AAS 19-692

ELM-BASED ACTOR-CRITIC APPROACH TO LYAPUNOV VECTOR FIELDS RELATIVE MOTION GUIDANCE IN NEAR-RECTILINEAR ORBITS

Andrea Scorsoglio*, Roberto Furfaro†

In this paper, we present a new feedback guidance algorithm for autonomous docking maneuvers in the cislunar environment. In particular, we propose a closed-loop optimal guidance algorithm that is capable of taking path constraints and collision avoidance into account while being on a Near Rectilinear Orbit (NRO) around the L2 Lagrangian point in the Earth-Moon system. The algorithm is based on the Lyapunov vector field guidance where the acceleration command is derived from a desired velocity vector field. We use reinforcement learning to learn the shape of the field as a function of the state of the system, allowing for increased flexibility in terms of constraint shapes and better performance in terms of fuel consumption with respect to classical Lyapunov vector field guidance.

INTRODUCTION

Accurate guidance algorithms have always played a pivotal role in space exploration. With the increasing interest in Moon exploration and eventually Mars, both with robotic and manned missions, the interest in new autonomous guidance solutions is also increasing. In particular, autonomous proximity operations for docking to a space station is an active field of research. The applications are various: among them are the transfer of fuel and astronauts and on-orbit assembly. With the Lunar Orbital Platform-Gateway (LOP-G)[1] under development, meant to become the new establishment for human exploration of the solar system, relative dynamics guidance in the cislunar environment will become of pivotal importance in the future. NASA has shown that near rectilinear orbits (NRO) have some advantages over other cislunar orbits[2] and therefore will probably be selected for this kind of missions. Their peculiar shape allows them to have continuous coverage of either one of the sides of the Moon, being in the meantime continuously visible from Earth. Moreover, they are advantageous in terms of ΔV for transfer to and from Earth and lunar surface, and they are within the launching capabilities of an SLS-Orion mission. For this reason the guidance algorithm developed in this work has been applied to docking maneuvers in NRO.

Traditionally, relative motion maneuvers are performed using open-loop planning techniques,[3] in general by solving the associated optimal two-point boundary value problem (TPBVP) either via direct or indirect methods and then use a controller to track the optimal trajectory. So ad-hoc maneuver corrections must be employed to compensate for errors inherent to open-loop control. Although powerful, the introduction of path constraints in this kind of architectures is cumbersome

*PhD student, System & Industrial Engineering Department, University of Arizona, 1127 E. James E. Rogers Way, Tucson, AZ 85721

†Professor, System & Industrial Engineering Department, University of Arizona, 1127 E. James E. Rogers Way, Tucson, AZ 85721

and are in general limited to simple shapes (conical, spherical). Open-loop architectures are also not robust against perturbations on the environmental conditions and are in general not very responsive to changes in the mission profile. Recently, interest has been increasing around closed-loop maneuvering, especially for missions that involve formation flying or automated rendezvous, docking, and proximity operations. During the process of spacecraft rendezvous and docking, especially in the final phase, flight safety is an essential requirement and a prerequisite for all other on-orbit tasks. There are two main safety concerns during docking maneuvers. First of all, due to the relative overcrowding of some areas of space, the spacecraft may become the victim of an impact with some non-cooperative object/obstacle, such as space debris, which may potentially cause task failure and economic loss. It is thus necessary for spacecraft to have autonomous obstacle avoidance ability. A high-precision control law for spacecraft formation reconfiguration was proposed by Zhou.[4] Li et al.[5] introduced an optimal control law, which also considers obstacle avoidance for spacecraft formation for spacecraft near libration points. Moreover, Breger and How[6] proposed a method to generate safe, fuel-optimized rendezvous trajectories with a collision-avoidance ability. On the other hand, a large scale target like a space station may have many components extending outwards with respect to the main body. In the final phase of rendezvous and docking when the relative distance between the target and the pursuer is small, this may lead to the pursuer colliding with the target while approaching the docking port. To solve this problem, a keep-out sphere is introduced, which contains all the components of the target and only allows the pursuer approaches the docking port from a particular approach corridors. Trajectory planning[7,8] is one of the effective methods to solve this problem, but when the pursuer spacecraft is simultaneously also threatened by a possible non-cooperative obstacle, the non-cooperative nature of the obstacle renders trajectory-planning methods (which always need comprehensive information of constraints) very difficult to be applied from a real-time-implementation standpoint.

The artificial potential function (APF) method is an interesting approach that has been considered for this problem in the past; its clear physical meaning makes it very suitable to describe the motion constraints, such as keep-out zones and obstacles, and so it has aroused extensive interests[9,10] , and has been tentatively implemented in the spacecraft rendezvous and docking operations[11] recently. It is important to note that this approach proposed in some recent papers[12–14] can only deal with spatially fixed obstructions and the stability of such methods have not been analyzed, and so the approaches cannot be readily applied to most practical problems given the lack of stability and robustness guarantees. A novel approach in this field was introduced by Dong et al.[15] in which the APF method is used to solve a docking problem with path constraints and collision avoidance capabilities trying to solve the problems arose from previous papers on the matter.

All these methods, although effective, have a common drawback: they lack flexibility. Even APF methods rely on the fact that the constraints must be formalized as potential fields with a know analytical formula which, for complex shapes like a space station, may only work if one wanted to model its overall approach keep-out zones rather than the actual shape. It is in these cases that the power and flexibility of machine learning can be exploited to design algorithms that can adapt to very different scenarios. Machine learning is gaining its recognition in many fields besides computer science, where it was born. This is because many problems that seem impossible some years ago can now be tackled with the help of neural networks. Reinforcement learning (RL) has already been used mainly in robotics motion tasks [16–21] . There are examples of machine learning and reinforcement learning being applied to landing problems[22–24] and relative dynamics guidance[25] . The idea for this paper is to create a new guidance algorithm based on Lyapunov vector

field for relative guidance. Lyapunov vector fields are not new to space applications as they were already applied to some collision avoidance and docking problems [26,27] . Although interesting, those studies are limited to very basic constraint scenarios and lack the flexibility needed in a more realistic mission scenario. With this work, we want to demonstrate that reinforcement learning can be used to create an algorithm that is closed-loop and flexible enough to comply with virtually any constraint shape, minimizing in the meantime the fuel consumption. In particular, an actor-critic algorithm is used to learn the shape of the desired velocity field as a function of the state. This allows for great flexibility in terms of tasks to be carried out and, if tuned correctly, can be applied to many different environments and constraint scenarios.

PROBLEM FORMULATION

Physical model

NRO Considering two main bodies, or primaries (in this case Earth and Moon), m_1 and m_2 where the only force acting between the particles is the gravitational attraction, the motion of a particle (i.e., a satellite) is described in general by the Circular Restricted Three Body Problem (CRTBP). The dynamics of the problem are expressed in the absolute synodic reference frame that in the case of the Earth-Moon system will be called $\mathcal{R}_{em}$. The origin of this frame is in the center of mass of the system G, the x-axis is aligned with the line connecting the two primaries, the z-axis is parallel to the angular momentum vector of the primaries, and the y-axis completes the orthonormal triad. The frame rotates with an angular velocity equal to the mean angular motion of the two primaries around their center of mass. Moreover, quantities in this reference frame are made non-dimensional by introducing some normalization parameters.

In this reference system, the equations of motion describing the dynamics of the particle are the following:

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

with

$$\begin{aligned} r_1 &= \sqrt{(x+\mu)^2 + y^2 + z^2} \\ r_2 &= \sqrt{(x-(1-\mu))^2 + y^2 + z^2} \end{aligned} \tag{2}$$

and $\mu = \dfrac{m_2}{m_1 + m_2}$ being the mass parameter. A more in-depth study on the problem and the origin of the equations of motion can be found in the references.[28]

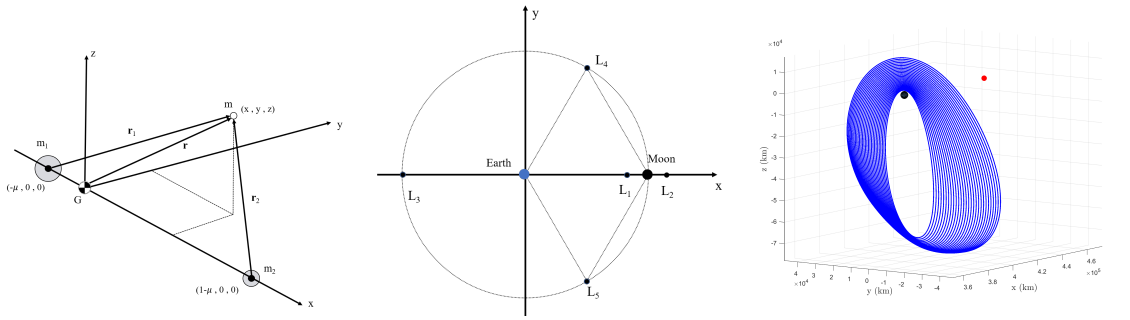
The equilibrium points, or Lagrangian points, or libration points are stationary points of the potential function U of the CRTBP defined as

$$U = \frac{1}{2}\left(x^2 + y^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{3}$$

and are the solutions of the equation

$$\nabla U = 0 \tag{4}$$

In the CRTBP framework, there exist a wide variety of trajectories that result in a periodical motion around the libration points. Near Rectilinear Orbits, or NROs, are a degenerate subset of Halo



(a) Synodic reference frame

(b) Lagrangian Points

(c) L_2 southern family. The red dot is the lagrangian point, the black dot is the Moon

Figure 1. Circular Restricted Three Body Problem (CR3BP)

Orbits whose projection on the x-y plane of the closest point to one of the primaries lies inside the circle defined by the projection on the same plane of the aforementioned primary. The generation of periodical orbits in this framework is carried out using a shooting algorithm based on a multi-variable newton method. The whole process of finding those orbits is described thoroughly in the references[29,30] and since it is not the main subject of this project, it will not be described in details. A representation of the NROs family taken into consideration for this project can be seen in Figure 1.

Rendezvous in NRO The problem addressed in this project is the creation of a guidance algorithm for performing rendezvous in the context of cislunar NRO. The operations guidelines for this kind of mission have already been formalized in the past.[25,31] Noted that the cislunar short-term relative dynamics are quasi-straight, the constraints and safety procedures developed for the faster dynamics of the problem in the neighborhood of a strong central body, are no longer valid. So the new regulations define four areas around the target related to different phases of the rendezvous procedure: the Keep-Out Sphere (KOS), the Approach/Departure Corridors, the Approach Sphere (AS) and Rendezvous Sphere (RS). See reference for details[31] . In this project, the focus is put only on the close approach part of the problem. Precisely, it is assumed that the most critical part is the one related to precision guidance inside the AS, so this is the environment considered in this study. The motion of chaser and target is described by equations 1 in the non-dimensional synodic reference frame. These equations, however, are not ideal for describing the relative guidance and control problem, so the introduction of relative reference frames and relative dynamics equations is necessary. The motion of the chaser as seen from the target centered reference frame is defined as relative motion. In the cislunar environment, the problem of relative motion has not been studied as extensively as in the two-body problem. A new set of reference frames and quations of motion must be introduced.

Reference frames The dynamics in the Earth-Moon CRTBP are developed in the absolute synodic non-dimensional frame $\mathcal{R}_{em}$. In two-body dynamics the relative reference frame that is commonly used is the Local-Vertical-Local-Horizon (LVLH) frame. The LVLH frame ($\mathcal{R}_l$) has been defined also for the CRTBP[31] . Although the problem is different with respect to the two-body case, it has been demonstrated that the short term NRO dynamics can be described in a LVLH defined with

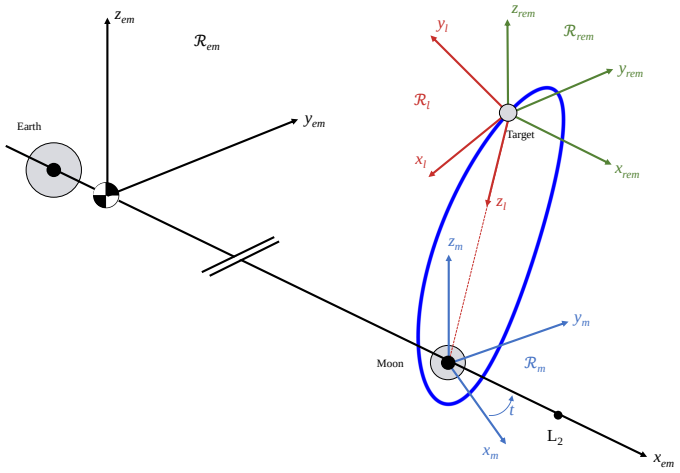


Figure 2. Reference systems (from Scorsoglio[25])

respect to a Moon Centered Inertial (MCI) frame ($\mathcal{R}_m$). Moreover, the Earth-Moon relative synodic (EMRS) frame ($\mathcal{R}_{rem}$) is an additional reference frame used in the project, defined as the relative version of the absolute synodic frame, aligned with the latter at all time and centered on the target. An extensive explanation of the reference systems and the change of coordinates between them can be found in the references[31] . A representation of all the reference systems on a NRO can be seen in Figure 2.

Relative equations of motion The relative motion in NROs can be described using two models depending on the position along the orbit. It has been shown[31] that in areas of the orbit where the gravitational influence of the Moon is strong, so close to the periselene*, the problem dynamically resembles the two-body counterpart hence using the *Clohessy-Wiltshire equation* (CW) expressed in the LVLH frame introduce a relatively small error. In the region close to the aposelene † and in general in any other region of the orbit, the *Non-Linear Relative equations* (NLR) defined in the relative synodic reference system (EMRS) should be employed instead.

- The Clohessy-Wiltshire equations (**CW**) are a set of equations for describing the relative motion of the chaser with respect to the target in the two-body LVLH frame. In this frame the equations take the form:

$$\begin{aligned} \ddot{x} - 2n\dot{z} &= 0 \\ \ddot{y} + n^2 y &= 0 \\ \ddot{z} + 2n\dot{x} - 3n^2 z &= 0 \end{aligned} \tag{5}$$

  where

$$n = \sqrt{\frac{\mu}{r_0^3}} \tag{6}$$

  Where r_0 is the distance from the center of the second primary.

*The closest point on the orbit from the Moon

†The furthest point on the orbit from the Moon

- The Non-linear Relative Equations in synodic reference system (**NLR**) are obtained by subtraction of the absolute equations of motion in CRTBP for the target and the chaser and are expressed in the $\mathcal{R}_{rem}$ reference frame:

$$\begin{aligned}
\ddot{x} - 2\dot{y} - x &= (1-\mu)\left[\frac{x_T+\mu}{\|r_{1T}\|^3} - \frac{x_T+x+\mu}{\|r_{1T}+\rho\|^3}\right] + \mu\left[\frac{x_T+\mu-1}{\|r_{2T}\|^3} - \frac{x_T+x+\mu-1}{\|r_{2T}+\rho\|^3}\right] \\
\ddot{y} + 2\dot{x} - y &= (1-\mu)\left[\frac{y_T}{\|r_{1T}\|^3} - \frac{y_T+y}{\|r_{1T}+\rho\|^3}\right] + \mu\left[\frac{y_T}{\|r_{2T}\|^3} - \frac{y_T+y}{\|r_{2T}\rho\|^3}\right] \\
\ddot{z} &= (1-\mu)\left[\frac{z_T}{\|r_{1T}\|^3} - \frac{z_T+z}{\|r_{1T}+\rho\|^3}\right] + \mu\left[\frac{z_T}{\|r_{2T}\|^3} - \frac{z_T+z}{\|r_{2T}\rho\|^3}\right]
\end{aligned} \tag{7}$$

where

$$\mathbf{x} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}] = \mathbf{x}_C - \mathbf{x}_T \tag{8}$$

is the synodic relative state,

$$\rho = [x \quad y \quad z]^{\mathrm{T}} \tag{9}$$

is the relative position,

$$\begin{aligned}
\mathbf{x}_T &= [x_T \quad y_T \quad z_T \quad \dot{x}_T \quad \dot{y}_T \quad \dot{z}_T]^T \\
\mathbf{x}_C &= [x_C \quad y_C \quad z_C \quad \dot{x}_C \quad \dot{y}_C \quad \dot{z}_C]^T
\end{aligned} \tag{10}$$

are the target and chaser synodic absolute positions,

$$\begin{aligned}
\mathbf{r}_{1T} &= [(x_T+\mu) \quad y_T \quad z_T]^T \\
\mathbf{r}_{2T} &= [(x_T+\mu-1) \quad y_T \quad z_T]^T
\end{aligned} \tag{11}$$

are the absolute non-dimensional distances of the target from the Earth and the Moon. They can be used in any region of the NRO, being them derived directly from the absolute equations of motion of a particle in the CRTBP. In this case they are used in a region close to the aposelene.

Lyapunov vector fields in rotating reference frames

Lyapunov vector fields guidance is presented here to introduce the problem and to show how this can be improved with reinforcement learning. Consider a dynamical closed-loop model:

$$\dot{\mathbf{x}} = f(\mathbf{x}) \tag{12}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a state vector. Historically, Lyapunov functions are found relative to the closed-loop system to prove its stability. In control, however, they can be used to create control laws that are inherently asymptotically stable, which makes those types of control very appealing. An attractor $\mathbf{C} \in \mathbb{R}^n$ is the set of $\mathbf{x}$ within the Lyapunov function $V \in \mathbb{R}$ that has the lowest energy state. In other words, $\mathbf{C} = \mathbf{x} \in \mathbb{R}^n | V(\mathbf{x}) = 0$. The shape of V, and consequently of C, is the parameter that must be tuned to obtain the desired control. The conditions for global asymptotic stability of the guidance law are:

- $V(\mathbf{C}) = 0$ otherwise $V(\mathbf{r}) > 0$
- V is radially unbounded: as $\mathbf{r} \to \infty, V \to \infty$
- V is continuously differentiable on an open domain $D \subset \mathbb{R}$ and $\frac{\delta V}{\delta \mathbf{r}}$ is only 0 on the attractor $\mathbf{C}$
- V is not an explicit function of time

Where $\mathbf{r}$ is the position vector. Once a V function is defined, an option for a guidance law is:

$$\mathbf{h}(\mathbf{r}) = \left[-\frac{\delta V}{\delta \mathbf{r}} \mathbf{\Gamma}(\mathbf{r}) \right]^T + \mathbf{S}(\mathbf{r}) \tag{13}$$

which represents the desired velocity. A control law with proven asymptotic tracking stability of Lyapunov vector fields in non-rotating frames is[26] :

$$\ddot{\mathbf{r}} = -\beta(\mathbf{h}(\mathbf{r}) - \dot{\mathbf{r}}) + \ddot{\mathbf{r}}_d \tag{14}$$

where $\beta \in \mathbb{R}$ is a gain value and $\ddot{\mathbf{r}}_d$ being the time derivative of the desired velocity $\ddot{\mathbf{r}}_d = \frac{\delta \mathbf{h}}{\delta \mathbf{r}} \dot{\mathbf{r}} + \frac{\delta \mathbf{h}}{\delta t}$. In this case the relative motion problem is formalized in a rotating relative reference system like the Local Vertical Local Horizon (LVLH) frame. This system, while being convenient to describe the motion in this environment, does not allow the direct application of the control law in 14. As described above, in fact, it only works in non-rotating frames. The problem is overcome by implementing the guidance in the rotating frame as if it were in an inertial frame where an input of the relative position vector $\rho \in \mathbb{R}^3$ produces the output of desired relative velocity $\mathbf{h}(\rho)$. However, the control is obtained as in 14 using the relative state as input and subtracting the uncontrolled relative acceleration a_r from the acceleration command to counteract its effect:

$$\ddot{\rho} = -\beta(\mathbf{h}(\rho) - \dot{\rho}) + \ddot{\rho}_d - a_r \tag{15}$$

The idea is to learn the field $\mathbf{h}(\mathbf{r})$ in 13 as function of the state. Specifically, the method relies on training an agent that describes a parametrized policy $\pi_\theta(u|x)$ that represents the shape of the field as a function of the state. In this case the parameters V, Γ and $\mathbf{S}$ in 13 are condensed in a single component H:

$$\pi_\theta(u|x) = \mathbf{h}(\mathbf{r}) = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} \tag{16}$$

as the complex structure of the field can be captured by the neural network. This increases a lot the flexibility of the method. One of the biggest downsides of the classical Lyapunov vector fields guidance methods is the difficulty in creating fields with complex shapes. Adapting the shape of the field to the environment is the ultimate goal for this kind of problems. This is achieved through reinforcement learning. By using an actor-critic algorithm, it is possible to learn the parameters of the field in virtually any constraint scenario (e.g., space station approach, collision avoidance, proximity operations around asteroids).

Guidance law formulation

In this project, the guidance law is derived using reinforcement learning. Specifically, an actor-critic (AC) algorithm is used to learn a control policy to dock to the target avoiding collision with obstacles and respecting path constraints. The algorithm was developed starting from the REINFORCE algorithm[18] introducing a critic network based on Extreme Learning Machines (ELM)[19,21] for estimating the value function. This method specifically was first introduced in,[25] and it is well suited to problems like this where the states and actions spaces are continuous. Literature on actor-critic algorithms and reinforcement learning in general is copious.[17,32–36] Focus will be put in the following on the explanation of this specific algorithm applied to this problem rather than the basic concepts of reinforcement learning. It is sufficient for the scope of this paper to say that an actor-critic algorithm is generally based on an agent (the spacecraft) that interacts with an environment using a parametric policy $\pi_\theta(u|x)$ depending on state x and action u and is assigned rewards depending on the actions it takes. The actor's goal is to update the policy in a way that maximizes the objective function $J(\pi_\theta) = \mathbb{E}\left[r(x,u)\right]$ which is the expectation of the return $r(x,u)$ which is in turn a function of the reward. Policy gradient algorithms optimize the policy by adjusting its parameters in the direction of the gradient $\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(u|x) Q^\pi(x,u)\right]$, where $\nabla_\theta \log \pi_\theta(u|x)$ is the gradient of the log-probability of $\pi(x,u)$ and $Q^\pi(x,u)$ is the action-value function, which is a function of the state and the action. The computation of this gradient involves an expectation that is cumbersome to compute exactly, especially with continuous state and action spaces. This is why in stochastic policy gradient, this is substituted by a sample-based approximation of it. Moreover, the introduction of a critic network allows for the approximation of $Q^\pi(x,u)$ to be used instead of its real counterpart, reducing the complexity of the task even more. The algorithm has a main loop composed by the following subroutines: sample generation, critic neural network fitting and policy update. A brief explanation of each block will be given in the following.

Samples generation At each global iteration, a batch of trajectories is generated by letting the agent moving within the simulated environment using policy $\pi_\theta(u|x)$ that maps the field shape in a features space giving a series of samples $(x_{i,t}, u_{i,t}, c_{i,t}, x_{i,t+1})^*$. The time is discretized in a fixed number of time-steps: at the beginning of each time-step the policy is sampled which gives the local field. The acceleration command is then obtained and the equations of motion integrated forward in time. The acceleration command is kept constant during the time interval. A reward is assigned at each time step depending on the final state and the mass burned. The agent runs until the end time is reached unless an impact with the constraint is detected in which case the episode ends.

Policy The policy is described by a gaussian distribution with a fixed variance σ^2. The stochasticity of the policy is essential to be able to apply the machinery involved in the stochastic policy gradient methods. Moreover, this ensures the exploration of the action space, which is important to insure convergence towards the global minimum avoiding falling into local minima. The policy that is used to test the algorithm and that could be used in practice is the deterministic version of it, which is represented by the mean of the above mentioned gaussian policy alone. The policy represents the 3 components of the Lyapunov field at the specific position. Each component is dependent

*$x_{i,t}$ is the state at time-step *t*, $u_{i,t}$ is the action at time-step *t*, $r_{i,t}$ is the reward associated to time-step *t* and $x_{i,t+1}$ is the next state

on a different set of parameters θ. The policy can be expressed as:

$$\begin{aligned} H_x &= \pi_{\theta_{H_x}} = \mathcal{N}(\mu_{H_x}, \sigma^2) \\ H_y &= \pi_{\theta_{H_y}} = \mathcal{N}(\mu_{H_y}, \sigma^2) \\ H_z &= \pi_{\theta_{H_z}} = \mathcal{N}(\mu_{H_z}, \sigma^2) \end{aligned} \tag{17}$$

where:

$$\begin{aligned} \mu_{H_x} &= \phi(\mathbf{x})^T \theta_{H_x} \\ \mu_{H_y} &= \phi(\mathbf{x})^T \theta_{H_y} \\ \mu_{H_z} &= \phi(\mathbf{x})^T \theta_{H_z} \end{aligned} \tag{18}$$

$\phi(\mathbf{x})$ is the vector of feature functions evaluated in state $\mathbf{x}$ and θ_{H_x}, θ_{H_y} and θ_{H_z} are the weight vectors associated with each output.

Features The features are a set of three-dimensional radial basis functions (RBF) with centers distributed evenly across the position space. The RBFs are represented by the expression:

$$\phi(\mathbf{r}) = e^{-\beta_R \|\mathbf{r} - \mathbf{c_r}\|^2} \tag{19}$$

with β_R being a constant parameter related to the variance of the radial functions which is set according to the distance of the centers, $\mathbf{r}$ being the position and $\mathbf{c_r}$ the centers of the RBFs. The centers are generated by dividing the position space of the problem in a set of intervals, creating a grid of equally spaced points in the position space. The deterministic part of this policy can be seen as a neural network with a three-dimensional input ($\mathbf{r}$), a single hidden layer of neurons with radial basis activation functions and a three-dimensional output layer (the three components of the Lyapunov field). See Figure 3 for reference.

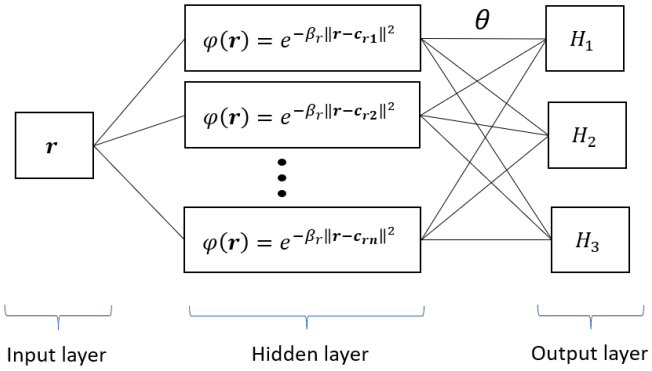


Figure 3. Policy

Critic neural network In actor-critic algorithms based on stochastic policy gradient, the expectation in the definition of the gradient of the performance parameter is not computed exactly, it is instead obtained using an approximated action-value function $Q^w(x, u)$ or, as in this case, an approximation of the advantage function $A^\pi(x, u) = Q^\pi(x, u) - V^\pi(x)$. The approximated advantage function can be rewritten, using the definition of Q, as function of V only:

$$Q^w(x, u) = \hat{A}^\pi(u, x) = \hat{Q}^\pi(x, u) - \hat{V}^\pi(x) = r(x, u) + \hat{V}^\pi(x_{t+1}) - \hat{V}^\pi(x) \tag{20}$$

where $\hat{A}^\pi(u,x)$, $\hat{Q}^\pi(u,x)$ and $\hat{V}^\pi(x)$ are the approximated versions of $A^\pi(u,x)$, $Q^\pi(u,x)$ and $V^\pi(x)$. Using this formulation, only $\hat{V}^\pi(x)$ must be obtained. This is done using an Extreme Learning Machine (ELM) with a *sigmoid* activation function. The ELM is used as a function approximator that maps the input, in this case the 6D state, into the scalar representing the discounted reward. This is done by generating at each global iteration step, a training set defined using the Monte Carlo (MC) formulation: the value function is approximated at any given state by the return, which is the discounted reward-to-go. So the training set is represented by the couples:

$$\left\{ \left(x_{i,t}, \sum_{t'=t}^{T} \gamma^{t'-t} r(x_{i,t'}, u_{i,t'}) \right) \right\} \tag{21}$$

Details about ELM are out of the scope of this paper and can be found in the references.[19,21]

Policy update The approximated value function is used to estimate the gradient of the objective function $J(\pi_\theta)$. The expression of the approximated gradient in stochastic policy gradient becomes:

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(u_{i,t}|x_{i,t}) \hat{A}^\pi(u_{i,t}, x_{i,t}) \tag{22}$$

where N is the number of sample trajectories in the batch, T is the number of time instants in each trajectory, $\nabla_\theta \log \pi_\theta(u|x)$ is the gradient of the log-probability of the stochastic policy which, for a gaussian policy like 17, is obtained analytically as:

$$\nabla_\theta \log \pi_\theta = \frac{\pi_\theta - \mu}{\sigma^2} \phi(\mathbf{x}) \tag{23}$$

and $\hat{A}^\pi(u_t, x_t)$ is the approximated advantage function and is an indication of how much better action u is with respect to the average action. Using the Monte-Carlo formulation of the advantage function, the expression becomes:

$$\hat{A}^\pi_n(u_{i,t}, x_{i,t}) = \sum_{t'=t}^{T} \gamma^{t'-t} r(x_{i,t'}, u_{i,t'}) - \hat{V}^\pi(x_{i,t}) \tag{24}$$

having introduced also the discount factor $0 < \gamma < 1$. The update then is done according to stochastic gradient ascent taking a step in the direction of the gradient $\nabla_\theta J(\pi_\theta)$:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_\theta) \tag{25}$$

where α is the bounded learning rate.

NUMERICAL RESULTS

The algorithm was tested on a constrained motion task. Focus of the project is on the computation of the guidance law regardless of the actuation on board so the model of the spacecraft is not address in details. Suffice to say that it has an initial mass $m_0 = 1500$ kg and the propulsion unit has specific impulse $I_{sp} = 220$ s and maximum thrust $T_{max} = 4$ N. It is assumed that the control thrust can be applied along any specific direction at any time meaning that the attitude control system can completely control the spacecraft along each axis. Chaser and target are assumed to be initially on the same NRO, selected among the families presented above, separated by a small distance. In

particular, it is an orbit of the southern L_2 family, the periselene has an altitude of 2439 km over the Moon surface while the aposelene has an altitude of 69758 km and the period is 6.89 days. The chaser is assumed to be inside the approach sphere and following the target along the orbit. The algorithm is tested on two different regions of the orbit. In the first one, the two satellites are considered to be in a region across the periselene of the NRO where the CW equations can be employed, in the second, they are across the aposelene and the NLR equations of motion are used instead.

Collision avoidance and keep-out sphere with approach corridor path constraint task

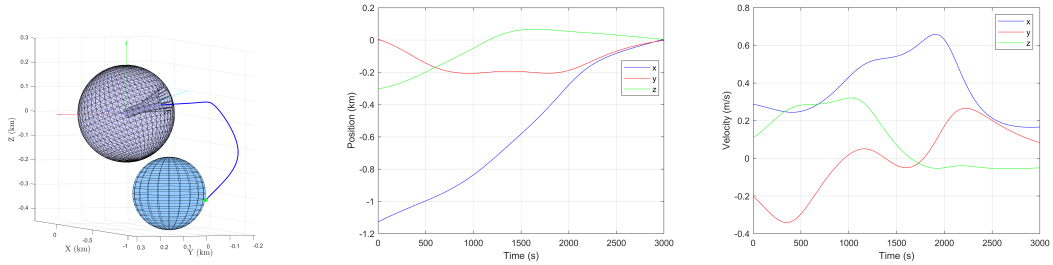
The task is to dock autonomously with a hypothetical docking port at the center of the relative reference frame avoiding the forbidden areas. Specifically, the agent must avoid collisions with a spherical constraint representing a non-cooperative object standing inside the approach sphere. Moreover, it has to approach the docking port through an approach corridor aligned with the axis of such port and avoid entering the keep-out zone. The non-cooperative object is at position $[-0.6, 0, -0.3]$ km and has a radius of 150 m. The keep-out zone is a sphere of radius $r = 200$ m centered on the target with an approach corridor that is conical with half-cone angle of $15\,\mathrm{deg}$ up to where it becomes cylindrical with a radius of 20 m. The axis of the docking port has a component in all directions to keep things as general as possible. See figures for reference.

The reward function is represented by the expression 26. It is composed by two terms related to the end position and velocity errors with respect to the nominal target state that are added only if the agent reaches the end time ($t = t_f$), and one term related to the position error of the impact point, if present, with respect to the target state.

$$R(t) = -\delta(t - t_f)\left[w_r^f \|r_t - r_f\|^2 + w_v^f \|v_t - v_f\|^2\right] - \delta(t - t_i)\left[w_r^i \|r_t - r_f\|^2\right] \tag{26}$$

Where w_r^f, w_v^f and w_r^i are weights associated with the burned mass, the end position and velocity errors and the impact point position error respectively.

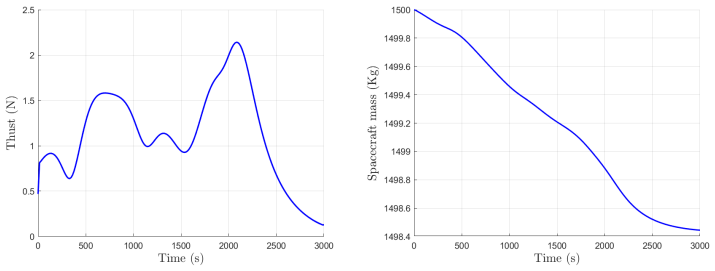
The policy rollouts are generated selecting the starting state from a uniform distribution around the nominal starting state on the Y-Z plane in the interval ± 0.15 km. The agent runs the policy until the final time of 3000 s is reached unless a collision with one of the boundaries is detected in which case the episode ends.



(a) Final trajectory in LVLH frame (b) Position in LVLH frame (c) Velocity in LVLH frame

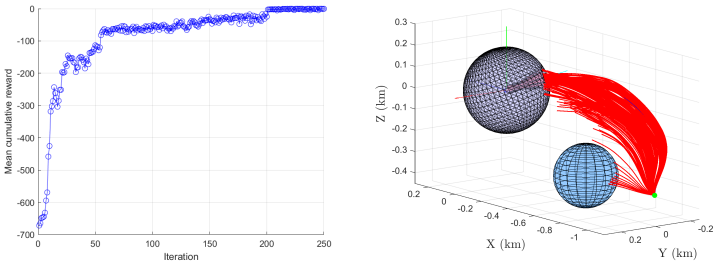
Figure 4. Final trajectory (periselene)

Figure 4 and Figure 8 show how the agent manages to find a solution that drives the spacecraft to the target avoiding collisions with the obstacles. Figure 6 and Figure 10 show the training per-



(a) Thrust profile (b) Mass profile

Figure 5. Thrust and Mass profiles (periselene). Consumed mass: 1.55 **kg**



(a) Cumulative average reward (b) Test trajectories

Figure 6. Training (periselene)

formances where the evolution of the test trajectories can be easily correlated to the behaviour of the average cumulative reward among rollouts. This clearly shows that the agent first learn how to avoid colliding with the non-cooperative object and then to converge towards the target through the approach corridor. The final position deviation for the periselene case is $[4.62, -2.81, 3.6]$ m while the final velocity deviation is $[0.16, 0.08, -0.05]$ m/s. The final position deviation for the aposelene is instead $[-4.42, 1.86, -5.12]$ m while the final velocity deviation is $[0.17, 0.11, -0.06]$ m/s.

It should be noted that at the present state, given the final state error, this type of guidance should be considered feasible for the final translation phase but not for proximity operations where a different control scheme could be used to finalize the docking maneuver. We are confident that better performances are obtainable with this architecture if more time is invested in tweaking the parameters of the learning algorithm and will be the subject of future development. The fuel consumed by the spacecraft during the entire mission is 1.15 kg and 1.55 kg at periselene and aposelene respectively as shown in Figure 5 and Figure 9.

Lastly, Figure 7 shows how the initial field gets modified by the agent during training in order to comply with the constraint and execute the task. The field is initialized as semi-random, with a overall direction pointing towards the direction of the target with random noise added in all directions. This allows for a slightly faster learning procedure without loosing in generality.

From a machine learning perspective, the proposed ELM-based actor-critic algorithm performs well, with good convergence capabilities. Table 1 shows some information about the performances of the method. It is interesting to note that the training time of the ELM accounts for not more than 5 % of the average iteration time. This along with their straightforward implementation shows that

ELM are a good alternative to classical neural networks.

Table 1. Critic network performance

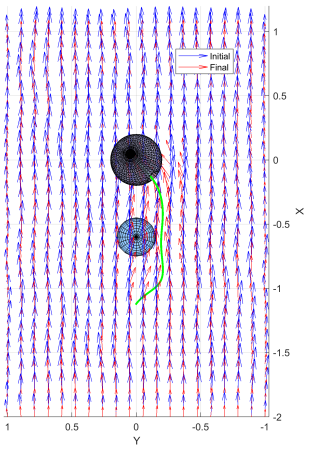
Model	N^o iterations	Total training time (s)	Average iteration time (s)	Average critic training time (s)
CW	250	2897	11.59	0.54
NLR	500	5608	16.02	0.64

CONCLUSION AND FUTURE WORK

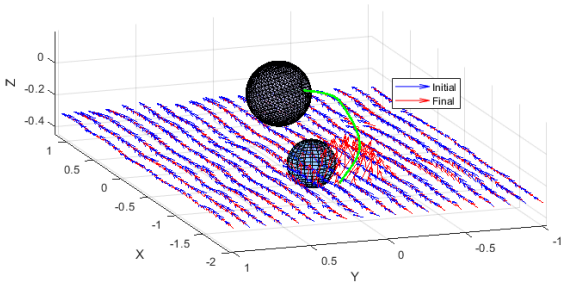
In this work we presented a closed-loop guidance algorithm with collision avoidance capabilities and the possibility to introduce path constraints. The trained agent is able in fact to reach the target with a good level of accuracy avoiding collisions with non-cooperative objects modeled as spheres and with a keep-out zone of arbitrary shape. The use of Lyapunov vector fields allows for a very flexible guidance algorithm that can be adapted to virtually any constraint scenario. Reinforcement learning is very effective at acquiring knowledge about the environment and shape the field accordingly. Future work may be focused on expanding its capabilities to comply with smaller-scale features in the constraints shapes such as relative navigation in the proximity of extended objects (i.e., space stations with appendages).

From a machine learning perspective, the ELM-based actor-critic algorithm shows good performances. Convergence is achieved with a relatively small iteration number. Moreover, ELM have not been used often as critic.[25] This work proves that they are effective at approximating the value function both in terms of accuracy and computation time. The critic training time is in fact generally small when compared to the other subroutines within a single global iteration (about 5 % of the cycle time).

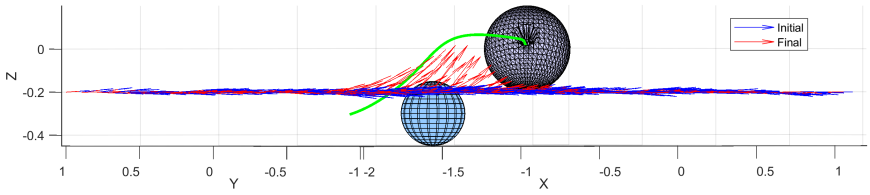
This work fits into the growing research field of autonomous guidance for docking and proximity operations, proving that reinforcement learning can be applied to a wide variety of tasks. This can be used as a basis for future works aimed at designing closed-loop guidance for autonomous operations. One of the crucial features of this method is the fact that it is model-free, which allows for its application to any environment. A model of the environment must still be present to provide the policy rollouts but, once the agent is trained, it doesn't need knowledge of it to provide the control command, paving the way for true autonomous relative motion guidance.



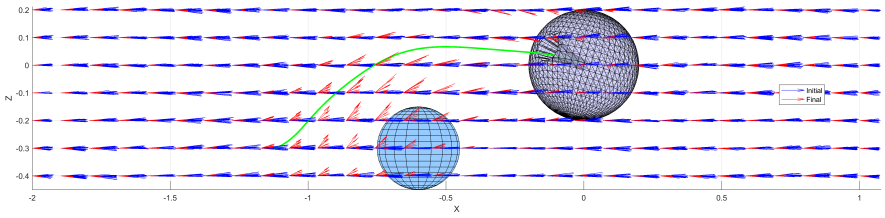
(a) Z = 0 km



(b) Z = -0.3 km

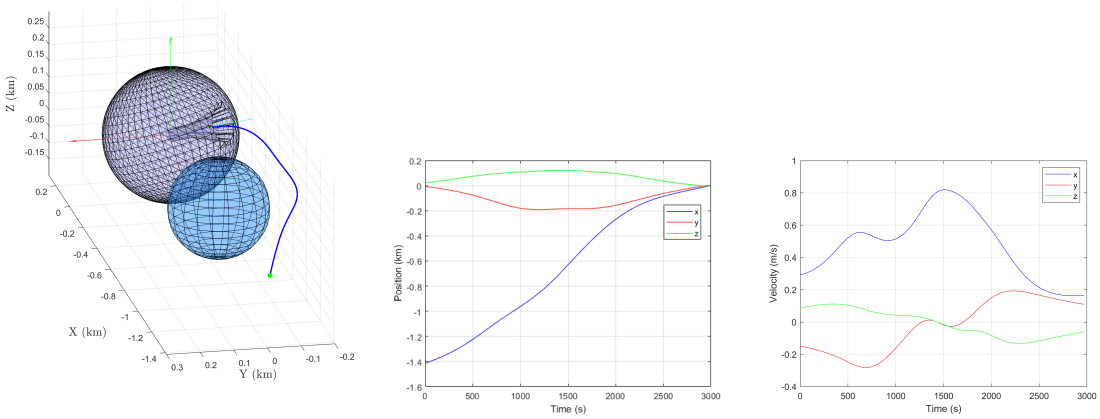


(c) Z = -0.2 km



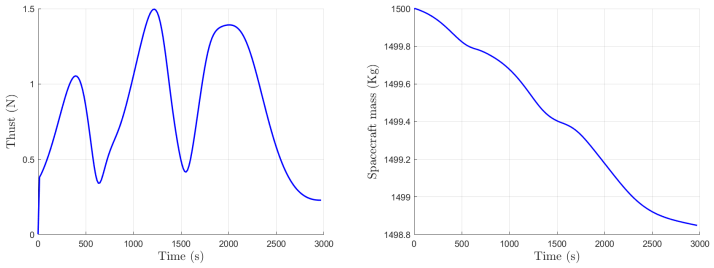
(d) Field

Figure 7. Some views of the initial and final velocity fields (periselene)



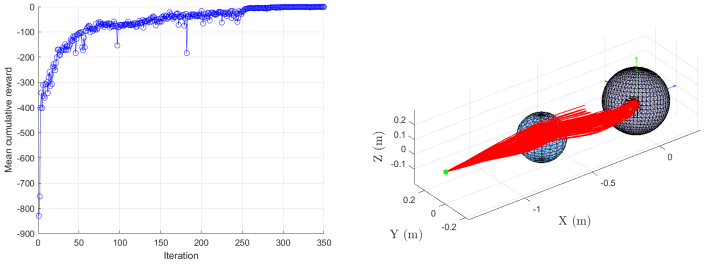
(a) Final trajectory in LVLH frame (b) Position in LVLH frame (c) Velocity in LVLH frame

Figure 8. Final trajectory (aposelene)



(a) Thrust profile (b) Mass profile

Figure 9. Thrust and Mass profiles (aposelene). Consumed mass: 1.15 **kg**



(a) Cumulative average reward (b) Test trajectories

Figure 10. Training (aposelene)

REFERENCES

[1] T. Gill, "NASA's Lunar Orbital Platform-Gatway," 2018.

[2] R. Whitley and R. Martinez, "Options for staging orbits in cislunar space," *Aerospace Conference, 2016 IEEE. IEEE*, 2016.

[3] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[4] N. Zhou and Y. Q. Xia, "Coordination Control Design for Formation Reconfiguration of Multiple Spacecraft," *IET Control Theory and Applications*, Vol. 9, No. 15, 2015, pp. 2222–2231.

[5] M. W. Li, H. J. Peng, and W. X. Zhong, "Optimal Control of Loose Spacecraft Formations near Libration Points with Collision Avoidance," *Nonlinear Dynamics*, Vol. 83, No. 4, 2016, pp. 2241–2261.

[6] L. Breger and J. P. How, "Safe Trajectories for Autonomous Rendezvous of Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1478–1489.

[7] P. Lu and X. F. Liu, "Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389.

[8] X. F. Liu and P. Lu, "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.

[9] A. B. Roger and C. R. McInnes, "Safety Constrained Free-Flyer Path Planning at the International Space Station," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 4, 2000, pp. 971–979.

[10] C. M. Saaj, V. Lappas, and V. Gazi, "Spacecraft Swarm Navigation and Control Using Artificial Potential Field and Sliding Mode Control," *IEEE International Conference on Industrial Technology, Inst. of Electrical and Electronics Engineers, New York*, 2006, pp. 2646–2651.

[11] R. M. N. S. E. C. M. McCamish, S. B. and D. W. Miller, "Flight Testing of Multiple-Spacecraft Control on SPHERES During Close-Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 46, No. 4, 2009, pp. 1202–1213.

[12] I. Lopez and C. R. McInnes, "Flight Testing of Multiple-Spacecraft Control on SPHERES During Close-Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 18, No. 2, 1995, pp. 237–241.

[13] S. S. Zhang, D. and R. Pei, "Flight Testing of Multiple-Spacecraft Control on SPHERES During Close-Proximity Operations," *AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2010-7592*, 2010.

[14] K. L. E. Phillips, J. M. and N. Bedrossian, "Spacecraft Rendezvous and Docking with Real-Time, Randomized Optimization," *AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2003-5511*, 2003, pp. 237–241.

[15] H. Dong, Q. Hu, and M. R. Akella, "Safety control for spacecraft autonomous rendezvous and docking under motion constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 7, 2017, pp. 1680–1692.

[16] H. B. Ammar, E. Eaton, P. Ruvolo, and M. Taylor, "Online multi-task learning for policy gradient methods.," *International Conference on Machine Learning (pp. 1206-1214)*, 2014.

[17] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," *ICML*, 2014.

[18] R. J. Williams, *Reinforcement learning*. Springer, 1992.

[19] G. B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey.," *International journal of machine learning and cybernetics, 2(2), 107-122.*, 2011.

[20] E. Cambria, G. B. Huang, L. L. C. Kasun, H. Zhou, C. M. Vong, J. Lin, and V. C. Leung, "Extreme learning machines [trends and controversies]," *IEEE Intelligent Systems, 28(6), 30-59*, 1999.

[21] G. B. Huang, "What are extreme learning machines? Filling the gap between Frank Rosenblatts dream and John von Neumanns puzzle," *Cognitive Computation 7.3*, 2015.

[22] R. Furfaro and R. Linares, "Waypoint-Based Generalized ZEM/ZEV Feedback Guidance for Planetary Landing via a Reinforcement Learning Approach," *3rd IAA Conference on Dynamics and Control of Space Systems, Moscow, Russia*, 2017.

[23] B. Gaudet and R. Furfaro, "Adaptive pinpoint and fuel efficient mars landing using reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, Vol. 1, No. 4, 2014, pp. 397–411.

[24] B. Gaudet, R. Linares, and R. Furfaro, "Deep reinforcement learning for six degree-of-freedom planetary powered descent and landing," *arXiv preprint arXiv:1810.08719*, 2018.

[25] A. Scorsoglio, R. Furfaro, R. Linares, and M. Massari, "Actor-Critic Reinforcement Learning Approach to Relative Motion Guidance in Near-Rectilinear Orbit," *In 29th AAS/AIAA Space Flight Mechanics Meeting*, 2019, pp. 1–20.

[26] D. A. Lawrence and a. P. W. J. Frew, E. W., "Lyapunov vector fields for autonomous unmanned aircraft flight control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1220–1229.

[27] C. R. McInnes, "Autonomous path planning for on-orbit servicing vehicles," *Journal of the British Interplanetary Society*, Vol. 53, No. 1/2, 2000, pp. 26–38.

[28] W. S. Koon, M. W. Lo, and J. E. Marsden, *Dynamical Systems, the Three-Body Problem and Space Mission Design*. 2011.

[29] D. J. Grebow, *Trajectory design in the Earth-Moon system and lunar South Pole coverage (Doctoral dissertation)*, 2010.

[30] T. A. Pavlak, *Mission design applications in the earth-moon system: Transfer trajectories and station-keeping. (MSAA Thesis)*, 2010.

[31] A. Campolo, *Safety Analysis for Near Rectilinear Orbit Close Approach Rendezvous in the Circular Restricted Three-Body Problem (MSAA Thesis)*, 2017.

[32] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvri, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," *Proceedings of the 26th Annual International Conference on Machine Learning. (pp. 993-1000) ACM*, 2009.

[33] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction (Vol. 1, No. 1)*. Cambridge: MIT press, 1998.

[34] J. Peters and S. Schaal, "Policy gradient methods for robotics," *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE*, 2006.

[35] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing 71.7-9*, 2008.

[36] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," *IEEE International Conference on. Vol. 4. IEEE*, 2002.