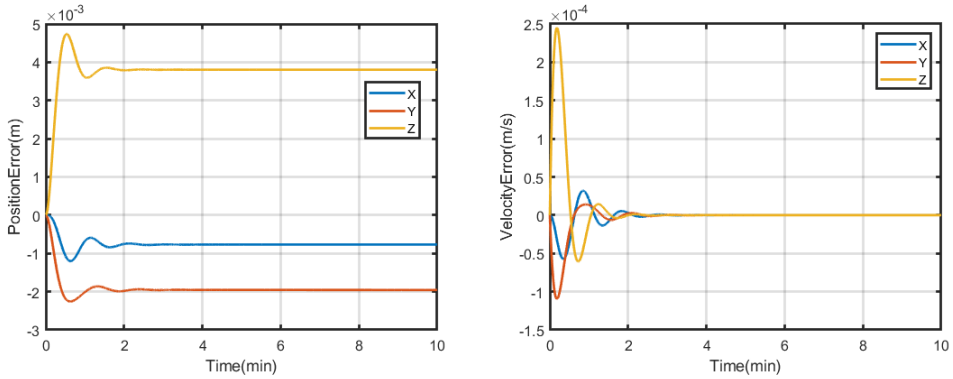

Fig.12 The deviation of the state with policy π_2 in environment env_2

Employing the policy π_2 in the environment env_2, the result is illustrated in Fig.12. The steady-state error is 0.0043469m in position and 7.7531×10^{-9}m/s^2 in velocity. Compared with the result in previous, the trained model can decrease the error caused by the change of the environment. Consequently, when the model is training during the mission, the policy could suppress the fluctuation in Fig.9.

Implementation Details

In order to facilitate reproduction of our results, we include in this section several techniques we used in our implementation. We use the ADAM optimizer to adjust the learning rate for both the policy and value function networks. The feature scaling is applied to the state, action and reward. The variables are assumed bounded and a min-max normalization is employed.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{24}$$

where x is an original value, x' is the normalized value. According to the assume before, this normalization can be written in

$$x' = \frac{x}{x_{bound}} \tag{25}$$

where x_{bound} is the boundary value. By means of feature scaling, the variables can be normalized to the range in [0,1]. Feature scaling includes a series of methods and the hyperparameter could be adjusted for different scenarios. Table 3 shows the training hyperparameter in this paper.

Table 3 Hyperparameter

Hyperparameter	Magnitude	Unit
The boundary value of position	5	m
The boundary value of velocity	1	m/s
The boundary value of action	1	N
The discount rate	0.9	
The learning rate of the actor	0.0001	
The learning rate of the critic	0.001	
k_r	0.8	
k_v	0.2	

Conclusion and Discussion

This paper proposes that Reinforcement Learning(RL) could help the Gravity Tractor(GT) to maintain the hovering state and adapt to the change of the environment. The binary asteroid system is modeled as a double ellipsoid system and the gravity model of the triaxial ellipsoid is a second degree and order gravity field. The relationship mapping the Markov Decision Process(MDP) and the hovering control problem is established. The actor-critic frame is explained as well. The simulation results have demonstrated that the RL model could adapt to the change of the attraction on the hovering position. The RL algorithm employed here is Asynchronous Advantage Actor-Critic. It belongs to on-policy algorithm, which supports learn the data and update the policy during

the mission. This feature makes the agent evolution with the environment. As a long-term mission, this operation can produce lots of samples to train the model. On the other hand, learning online helps the agent to maintain the control accuracy. The RL model could adapt the evolution of the environment. Moreover, a little of researchers study the system of orbiting multiple gravity tractors to acquire larger total velocity increasement of the asteroid. Each single GT is treated as the asynchronous actor-leaners. Under this asynchronous frame, the muti-GT can take full advantage of the formation to sample the training data effectively.

References

[1] Lu, Edward T., and Stanley G. Love. "Gravitational tractor for towing asteroids." Nature 438.7065 (2005): 177-178.

[2] Wie, Bong. "Hovering control of a solar sail gravity tractor spacecraft for asteroid deflection." Proceedings of the 17th AAS/AIAA Space Flight Mechanics Meeting, AAS. Vol. 7. 2007.

[3] Wie, Bong. "Dynamics and control of gravity tractor spacecraft for asteroid deflection." Journal of guidance, control, and dynamics 31.5 (2008): 1413-1423.

[4] Furfaro, Roberto. "Hovering in asteroid dynamical environments using higher-order sliding control." Journal of Guidance, Control, and Dynamics 38.2 (2015): 263-279.

[5] Guzzetti, Davide. "Reinforcement learning and topology of orbit manifolds for station-keeping of unstable symmetric periodic orbits." AAS/AIAA Astrodynamics Specialist Conference. 2019.

[6] Gaudet, Brian, and Roberto Furfaro. "Robust spacecraft hovering near small bodies in environments with unknown dynamics using reinforcement learning." AIAA/AAS Astrodynamics Specialist Conference. 2012.

[7] Woo, Pamela, Arun K. Misra, and Mehdi Keshmiri. "On the planar motion in the full two-body problem with inertial symmetry." Celestial Mechanics and Dynamical Astronomy 117.3 (2013): 263-277.

[8] Hu, W., and Daniel Jay Scheeres. "Numerical determination of stability regions for orbital motion in uniformly rotating second degree and order gravity fields." Planetary and Space Science 52.8 (2004): 685-692.

[9] Scheeres, Daniel Jay. "Dynamics about uniformly rotating triaxial ellipsoids: applications to asteroids." Icarus 110.2 (1994): 225-238.

[10] Sutton, R. and Barto, A. Reinforcement Learning: an Introduction.MIT Press, 1998.
[11] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. PMLR, 2016.