AAS 19-441

ACTOR-CRITIC REINFORCEMENT LEARNING APPROACH TO RELATIVE MOTION GUIDANCE IN NEAR-RECTILINEAR ORBIT

Andrea Scorsoglio[*], Roberto Furfaro[†], Richard Linares[‡], and Mauro Massari[§]

This paper aims a developing a new feedback guidance algorithm for docking maneuvers in the cislunar environment. In particular, the goal is to create an algorithm that is lightweight, closed-loop and capable of taking path constraints into account. The problem has been solved starting from the well know Zero-Effort-Miss/Zero-Effort-Velocity (ZEM/ZEV) guidance using machine learning to improve its capabilities and widen its field of application. The algorithm has been developed in the circular restricted three body problem (CRTBP) framework for Near Rectilinear Orbits (NRO) in the Earth-Moon system but the results can be easily generalized to many more guidance problems. The results are satisfactory and show that reinforcement learning can be effectively used to solve constrained relative spacecraft guidance problems.

INTRODUCTION

Accurate feedback guidance algorithms have always been of utmost importance for space exploration. Specifically, precise maneuvering around lagrangian points has always been important since the beginning of solar system exploration. Examples of spacecraft that make use of the advantageous position of these particular points are the solar wind monitoring probes (ACE, SOHO, DSCVR, WIND) that are positioned in the L_1 Earth-Sun lagrangian point. The importance of these locations has also been raised by the announcement of the James Webb telescope, scheduled to launch in March 2021 and directed to an halo orbit originating from the L_2 Earth-Sun lagrangian point[1] . Moreover with the Lunar Orbital Platform-Gateway (LOP-G)[2] set to become the new establishment for human exploration of the solar system, relative dynamics guidance in cislunar environment will be of pivotal importance in the near future. NASA has stated that in the next decade, the Moon will be one of the primary objectives for space exploration, both for its scientific value and as 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)[3] in the Earth-Moon three-body framework are considered the most promising environment for this kind of missions. An important study by NASA[3] , has shown some

[*]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

[‡]Charles Stark Draper Assistant Professor, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue Cambridge, MA 02139, Senior Member AIAA, E-mail:linaresr@mit.edu

[§]Professor, Aerospace Science and Technology Department, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

advantages of using these kind of orbits over different cislunar orbits. Their particular 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; it has been proven in fact by the same study that they are within the launching capabilities of an SLS-Orion mission. Finally they have a small ΔV requirement for station keeping and the thermal characteristics are favorable. Many of the operations in this environment will rely on precise relative guidance. Up until now, guidance algorithms for this kind of problems almost always have relied on open-loop architectures that are either defined beforehand on the ground or depend on direct human intervention in case of manned missions. Examples of almost automated docking are ESA's ATV [4] and Roscosmos' Progress[5] . Although well performing they still rely heavily on planning from ground and do not use a completely autonomous guidance approach. Moreover, they work well for docking to the ISS, so in Low Earth Orbit (LEO) but 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. Luckily this has already been studied and formalized[6] but there is little to no literature on the guidance and control side of the problem. The aim of this work is to propose a new guidance algorithm capable of operating in such environment. The idea is to use some concepts of artificial intelligence, especially reinforcement learning[7–9] and extreme learning machines [10–12] , to create a zero-effort-miss/zero-effort-velocity [13] based closed-loop algorithm able to solve this kind of problems.

ZEM/ZEV feedback guidance and reinforcement learning

Over the past few years, researchers have been exploring the performances of the generalized Zero-Effort-Miss/Zero-Effort-Velocity (ZEM/ZEV) feedback guidance for soft landing, intercept and rendezvous problems[13–15] . The ZEM/ZEV feedback guidance is attractive because of its analytical simplicity and accuracy: guidance mechanization is straightforward, and it can theoretically drive the spacecraft to a target autonomously and with minimal guidance errors, regardless of the equations of motion. Moreover, it has been shown to be globally finite time stable and robust to perturbations and uncertainties in the model if a proper sliding parameter is added (Optimal Sliding Guidance)[16] . One of the biggest strengths is its closed-loop nature: the guidance action in a particular state is derived directly from information on the current state and the target state. This is powerful because there is no need of integrating ground operations in the control loop as it is done with open-loop architectures. It is all in all a very straightforward way of solving closed-loop guidance problems when constraints are not an issue. Nevertheless, the algorithm has two major limitations: 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 it is virtually impossible to take path constraints into account. These are strong limitations. Especially the second one makes the algorithm not suitable for relative motion operations for docking that normally have path constraints to be enforced. The aim of this project is to create a new algorithm that retains the strengths of classical ZEM/ZEV and overcomes its major limitations by making use of machine learning techniques.

Historically machine learning is divided in three branches: supervised learning, unsupervised learning and reinforcement learning[17,18] . In this case Reinforcement Learning (RL) has been used to create a new algorithm suitable for the solution of a continuous-state/continuous-action guidance problem. Although reinforcement learning algorithms have been already used to solve many robotic motion tasks[19–24] , they have not been used frequently in spacecraft guidance. There is an example, involving pinpoint soft landing in which RL has been used to select the optimal sequence

of waypoints in a waypoint-based ZEM/ZEV algorithm[25] but it has never been used, to the author knowledge, to solve directly the continuous guidance problem. The idea behind this project is to use RL, specifically an actor-critic algorithm, to solve the constrained guidance problem, paving the way for true autonomous spacecraft guidance.

NRO RENDEZVOUS PROBLEM FORMALIZATION

Circular restricted three body problem and NROs

The motion of a particle in presence of two main bodies, or primaries, m_1 and m_2 where the only mean of interaction between the particles is the gravitational attraction, is described in general by the Circular Restricted Three Body Problem (CRTBP). In this framework the primaries are considered orbiting around the center of mass of the system in circular orbits. Historically the dynamics of the problem are expressed in the absolute synodic reference frame that in the case of the Earh-Moon system and for the remainder of this project 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 is non-inertial, it is in fact rotating 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.[26]

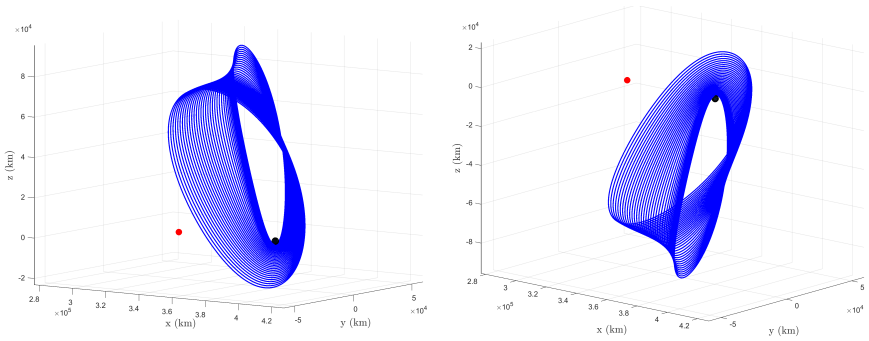
Equilibrium solutions Although equations 2 do not have a closed form analytical solution, it is possible to determine the location of equilibrium points of the CRTBP. The equilibrium points, or lagrangian points, or libration points are stationary points of the potential function U defined as

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

and are the solutions of the equation

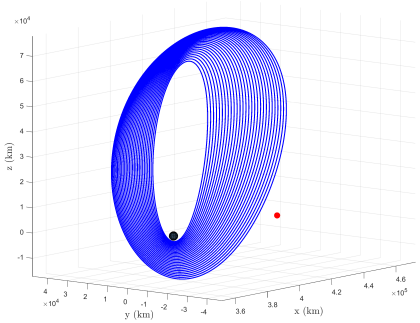
$$\nabla U = 0 \tag{5}$$

The equilibrium points are locations in which the secondary mass m would appear motionless in the rotating synodic frame.

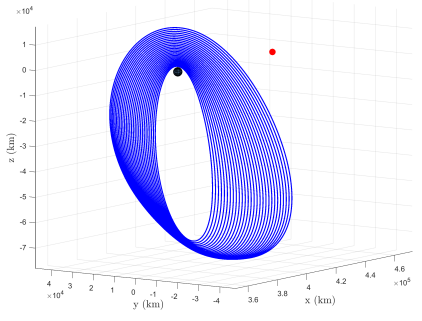


(a) L_1 northern family

(b) L_1 southern family



(c) L_2 northern family



(d) L_2 southern family

Figure 1. NRO families. The red dot is the lagrangian point, the black dot is the Moon

Near Rectilinear Orbits In the CRTBP framework, there exist a wide variety of trajectories that result in a periodical motion. They can be divided in 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. The generation of periodical orbits in this framework is not straightforward. Closed trajectories were in fact found using a shooting algorithm based on a multi-variable newton method. The whole process of finding those orbits is described thoroughly in the references[27,28] and since it is not the main subject of this project, it will not be described in details. A representation of all the NROs families that were considered for this study 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 by Campolo[6] . They are divided in two sections, 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 a ”close approach” phase, starting at the end of the first phase and ending with docking. 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

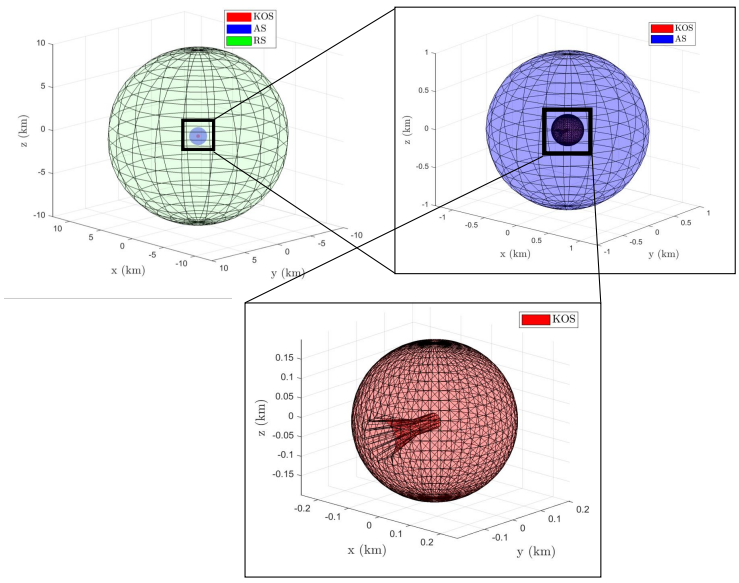


Figure 2. Rendezvous areas

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). In Figure 2 the areas defined above are shown. In this project, 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 in which the algorithm is developed and tested. The motion of chaser and target is described by equations 2 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.

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 is something that has not been studied as extensively as for LEOs. Campolo proposed an interesting solution that is used as starting point for this project and will be summarized in the following.

Reference frames The absolute dynamics in the Earth-Moon CRTBP are developed in the absolute synodic non-dimensional frame $\mathcal{R}_{em}$. The description of rendezvous dynamics however is normally done using a reference frame relative to the target. In case of two-body dynamics this is generally the Local-Vertical-Local-Horizon frame (LVLH). The LVLH frame ($\mathcal{R}_l$) has been defined also for the CRTBP[6] . The problem is intrinsically different with respect to the two-body case. It has been demonstrated however that the short term NRO dynamics can be described in a LVLH defined with 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[6] . A representation of all the reference systems on a sample

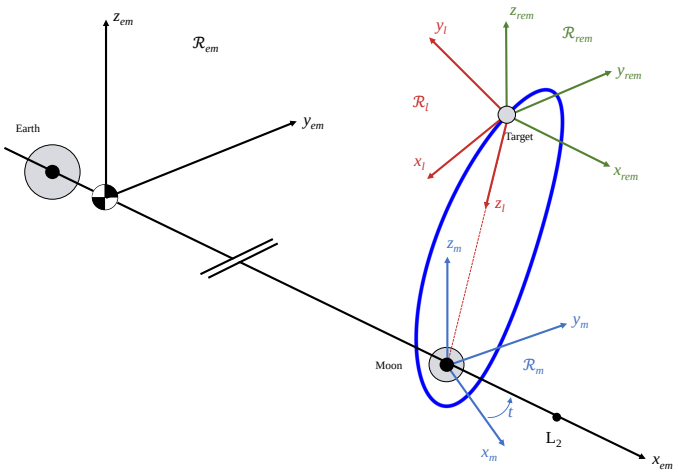


Figure 3. Reference systems

NRO can be see in Figure 3.

Relative equations of motion The relative motion in NROs can be fundamentally described using two models depending on the position along the orbit. It has been shown[6] that in portions 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 the *Clohessy-Wiltshire equation* (CW) expressed in the LVLH frame can be employed with little error. In other region 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).

- 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

*The closest point on the orbit with respect to the Moon

expressed in the $\mathcal{R}_{rem}$ reference frame:

$$\begin{aligned}
\ddot{x} - 2\dot{y} - x &= (1-\mu)\left[\frac{x_T+\mu}{\|r_{1T}\|^3} - \frac{x_T+x+\mu}{\|r_{1T}+\rho\|^3}\right] + \mu\left[\frac{x_T+\mu-1}{\|r_{2T}\|^3} - \frac{x_T+x+\mu-1}{\|r_{2T}+\rho\|^3}\right] \\
\ddot{y} + 2\dot{x} - y &= (1-\mu)\left[\frac{y_T}{\|r_{1T}\|^3} - \frac{y_T+y}{\|r_{1T}+\rho\|^3}\right] + \mu\left[\frac{y_T}{\|r_{2T}\|^3} - \frac{y_T+y}{\|r_{2T}\rho\|^3}\right] \\
\ddot{z} &= (1-\mu)\left[\frac{z_T}{\|r_{1T}\|^3} - \frac{z_T+z}{\|r_{1T}+\rho\|^3}\right] + \mu\left[\frac{z_T}{\|r_{2T}\|^3} - \frac{z_T+z}{\|r_{2T}\rho\|^3}\right]
\end{aligned} \tag{8}$$

where

$$\mathbf{x} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}] = \mathbf{x}_C - \mathbf{x}_T \tag{9}$$

is the synodic relative state,

$$\rho = [x \quad y \quad z]^{\mathrm{T}} \tag{10}$$

is the relative position,

$$\begin{aligned}
\mathbf{x}_T &= [x_T \quad y_T \quad z_T \quad \dot{x}_T \quad \dot{y}_T \quad \dot{z}_T]^T \\
\mathbf{x}_C &= [x_C \quad y_C \quad z_C \quad \dot{x}_C \quad \dot{y}_C \quad \dot{z}_C]^T
\end{aligned} \tag{11}$$

are the target and chaser synodic absolute positions,

$$\begin{aligned}
\mathbf{r}_{1T} &= [(x_T+\mu) \quad y_T \quad z_T]^T \\
\mathbf{r}_{2T} &= [(x_T+\mu-1) \quad y_T \quad z_T]^T
\end{aligned} \tag{12}$$

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*.

ADAPTIVE ZEM/ZEV

The Adaptive-ZEM/ZEV (A-ZEM/ZEV) was developed starting from the *classical ZEM/ZEV* 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). 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:

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

*The furthest point on the orbit with respect to the Moon

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:

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

where the ZEM distance and the ZEV error 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.

In any other case in which $\mathbf{f}(\mathbf{r}, \mathbf{v}) \neq \mathbf{g}(t)$ as it is in this project, the control law is still usable but it will not be necessarily optimal. ZEM and ZEV must be defined in a different way though. 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:[13]

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

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 project justify the use of this generalized form of the guidance algorithm. The fundamental idea behind A-ZEM/ZEV is to use reinforcement learning to learn the parameters K_R and K_V as function of the state x.

Learning is achieved via an actor-critic policy gradient algorithm that was developed specifically for this problem starting from the REINFORCE algorithm[9] 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 $J(\pi_\theta) = \mathbb{E}\left[r(x, u)\right]$ 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 $\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(u|x) Q^\pi(x,u)\right]$, 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. In stochastic policy gradient, this is substituted by a sample-based approximation of it 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, reducing the complexity of the task even more. The algorithm can be broken down in three blocks that are run sequentially at each global iteration: sample generation, critic neural network fitting and policy update.

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 is a representation of the guidance gains in equation 19, giving a series of samples $(x_{i,t}, u_{i,t}, c_{i,t}, x_{i,t+1})^*$. The starting position is randomly chosen by sampling a gaussian distribution around the nominal one. This ensures exploration of the state space and increases robustness against uncertainties on the starting state. 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 19, and the equations of motion integrated forward in time. 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. The agent runs until the end time is reached unless an intersection with the constraint is detected in which case the episode ends.

Policy The policy is described by a gaussian distribution with fixed variance σ^2 from which actions are sampled. The stochasticity of the policy is essential for a successful learning because it ensures exploration of the action space. It should be clear however, that the stochasticity is introduced only for the sake of exploration and because the machinery developed for stochastic policy gradient can then be applied. The policy that is then 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. This is why the algorithm only learns the mean of the policy and keeps the variance constant. The policy is divided in two separate parts, each dependent on a different set of parameters (θ_{K_R} and θ_{K_V}) related to the two parameters to learn K_R and K_V. The policy can be formally expressed as:

$$\begin{aligned} K_R &= \pi_{\theta_{K_R}} = \mathcal{N}(\mu_{K_R}, \sigma^2) \\ K_V &= \pi_{\theta_{K_V}} = \mathcal{N}(\mu_{K_V}, \sigma^2) \end{aligned} \tag{20}$$

where:

$$\begin{aligned} \mu_{K_R} &= \phi(\mathbf{x})^T \theta_{K_R} \\ \mu_{K_V} &= \phi(\mathbf{x})^T \theta_{K_V} \end{aligned} \tag{21}$$

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

*$x_{i,t}$ 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

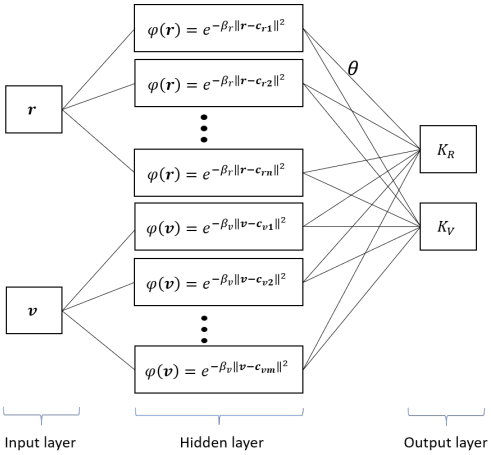


Figure 4. Policy neural network

Features The features are two sets of three dimensional radial basis functions (RBF) with centers distributed evenly across the position and velocity spaces. They are represented by the expression:

$$\begin{aligned} \phi(\mathbf{r}) &= e^{-\beta_R \|\mathbf{r}-\mathbf{c_r}\|^2} \\ \phi(\mathbf{v}) &= e^{-\beta_V \|\mathbf{v}-\mathbf{c_v}\|^2} \end{aligned} \tag{22}$$

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 4.

Critic neural network

A key part of the algorithm is the fitting of the neural network that approximates the value function. 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 $Q^w(x,u)$. 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 function of V only:

$$Q^w(x,u) = \hat{A}^\pi(u,x) = \hat{Q}^\pi(x,u) - \hat{V}^\pi(x) = r(x,u) + \hat{V}^\pi(x_{t+1}) - \hat{V}^\pi(x) \tag{23}$$

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, 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 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:

$$\left\{ \left(x_{i,t}, \sum_{t'=t}^{T} \gamma^{t'-t} c(x_{i,t'}, u_{i,t'}) \right) \right\} \tag{24}$$

This is an unbiased way of expressing the value function but could suffer from high variance. It should be noted though that the samples come from all the generated episodes, so the learned value function is an *average* of the expected cost-to-go, which is already a better estimate of the value function with respect to the single sample estimate. Note also 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 with the goal of minimizing the cumulative cost.

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:

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(u_{i,t}|x_{i,t}) \hat{A}^\pi(u_{i,t}, x_{i,t}) \tag{25}$$

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 20, is obtained analytically as:

$$\nabla_\theta \log \pi_\theta = \frac{\pi_\theta - \mu}{\sigma^2} \phi(\mathbf{s}) \tag{26}$$

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 the use of 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:

$$\hat{A}_n^\pi(u_t, x_t) = \sum_{t'=t}^{t+n} \gamma^{t'-t} c(s_{t'}, u_{t'}) - \hat{V}^\pi(x_t) + \gamma^n \hat{V}^\pi(x_{t+n}) \tag{27}$$

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 with respect to the gradient $\nabla_\theta J(\pi_\theta)$:

$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta J(\pi_\theta) \tag{28}$$

for k = 1: n° max iterations
  for i = 1 : n° episodes per batch
    for t = 1 : n° time steps per episode - 1
      - Sample policy $\pi \rightarrow K_R, K_V$
      - generate samples $(x_{i,t}, u_{i,t}, c_{i,t}, x_{i,t+1})$
    end for
  end for
  - fit $\hat{V}^{\pi}(x)$ to sampled cost-to-go $\{(s_{i,t}, \sum_{t'=t}^{T} \gamma^{t'-t} c(x_{i,t}, u_{i,t}))\}$
  for i = 1 : n° episodes per batch
    for t=1 : n° time steps per episode - 1
      - evaluate $\hat{A}_n{}^{\pi}(u_{i,t}, x_{i,t}) = \sum_{t'-t}^{t+n} \gamma^{t'-t} c(x_{i,t'}, u_{i,t'}) - \hat{V}(x_{i,t}) + \gamma^n \hat{V}(x_{i,t+n})$
      - evaluate $\nabla_\theta log \pi_\theta(u_{i,t}|x_{i,t})$
    end for
  end for
  - evaluate $\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta log \pi_\theta(u_{i,t}|x_{i,t}) \hat{A}_n{}^{\pi}(u_{i,t}, x_{i,t})$
  - update policy $\vartheta_k = \vartheta_{k-1} - \alpha \nabla_\vartheta J(\pi_\vartheta)$
  - test new policy $\pi_k \rightarrow$ obtain cumulative cost $C_k = \sum_{t=0}^{T} c(x_{k,t}, u_{k,t})$
  - calculate average change E in C among last 5 iteration
    if E < ε
      break
end for

Figure 5. Summary of the A-ZEM/ZEV algorithm

where α is the bounded learning rate. After each update, the algorithm is tested and the cumulative cost is computed

$$C_k = \sum_{t=0}^{T} c(x_t, u_t) \tag{29}$$

where k stands for k-th iteration. The algorithm stops if the average cumulative cost difference among the last 5 iteration is less than a tolerance ϵ or it has reached the maximum number of iterations. A summary of the algorithm in form of pseudo-code is given in Figure 5.

NUMERICAL RESULTS

The A-ZEM/ZEV algorithm is tested on the scenarios described below. Chaser and target are assumed to be on the same NRO, selected among the families presented above. In particular it is an orbit of the northern L_2 family, the periselene has an altitude of 1617 km over the Moon surface and the period is 6.17 days. The chaser is assumed to be inside the approach sphere and following the target along the orbit. 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 tested on two constraint scenarios:

- spherical objects standing between chaser and target
- a Keep-Out Sphere (KOS) with a conical approach corridor

The dynamics are described by the Clohessy-Wiltshire (CW) equations in proximity of the periselene and non-linear relative equations (NLR) in proximity of the aposelene. The results are presented separately for the two cases. The results obtained in the periselene region are also compared to an optimal solution obtained with GPOPS*. It should be noted that the possibility of solving the optimal problem with GPOPS also when the NLR equations are used was explored. The problem in that case becomes much more difficult to solve, mainly because of its number of states (both the

*General Pseudospectral OPtimal Control Software

chaser and the target absolute position must be tracked). This has proven to be a very cumbersome task to solve for the explicit method used by GPOPS and no acceptable solutions where obtained. To 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 algorithm is trained drawing the starting state of the samples from a gaussian distribution centered around the nominal starting state. At each iteration, the resulting guidance is tested 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 $K_V = -2$).

Spherical constraints

The constraints are represented by two spheres, assumed to be fixed in the LVLH frame. Their radii are 100 m and 70 m respectively. Their position is defined so that the Classical-ZEM/ZEV would not be acceptable. The cost function is represented by the expression 30. It is composed by a term related to the mass of propellant burned during the time-step (dm_t), two terms related to the end position and velocity errors with respect to the nominal target state that are added only in case $t = t_f$, and one term related to the position error of the impact point, if present, with respect to the target state.

$$C(t) = w_m dm_t + \delta(t - t_f)\left[w_r^f \|r_t - r_f\|^2 + w_v^f \|v_t - v_f\|^2 + b_f\right] + \delta(t - t_i)\left[w_r^i \|r_t - r_f\|^2 + b_i\right] \quad (30)$$

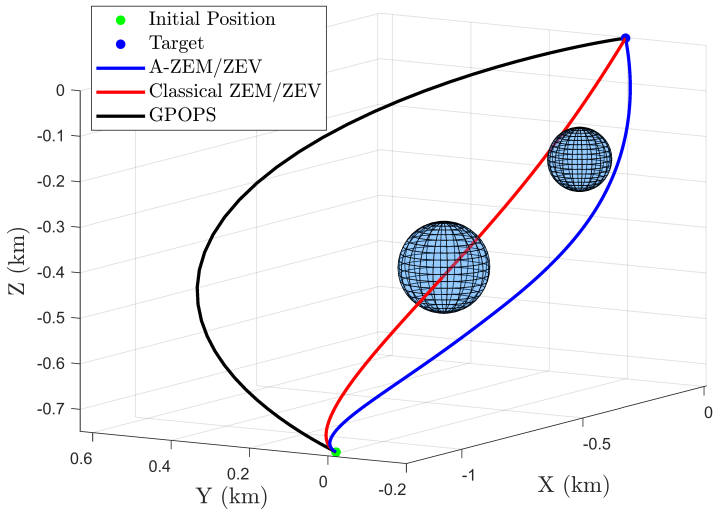
Where w_m, w_r^f, w_v^f and w_r^i are weights associated with the burned mass, the end position and velocity errors and the impact point position error respectively and b_f and b_i are biases added at the end of episodes with $b_i > b_f$. This ensures that the collision-less solution has a lower cost than a solution that impacts on the constraint. $b_f > 0$ instead ensures that the value function close to the target doesn't get too close to 0 which may cause problems during training because the error introduced by the function approximator might be big relatively to the actual value. The time-of-flight is set to be 6000 seconds at the periselene and 40000 at the aposelene.

Keep Out Sphere

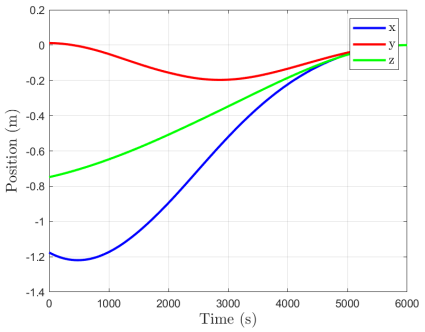
The second test scenario is represented by a keep-out sphere (KOS) described above, with an approach corridor aligned with an hypothetical docking port. The approach corridor is conical with a half cone angle equal to 15 degrees up until it becomes a 20 m radius cylinder with the same axis closer to the target. The cost function in this case is represented by equation 31. It is composed by the same kind of terms as for the spherical constraint case. The only difference is in the impact point error term that is related to the angular error of the impact point with respect to the axis of the approach corridor.

$$C(t) = w_m dm_t + \delta(t - t_f)\left[w_r^f \|r_t - r_f\|^2 + w_v^f \|v_t - v_f\|^2 + b_f\right] + \delta(t - t_i)\left[w_\theta \text{acos}\left(\frac{\mathbf{r}_i \cdot \mathbf{n}}{|\mathbf{r}_i||\mathbf{n}|}\right) + b_i\right] \quad (31)$$

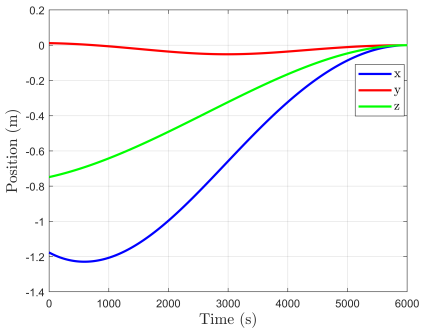
w_θ is a weight associated with the impact point angular error. $\mathbf{r}_i$ is the position vector of the impact point and $\mathbf{n}$ is the vector aligned with the approach corridor axis. In case the impact point is inside the approach corridor, the cost relative to the impact is instead associated with the distance of the



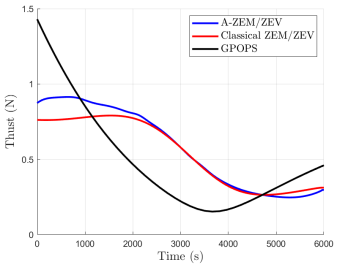
(a) Trajectory



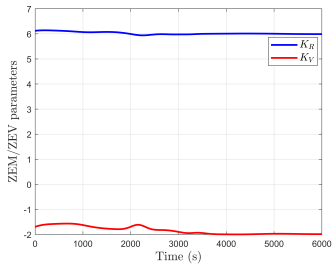
(b) Position



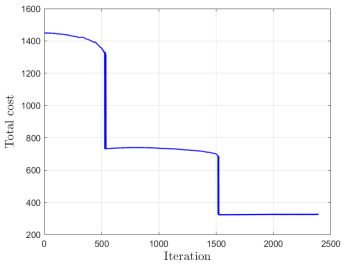
(c) Velocity



(d) Thrust

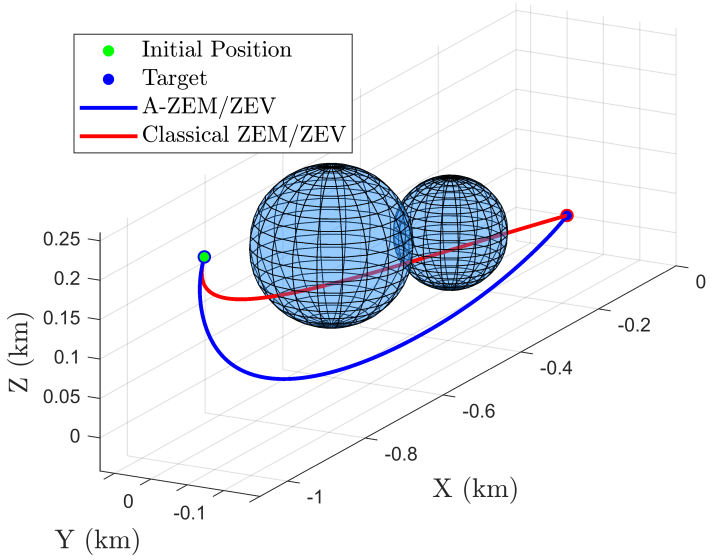


(e) Guidance gains

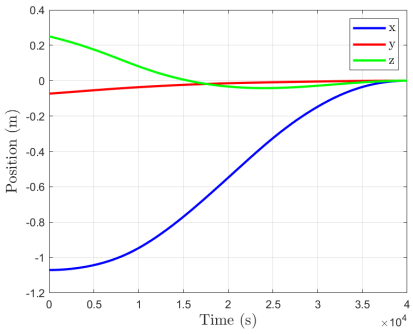


(f) Cost

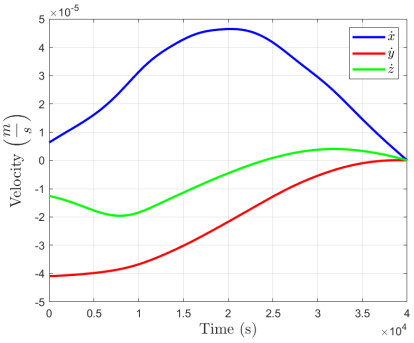
Figure 6. Spherical constraints problem at periselene. Fuel usage: A-ZEM/ZEV - 1.602 kg, Classical-ZEM/ZEV - 1.511 kg, GPOPS - 1.341 kg



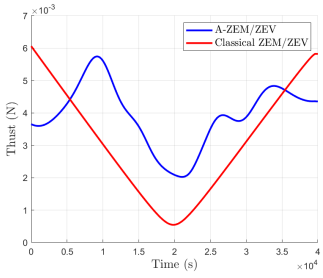
(a) Trajectory



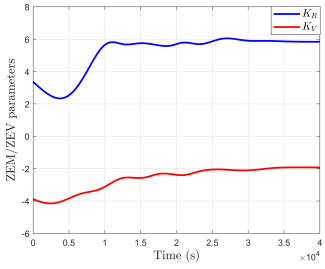
(b) Position



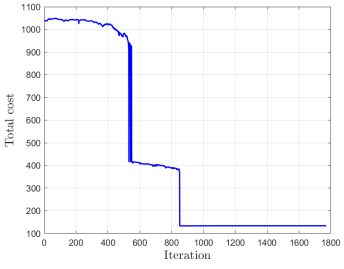
(c) Velocity



(d) Thrust



(e) Guidance gains



(f) Cost

Figure 7. Spherical constraints problem at aposelene. Fuel usage: A-ZEM/ZEV - 0.0728 kg, Classical-ZEM/ZEV - 0.0576 kg

impact point with respect to the target position as in the spherical constraint case (30). The time-of-flight is set to be 6000 seconds at the periselene and 40000 seconds at the aposelene.

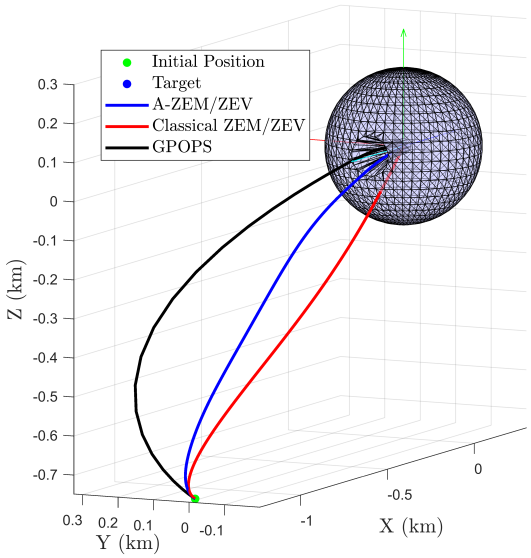
Analysis

As shown in Figures 6,7,8,9, A-ZEM/ZEV manages to find a solution that respects 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 doesn't respect the constraints in any case. In the spherical constraints case at the periselene A-ZEM/ZEV manages to first avoid the first sphere and then the second, as clearly shown by the behaviour of the cost in 6(f). In the KOS constraint cases, both at periselene and aposelene, A-ZEM/ZEV performs well managing to first align with the approach corridor and then avoid collisions with its "walls".

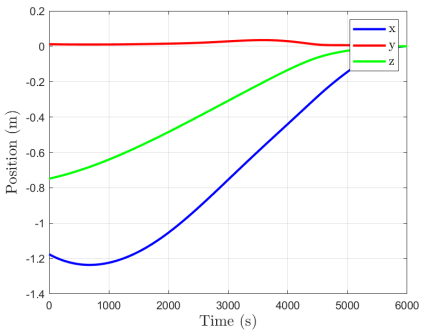
The task seems in general much more cumbersome in the aposelene region. In this case the dynamics are very slow and a much bigger change in the gains K_R and K_V is needed to obtain a reasonable change in the guidance acceleration. In particular the spherical constraints have proven to be the biggest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

The big jumps in the cost during training are due to the fact that in order to avoid falling in local minima and achieve a successful learning, the cost weights had to be tuned in order to have a clear separation in terms of cost between solutions that led to an impact with the constraints and collision-less solutions. For this reason, once the obstacle is avoided there is an abrupt change in cumulative cost. This doesn't allow for really satisfying results in terms of fuel consumption without the introduction of variable weights. Putting emphasis on fuel consumption minimization, so increasing the weight associated to the mass burned, would in fact degrade the collision avoidance capabilities in most cases. We decided to favor collision avoidance because the ability to embed constraints directly into the guidance law was considered more valuable than a further minimization of the fuel consumption.

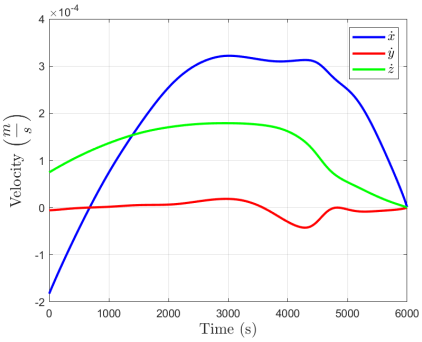
From a machine learning prospective, the usage of ELMs as critic has proven to be feasible. Figure 10 shows examples of value function approximation regression plots referred to single iterations. It clearly show that the ELM does a good job at capturing the variations in the value function. In particular it shows that even when the set has samples coming from both trajectories that hit the constraint and have big 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 parameters of the algorithm according to the case to solve, it was possible to maintain the regression accuracy high at all times to keep the bias controlled. Specifically, a 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 in reducing the regression error of the critic. This is due to the fact that this way the samples cover more densely the state space and allow for a smoother function to approximate. Moreover, the discount factor γ 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 most the overall cost-to-go, is valued less, which leads to a less stiff value function to approximate. Of course different values of γ also mean different results in terms of learning of the overall algorithm so its value was



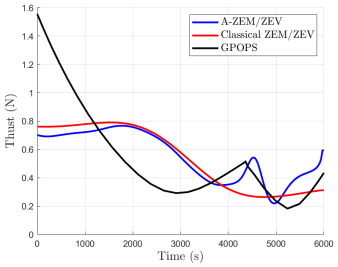
(a) Trajectory



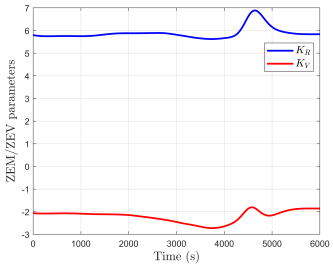
(b) Position



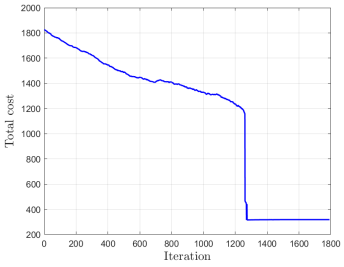
(c) Velocity



(d) Thrust

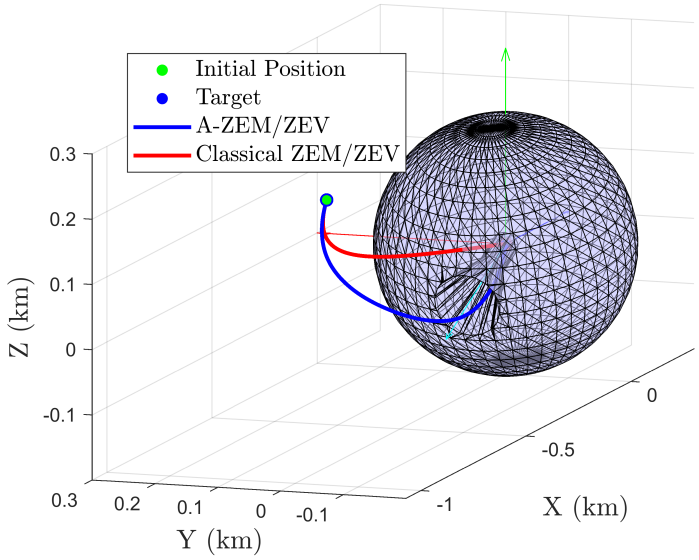


(e) Guidance gains

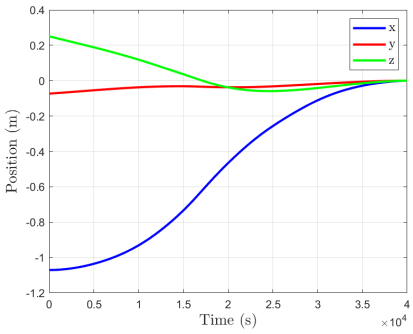


(f) Cost

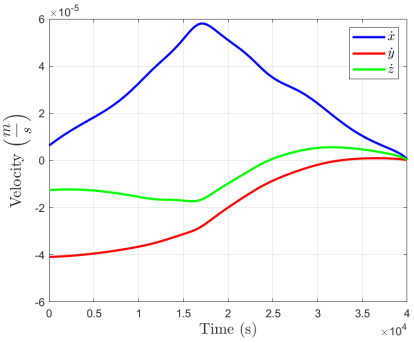
Figure 8. KOS constraint problem at periselene. Fuel usage: A-ZEM/ZEV - 1.543 kg, Classical-ZEM/ZEV - 1.511 kg, GPOPS - 1.485 kg



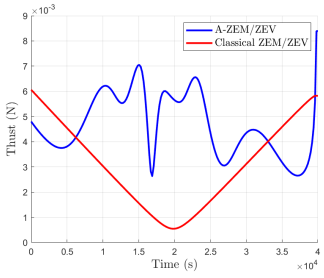
(a) Trajectory (top view)



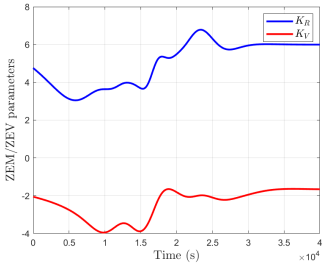
(b) Position



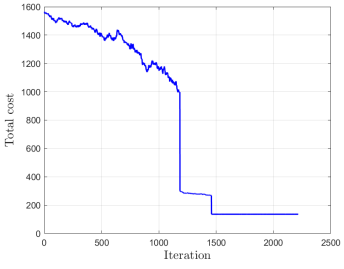
(c) Velocity



(d) Thrust



(e) Guidance gains



(f) Cost

Figure 9. KOS constraint problem at aposelene. Fuel usage: A-ZEM/ZEV - 0.0862 kg, Classical-ZEM/ZEV - 0.0576 kg

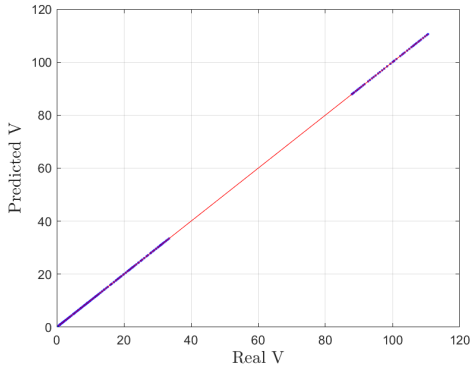


Figure 10. ELM regression plot for a random iteration

always the product of a compromise. Table 1 is a summary of the learning process as a whole where the fast training time of the ELM can be appreciated. It clearly shows that the training time for the ELM is in general 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 feasible and efficient.

Table 1. Critic network performance

Constraint	Model	N^o iterations	Total training time (hours)	Average iteration time (s)	Average critic training time (s)	Average critic NRMSE
Spherical	CW	2391	4.62	7.018	0.32	0.021
	NLR	1768	2.19	5.222	0.29	0.0029
KOS	CW	1787	6.64	13.300	0.39	0.0060
	NLR	2213	9.90	16.112	0.35	0.0017

CONCLUSION

While Classical-ZEM/ZEV has in its closed-loop nature and ease of implementation its biggest strengths, the fact that it is impossible to impose path constraints directly into the algorithm has limited its possible applications. This work has shown that using machine learning, in particular an actor-critic algorithm based on policy gradient, with advantage function estimation, it is possible to expand its capabilities way beyond what it was first designed for. The resulting *adaptive* algorithm (A-ZEM/ZEV) is capable of improving its performance in terms of collision avoidance capabilities in a variety of constraint scenarios. The A-ZEM/ZEV has demonstrated that an adaptive guidance algorithm based on ZEM/ZEV could be feasible for future missions in cislunar environment where non-linear equations of motion are used. Moreover, it has been shown to work extremely well in LVLH frame when the Clohessy-Wiltshire equations describe the dynamics of the system, which means that it could also be used for relative motion in any nearly circular orbits in the restricted two-body problem framework. Finally, from a machine learning prospective, it has been shown that an actor-critic algorithm that uses an Extreme Learning Machine as critic network is possible and learns efficiently. All in all, this shows that machine learning and artificial intelligence in general are valuable assets that should be taken into consideration in the design of new guidance algorithm for spacecraft guidance when autonomy and flexibility are pivotal.

REFERENCES

[1] P. A. Lightsey, C. B. Atkinson, M. C. Clampin, and L. D. Feinberg, "James Webb Space Telescope: large deployable cryogenic telescope in space," *Optical Engineering, 51(1), 011003*, 2012.

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

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

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

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

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

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

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

[14] Y. Zhang, Y. Guo, G. Ma, and T. Zeng, "Collision avoidance ZEM/ZEV optimal feedback guidance for powered descent phase of landing on Mars," *Advances in Space Research, 59(6), 1514-1525*, 2017.

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

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

[17] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

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

[19] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research 32.11*, 2013.

[20] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska, "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*, 2012.

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

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

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

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

[25] 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.

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

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

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