A. Scorsoglio, R. Furfaro, R. Linares, M. Massari
Relative Motion Guidance for Near-Rectilinear Lunar Orbits with Path Constraints via Actor-Critic Reinforcement Learning
Advances in Space Research, Published online 10/08/2022
doi:10.1016/j.asr.2022.08.002

Relative motion guidance for near-rectilinear lunar orbits with path constraints via actor-critic reinforcement learning

Andrea Scorsoglio[a,1,*], Roberto Furfaro[a,2], Richard Linares[b,3], Mauro Massari[c,4]

[a]*The University of Arizona, 1127 E. James E. Rogers Way, Tucson, AZ 85721, USA*
[b]*Massachusetts Institute of Technology, 77 Massachusetts Avenue Cambridge, MA 02139, USA*
[c]*Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy*

R

Abstract

This paper presents a feedback guidance algorithm for proximity operation in cislunar environment based on actor-critic reinforcement learning. The algorithm is lightweight, closed-loop, and capable of taking path constraints into account. The method relies on reinforcement learning to make the well known Zero-Effort-Miss/Zero-Effort-Velocity guidance state dependent and allow for path constraints to be directly embedded. The algorithm is tested in the circular restricted three-body problem (CRTBP) framework for Near Rectilinear Orbits (NRO) in the Earth-Moon system. It shows promising results in terminal guidance error and satisfies path constraints in constraint scenarios comprising spherical constraints and keep-out-spheres with approach corridors. Furthermore, this approach indicates that reinforcement learning can be effectively used to solve constrained relative spacecraft guidance problems in complex environments and thus can be effective for autonomous relative motion operations in the Earth-Moon dynamical environment.

Keywords: Near Rectilinear Orbits ; Spacecraft Guidance ; Machine Learning ; Reinforcement Learning ; Actor-Critic

1. Introduction

Accurate feedback guidance algorithms have always been of utmost importance for space exploration. Lately, Lagrangian points have gained much attention due to future missions using the advantageous position of these particular points. Many successful missions in lagrangian points include the solar wind monitoring probes (ACE, SOHO, DSCVR, WIND) positioned in the L_1 Earth-Sun Lagrangian point. More recently, James Webb Telescope has been launched, directed towards a halo orbit around the L_2 Earth-Sun Lagrangian point (Lightsey et al., 2012). Moreover, with the Lunar Orbital Platform-Gateway (LOP-G) (Gill, 2018) set to become the new establishment for human exploration of the solar system, relative dynamics guidance in the cislunar

*Corresponding author: Tel.: +1-520-312-7440
Email addresses: andreascorsoglio@email.arizona.edu (Andrea Scorsoglio), robertof@email.arizona.edu (Roberto Furfaro), linaresr@mit.edu (Richard Linares), mauro.massari@polimi.it (Mauro Massari)
[1]Ph.D. Student, Department of System and Industrial Engineering
[2]Professor, Department of System and Industrial Engineering, Department of Aerospace and Mechanical Engineering
[3]Professor, Department of Aeronautics and Astronautics
[4]Professor, Dipartimento di Scienze e Tecnologie Aerospaziali

environment will be of pivotal importance in the near future. By introducing the Artemis program, NASA has made clear that in the next decade, the Moon will be one of the primary objectives for space exploration, both for its scientific value and as a proving ground for further advancements in human exploration (i.e. Mars). In this context, the LOP-G will serve NASA and its commercial and international partners as a valuable staging point and telecommunications relay for exploration and science missions in deep space. Near Rectilinear Halo Orbits (NRHO or NRO) in the Earth-Moon three-body framework are considered the most favorable environment for this kind of mission. A critical study by NASA (Whitley & Martinez, 2016), has shown some advantages of using these kinds of orbits over different cislunar orbits. Their particular shape allows continuous coverage of either side of the Moon while being continuously visible from Earth. Moreover, they are advantageous in terms of ΔV for transfer to and from Earth and the lunar surface. Indeed, the same study shows that they are within the launching capabilities of an SLS-Orion mission. Finally, such orbits exhibit a small ΔV requirement for station-keeping while maintaining favorable thermal characteristics. Many of the operations in the cislunar environment will rely on precise relative guidance. Historically, guidance algorithms for this kind of problem have almost always relied on open-loop architectures that are either defined beforehand on the ground or are dependent on direct human intervention, e.g., manned missions. Examples of nearly automated docking procedures are ESA's ATV (Pinard et al., 2007) and Roscosmos' Progress (Zimpfer et al., 2005). Although the deployed methods are demonstrated to work well for docking to the ISS, i.e., in Low Earth Orbit (LEO), there is no assurance that they would work in a cislunar environment. For this reason, the rendezvous problem in CRTBP had to be redefined from the ground up. Although an initial study on the rendezvous problem in cislunar orbits has been recently presented (Campolo, 2017), there is currently little literature on the guidance and control side of the problem. For example, free drift trajectories and invariant manifolds were used to prove the feasibility of a multiple impulse guidance (Ueda & Murakami, 2015), while non-linear optimal control algorithm and surrogate-based parameter optimization method were proposed to solve the rendezvous problem between different halo orbits (Peng et al., 2013). A constant-thrust glideslope algorithm for halo rendezvous is also present in the literature (Lian et al., 2012): it achieves good performances in terms of final velocity but the errors in the final position are in the order of meters. Finally, a work by NASA (Williams et al., 2017) shows an approach based on forward and backward shooting methods combined to create ballistic trajectories in the CRTBP environment and introduces the principal guidelines for relative approach, eclipse avoidance, and end of life operations which will be considered for this paper.

When considering closed-loop guidance algorithms, Zero-Effort-Miss/Zero-Effort-Velocity (ZEM/ZEV) feedback guidance has been applied to a variety of problems, from soft landing to intercept and rendezvous (Zhang et al., 2017; Guo et al., 2011, 2013; Furfaro et al., 2018). The ZEM/ZEV feedback guidance is attractive because of its analytical simplicity and accuracy in real-world scenarios and its ease of implementation. Guidance mechanization is straightforward, and it can theoretically drive the spacecraft to a target autonomously and with minimal guidance errors, regardless of the environmental dynamics. Moreover, it is globally finite-time stable and robust to perturbations and uncertainties in the model if supported by a sliding parameter (Optimal Sliding Guidance) (Furfaro & Wibben, 2016; Wibben & Furfaro, 2016) . One of the biggest strengths resides in its closed-loop nature. Indeed, the algorithm usually computes online the desired acceleration as a function of the current spacecraft state (i.e., position and velocity, generally provided by the navigation system). The latter is an enabler for autonomy because there is no need to integrate ground operations in the control loop. Nevertheless, the algorithm has two significant limitations, i.e., 1) it solves the guidance problem optimally only in cases where the gravity field and the acceleration components, in general, are constant or solely dependent on time, and 2) the algorithm does not take path constraints into account. The latter represents a severe limitation for the classical ZEM/ZEV, especially when implemented in environments where the guidance algorithm must specify strict path

constraints. Indeed, the classical algorithm is generally not suitable for relative motion operations and docking that usually requires path constraints to be enforced. Here, we propose a new algorithm that retains the strengths of classical ZEM/ZEV and overcomes its above-mentioned significant limitations by using machine learning techniques.

Reinforcement learning (RL) has grown in importance in recent years thanks to the advancements in computing power. It has been shown to work well in many robotic motion tasks (Kober et al., 2013; Grondman et al., 2012; Nakanishi et al., 2004; Peters & Schaal, 2006, 2008; Smart & Kaelbling, 2002) , yet its use has not been sufficiently explored for closed-loop spacecraft guidance. For example, Furfaro and Linares (Furfaro & Linares, 2017) show that RL has been recently employed to select the optimal sequence of waypoints in a waypoint-based ZEM/ZEV algorithm for planetary landing. Recent papers have shown interest in both Deep RL and Deep Meta-RL. Topics include path planning for asteroid hopping rovers (Jiang et al., 2020), planetary landing guidance (Gaudet et al., 2020c,b), small bodies proximity operations (Gaudet et al., 2020d) and intercept guidance (Gaudet et al., 2020a). Gaudet, Linares and Furfaro (Gaudet et al., 2020c,b,d) have shown good performances of a meta-reinforcement learning algorithm based on Proximal Policy Optimization (PPO) applied to a 6-DOF Mars pinpoint landing with uncertain dynamics and an asteroid hovering problem with flash LIDAR observations. Moreover, Scorsoglio et al. (Scorsoglio et al., 2022) demonstrated good performance of a similar algorithm in a lunar landing scenario with uncertain dynamics and actuator failure using sequences of images as inputs. Moreover, RL has been successfully applied to the problem of docking with rotating and non-rotating targets (Oestreich et al., 2021), even in presence of simple spherical obstacles (Hovell & Ulrich, 2021). The latter also proved the capabilities of the proposed method on a physical test-bed.

This paper aims to propose a guidance algorithm capable of operating in the more complex non-keplerian environment in presence of more complex constraints when compared to previous works. Reinforcement learning (Ammar et al., 2014; Silver et al., 2014; Williams, 1992) and extreme learning machines (Huang et al., 2011; Cambria et al., 1999; Huang, 2015) are used to create a zero-effort-miss/zero-effort-velocity (ZEM/ZEV) (Guo et al., 2013) based closed-loop algorithm (Scorsoglio, 2018; Scorsoglio et al., 2019), that is able to solve this kind of guidance problems. This kind of customized actor-critic algorithm has been formulated and tested by our team in a planetary landing problem (Furfaro et al., 2020). This paper extends its capabilities to the much more complex problem of relative motion in NROs with path constraints. Specifically, we demonstrate the capabilities of the method on a rendezvous problem with two constraint scenarios: one with two spherical keep-out zones in the vicinity of the target point (i.e., simulating appendages or third bodies in the approach area), and one with a keep-out sphere with a conical approach corridor centered on the target. These constraint scenarios are inspired by previous work in the field of optimal path planning for spacecraft docking (Zappulla et al., 2018; Dong et al., 2017; Cui et al., 2017), although none of them produce a closed-loop guidance or make use of RL. The peculiar dynamical environment of NROs and the proximity operation constraint scenarios pose new challenges to the RL method, which requires expanding its capabilities beyond its initial application. As shown in (Campolo, 2017), the dynamics of NROs can be described using both linearized equations of motion in the vicinity of the Moon and full non-linear relative equations of motion when far from the Moon, which poses new design challenges related to choosing the correct dynamical model depending on the position on the orbit. It also increases the overall difficulty from the agent perspective, as thrust commands in a specific direction do not necessarily cause motion in the same direction, and might produce effects far into the future, which are difficult to account for. Additionally, the slow dynamics of the problem, especially in the region close to the aposelene, makes it challenging for the agent to produce a control command high enough to effectively avoid violating the constraints. Moreover, the complex path constraints pose new challenges that demand a new definition of the reward function, as well as extensive parameter tuning, to function properly. The result is a closed-loop guidance algorithm capable of driving

a spacecraft to the target in the presence of complex path constraints, with minimal final guidance error, and regardless of the dynamical model used to train it.

The remainder of this paper is organized as follows. Section 2 introduces the problem of rendezvous in lunar NRO and the approach used to describe the relative motion between target and chaser. Section 3 describes the guidance optimization method. In Section 4 the results for two different constraint scenarios are presented. Finally, Section 5 is dedicated to the conclusions and final thoughts.

2. Background

2.1. Circular restricted three-body problem and NROs

The particle's motion in the presence of two main bodies with masses m_1 and m_2 where the only mean of interaction between the particles is the gravitational attraction is generally described by the Circular Restricted Three-Body Problem (CRTBP). In this framework, the primaries are considered orbiting around the system's center of mass in circular orbits. The dynamics of the problem are expressed in the absolute synodic reference frame, which in the case of the Earth-Moon system will be called $\mathcal{R}_{em}$. The origin of this frame is positioned 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. The only parameter governing the dynamics of the system is the mass parameter

$$\mu = \frac{m_2}{m_1 + m_2} \tag{1}$$

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

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

with

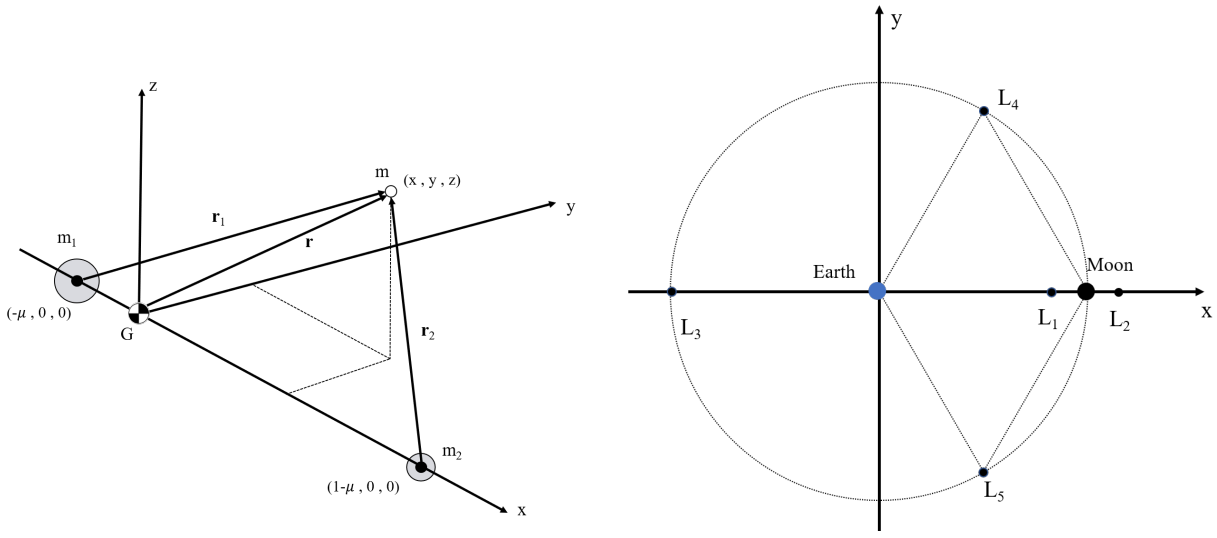
$$\begin{aligned} r_1 &= \sqrt{(x+\mu)^2 + y^2 + z^2} \\ r_2 &= \sqrt{(x-(1-\mu))^2 + y^2 + z^2} \end{aligned} \tag{3}$$

A more comprehensive study on the problem and the procedure to derive the equations of motion can be found in the references (Koon et al., 2011).

2.1.1. Equilibrium solutions

Equations (2) do not have a closed-form analytical solution. However, it is possible to determine the location of equilibrium points of the CRTBP. The equilibrium points, also named lagrangian points, or libration points, are stationary points of the potential function U defined as:

$$U = \frac{1}{2}\left(x^2 + y^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{4}$$



(a) Synodic frame (b) Lagrangian points

Fig. 1. Synodic frame and lagrangian points

and are the solutions of the equation:

$$\nabla U = 0 \tag{5}$$

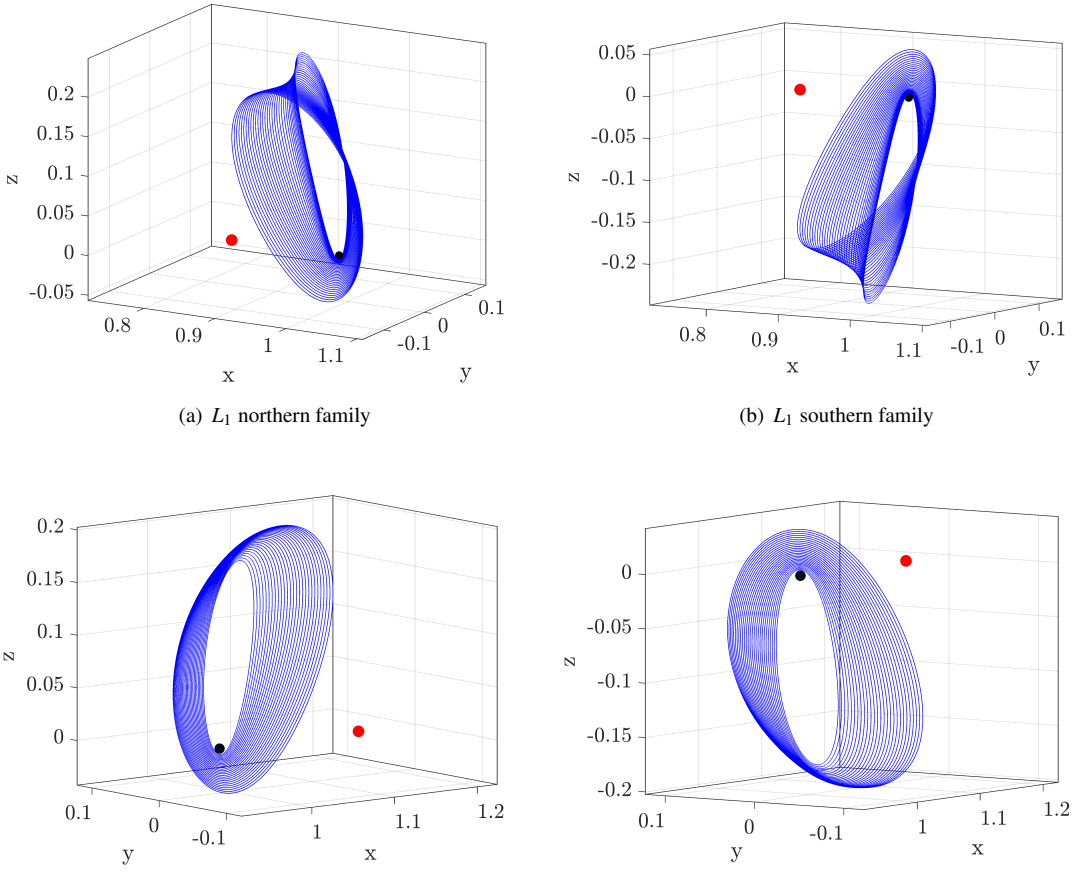
The equilibrium points are locations in which the secondary mass m would appear motionless in the rotating synodic frame. A qualitative representation of the lagrangian points in the Earth-Moon system expressed in the synodic reference frame is given in Figure 1.

2.1.2. Near Rectilinear Orbits

In the CRTBP framework, a wide variety of trajectories result in a periodical motion. They can be divided into two main groups: in-plane and out-of-plane orbits. Near Rectilinear Orbits, or NROs, belong to the second group. More specifically, they are a degenerate subset of Halo 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. Closed trajectories were found using a single shooting algorithm based on a multi-variable newton method. The process used is described thoroughly in the references (Scorsoglio, 2018; Grebow, 2010; Pavlak, 2010). A representation of all the NROs families that were considered for this study can be seen in Figure 2. Specifically, we used orbits from the southern L2 NRO family.

2.2. Rendezvous in NRO

As mentioned earlier, the overall goal is to design and demonstrate a closed-loop algorithm capable of performing rendezvous and docking in the context of cislunar NRO. The operations guidelines for this kind of mission have been formalized by Campolo (Campolo, 2017) . They are divided into two segments, i.e., 1) a ”far approach” phase, starting at the departure of the chaser from the phasing orbit and ending at the beginning of robust relative navigation and 2) a ”close approach” phase, starting at the end of the first phase and ending with docking. Noting 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 gravitational attraction from the central body are no longer valid. Therefore, the new regulations define four areas around the target related to different phases of the rendezvous procedure:



(a) L_1 northern family

(b) L_1 southern family

(c) L_2 northern family

(d) L_2 southern family

Fig. 2. NRO families. The red dot is the lagrangian point, the black dot is the Moon

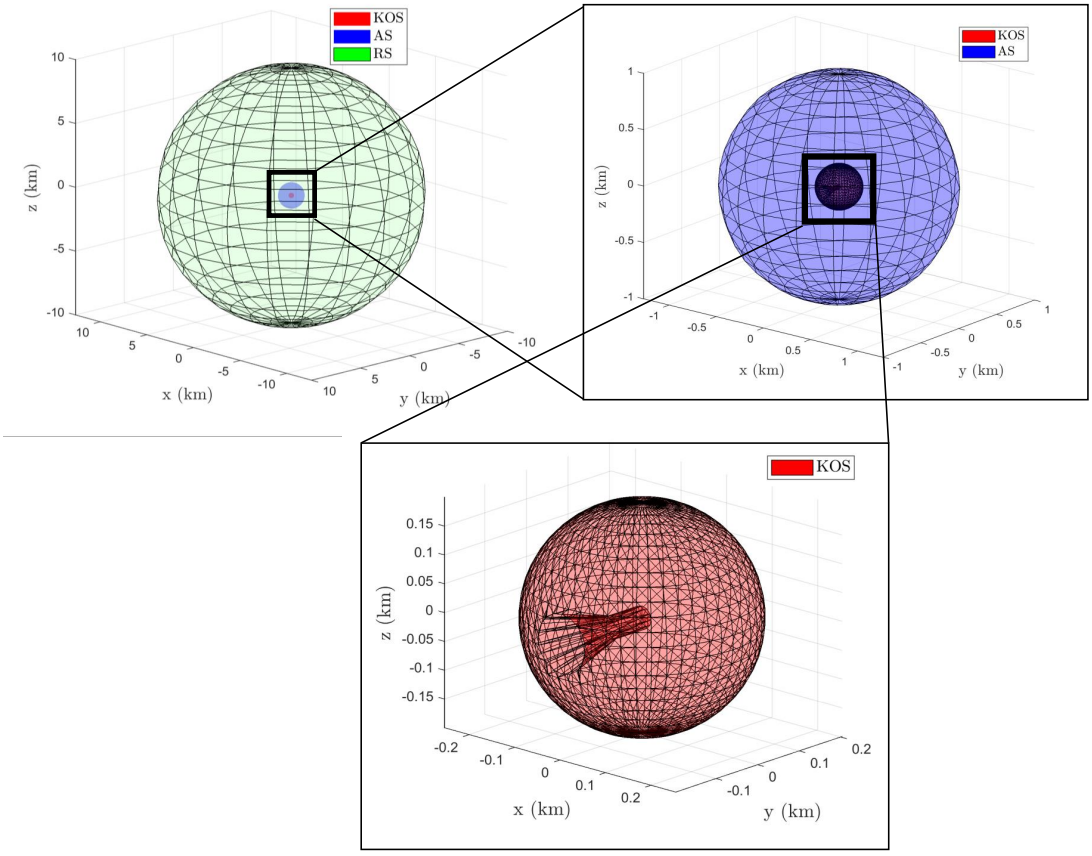


Fig. 3. Rendezvous areas

- **Keep-Out Sphere (KOS)**: a sphere of 200 meters of radius centered on the target's center of mass
- **Approach/Departure corridors**: shapes defined within the KOS. They are defined according to the target spacecraft shape. In the case of the ISS, they are conical with a half cone angle of 15°.
- **Approach Sphere (AS)**: a 1 *km* radius sphere centered on the target's center of mass. The Approach Initiation (AI) burn is the first burn allowed to target within the AS. Integrated operations must begin before the chaser is on a trajectory that would enter the AS.
- **Rendezvous Sphere (RS)**: a 10 *km* radius sphere centered on the target's center of mass and is used to govern the Rendezvous Orbit Entry (ROE) decision.

In Figure 3 the areas defined above are shown. Here, the focus is on the close approach part of the problem. It is assumed that the most critical aspect is the one related to precision guidance inside the AS, so this is the environment in which the algorithm is trained and tested. Eqs (2) describe the motion of the chaser and target. In the non-dimensional synodic reference frame. These equations, however, are not feasible for describing the relative guidance and control problem, so the introduction of relative reference frames and relative dynamics equations is necessary.

2.3. NRO relative motion

The motion of the chaser, as seen from the target-centered reference frame, is defined as relative motion. In the three-body environment and for NROs in particular, relative motion has not been studied as extensively as for LEOs. Campolo proposed an

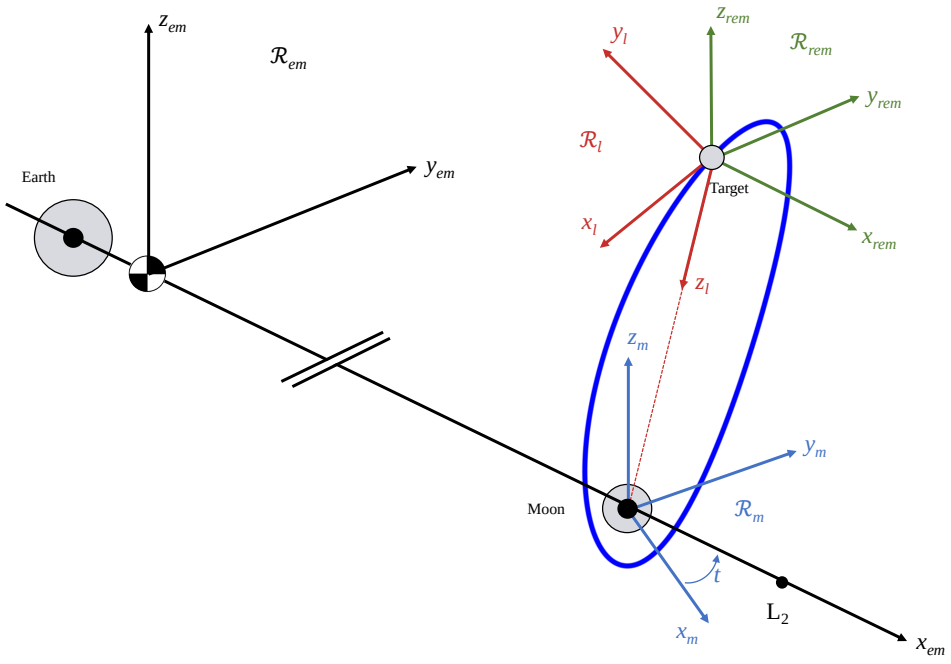


Fig. 4. Reference systems

interesting solution (Campolo, 2017) that will be summarized in this section.

2.3.1. *Reference frames*

The absolute dynamics in the Earth-Moon CRTBP are developed in the absolute synodic non-dimensional frame $\mathcal{R}_{em}$. However, the description of rendezvous dynamics is usually done using a reference frame relative to the target. In the case of two-body dynamics, this is generally the Local-Vertical-Local-Horizon frame (LVLH). The LVLH frame ($\mathcal{R}_l$) has also been defined for NROs (Campolo, 2017), with the z-axis directed towards the center of the Moon, the x-axis parallel to the velocity, and the x-axis completing the orthonormal triad. Although the dynamical environment is different, it has been demonstrated that the short-term NRO dynamics can be described in an LVLH frame defined with respect to a Moon Centered Inertial (MCI) frame ($\mathcal{R}_m$) defined exactly as in the two-body case. An additional reference frame is used in this paper, the Earth-Moon relative synodic (EMRS) frame ($\mathcal{R}_{rem}$), 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 (Campolo, 2017) . A representation of all the reference systems on a sample NRO can be see in Figure 4.

2.3.2. *Relative equations of motion*

The relative motion in NROs can be described using two models depending on the spacecraft's position along the orbit. It has been shown (Campolo, 2017) that in portions of the orbit where the gravitational influence of the Moon is strong, i.e., close to the periselene (the closest point to the Moon), the problem dynamically resembles the two-body problem, hence the *Clohessy-Wiltshire equation* (CW) expressed in the LVLH frame defined above can be employed with a minor error. However, in other regions of the orbit, the *Non-Linear Relative equations* (NLR) defined in the relative synodic reference system (EMRS) must be employed instead.

- The Clohessy-Wiltshire equations (**CW**) are a well known 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 this familiar form:

$$\begin{aligned} \ddot{x} - 2n\dot{z} &= 0 \\ \ddot{y} + n^2 y &= 0 \\ \ddot{z} + 2n\dot{x} - 3n^2 z &= 0 \end{aligned} \tag{6}$$

where

$$n = \sqrt{\frac{\mu}{r_{2T}^3}} \tag{7}$$

Where r_{2T} is the distance from the center of the second primary (Moon in this case), and μ is the gravitational parameter associated with the same object.

- 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. Considering the synodic relative state:

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

and position:

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

and the target and chaser absolute positions:

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

and the absolute non-dimensional distances of the target from the Earth and the Moon.

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

The equations of motion are:

$$\begin{aligned} \ddot{x} - 2\dot{y} - x &= (1-\mu)\left[\frac{x_T + \mu}{\|r_{1T}\|^3} - \frac{x_T + x + \mu}{\|r_{1T} + \rho\|^3}\right] + \mu\left[\frac{x_T + \mu - 1}{\|r_{2T}\|^3} - \frac{x_T + x + \mu - 1}{\|r_{2T} + \rho\|^3}\right] \\ \ddot{y} + 2\dot{x} - y &= (1-\mu)\left[\frac{y_T}{\|r_{1T}\|^3} - \frac{y_T + y}{\|r_{1T} + \rho\|^3}\right] + \mu\left[\frac{y_T}{\|r_{2T}\|^3} - \frac{y_T + y}{\|r_{2T}\rho\|^3}\right] \\ \ddot{z} &= (1-\mu)\left[\frac{z_T}{\|r_{1T}\|^3} - \frac{z_T + z}{\|r_{1T} + \rho\|^3}\right] + \mu\left[\frac{z_T}{\|r_{2T}\|^3} - \frac{z_T + z}{\|r_{2T}\rho\|^3}\right] \end{aligned} \tag{12}$$

They can be used in any region of the NRO, being 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 (i.e., the furthest point on the orbit with respect to the Moon).

3. State-dependent ZEM/ZEV

The guidance algorithm used here is derived from the adaptive ZEM/ZEV (A-ZEM/ZEV(Furfaro et al., 2020)), developed starting from the *classical ZEM/ZEV*(Guo et al., 2013, 2011; Zhang et al., 2017; Hawkins et al., 2012; Furfaro & Wibben, 2016; Wibben & Furfaro, 2016) feedback guidance algorithm, which is a particular kind of closed-loop guidance law based on the definition of two errors, zero-effort-miss (ZEM) and the zero-effort-velocity (ZEV) and will be here summarized. Considering a mission from time t_0 to t_f, the optimal control acceleration $\mathbf{a}$ is the solution that minimizes the performance index:

$$J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{a}^{\mathbf{T}}\mathbf{a} \quad \mathrm{dt} \tag{13}$$

for a body subjected to the following general dynamic equations, valid even for non-inertial systems:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a} + \mathbf{f}(\mathbf{r}, \mathbf{v}) \\ \mathbf{a} &= \mathbf{T}/m \end{aligned} \tag{14}$$

with $\mathbf{r}$, $\mathbf{v}$, $\mathbf{T}$ and $\mathbf{a}$ position, velocity, thrust and acceleration command vectors respectively and $\mathbf{f}(\mathbf{r}, \mathbf{v})$ being the generalized acceleration terms in which the gravitational and non-inertial acceleration contributions are present, with the following given boundary conditions:

$$\mathbf{r}(t_0) = \mathbf{r_0}, \qquad \mathbf{r}(t_f) = \mathbf{r_f} \tag{15}$$

$$\mathbf{v}(t_0) = \mathbf{v_0}, \qquad \mathbf{v}(t_f) = \mathbf{v_f} \tag{16}$$

The guidance law, assuming this is a problem for which $\mathbf{f}(\mathbf{r}, \mathbf{v}) = \mathbf{g}(t)$, is obtained solving the associated two point boundary value problem as:

$$\mathbf{a} = \frac{6}{t_{go}^2}\mathbf{ZEM} - \frac{2}{t_{go}}\mathbf{ZEV} \tag{17}$$

Where the ZEM and ZEV errors are defined respectively as the difference between the desired final position and velocity and the projected final position and velocity if no additional control is commanded from time t onward and can be computed analytically (see reference (Guo et al., 2013, 2011; Zhang et al., 2017; Hawkins et al., 2012; Furfaro & Wibben, 2016) for details).

In any other case in which $\mathbf{f}(\mathbf{r}, \mathbf{v}) \neq \mathbf{g}(t)$, as it is in this paper, the control law is still usable, but it will not be necessarily optimal, and ZEM and ZEV must be defined differently. The projected position and velocity cannot be recovered analytically: they must be obtained through an integration of the equations of motion from the current time instant to the end of the mission with control actions set to zero:

$$\begin{aligned} \mathbf{ZEM} &= \mathbf{r}_f - \mathbf{r}_{nc} \\ \mathbf{ZEV} &= \mathbf{v}_f - \mathbf{v}_{nc} \end{aligned} \tag{18}$$

where $\mathbf{r}_{nc}$ and $\mathbf{v}_{nc}$ are, respectively, the position and velocity at the end of mission if no control action is given from the considered time onward. It should be noted that using the formulation in (17), which will be called *Classical-ZEM/ZEV* from now on, can

result in valid trajectories even for cases when the generalized acceleration term is arbitrary. In these types of environment however, using a definition of ZEM and ZEV as in (18), the control gains that solve the optimal problem are no longer the ones in (17). This leads to the definition of the *Generalized-ZEM/ZEV* algorithm (Guo et al., 2013):

$$\mathbf{a} = \frac{K_R}{t_{go}^2}\mathbf{ZEM} + \frac{K_V}{t_{go}}\mathbf{ZEV} \tag{19}$$

where K_R and K_V are arbitrary. The non-linear acceleration components in the relative equations of motion of the problem under investigation in this paper justify the use of this generalized form of the guidance algorithm. The fundamental idea behind A-ZEM/ZEV is to use reinforcement learning (RL) to learn the parameters K_R and K_V as function of the state in order to enforce path constraints.

3.1. Reinforcement learning

Learning is achieved via an actor-critic (A-C) policy gradient algorithm that was developed for this paper based on previous works (Scorsoglio, 2018; Scorsoglio et al., 2019) starting from the REINFORCE algorithm (Williams, 1992) introducing a critic network based on Extreme Learning Machines (ELM) for estimating the value function. Literature on actor-critic algorithms and reinforcement learning, in general, is extensive, so in the following, we will focus on the explanation of this specific algorithm rather than the basic concepts of reinforcement learning. Suffice to say that an actor-critic algorithm is generally based on an agent (the spacecraft) that interacts with an environment (relative dynamics NRO environment) using a parametric policy $\pi_\theta(u|x)$ depending on state x and action u and is assigned rewards (or costs) depending on the actions it takes. The actor's goal is to update the policy in a way that maximizes (or minimizes) the objective function

$$J(\pi_\theta) = \mathbb{E}\left[r(x,u)\right] \tag{20}$$

which is the expectation of the return $r(x,u)$, which is, in turn, a function of the reward (or cost). Policy gradient algorithms optimize the policy by adjusting its parameters in the direction of the gradient

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(u|x) Q^\pi(x,u)\right] \tag{21}$$

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. Compounding this gradient involves an expectation that is cumbersome to compute exactly, especially with continuous state and action spaces. In stochastic policy gradient, this is substituted by a sample-based approximation that we will later discuss. Moreover, the introduction of a critic network allows for the approximation of $Q^\pi(x,u)$ to be used instead of its real counterpart, further reducing the complexity of the optimization task. The algorithm can be broken down into three blocks that are run sequentially at each global iteration: sample generation, critic neural network fitting, and policy update.

3.2. Samples generation

At each global iteration, a batch of trajectories is generated by letting the agent interact with the environment using policy $\pi_\theta(u|x)$, which represents the guidance gains in equation (19), creating a series of samples $(x_{i,t}, u_{i,t}, c_{i,t}, x_{i,t+1})$, where $x_{i,t}$ is the state at time-step t, $u_{i,t}$ is the action at time-step t, $c_{i,t}$ is the cost associated to time-step t and $x_{i,t+1}$ is the next state. The initial state is randomly chosen by sampling a uniform distribution around the nominal one. This allows the network to learn on a larger state space as it will be discussed in Section 4. The time is discretized in a fixed number of time-steps: at the beginning of each time-step,

the policy is sampled and K_R and K_V obtained, the acceleration command calculated with Eq. (19), and the equations of motion integrated for the length of a time-step. The acceleration command is kept constant during the time interval. A cost is assigned at each time-step depending on the final state and the mass burned (more details in Section 4). The agent runs until the final time is reached unless a violation of the constraint is detected in which case the episode ends.

3.2.1. Policy

The policy is described by a gaussian distribution with fixed variance, representing the guidance gains. The policy must be stochastic for the learning to be successful because it ensures exploration of the action space. The agent learns from repeated interaction with the environment and has to try different actions before figuring out the best way of interacting with said environment. However, it should be clear that the stochasticity of the policy is introduced only to implement exploration and because the machinery developed for stochastic policy gradient can then be applied. The version of the policy used to test the algorithm that could then be implemented in practice is its deterministic version, represented by the mean of the above-mentioned gaussian policy. For this reason, only the mean of the policy is learned. The policy is divided into two separate sections, each dependent on a different vector of parameters (θ_{K_R} and θ_{K_V}) related to the two guidance gains to learn K_R and K_V. The policy can be formally expressed as:

$$\begin{aligned} K_R &= \pi_{\theta_{K_R}} = \mathcal{N}(\mu_{K_R}, \sigma^2) \\ K_V &= \pi_{\theta_{K_V}} = \mathcal{N}(\mu_{K_V}, \sigma^2) \end{aligned} \tag{22}$$

where:

$$\begin{aligned} \mu_{K_R} &= \phi(\mathbf{x})^T \theta_{K_R} \\ \mu_{K_V} &= \phi(\mathbf{x})^T \theta_{K_V} \end{aligned} \tag{23}$$

σ^2 is the variance of the distribution, $\phi(\mathbf{x})$ is the vector of feature functions evaluated at state $\mathbf{x}$ and θ_{K_R} and θ_{K_V} are the weight vectors associated with each output.

3.2.2. Features

The features are two collections of three-dimensional radial basis functions (RBF) with centers distributed across the position and velocity spaces. They are represented by the expressions:

$$\begin{aligned} \phi(\mathbf{r}) &= e^{-\beta_r \|\mathbf{r}-\mathbf{c_r}\|^2} \\ \phi(\mathbf{v}) &= e^{-\beta_v \|\mathbf{v}-\mathbf{c_v}\|^2} \end{aligned} \tag{24}$$

With β_R and β_V being constant parameters related to the variance of the radial functions, which is set according to the distance of the centers, $\mathbf{r}$ and $\mathbf{v}$ being respectively the position and velocity and $\mathbf{c_r}$ and $\mathbf{c_v}$ the centers of the RBFs. The centers are generated by dividing the state space of the problem in a set of intervals, creating a grid of equally spaced points in the position and velocity spaces. The deterministic part of this policy can be seen as a neural network with two three-dimensional inputs ($\mathbf{r}$,$\mathbf{v}$), a single hidden layer of neurons with radial basis activation functions, and a two-dimensional output layer (K_R and K_V). A representation can be seen in Figure 5.

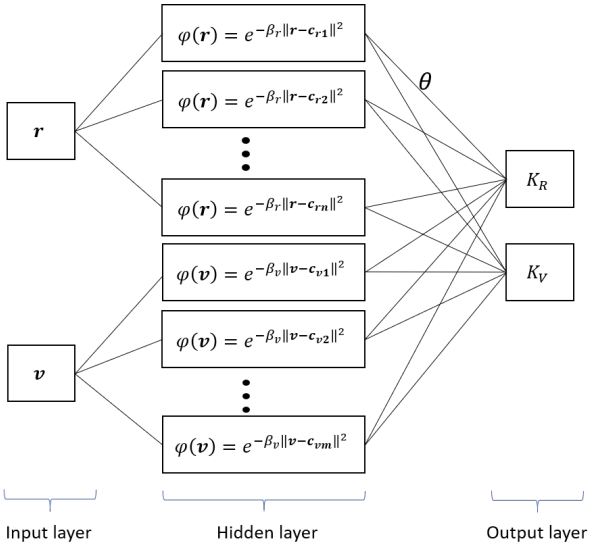


Fig. 5. Policy neural network

3.3. *Critic neural network*

For the algorithm to learn efficiently, a second neural network is introduced called critic. In actor-critic algorithms based on stochastic policy gradient, the expectation in the definition of the gradient of the performance index is not computed exactly, it is obtained using an approximated action-value function $Q^w(x,u)$. It has actually been shown that it is better to think of $Q^w(x,u)$ as an approximation of the advantage function $A^\pi(x,u) = Q^\pi(x,u) - V^\pi(x)$ rather than $Q^\pi(x,u)$. The approximated advantage function can be rewritten, using the definition of Q, as a function of V only:

$$Q^w(x,u) = \hat{A}^\pi(u,x) = \hat{Q}^\pi(x,u) - \hat{V}^\pi(x) = r(x,u) + \hat{V}^\pi(x_{t+1}) - \hat{V}^\pi(x) \tag{25}$$

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 $V^\pi(x)$. This shows that, in order to compute the approximated advantage function, $\hat{V}^\pi(x)$ is the only quantity that needs to be estimated. This is done using an Extreme Learning Machine (ELM(Huang et al., 2006)) with a *sigmoid* activation function. ELMs are a particular kind of single-layer feed-forward network in which the input weights and biases are assigned at random, and the only learned parameters are the output weights. This allows for learning in a single step via least square methods, which is very quick for small training sets as in this case. The ELM is used to map the 6D state into the scalar representing the discounted cost. 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 cost-to-go. So the training set is represented by the couples:

$$\left\{\left(x_{i,t}, \sum_{t'=t}^{T} \gamma^{t'-t} c(x_{i,t'}, u_{i,t'})\right)\right\} \tag{26}$$

Choosing the number of neurons is generally challenging when designing neural networks. It is important to remember that since the number of samples in a trajectory is not known a priori in this case, the number of training samples fed to the critic network is also not known a priori, so it is impossible to optimize the network size in advance. This being said, ELMs are known for being very stable at a wide range of hidden nodes, as demonstrated in the original paper and some successive studies (Huang et al., 2011, 2006, 2015; Wang & Huang, 2005). When tested on a variety of datasets of very different sizes, a common finding is that the

optimal number of neurons spans between 1/4 and 1/66 of the number of training points, excluding some extreme cases. It is also observed that instability is observed mainly when the number of neurons is too small. For this reason, 1/10 is chosen as a middle ground that will work in most cases. Note that the value function approximates the expected cost-to-go instead of the more common reward-to-go because, in this case, the goodness of an action is more clearly represented by a cost instead of a reward. The policy is optimized accordingly using gradient descent to minimize the cumulative cost.

3.4. Policy update

Once the value function is approximated by the critic net, it is used to estimate the gradient of the objective function $J(\pi_\theta)$. The expression of the approximated gradient in stochastic policy gradient becomes:

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(u_{i,t}|x_{i,t}) \hat{A}^\pi(u_{i,t}, x_{i,t}) \tag{27}$$

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 in Eq. (22), is obtained analytically as:

$$\nabla_\theta \log \pi_\theta = \frac{\pi_\theta - \mu}{\sigma^2} \phi(\mathbf{x}) \tag{28}$$

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. This approximation introduces bias. A way to reduce the effect of bias is to use a slightly different definition of the advantage function, often referred to as *n-step returns*. The idea is to use a sample-based estimation of the expected cost-to-go only for the first n steps into the future and then use the approximated value function for the rest of the time steps. This implies a reduction of the variance thanks to using the value function for time steps far into the future but ensures an unbiased estimate for the time steps close to the considered one. Using this definition, the expression becomes:

$$\hat{A}_n^\pi(u_t, x_t) = \sum_{t'=t}^{t+n} \gamma^{t'-t} c(x_{t'}, u_{t'}) - \hat{V}^\pi(x_t) + \gamma^n \hat{V}^\pi(x_{t+n}) \tag{29}$$

having introduced also the discount factor $0 < \gamma < 1$. n is the number of time steps into the future for which the unbiased cost-to-go is used. The update then is simply done according to stochastic gradient descent taking a step in the opposite direction of the gradient $\nabla_\theta J(\pi_\theta)$:

$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta J(\pi_\theta) \tag{30}$$

Where α is the bounded learning rate. After each update, the algorithm is tested starting from 15 random initial states sampled from the same distribution as the one used for training. The cumulative cost is computed along each trajectory and then averaged among all the test trajectories:

$$C_k = \frac{1}{N_t} \sum_{i=0}^{N_t} \sum_{t=0}^{T} c(x_{i,t}, u_{i,t}) \tag{31}$$

where k stands for k-th global iteration, N_t is the number of test trajectories and T is the number of time-steps per trajectory. A summary of the algorithm in form of pseudo-code is given in Algorithm 1 while the hyperparameters relative to the learning algorithm are shown in Table 1.

Table 1. Hyperparameters

Model	# policy neurons	β_r	β_v	σ^2	α	γ	N_t	T
CW	66087	300	65	0.6	2e-5	0.99	15	200
NLR	32262	300	65	1	1e-5	0.9999	15	200

Algorithm 1 ELM-based Actor-Critic

for $k = 1$: # global iterations **do**
  for $i = 1$: # episodes per batch **do**
    for $t = 1$: # time-steps per episode - 1 **do**
      sample policy $\pi_{\theta_k} \rightarrow K_r, K_v$
      calculate command action with 19
      obtain sample $(x_{i,t}, u_{i,t}, c_{i,t}, x_{i,t+1})$
    end for
  end for
  fit $\hat{V}^{\pi}(x)$ to sampled discounted cost-to-go $\left\{ \left(x_{i,t}, \sum_{t'=t}^{T} \gamma^{t'-t} c(x_{i,t'}, u_{i,t'}) \right) \right\}$
  for $i = 1$: # episodes per batch **do**
    for $t = 1$: # time-steps per episode - 1 **do**
      evaluate $\hat{A}_n^{\pi}(u_{i,t}, x_{i,t}) = \sum_{t'=t}^{t+n} \gamma^{t'-t} c(x_{i,t'}, u_{i,t'}) - \hat{V}^{\pi}(x_{i,t}) + \gamma^n \hat{V}^{\pi}(x_{i,t+n})$
      evaluate $\nabla_\theta \log \pi_\theta(u_{i,t}|x_{i,t})$
    end for
  end for
  evaluate $\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(u_{i,t}|x_{i,t}) \hat{A}^{\pi}(u_{i,t}, x_{i,t})$
  update policy $\theta_{k+1} = \theta_k - \alpha \nabla_\theta J(\pi_\theta)$
  test new policy $\pi_{\theta_{k+1}} \rightarrow C_k = \frac{1}{N_t} \sum_{i=0}^{N_t} \sum_{t=0}^{T} c(x_{i,t}, u_{i,t})$
end for

4. Training and Testing Results

The proposed algorithm is developed, trained, and tested on the scenarios described below. Chaser and target are assumed to be initially on the same NRO, selected among the families presented in the sections above, with the chaser trailed behind the target and within the approach sphere. It is assumed that the chaser has gotten to this position through a phasing process prior to initiating the final approach. The orbit belongs to the southern L_2 family, the periselene has an altitude of 1617 km over the Moon surface and the period is 6.17 days. The spacecraft 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. The algorithm is trained and tested on two constraint scenarios:

- spherical keep-out zones randomly positioned in the approach sphere
- a Keep-Out Sphere (KOS) with a conical approach corridor centered on the target

The test cases are selected to be challenging constraints scenarios that showcase the capabilities of the proposed method. The spherical constraints represent spherical keep-out zones: they can represent appendages of the target body or separate bodies. The KOS constraint represents a keep-out sphere surrounding the target with an approach corridor aligned with the docking port. This is a typical constraint scenario as reported in the references (Campolo, 2017; Dong et al., 2017; Zhao & Zhang, 2021; Lu & Liu, 2013). The Clohessy-Wiltshire (CW) equations describe the dynamics in the proximity of the periselene and non-linear relative equations (NLR) in the proximity of the aposelene. The results are presented separately for the two cases. The solution is compared with a waypoint-based ZEM-ZEV guidance using the optimal guidance parameters and user-defined waypoints. The results obtained in the periselene region are also compared with an optimal solution obtained with GPOPS (**G**eneral **P**seudospectral **OP**timal Control

Table 2. Spheres radii and position vectors in LVLH frame.

	Sphere #	R [m]	x [km]	y [km]	z [km]
Periselene	1	100	-0.75	0	-0.4
	2	70	-0.4	-0.13	-0.2
Aposelene	1	100	-0.7	-0.05	0.16
	2	70	-0.3	0	0.06

Software). It should be noted that numerically solving the optimal problem with GPOPS when the NLR equations are used was explored. However, in such a case, due to the number of states, the problem becomes prohibitively difficult to solve for the direct transcript method used by GPOPS and did not converge to any acceptable solutions. To the best of the authors' knowledge, there are no examples in the literature of optimal solutions for the relative control problem in the CRTBP environment, so this remains an open point that should be addressed in future works. The A-ZEM/ZEV algorithm is trained by sampling the starting state from a gaussian distribution centered around the nominal starting state. The resulting guidance law is tested at each iteration starting from 15 random initial states within the distribution used for training and the cumulative cost computed. The policy is initialized with K_R and K_V equal to the classical ZEM/ZEV solution ($K_R = 6$ and $K_V = -2$).

4.1. Spherical constraints

The constraints are represented by two spheres, assumed to be fixed in the LVLH frame. Their positions and radii are reported in Table 2. The constraint is considered violated if the following condition is true:

$$||\mathbf{r}_t - \mathbf{r}_s^j|| \leq R_s^j \tag{32}$$

where $\mathbf{r}_t$ is the current position of the satellite, $\mathbf{r}_s^j$ is the position of the j-th sphere's center and R_s^j is the j-th sphere's radius. The cost function is represented by Eq. (33). It comprises of 1) a term related to the mass of propellant burned during the time-step (Δm_t), 2) two terms related to the end position and velocity errors with respect to the nominal target state that are added only in case $t = t_f$, t_f being the final time; and 3) a term related to the position error of the impact point, if present, with respect to the target state:

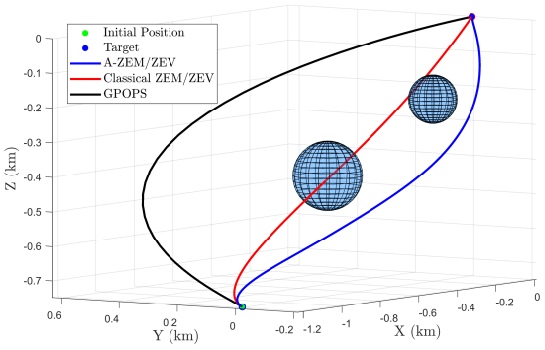
$$C(t) = w_m \Delta m_t + \delta(t - t_f)\left[w_r^f ||\mathbf{r}_t - \mathbf{r}_f||^2 + w_v^f ||\mathbf{v}_t - \mathbf{v}_f||^2 + b_f\right] + \delta(t - t_i)\left[w_r^i ||\mathbf{r}_t - \mathbf{r}_f||^2 + b_i\right] \tag{33}$$

Here, w_m, 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 and b_f and b_i are biases added at the end of episodes with $b_i > b_f$. The latter ensures that a collision-less solution has a lower cost than a solution that violates the constraint. The condition $b_f > 0$ instead ensures that the value function, as the agent gets near the target, does not get too close to 0, which avoids an issue related to the relative error introduced by the value function approximation getting too big. The time-of-flight is set to 6000 seconds at the periselene and 40000 seconds at the aposelene. Refer to table 6 for the values of the parameters.

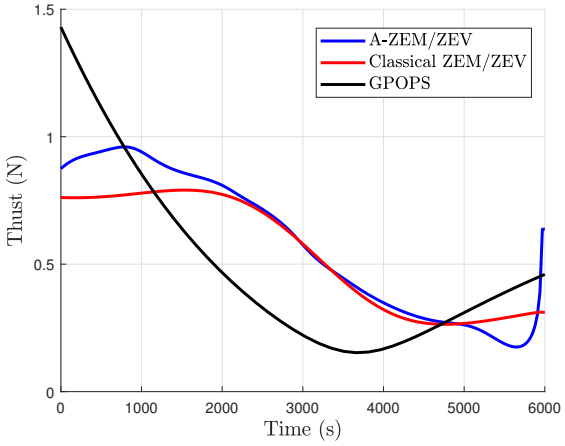
The waypoint-based solution can be seen in Figure 8, both in the periselene and the aposelene regions. The original starting state is assumed to be the same as the proposed method. An arbitrary intermediate waypoint was then placed in a position that would be more convenient for docking. The waypoint states are reported in Table 3. The solution of the independent legs was then found using classical ZEM-ZEV.

4.2. Keep-Out Sphere

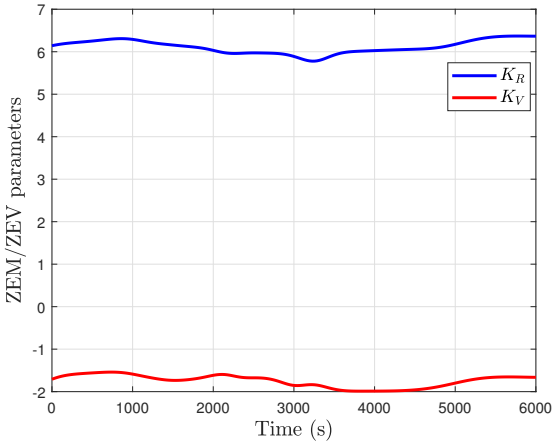
The second test scenario is represented by a keep-out sphere (KOS) with an approach corridor aligned with a hypothetical docking port. The approach corridor is conical with a half cone angle equal to 15 degrees, intersected with a cylinder with the



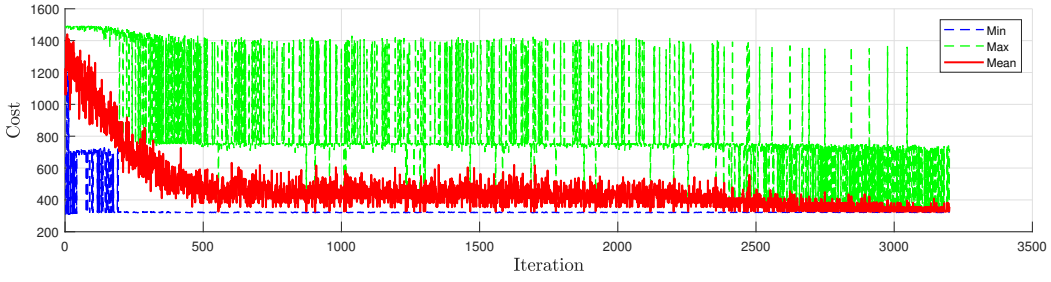
(a) Trajectory



(b) Thrust



(c) Guidance gains

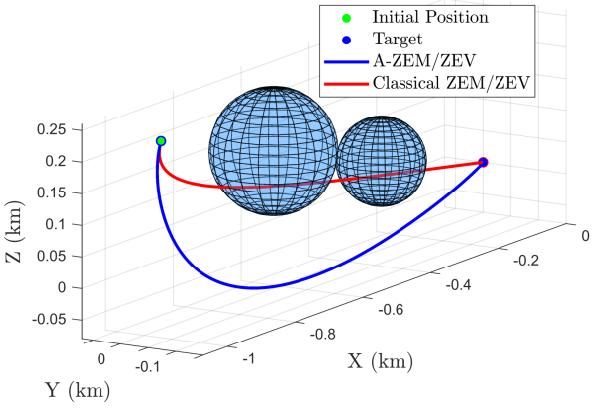


(d) Cost

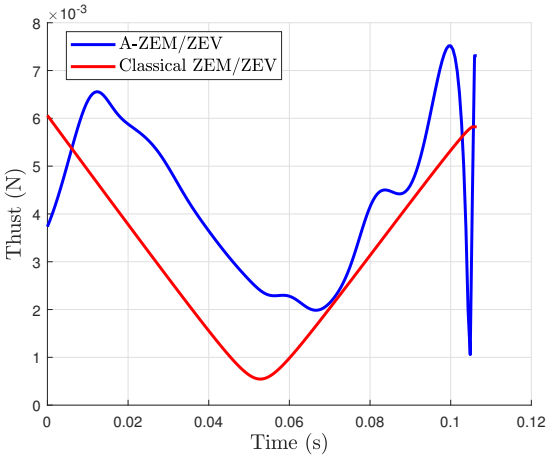
Fig. 6. Spherical constraints problem at periselene. Fuel usage: A-ZEM/ZEV - 1.6202 kg, Classical-ZEM/ZEV - 1.5108 kg, GPOPS - 1.4985 kg

Table 3. Waypoints. Spherical constraints.

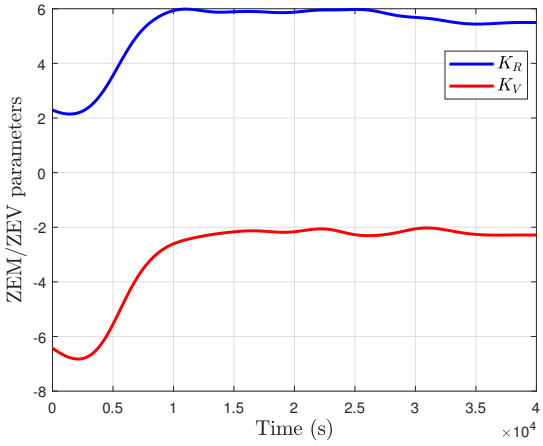
	x [km]	y [km]	z [km]	$\dot{x}$ [km/s]	$\dot{y}$ [km/s]	$\dot{z}$ [km/s]
Periselene	-0.8	0	0	0	0	0
Aposelene	-0.5	0	-0.1	0	0	0



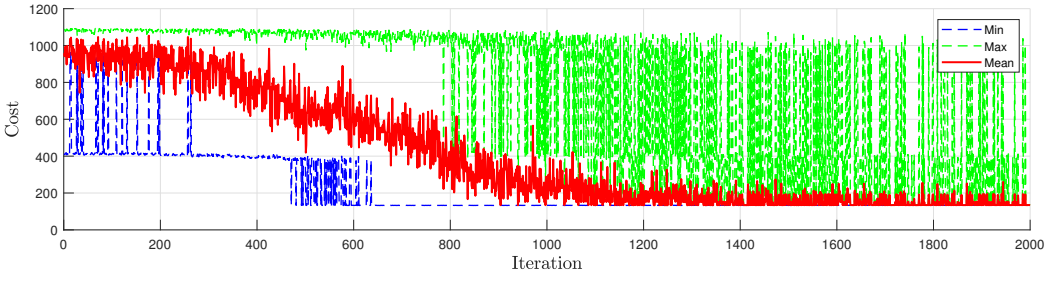
(a) Trajectory



(b) Thrust

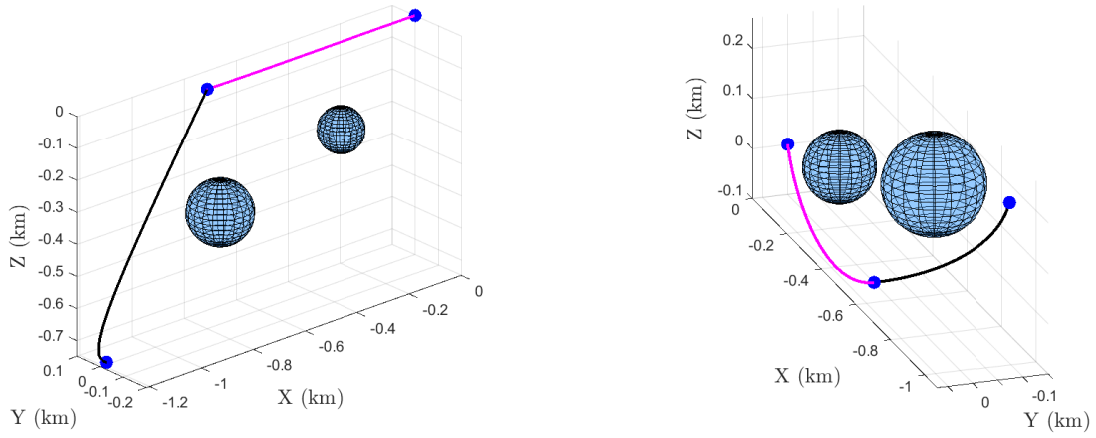


(c) Guidance gains



(d) Cost

Fig. 7. Spherical constraints problem at aposelene. Fuel usage: A-ZEM/ZEV - 0.0795 kg, Classical-ZEM/ZEV - 0.0576 kg



(a) Periselene. Fuel usage: 2.1691 kg

(b) Aposelene. Fuel usage: 0.1291 kg

Fig. 8. Spherical constraints waypoints

Table 4. Approach corridor axis direction in LVLH frame.

	x [km]	y [km]	z [km]
Periselene	-1	0	0
Aposelene	-0.944	0	-0.330

same axis and radius of 10 meters. The axis direction is arbitrarily chosen and depends on the problem to solve and can be seen in Table 4. It is important to note that the way this is implemented in practice does not make use of mathematical relationships. We instead created a triangular mesh of the keep-out sphere and approach corridor and detected collisions with it by identifying if a point was inside or outside the polyhedron. This was designed to demonstrate the possibility of using virtually any arbitrarily shaped constraints with this algorithm. The cost function in this case is represented by equation 34. It is composed of the same kind of terms as the spherical constraint case. The difference is in the impact point error term related to the angular error of the impact point with respect to the axis of the approach corridor.

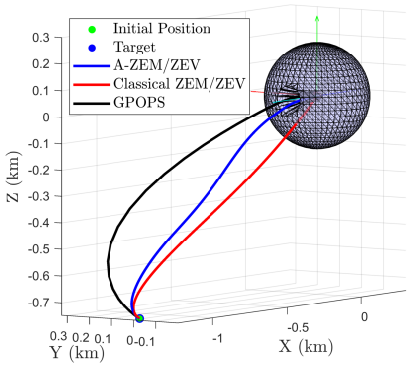
$$C(t) = w_m \Delta m_t + \delta(t - t_f)\left[w_r^f \|\mathbf{r}_t - \mathbf{r}_f\|^2 + w_v^f \|\mathbf{v}_t - \mathbf{v}_f\|^2 + b_f\right] + \delta(t - t_i)\left[w_\theta \text{acos}\left(\frac{\mathbf{r}_i \cdot \mathbf{n}}{|\mathbf{r}_i||\mathbf{n}|}\right) + b_i\right] \tag{34}$$

w_θ is a weight associated with the impact point angular error, $\mathbf{r}_i$ is the position vector of the impact point and $\mathbf{n}$ is the vector aligned with the approach corridor axis. In case the impact point is inside the approach corridor, the cost relative to the impact is instead associated with the distance of the impact point with respect to the target position as in Eq. (33). The time-of-flight is set to be 6000 seconds at the periselene and 40000 seconds at the aposelene. Again refer to table 6 for the values of the parameters.

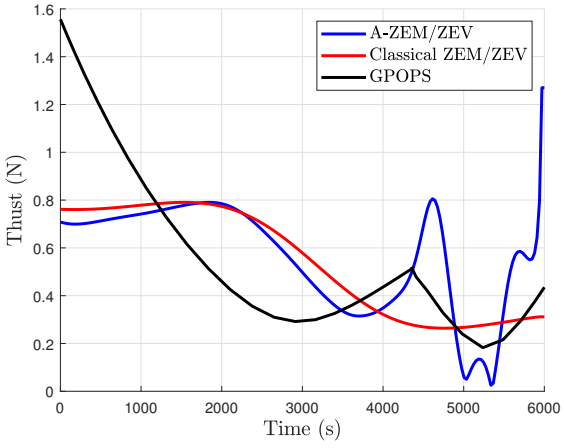
The waypoint-based solution can be seen in Figure 11 both in the periselene and the aposelene regions. The starting states are the same as for the proposed algorithm. The intermediate waypoints are positioned on the axis of the approach corridor, at a range equal to the keep-out sphere radius. The waypoint states are reported in Table 5.

Table 5. Waypoints. Keep-out sphere

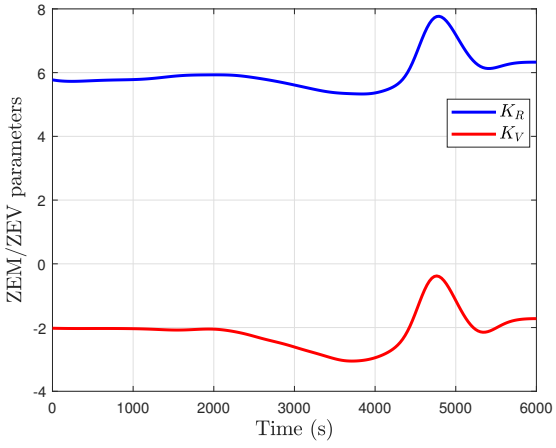
	x [km]	y [km]	z [km]	$\dot{x}$ [km/s]	$\dot{y}$ [km/s]	$\dot{z}$ [km/s]
Periselene	-0.8	0	0	0	0	0
Aposelene	-0.5	0	-0.1	0	0	0



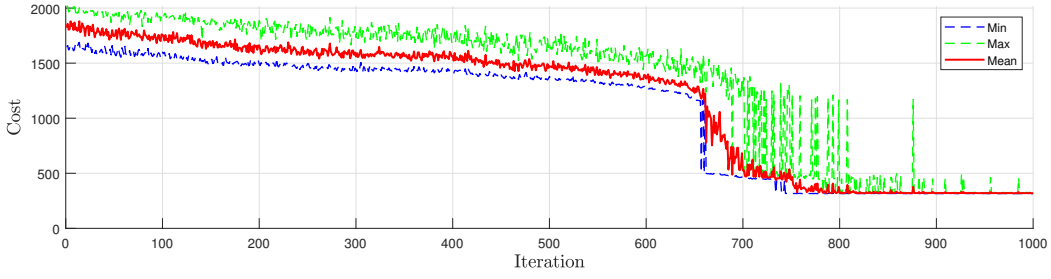
(a) Trajectory



(b) Thrust



(c) Guidance gains

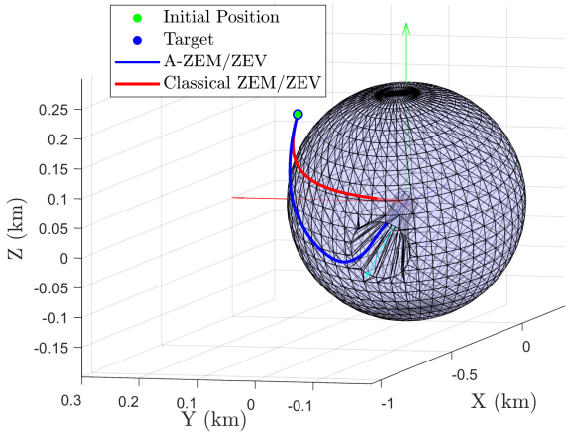


(d) Cost

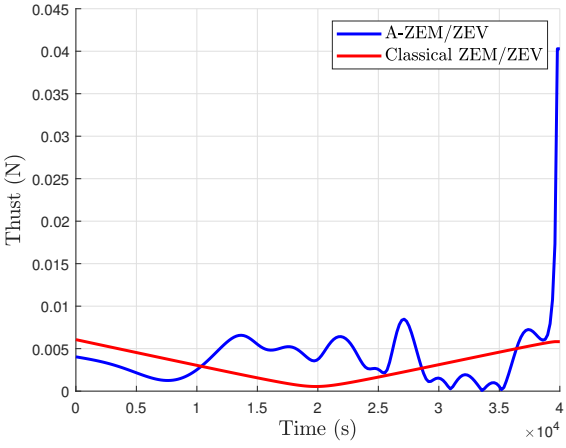
Fig. 9. Keep-out sphere constraint problem at periselene. Fuel : A-ZEM/ZEV - 1.565 kg, Classical-ZEM/ZEV - 1.511 kg, GPOPS - 1.485 kg

Table 6. Cost parameters

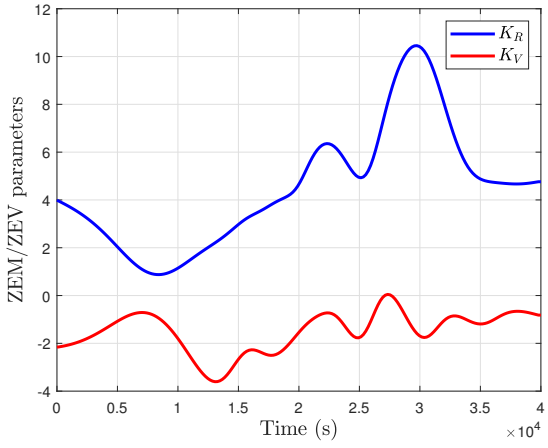
Constraint	w_m	w_r^f	w_v^f	b_f	w_r^i	b_i	w_θ
Spherical	125	125	125	125	1250	250	-
KOS	125	125	125	125	-	250	100



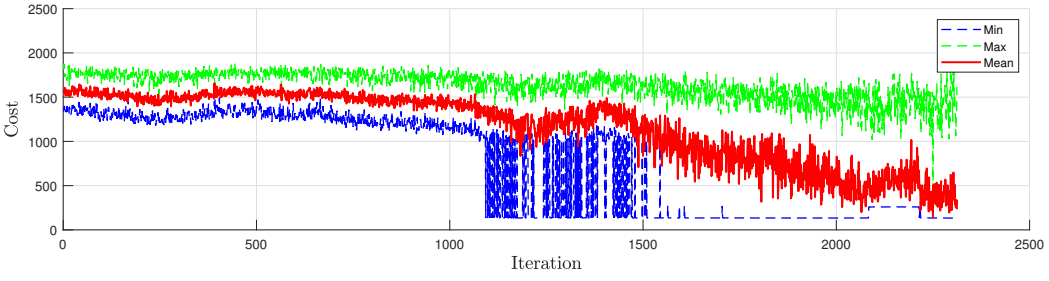
(a) Trajectory



(b) Thrust

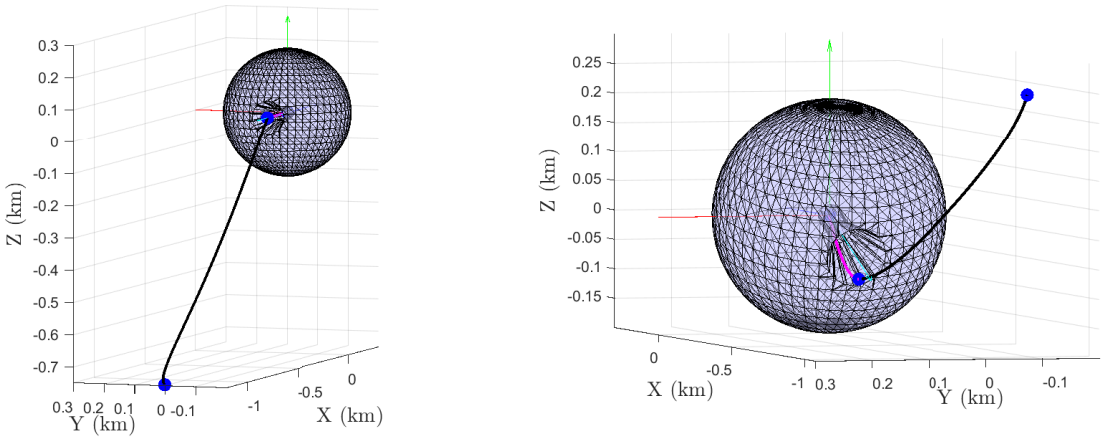


(c) Guidance gains



(d) Cost

Fig. 10. Keep-out sphere constraint problem at periselene. Fuel usage: A-ZEM/ZEV - 0.0737 kg, Classical-ZEM/ZEV - 0.0576 kg



(a) Periselene. Fuel usage: 1.8190 kg

(b) Aposelene. Fuel usage: 0.1120 kg

Fig. 11. Keep-out sphere waypoints

4.3. Results analysis

As shown in Figures 6,7,9,10, A-ZEM/ZEV manages to find a solution that is compliant with the path constraints in all cases. The classical-ZEM/ZEV solution is represented only for comparison purposes: it should be noted in fact that it violates the constraints in all cases. In the spherical constraints case at the periselene, A-ZEM/ZEV manages first to avoid the first sphere and then the second, as clearly shown by the behavior of the cost in 6(d). In the KOS constraint cases, both at periselene and aposelene, A-ZEM/ZEV performs well, first aligning with the approach corridor and then avoiding collisions with its inner sides.

The task seems, in general, much more difficult in the aposelene region. In this case, the dynamics are very slow, and more significant changes in the gains K_R and K_V are needed to obtain the necessary change in the guidance command. In particular, the spherical constraints have proven to be the biggest hurdles for the algorithm. In this case, both the n parameter and the discount factor γ were increased so that the algorithm could sense the presence of the constraint from further away and effectively steer away from it. Although this is more evident in this case, the same phenomenon was observed in the KOS constraint case when the same equations of motion are used.

When compared to the previous paper where A-ZEM/ZEV was first introduced (Furfaro et al., 2020), the more complex dynamical environment seems to have an impact on the training speed. The total number of iterations needed for a successful training increase from the landing problem in (Furfaro et al., 2020), to the problem at periselene using linearized equations to the aposelene case with full non-linear equations. Parameters tuning was also considerably harder. The big jumps in the cost during training are due to the fact that, to avoid falling in local minima and achieve successful learning, the cost weights had to be tuned to have a clear separation in terms of cost between solutions that violate the constraints and collision-less solutions. For this reason, once the obstacle is avoided, there is an abrupt change in cumulative cost. The latter does not allow for good results in terms of fuel consumption without the introduction of variable weights. Putting too much emphasis on fuel consumption minimization, i.e., increasing the weight associated with the mass burned, would, in fact, degrade the collision avoidance capabilities in most cases. Therefore, we decided to favor collision avoidance because the ability to embed constraints directly into the guidance law was considered more valuable than further minimizing fuel consumption.

From a machine learning perspective, embedding ELM theory in the critic design has proven feasible and efficient. Figure 12 shows an example of value function approximation regression plot referred to as a single iteration. It clearly shows that the ELM

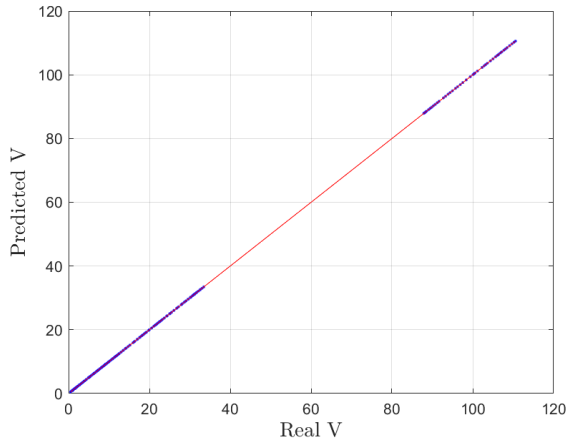


Fig. 12. ELM regression plot for a random iteration

Table 7. Learning algorithm performance (NRMSE: Normalized Root Mean Squared Error)

Constraint	Model	N^o iter.	Total train time (hours)	Mean iter. time (s)	Mean critic train time (s)	Mean critic NRMSE
Spherical	CW	3200	27.761	31.231	0.162	0.0142
	NLR	2000	9.901	17.822	0.154	0.0021
KOS	CW	1000	20.391	73.408	0.154	0.0077
	NLR	2312	37.405	58.243	0.140	0.0053

can capture the variations in the value function as a function of the state. In particular, it shows that even when the set has samples coming from both trajectories that hit the constraint and have significant cumulative costs (top right) and trajectories that arrive at the target that have lower cumulative cost (bottom left), the regression is still accurate which ultimately leads to successful learning. By adjusting the algorithm's parameters according to the case to solve, it was possible to maintain a high regression accuracy and, consequently, to keep the bias controlled. Specifically, the number of neurons of the ELM that proved to work well in any situation was found to be 1/10 of the number of total samples. Increasing the number of time-steps per episode and the variance of the gaussian distribution from which the initial state of the episodes is sampled also helped reduce the critic's regression error. This is because the samples cover more densely the state space and create a smoother function to approximate. Moreover, the discount factor γ also has an effect on the accuracy of the critic net: a smaller value, in general, leads to a higher accuracy because the end state cost, which is the one that affects the overall cost-to-go the most, is valued less, which leads to a less stiff value function to approximate. Of course, different values of γ also mean different learning results, so its selected value is the product of a compromise. Table 7 is a summary of the learning process performances. It clearly shows that the ELM training time is very low compared to the iteration time. This shows that the critic has a minor impact on the training time, from which we can infer that using an ELM as critic in an actor-critic algorithm is very efficient. It should also be noted that the validation phase takes up a significant part of the iteration time.

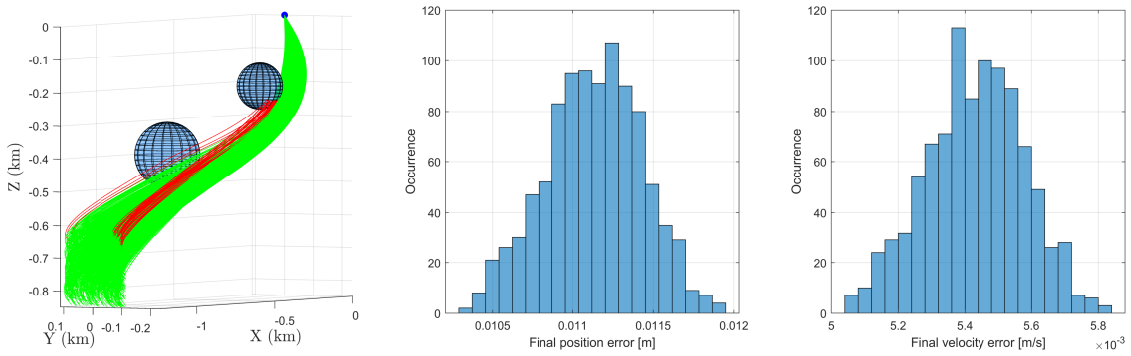
4.3.1. Monte Carlo analysis

A Monte Carlo analysis is carried out for every constraint and dynamics case. Figures 13 show the results. The initial state of the 1000 trials is sampled from the same distribution used during training. The results show that the neural network learns a guidance law that succeeds in reaching the target without colliding with the constraints in more than 95 % of the cases across all the dynamics and constraint scenarios. Figures 13 also show that the trajectories that fail are usually the ones that have initial

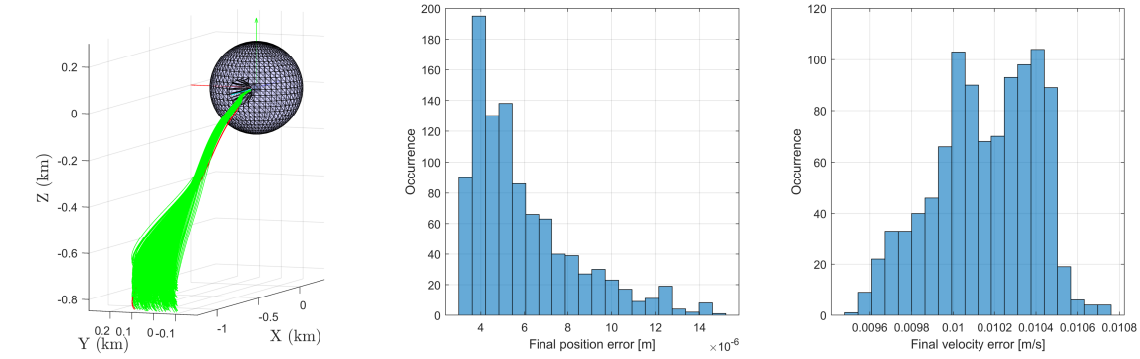
conditions near to the boundaries of the considered region. This means that these cases are relatively extreme and should not threaten the mission's success if the uncertainty on the initial state is well below the specified boundaries. The analysis shows good targeting performances, with final guidance relative position and velocity in the order of centimeters and centimeters per second, respectively.

5. Conclusions

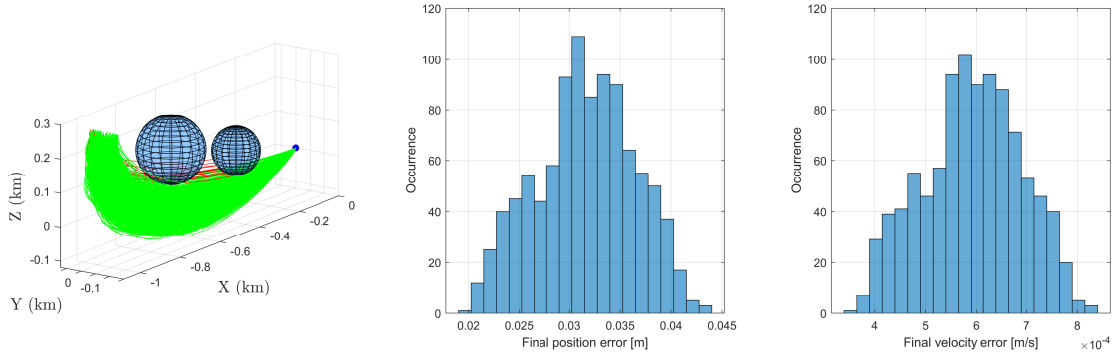
While the classical-ZEM/ZEV has its most significant strengths in its closed-loop nature and ease of implementation, the fact that it is impossible to impose path constraints directly into the algorithm has limited its possible applications. This work shows that using reinforcement learning, notably an actor-critic algorithm based on policy gradient, with advantage function estimation, it is possible to expand its capabilities. The resulting algorithm is trained and deployed for relative motion guidance in NRO scenarios. It is demonstrated that it can improve ZEM/ZEV performance in terms of collision avoidance capabilities in a variety of complex constraint scenarios. Furthermore, it increases autonomy when compared to simple waypoint-based guidance using classical ZEM/ZEV, as it can automatically satisfy the constraints, provided that the initial condition is within the training distribution. The proposed method demonstrates that an adaptive guidance algorithm based on ZEM/ZEV is feasible for guidance in cislunar space both when using non-linear CRTBP equations and Clohessy-Wiltshire equations. A Monte Carlo analysis performed on the trained network shows reasonable success rates, starting from a distribution of states centered on the nominal starting state and good terminal error performances. Finally, from a reinforcement learning perspective, it has been shown that ELM can be employed as critic network in an actor-critic algorithm. This paper shows that shallow networks work exceptionally well for simple regression problems when the input dimension is small. This work also demonstrates that machine learning and artificial intelligence, in general, are valuable assets that should be taken into consideration in the design of a new guidance algorithm for spacecraft guidance when autonomy and flexibility are pivotal and should be considered to achieve high degrees of autonomy for operations in the cislunar space.



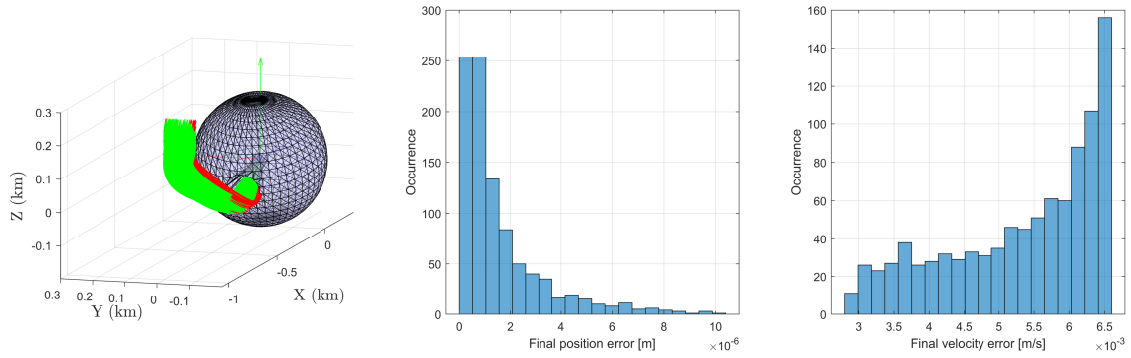
(a) CW, spherical constraints. Success rate: 96.3 %



(b) CW, KOS. Success rate: 99.8



(c) NLR, spherical constraints. Success rate: 98.1



(d) NLR, KOS. Success rate: 95.3

Fig. 13. Monte Carlo analysis. The green trajectories arrive successfully to the target, the red ones violate the constraints. Histograms refer only to successful trials.

References

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

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

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

Cui, P., Yan, W., & Wang, Y. (2017). Reactive path planning approach for docking robots in unknown environment. *Journal of Advanced Transportation*, *2017*.

Dong, H., Hu, Q., & Akella, M. R. (2017). Safety control for spacecraft autonomous rendezvous and docking under motion constraints. *Journal of Guidance, Control, and Dynamics*, *40*(7), 1680–1692.

Furfaro, R., & Linares, R. (2017). 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*, .

Furfaro, R., Ruggiero, R., Topputo, F., Lovera, M., Linares, R. et al. (2018). Waypoint-optimized closed-loop guidance for spacecraft rendezvous in relative motion. *Advances in the Astronautical Sciences*, *162*, 2651–2666.

Furfaro, R., Scorsoglio, A., Linares, R., & Massari, M. (2020). Adaptive generalized zem-zev feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica (Accepted)*, .

Furfaro, R., & Wibben, R. D. (2016). Robustification of a class of guidance algorithms for planetary landing: Theory and applications. *26th AAS/AIAA Space Flight Mechanics Meeting, 2016. Univelt Inc.*, .

Gaudet, B., Furfaro, R., & Linares, R. (2020a). Reinforcement learning for angle-only intercept guidance of maneuvering targets. *Aerospace Science and Technology*, *99*, 105746.

Gaudet, B., Linares, R., & Furfaro, R. (2020b). Adaptive guidance and integrated navigation with reinforcement meta-learning. *Acta Astronautica*, .

Gaudet, B., Linares, R., & Furfaro, R. (2020c). Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, .

Gaudet, B., Linares, R., & Furfaro, R. (2020d). Six degree-of-freedom hovering over an asteroid with unknown environmental dynamics via reinforcement learning. In *AIAA Scitech 2020 Forum* (p. 0953).

Gill, T. (2018). Nasa's lunar orbital platform-gatway, .

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

Grondman, I., Busoniu, L., Lopes, G. A., & Babuska, R. (2012). A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.6: 1291-1307*, .

Guo, Y., Hawkins, M., & Wie, B. (2011). Optimal feedback guidance algorithms for planetary landing and asteroid intercept. *AAS/AIAA astrodynamics specialist conference (pp. 2011-588). AAS*, .

Guo, Y., Hawkins, M., & Wie, B. (2013). Applications of generalized zero-effort-miss/zero-effort-velocity feedback guidance algorithm. *Journal of Guidance, Control, and Dynamics, 36(3), 810-820*, .

Hawkins, M., Guo, Y., & Wie, B. (2012). Zem/zev feedback guidance application to fuel-efficient orbital maneuvers around an irregular-shaped asteroid. *AIAA Guidance, Navigation, and Control Conference (p. 5045)*, .

Hovell, K., & Ulrich, S. (2021). Deep reinforcement learning for spacecraft proximity operations guidance. *Journal of spacecraft and rockets*, *58*(2), 254–264.

Huang, G., Huang, G.-B., Song, S., & You, K. (2015). Trends in extreme learning machines: A review. *Neural Networks*, *61*, 32–48.

Huang, G. B. (2015). What are extreme learning machines? filling the gap between frank rosenblatt's dream and john von neumann's puzzle. *Cognitive Computation 7.3*, .

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

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*(1-3), 489–501.

Jiang, J., Zeng, X., Guzzetti, D., & You, Y. (2020). Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures. *Acta Astronautica*, *171*, 265–279.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research 32.11*, .

Koon, W. S., Lo, M. W., & Marsden, J. E. (2011). *Dynamical Systems, the Three-Body Problem and Space Mission Design*.

Lian, Y., Meng, Y., Tang, G., & Liu, L. (2012). Constant-thrust glideslope guidance algorithm for time-fixed rendezvous in real halo orbit. *Acta Astronautica*, *79*, 241–252.

Lightsey, P. A., Atkinson, C. B., Clampin, M. C., & Feinberg, L. D. (2012). James webb space telescope: large deployable cryogenic telescope in space. *Optical Engineering, 51(1), 011003*, .

Lu, P., & Liu, X. (2013). Autonomous trajectory planning for rendezvous and proximity operations by conic optimization. *Journal of Guidance, Control, and Dynamics*, *36*(2), 375–389.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., & Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robotics and autonomous systems, 47(2-3), 79-91*, .

Oestreich, C. E., Linares, R., & Gondhalekar, R. (2021). Autonomous six-degree-of-freedom spacecraft docking with rotating targets via reinforcement learning. *Journal of Aerospace Information Systems*, *18*(7), 417–428.

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

Peng, H., Yang, C., Li, Y., Zhang, S., & Chen, B. (2013). Surrogate-based parameter optimization and optimal control for optimal trajectory of halo orbit rendezvous. *Aerospace Science and Technology*, *26*(1), 176–184.

Peters, J., & Schaal, S. (2006). Policy gradient methods for robotics. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE*, .

Peters, J., & Schaal, S. (2008). Natural actor-critic. *Neurocomputing 71.7-9*, .

Pinard, D., Reynaud, S., Delpy, P., & Strandmoe, S. E. (2007). Accurate and autonomous navigation for the atv. *Aerospace Science and Technology, 11(6), 490-498*, .

Scorsoglio, A. (2018). *Adaptive ZEM/ZEV feedback guidance for rendezvous in lunar NRO with collision avoidance)*.

Scorsoglio, A., D'Ambrosio, A., Ghilardi, L., Gaudet, B., Curti, F., & Furfaro, R. (2022). Image-based deep reinforcement meta-learning for autonomous lunar landing. *Journal of Spacecraft and Rockets*, *59*(1), 153–165.

Scorsoglio, A., Furfaro, R., Linares, R., Massari, M. et al. (2019). Actor-critic reinforcement learning approach to relative motion guidance in near-rectilinear orbit. *Advances in the Astronautical Sciences*, *168*, 1737–1756.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. *ICML*, .

Smart, W. D., & Kaelbling, L. P. (2002). Effective reinforcement learning for mobile robots. *IEEE International Conference on. Vol. 4. IEEE*, .

Ueda, S., & Murakami, N. (2015). Optimum guidance strategy for rendezvous mission in earth-moon l2 halo orbit. In *Proceedings of the 25th International Symposium on Space Flight Dynamics (ISSFD), Munich, Germany*.

Wang, D., & Huang, G.-B. (2005). Protein sequence classification using extreme learning machine. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* (pp. 1406–1411). IEEE volume 3.

Whitley, R., & Martinez, R. (2016). Options for staging orbits in cislunar space. *Aerospace Conference, 2016 IEEE. IEEE*, .

Wibben, D. R., & Furfaro, R. (2016). Optimal sliding guidance algorithm for mars powered descent phase. *Advances in space research*, *57*(4), 948–961.

Williams, J., Lee, D. E., Whitley, R. J., Bokelmann, K. A., Davis, D. C., & Berry, C. F. (2017). Targeting cislunar near rectilinear halo orbits for human space exploration, .

Williams, R. J. (1992). *Reinforcement learning*. Springer.

Zappulla, R., Park, H., Virgili-Llop, J., & Romano, M. (2018). Real-time autonomous spacecraft proximity maneuvers and docking using an adaptive artificial potential field approach. *IEEE Transactions on Control Systems Technology*, *27*(6), 2598–2605.

Zhang, Y., Guo, Y., Ma, G., & Zeng, T. (2017). Collision avoidance zem/zev optimal feedback guidance for powered descent phase of landing on mars. *Advances in Space Research, 59(6), 1514-1525*, .

Zhao, X., & Zhang, S. (2021). Adaptive saturated control for spacecraft rendezvous and docking under motion constraints. *Aerospace Science and Technology*, *114*, 106739.

Zimpfer, D., Kachmar, P., & Tuohy, S. (2005). Autonomous rendezvous, capture and in-space assembly: past, present and future. *1st Space exploration conference: continuing the voyage of discovery (p. 2523)*, .