(Preprint) AAS 21-630

CLOSED-LOOP NEURAL CONTROL FOR THE FINAL PHASE OF A CISLUNAR RENDEZVOUS

Giangregorio Tofanelli*, Giordana Bucchioni†, and Mario Innocenti‡

The paper presents the design of a closed loop controller based on neural networks for the final rendezvous in a three body problem scenario with full six degrees of freedom dynamics. The reference scenario is a rendezvous maneuver between an active chaser and a passive target located on a near rectilinear halo orbit around the L2 Lagrangian in the Earth - Moon system. The guidance system synthesis is performed using "imitation learning" with two neural networks for attitude and translation. The performance of the controllers is evaluated with respect to a standard PID approach, as well as a high frequency, high performance state dependent Riccati equation algorithm, in order to evaluate the size of the learning database and capabilities of the two networks.

INTRODUCTION

The currently programmed first step in the future space exploration is the return to the Moon, with the construction of an orbiting space station called Lunar Orbital Space Gateway (LOP-G)[1].[2] The Gateway will be located on a Near Rectilinear Halo Orbit(NRHO) about the L2 collinear Earth - Moon Lagrangian point[3].[4] It will be used as a solar-powered communication hub, science laboratory, short-term habitation module and holding area for rovers and other robots located on the surface of our satellite. In this scenario, rendezvous and docking maneuvers have a fundamental role for the success of the mission. Usually, these proximity operations are divided in several parts, starting from far range rendezvous (after phasing) and ending with close range rendezvous (under 1 km or closer, where closed loop control is active).[5]

This paper refers to the close range maneuver performed by an actively controlled chaser with an uncooperative target. The main objective is the study of the potential of artificial intelligence algorithms to be implemented in the guidance and control loop. Although space missions critical elements rarely if not ever rely on this type of guidance solutions, the authors feel that, from a

*MSc in Aerospace Engineering, University of Pisa, Via Caruso 8, 56122, Pisa, Italy.

†PhD candidate, currently with ISAE-SUPAERO, 10, avenue Édouard-Belin BP 54032 - 31055 Toulouse CEDEX 4, France.

‡Professor, University of Pisa, Department of Information Engineering, Largo Lucio Lazzarino, 56126 Pisa, Italy. Associate Fellow AIAA.

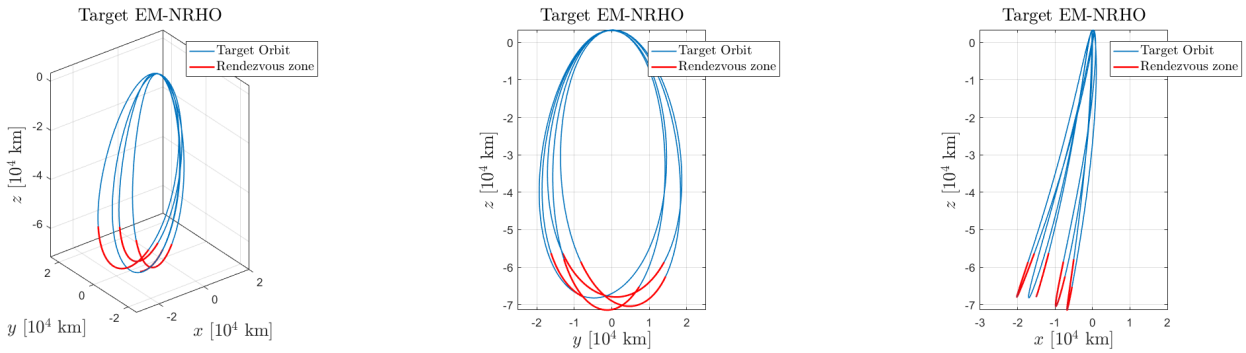
scientific standpoint, it is interesting to evaluate their performance against established and safe algorithms, for potential future implementation.
Closed loop guidance and control is necessary in the final approach phase of any automated rendezvous, and widely described especially for low Earth orbits (LEO), using Keplerian motion and relative dynamics[6][7][8] . In this context, the literature is very large, see,[9] and,[10] for instance.
In the case of the mission scenario of interest for this paper, the situation is different due to the many constraints involved, such as changes in sensors, safety issues and the need of fully automated operation, to mention a few. An additional problem is that, in this case, non Keplerian relative dynamics must be considered, at least at the restricted three body problem level. References[11] and[12] investigate a sliding mode approach and fixed optimization in close proximity of a halo orbit. Constant glideslope guidance was presented in,[13] and a state dependent Riccati equation (SDRE) approach, coupled with a standard PID control was presented in.[5]
This paper examines the use of ANN for the synthesis of a guidance system to be compared against some of the methods used in the above references. The paradigm followed to implement the closed-loop control with neural networks is based on the idea of *Imitation Learning*[14][15][16] . The aim of imitation learning is to learn a control policy by reproducing the behaviour of an *expert agent*. The policy synthesised from the agent's expertise should lead to the design of control solutions closely approximating those used to train the neural networks. The neural networks will be fed with state-action pairs originating from expert agent's controlled trajectories.

MISSION SCENARIO REVIEW

The Earth-Moon system considered in this study, refers to the Human Lunar Exploration Program (HLEPP). The study considers a human-assisted robotic mission on the lunar surface where rovers are tele-operated by astronauts on board a station in lunar orbit, the Deep Space Gateway (DSG). The access to the station by incoming vehicles from both the Earth and the lunar surface is one of the most critical aspects of the mission. The study of the relative dynamics for this scenario and the design of maneuvers for rendezvous and docking with the DSG will provide a step forward in the overall mission design. The candidate family of orbits for the DSG are the Earth-Moon L_2 9-2 south subfamily NRHO. An example is shown in Fig.1. The orbit was provided by ESA, and was obtained simulating the target motion considering the CR3BP equations for the Earth-Moon system, plus the influence of the solar radiation pressure.[17] In Fig.1 the region of the orbit suitable for rendezvous is highlighted in red. The rendezvous region is then identified by the *mean anomaly* of the NRHO:

$$M(t) = 2\pi \frac{t}{T}$$

where, in analogy with Keplerian orbits, *t* is the time from the *periselene* passage, and T is the orbit period. For the orbit in the figure, the average period is $T = 6.2834$ days, and the *aposelene* is 70,754 km from the Moon's surface.



((a)) Three-dimensional view **((b))** Projection on y-z plane **((c))** Projection on x-z plane

Figure 1: Reference Earth-Moon L_2 South NRHO Orbit for target, in the Moon centered reference frame $\mathcal{M}$. In red the preferable rendezvous zone.

Reference Frames

The relative motion of the chaser with respect to the passive DSG requires the use of three-body dynamics described using several reference frames. The most common are:

Inertial Frame Without loss of generality, the assumption on the primaries composite center of mass C is that is inertially fixed. Then, introducing the following *Inertial Frame*, centered in C:

$$\mathcal{I} = \{C; \hat{\boldsymbol{I}}, \hat{\boldsymbol{J}}, \hat{\boldsymbol{K}}\}$$

Three-Body Rotating Reference Frames A convenient frame for describing the motion of a spacecraft subjected to the gravitational influence of two rotating primaries is the *Synodic Reference Frame*:

$$\mathcal{S} = \{C; \hat{\boldsymbol{i}}_s, \hat{\boldsymbol{j}}_s, \hat{\boldsymbol{k}}_s\}$$

The frame is centered at the two primaries composite center of mass C, with unit vectors as follows:

- $\hat{\boldsymbol{i}}_s = -\dfrac{\boldsymbol{r}_{12}}{||\boldsymbol{r}_{12}||}$ where $\boldsymbol{r}_{12}$ is the position of M_2 with respect to M_1.
- $\hat{\boldsymbol{k}}_s$ is perpendicular to the plane where the primaries revolve, and is positive in the direction of the system angular velocity vector.
- $\hat{\boldsymbol{j}}_s = \hat{\boldsymbol{k}}_s \times \hat{\boldsymbol{i}}_s$ completes the right-handed coordinate systems.

As a consequence, the frame co-rotates with the primaries, and the latter appear stationary in the frame. In particular, the frame rotates with respect to the inertial frame with angular velocity

$\boldsymbol{\omega}_{s/i}$ that, in general, is not constant. By translating the synodic frame to the center of one of the primaries, one obtains the *Primary-Centered Rotating Frame*, which will be denoted as follows:

$$\mathcal{M}_i = \{O_i; \hat{\boldsymbol{i}}_m, \hat{\boldsymbol{j}}_m, \hat{\boldsymbol{k}}_m\}$$

where i is the index of the primary chosen for placing the coordinate frame. Figures (2) and (3) show the synodic reference frame and the primary-centered rotating frame, centered at M_2. This choice may be convenient when spacecraft measurements are taken with respect to the nearest primary.[18] If the Sun-Earth/Moon system is considered, the synodic frame is usually centered at the Earth.

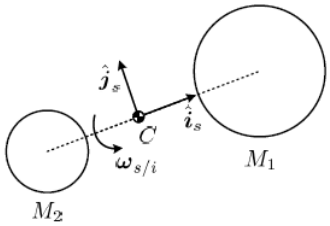


Figure 2: Synodic reference frame *S*

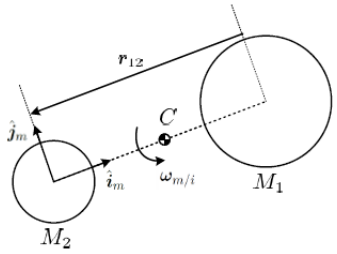


Figure 3: Primary-centered rotating reference frame $\mathcal{M}_2$

Local-Vertical Local-Horizon Frame Rendezvous trajectories are generally described in a frame local to the target. This facilitates the analysis and the trajectory monitoring of incoming vehicles, as well as the definition of keep-out zones and admissible approaching corridors. To this end the LVLH frame is usually employed and is denoted as:

$$\mathcal{L} : \{\boldsymbol{r}_{it}; \hat{\boldsymbol{i}}, \hat{\boldsymbol{j}}, \hat{\boldsymbol{k}}\}$$

The LVLH frame is defined with respect to the primary body around which the target is orbiting. Denoting with $\boldsymbol{r}_{it}$ the target position with respect to the primary *i*, with $[\dot{\boldsymbol{r}}_{it}]_{\mathcal{M}_i}$ the target velocity as seen from the primary, and with $\boldsymbol{h}_{it} = \boldsymbol{r}_{it} \times [\dot{\boldsymbol{r}}_{it}]_{\mathcal{M}_i}$ the target specific angular momentum with respect to the primary, the LVLH frame unit vectors are defined and named as follows:

- $\hat{\boldsymbol{k}} = -\dfrac{\boldsymbol{r}_{it}}{||\boldsymbol{r}_{it}||}$ points to the primary and is called the *R-Bar*.
- $\hat{\boldsymbol{j}} = -\dfrac{\boldsymbol{h}_{it}}{||\boldsymbol{h}_{it}||}$, is perpendicular to the target instantaneous orbital plane and is called *H-Bar*.
- $\hat{\boldsymbol{i}} = \hat{\boldsymbol{j}} \times \hat{\boldsymbol{k}}$ completes the right-handed reference frame, and is called *V-Bar*.

The above definition of the LVLH frame (with some abuse of notation) is consistent with the one given by Fehse in its classical reference book for spacecraft rendezvous and docking.[6] The LVLH frame for a target orbiting around M_2 is shown in Fig. 4.

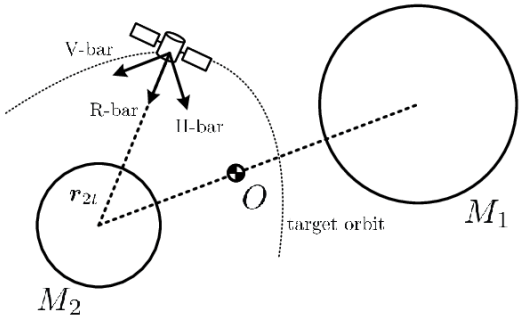


Figure 4: Local-vertical local horizon frame $\mathcal{L}_2$

Body fixed reference frames are also used, when dealing with the actual docking/berthing operations. Their selection depends on the hardware used for such operations and the location of the docking cone or hook for the arm gripper on the vehicles.

Dynamical System Modelling

In he final phase of the rendezvous both relative translational and rotational motions are considered. Since the target station is considered passive, the control input is that of the chaser only.
The relative translational dynamics in the LVLH frame and nondimensional units were modeled according to the work of one of the authors,[19] and repeated here for clarity's sake:

$$[\ddot{\boldsymbol{\rho}}]_{\mathcal{L}} = -2\boldsymbol{\omega}_{l/i} \times [\dot{\boldsymbol{\rho}}]_{\mathcal{L}} - \left[\dot{\boldsymbol{\omega}}_{l/i}\right]_{\mathcal{L}} \times \boldsymbol{\rho} - \boldsymbol{\omega}_{l/i} \times \left(\boldsymbol{\omega}_{l/i} \times \boldsymbol{\rho}\right) + \\ +\mu \left(\frac{\boldsymbol{r}}{r^3} - \frac{\boldsymbol{r}+\boldsymbol{\rho}}{||\boldsymbol{r}+\boldsymbol{\rho}||^3} \right) + (1-\mu) \left(\frac{\boldsymbol{r}+\boldsymbol{r}_{em}}{||\boldsymbol{r}+\boldsymbol{r}_{em}||^3} - \frac{\boldsymbol{r}+\boldsymbol{r}_{em}+\boldsymbol{\rho}}{||\boldsymbol{r}+\boldsymbol{r}_{em}+\boldsymbol{\rho}||^3} \right) \tag{1}$$

Where $\boldsymbol{\rho}$, is the relative position of the chaser with respect to the target, $\boldsymbol{r}$ is the position of the target with respect to the Moon, μ is the Earth-Moon mass ratio parameter, $\boldsymbol{r}_{em}$ is the position of the Moon with respect to the Earth and $\boldsymbol{\omega}_{l/i}$ is the angular velocity of $\mathcal{L}$ (LVLH reference frame) with respect to $\mathcal{I}$ (Inertial Frame), as shown in Fig. 5.

Relative rotational motion was also taken from the authors' previous work and details can be fount in.[20] The target's attitude dynamics are modeled considering the target to be uncooperative

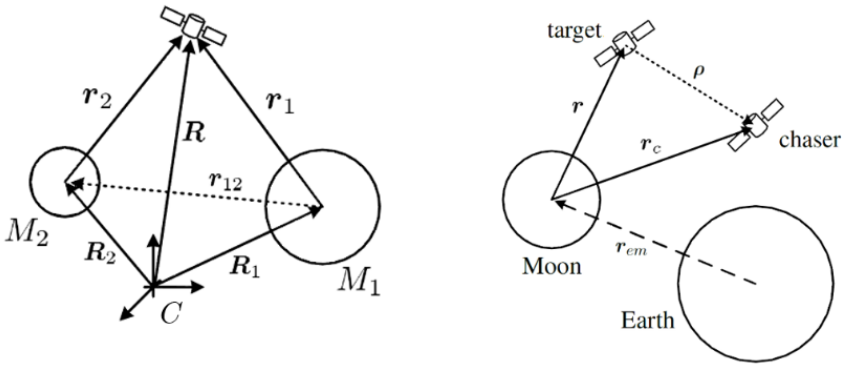


Figure 5: Three-Body Problem dynamic variables

following a sawtooth behaviour[7] jointly with the quaternion kinematic equation, given by:

$$\begin{cases} \dot{\boldsymbol{\omega}}_{t/l} = -\boldsymbol{K}_{q_t}\boldsymbol{q}_{t/l} \\ \dot{\boldsymbol{q}}_{t/l} = \dfrac{1}{2}\Xi(\boldsymbol{q}_{t/l})\boldsymbol{\omega}_{t/l} \end{cases} \tag{2}$$

The chaser's attitude dynamics are modeled considering Euler equations and the quaternion kinematic equation as:

$$\begin{cases} \dot{\boldsymbol{\omega}}_{c/l} = -\mathbb{I}^{-1}\left(\omega \times \mathbb{I}\omega\right) + \mathbb{I}^{-1}\boldsymbol{N} \\ \dot{\boldsymbol{q}}_{c/l} = \dfrac{1}{2}Q(\boldsymbol{\omega}_{c/l})\boldsymbol{q}_{c/l} \end{cases} \tag{3}$$

The relative attitude dynamics can then be computed as:

$$\boldsymbol{q}_{ra} = \boldsymbol{q}^*_{t/l} \otimes \boldsymbol{q}_{c/l} \tag{4}$$

Finally, port-to-port motion is given by:

$$\begin{cases} \boldsymbol{\rho}_{pp} = \boldsymbol{\rho} + \boldsymbol{r}_{pc} - \boldsymbol{r}_{pt} \\ \dot{\boldsymbol{\rho}}_{pp} = \dot{\boldsymbol{\rho}} + \boldsymbol{R}(\boldsymbol{q}_{c/l})\boldsymbol{\omega}_{c/l} \times \boldsymbol{r}_{pc} - \boldsymbol{R}(\boldsymbol{q}_{t/l})\boldsymbol{\omega}_{t/l}\boldsymbol{r}_{pt} \end{cases} \tag{5}$$

Since the final phase of the rendezvous occurs at the aposelene, the above equations can be linearized in order to obtain a representation feasible for closed loop control synthesis. In state space form, this leads to the standard form:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{cases} \tag{6}$$

Where:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\rho}^T & \dot{\boldsymbol{\rho}}^T & \boldsymbol{q}_{c/l}^T & \boldsymbol{\omega}_{c/l}^T & \boldsymbol{q}_{t/l}^T & \boldsymbol{\omega}_{t/l}^T & \boldsymbol{q}_{ra}^T \end{bmatrix}^T$$

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{F}^T & \boldsymbol{N}^T & [\boldsymbol{r_{pc}}]_{\mathcal{L}}^T & [\boldsymbol{r_{pt}}]_{\mathcal{L}}^T \end{bmatrix}^T$$

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{\rho}^T & \dot{\boldsymbol{\rho}}^T & \boldsymbol{q}_{c/l}^T & \boldsymbol{\omega}_{c/l}^T & \boldsymbol{q}_{t/l}^T & \boldsymbol{\omega}_{t/l}^T & \boldsymbol{\rho}_{pp}^T & \dot{\boldsymbol{\rho}}_{pp}^T & \boldsymbol{q}_{ra}^T & \boldsymbol{\omega}_{ra}^T \end{bmatrix}^T$$

For the sake of simplicity and brevity the state and control matrix, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ have been omitted and can be found in.[20]

GUIDANCE AND CONTROL SOLUTIONS

This section reviews briefly the guidance and control algorithms used for training the neural networks described in the next section. The algorithms are selected based on the simulation tool described in.[21] As a remainder we are only dealing with the final part of the rendezvous, where the relative distance decreases starting from 1 km as initial condition.

SDRE Approach

The rationale behind the selection of a SDRE controller stems from its capabilities of handling a set of nonlinear systems (affine in the control), and to produce a control law based on a LQR-like structure, once the dynamics are parametrized to yield a state dependent plant and input matrices as shown in Eq. (7). The parametrization is not unique and provides added freedom to the designer. Local controllability, observability, and stability properties can also be verified. In addition, since we are dealing with a continuous system, the resulting controller may require high bandwidth (in fact theoretically infinite), thus stressing the neural network design and the size of the database needed for a successful training.

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \\ \min\limits_{\boldsymbol{u}} \boldsymbol{\mathcal{J}} = \frac{1}{2}\int_0^\infty \boldsymbol{x}^T\boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}(\boldsymbol{x})\boldsymbol{u}dt + \frac{1}{2}\int_0^\infty \boldsymbol{z}^T\boldsymbol{W}_z(\boldsymbol{x})\boldsymbol{z}dt \\ l(\boldsymbol{x}) = -\left[\boldsymbol{\rho}^T\right]_{\mathcal{T}} \cdot [\boldsymbol{p}]_{\mathcal{T}} + \rho\cos\beta \end{cases} \tag{7}$$

For details on SDRE the reader can refer to the tutorial,[22] and the application to our scenario described in.[5]

The SDRE structure shown in Eq.(7) results in a controller that minimizes a state dependent quadratic cost $\boldsymbol{\mathcal{J}}$, where the third term incorporates the final approach state constraints[23] $l(\boldsymbol{x})$. The term $\boldsymbol{z}$ is a fictitious output to include the presence of the constraint in the cost function, $\boldsymbol{W}_z$ is a weight function that penalizes the state when it is far from the imposed constraint. The constraint expressed in the target's body frame is composed by the target's path unit vector $\boldsymbol{p}$, while β is the maximum approach angle of the final desired corridor[24],[5] and qualitatively shown in Fig. 6.

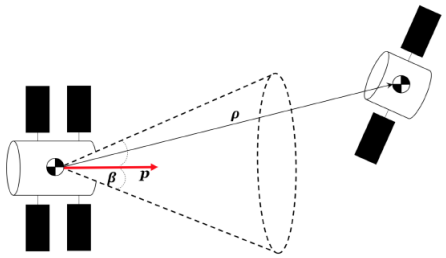


Figure 6: Proximity example scenario.

The final expression of the controller is given by:

$$\boldsymbol{u} = -\boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} = -\left[\boldsymbol{K}_0(\boldsymbol{x}) + \boldsymbol{K}_\Omega(\boldsymbol{x})\right]\boldsymbol{x}$$

Where:

$$\boldsymbol{K}_0(\boldsymbol{x}) = \bar{\boldsymbol{R}}^{-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\bar{\boldsymbol{P}}(\boldsymbol{x})$$
$$\boldsymbol{K}_\Omega(\boldsymbol{x}) = \bar{\boldsymbol{R}}^{-1}(\boldsymbol{x})\boldsymbol{B}^T(\boldsymbol{x})\boldsymbol{W}_z(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})$$

is the solution of the associated state dependent algebraic Riccati equation for the system augmented with the constraint.

$$\bar{\boldsymbol{A}}^T(\boldsymbol{x})\bar{\boldsymbol{P}}(\boldsymbol{x}) + \bar{\boldsymbol{P}}(\boldsymbol{x})\bar{\boldsymbol{A}}(\boldsymbol{x}) - \bar{\boldsymbol{P}}(\boldsymbol{x})\bar{\boldsymbol{B}}(\boldsymbol{x})\bar{\boldsymbol{R}}^{-1}(\boldsymbol{x})\bar{\boldsymbol{B}}^T(\boldsymbol{x})\bar{\boldsymbol{P}}(\boldsymbol{x}) + \bar{\boldsymbol{Q}}(\boldsymbol{x}) = 0$$

Classical PID Controller

In this case, a classical PID approach was used to compute translation and attitude controllers. A basic trial and error tuning yielded a PD controller for the translational dynamics and a proportional controller for the attitude dynamics. Fig.7 shows the block diagrams for translation while Fig.8 the attitude's one.

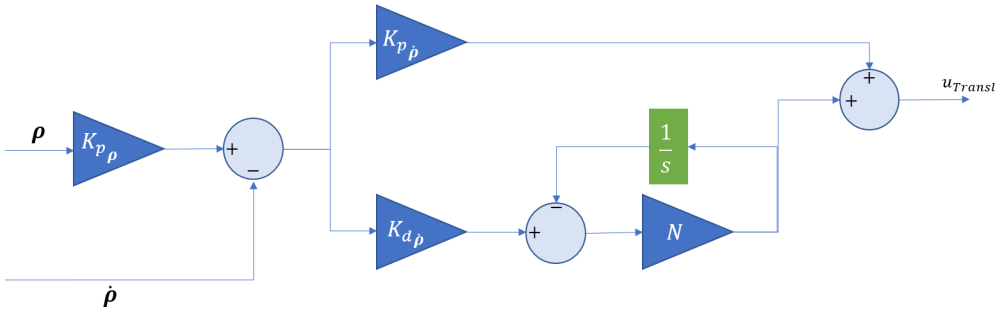


Figure 7: Translation: PD controller structure

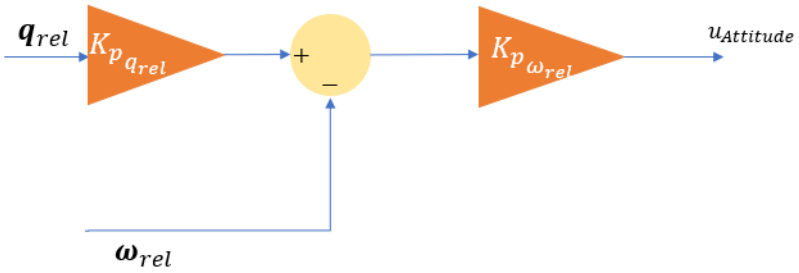


Figure 8: Attitude: Proportional controller structure

NEURAL NETWORK DESIGN

This section presents the approach to the synthesis of a neural network controller based on the training provided by the previously presented control laws.

Background

Thanks to the ever decreasing cost of computational resources and advances in research to train neural networks with many hidden layers,[16] there is a renewed interest in artificial neural networks (ANN), and in particular deep neural networks (DNN), in the context of optimal control. Successful applications of neural networks adopted for control purposes have been demonstrated for

simple problems involving control of continuous time, deterministic, non-linear systems.[14]
Artificial Neural Networks provide one of the most popular and capable applications in the field of machine learning in our decade. In particular DNNs, i.e. networks with a large number of hidden layers, are frequently deployed to learn a model from a database of examples using some form of gradient descent.
References,[15][25] for instance, provide a systematic study on how DNNs can be trained on optimal state feedback of continuous time, deterministic, non-linear systems. It was demonstrated that a neural network is capable of learning all tasks with remarkable accuracy and, for the multi-copter and spacecraft scenario, the network generalizes well outside the bounds of the training set whenever deep network topologies are deployed.
Supervised/imitation learning has been adopted for interplanetary trajectories,[26] showing how it is possible to train a DNN to guide a spacecraft optimally from an Earth orbit to a Mars orbit. The same authors introduced G&CNETs to indicate a generic deep architecture trained to perform optimal maneuvers using the imitation learning paradigm.[27] G&CNETs are one of the most promising Deep Learning based technologies that can potentially simplify the on board control and guidance software replacing it with one, relatively simple, trained neural model.
An extensive work has been done adopting Recurrent Neural Networks based on Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN)[28] for visual navigation purposes. Other solutions involving neural networks regard Reinforcement Learning (RL), a large literature regarding its application related to optimal control related to a space scenario is provided by the work of Furfaro and coworkers. Rendezvous problems were addressed considering a 3DOF[29] and a 6 DOF[30] dynamics.
To the best of the authors' knowledge "Imitation Learning" has never been applied in the context of proximity operations of interest in this paper.

Network Synthesis Outline

Imitation learning is a control design paradigm that seeks to learn a control policy reproducing demonstrations from expert agents. By substituting expert demonstrations for optimal behaviours, the same paradigm leads to the design of control policies closely approximating the ones provided by the agent. This approach requires training a machine learning algorithm (in our case artificial neural networks) directly on state-control pairs originating from agent's trajectories.
The first step in the design was the training dataset generation. For each algorithm in the previous section, we have propagated a set (N=1000) of trajectories using the ROSSONERO tool,[21] generating two datasets containing $\langle \boldsymbol{x}, \boldsymbol{u} \rangle$ pairs, where $\boldsymbol{x}$ is the state and $\boldsymbol{u}$ is the corresponding control action. The trajectories were split into two groups: 50 of them were randomly sampled to be used in a posteriori test, while the others were used to train the neural networks. In particular, 80% of the samples were the training set and the remaining 20% were the validation set. The data were then processed in order to feed them to the neural networks. In particular, the variables of interest were scaled to have 0-mean and 1-variance; additionally the target (control) data were also scaled between [-1, +1] in order to be represented by the hyperbolic tangent activation function in the

output layer. The activation functions selected for the network were: *ReLU* for the hidden layers, due its advantages solving the vanishing gradient problem, higher computational efficiency and better convergence performances, and *Tanh* for the output layer because of the bounded nature of the control signals.
After an initial tuning it was decided to separate the translational and attitude control tasks. So 2 ANNs one handling the translational control and the other the attitude control, were trained on the basis of the data generated from the trajectories controlled by the SDRE and by the PID control respectively. The translational ANN's input layer has a size of 6 neurons corresponding to $\boldsymbol{x}_{Translational} = \begin{bmatrix}\boldsymbol{\rho}^T & \dot{\boldsymbol{\rho}}^T\end{bmatrix}^T$ while the output layer has a size of 3 corresponding to the chaser's relative acceleration $\boldsymbol{u}_{Translational}$. For the attitude network the input layer has a size of 7 corresponding to attitude state $\boldsymbol{x}_{Attitudel} = \begin{bmatrix}\boldsymbol{q}_{rel}^T & {\boldsymbol{\omega}_{rel}}^T\end{bmatrix}^T$ and 3 for the output layer associated with the attitude control torques $\boldsymbol{u}_{Attitudel}$.

Training

The networks are trained for 100 epoch with *Adam*[31] and a batch size of 256 samples. The weights were initialized using Xavier's initialization method.[32] The optimization process aims to find the local minima of the *Mean Squared Error* (MSE)

$$\min_{\boldsymbol{u}} L(\boldsymbol{w}) = \frac{1}{2N}\sum_{q=1}^{256}\sum_{i=1}^{m}\left(u_j^q - \hat{u}_j^q\right)^2 \tag{8}$$

Where $\hat{u}_j^q$ is the control signal computed by the ANN related to the q_{th} mini batch's sample, u_j^q is control command computed by means of the SDRE/PID, m is the number of output and in the end $\boldsymbol{w}$ is the weight vector. Backpropagation was performed using the *Adam* optimization shown in Eq. (9).

$$w_{ji,t} = w_{ji,t-1} - \eta\frac{\hat{m}_t}{\sqrt{\hat{v}_t}} \tag{9}$$

The hyperparameters selected were: $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\eta = 0.001$, yielding:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_{\boldsymbol{w}} L(\boldsymbol{w}) \longrightarrow \hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{10}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\nabla_{\boldsymbol{w}}^2 L(\boldsymbol{w}) \longrightarrow \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{11}$$

Due to the non-convex optimization nature of the problem the attention was restricted to feed-forward, fully-connected networks with equal number of units in the hidden layers. 25 architectures were trained for each case, varying the number of layers and the number of neurons per layer, while keeping both the optimization hyper-parameters and activation function fixed. After the training, the ANNs with the lowest MSE for the training and validation data were in closed-loop replacing the respective controllers.

NEURAL NETWORK STRUCTURES

This section describes the structure of the neural networks such as number of neurons, hidden layers, etc. based on training and evaluation.

Initial Condition Dataset Generation

The initial conditions set was generated with the *Rossonero* simulation tool[17] limited to the close range rendezvous, with perfect models for the sensors and actuators. The elliptic linearized equations of relative motion, near the aposelene were used.
The maneuver starts when the relative distance between target and chaser is 1 km and the same initial conditions were used for both controllers. In particular:

- relative distance: $||\boldsymbol{\rho}||$ = 1 km,
- $||\dot{\boldsymbol{\rho}}|| \leq 2$ cm/s in which each $\dot{\rho}_i \in [-1, 1]$ cm/s,
- elevation and azimuth $El, Az \in [-0.2, 0.2]$ rad,
- $\boldsymbol{\theta_{c/l}}$ satisfying a target pointing strategy,[24]
- $\omega_{c/l_i} \in [-1, 1]$ deg/s,
- $\boldsymbol{\theta}_{t/l} = \boldsymbol{\omega}_{t/l} = \mathbf{0}$ and $M = 180$ deg.

All variables herein considered sampled using random uniform distribution in their range. The validity of the resulting trajectories, for both SDRE and classical controllers, was measured using the following performance measures:

$$\Delta V = \int_{t_0}^{t_f} ||\boldsymbol{u}_{Translational}(t)||dt, \Delta W = \int_{t_0}^{t_f} ||\dot{\boldsymbol{\omega}}(t)||dt \tag{12}$$

After propagation the trajectories performance parameters are shown in Tab.1. The results were satisfactory for both controllers, with rendezvous achieved.

Variable	SDRE Control	PID Control
T	2 Hours	1 $Hour$
$\bar{e}_{\boldsymbol{\rho}}$	~ 4.8 m	~ 5.6 m
$\sigma_{\boldsymbol{\rho}}$	0.16	0.032
$\bar{e}_{\dot{\boldsymbol{\rho}}}$	$\sim 1 \times 10^{-9}$ mm/s	~ 0.015 m/s
$\sigma_{\dot{\boldsymbol{\rho}}}$	$\sim 1 \times 10^{-10}$	$\sim 8.65 \times 10^{-6}$
$\bar{\Delta V}$	~ 0.74 m/s	~ 2.32 m/s
$\sigma_{\Delta V}$	~ 0.27	0.057
$\bar{\Delta W}$	~ 4.03 deg/s	~ 3.085 deg/s
$\sigma_{\Delta W}$	1.1454	~ 0.87

Table 1: Training trajectories comparison

The trajectory propagation used the **ode45** integration algorithm and the samples produced by each trajectory are shown in Tab. 2. Considering these large datasets generated, the following optimized libraries were used: *Tensorflow* and *Keras* executed on a processor: Intel(R) Core(TM) i7-10710U CPU 1.10 GHz.

	SDRE Control	PID Control
Integration Steps	~ 7500	~ 1500
Number of trajectories	1000	1000
Tot. Samples	7.5M	1.5M

Table 2: Trajectories statistical analysis

The training sets for both controllers are shown in Figures 9 and 10.

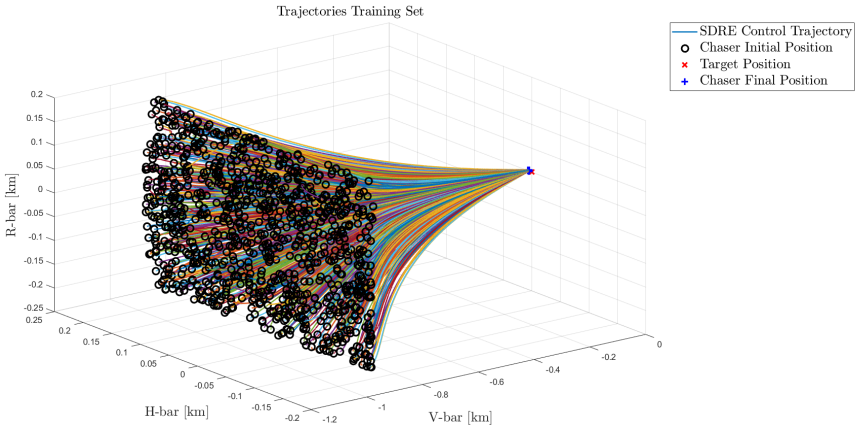


Figure 9: SDRE training set

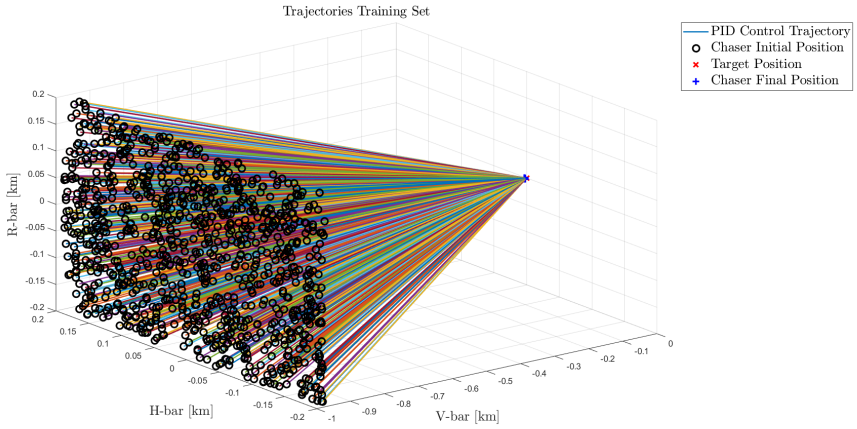


Figure 10: PID training set

SDRE-based Training Evaluation

The networks selected for this case have 4 layers and 50 neurons per layer for both the translational and the attitude (see Fig. 13). The results of the ANN training based on SDRE controller are shown in Fig.11.
The top row indicates a difference in the translational network performance, although the computed error is similar. This is an indication of an unrepresentative dataset. This is due to the fact that the control law generated by the solution of the algebraic Riccati equations produces high frequency content signals, which manifest themselves with impulses that are completely different in terms of amplitude and instant as seen in the validation curves.The attitude control shows a good fit instead. The validation curves follow the same profile of the training curves and again increasing the width and depth of the neural network increase its learning capabilities because of lower MSE reached.

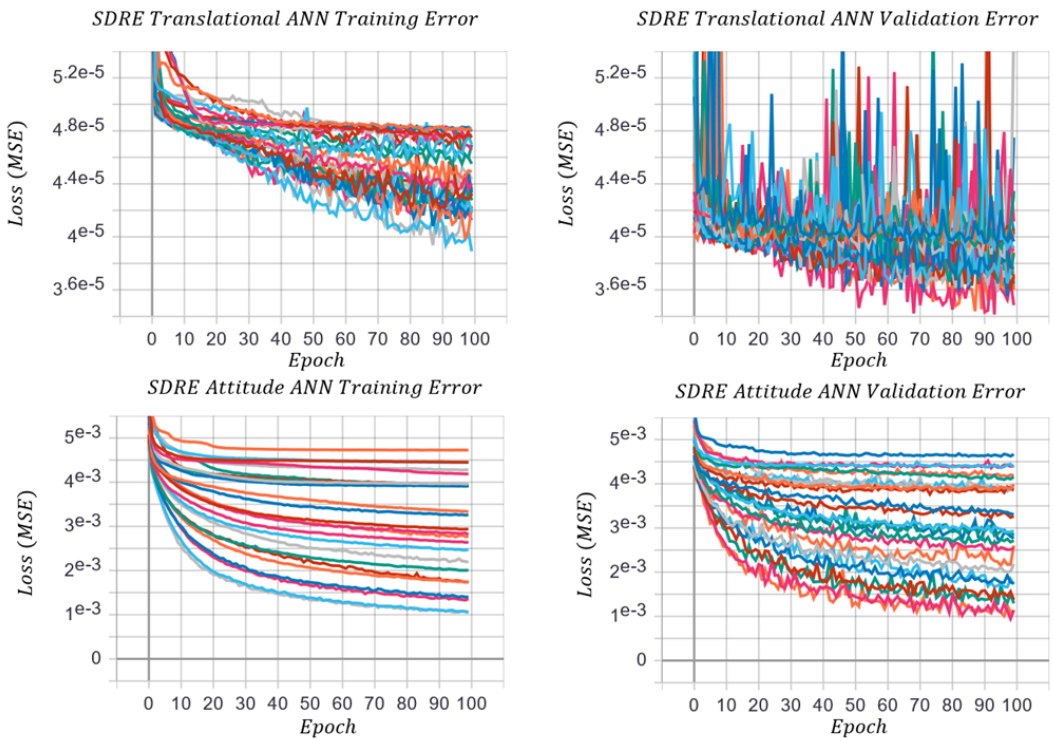


Figure 11: SDRE Training

PID-based Training Evaluation

The networks selected for this case consist of 2 layers and 50 neurons per layer for the translational ANN and 3 layers and 20 neurons per layer for the attitude ANN (see Fig. 13).
Training and validation results are presented in Fig. 12. In this case, both the translational and the attitude networks present a good fit since all architectures reach similar MSE level in almost all cases. Moreover, learning capabilities increase with network's width and length. In fact, in the PID translational learning curve, the first one (orange for the training data & blue for the validation data) corresponds to the ANN with 1 hidden layer and 10 neurons per layer. The increments of width and length boost the learning capabilities allowing to decrease the error.
In both figures the legend was omitted for simplicity reasons, different colours refer to different architectures, and there is no correspondence between the colour of training and validation curve.

NEURAL NETWORK VALIDATION

After the training, the selected ANN are inserted in the closed-loop system. 50 trajectories were randomly sampled from the dataset created for the training and were used to assess the ANN capabilities.

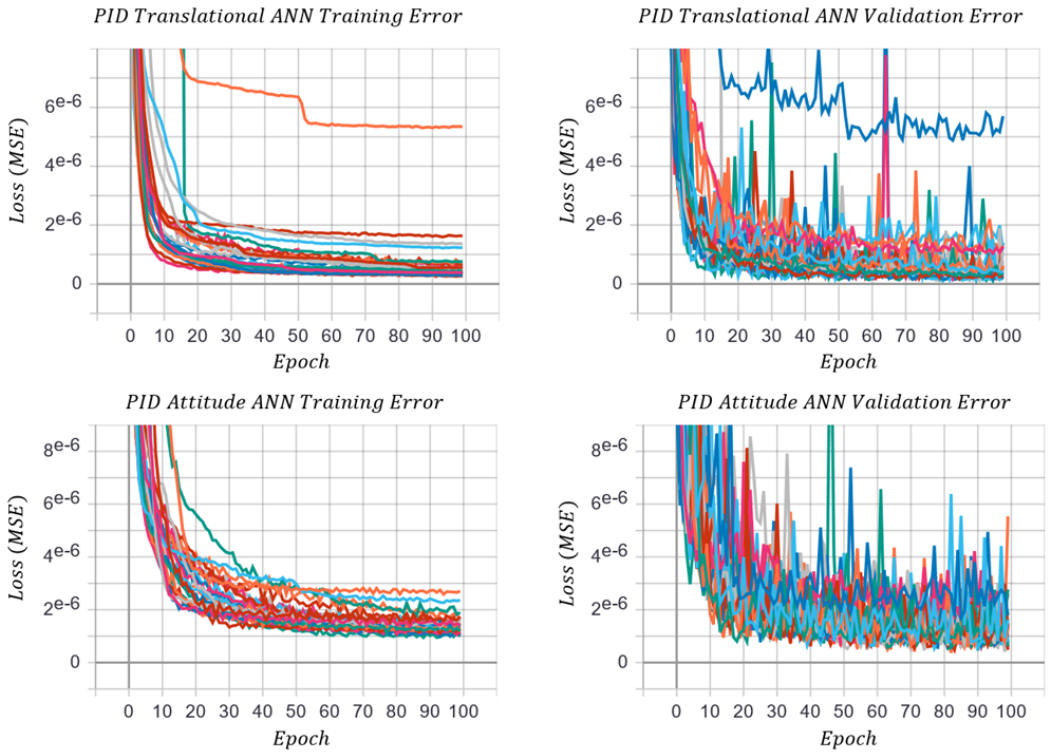
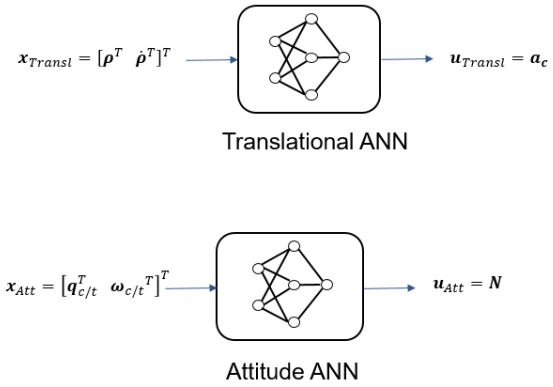


Figure 12: PID Training



ANN	*Layers*	*Neurons*
SDRE Tranlational	4	50
SDRE Attitude		
PID Translational	2	50
PID Attitude	3	20

Figure 13: Neural Networks Data

SDRE Network Performance

As anticipated in the training and validation results for translational control, the whole test set of trajectories propagated substituting the ANN synthesized on the SDRE basis with the SDRE itself

are unsatisfactory as indicated by Tab. 3. A subset of trajectories diverges from the expected target position in a random fashion, while a second subset does not provide control effort indicated by the original SDRE controller and depicted in Fig. 14.

Variable	SDRE Value	ANN_{SDRE} Value
T	2 Hours	2 Hours
$\bar{e}_{\boldsymbol{\rho}}$	~ 5 m	$\sim 4.3 \times 10^3$ km
$\sigma_{\boldsymbol{\rho}}$	0.03	8.4×10^6
$\bar{e}_{\dot{\boldsymbol{\rho}}}$	~ 0.015 m/s	~ 1.7 km/s
$\sigma_{\dot{\boldsymbol{\rho}}}$	$\sim 8.65 \times 10^{-4}$	$\sim 3.6 \times 10^3$
$\Delta\bar{V}$	~ 0.70 m/s	~ 1.67 m/s
$\sigma_{\Delta V}$	~ 0.32	$\sim 3.15 \times 10^3$
$\Delta\bar{W}$	3.971 deg/s	4.1637 deg/s
$\sigma_{\Delta W}$	~ 1	~ 1

Table 3: ANN_{SDRE} vs SDRE Trajectories Statistical Analysis

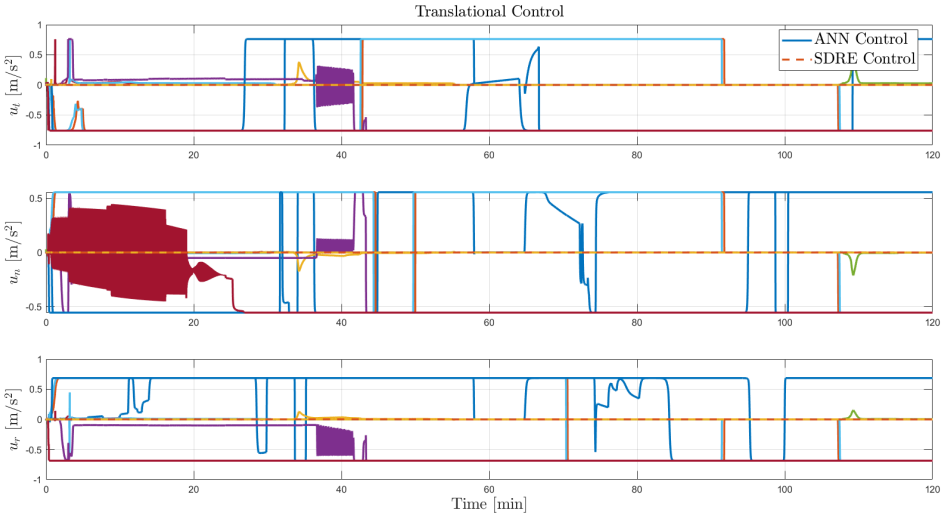


Figure 14: Complete Test Set Translational Control Behaviour

A somewhat improved performance can be seen by analyzing the 27 trajectories, which reach a final position that is close to the one of target. The final position error is or the order of 12 km, but a position error rate obviously too high as shown in Tab. 4. The inability of the SDRE network of replicating the required control effort is clear here as well, as shown in Fig. 15.

Variable	SDRE Value	ANN_{SDRE} Value
T	2 Hours	2 Hours
$\bar{e}_{\boldsymbol{\rho}}$	~ 5 m	~ 11.9 km
$\sigma_{\boldsymbol{\rho}}$	0.4	1.9×10^3
$\bar{e}_{\dot{\boldsymbol{\rho}}}$	~ 0.015 m/s	~ 6.75 m/s
$\sigma_{\dot{\boldsymbol{\rho}}}$	$\sim 4 \times 10^{-4}$	~ 0.76
$\Delta\bar{V}$	~ 0.70 m/s	~ 4 m/s
$\sigma_{\Delta V}$	~ 0.4	~ 0.43
$\Delta\bar{W}$	3.6675 deg/s	3.8775 deg/s
$\sigma_{\Delta W}$	~ 0.9	~ 1

Table 4: ANN_{SDRE} vs SDRE Trajectories Statistical Analysis, 27 Trajectories

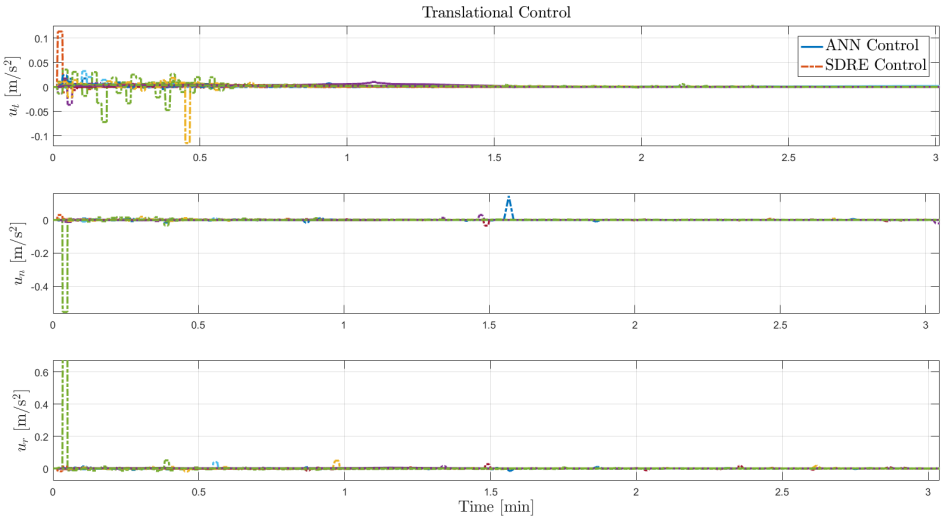


Figure 15: ANN test subset translational control behaviour

The SDRE attitude network, however, shows good results, and is capable of driving the relative Euler angles and the angular velocities to zero achieving its task. Of course this positive result is canceled by the translational performance above. The next two figures show the relative attitude and control effort time history comparison.

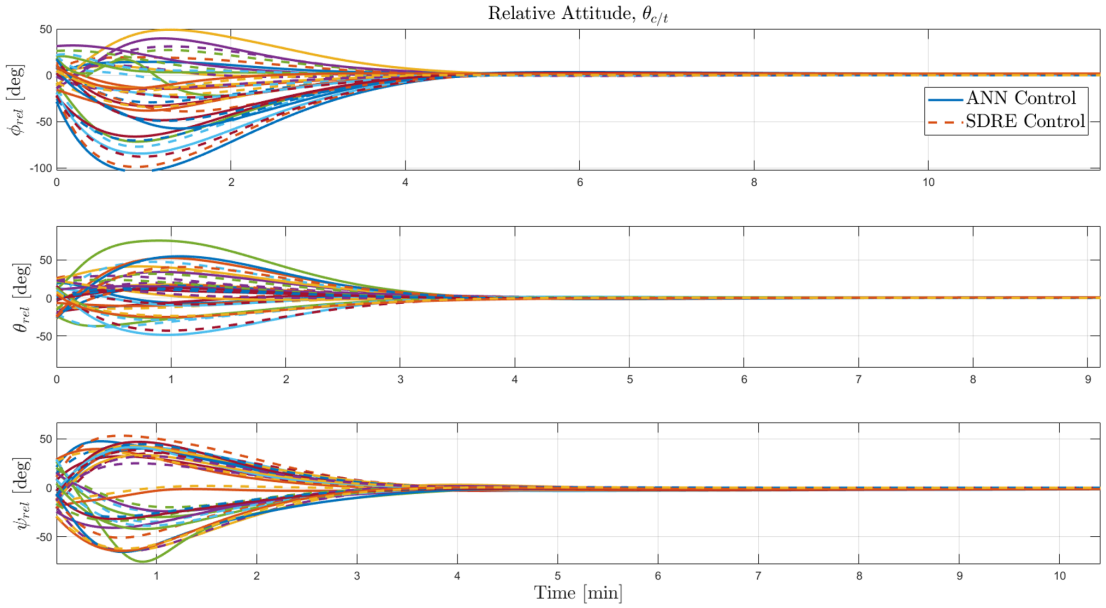


Figure 16: SDRE Attitude ANN angular position comparison

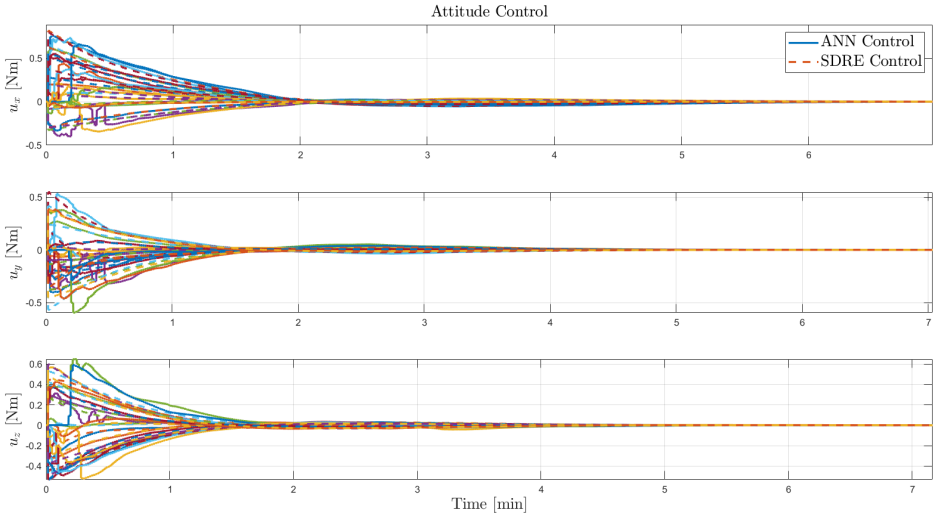


Figure 17: SDRE Attitude ANN control effort comparison

PID Network Performance

This section describes the behavior of the network structures based on classical PID control presented earlier and referred in Fig. 10. The capability of the ANN to replicate the classical controller trajectories is noteworthy as indicated by the Fig. 18.

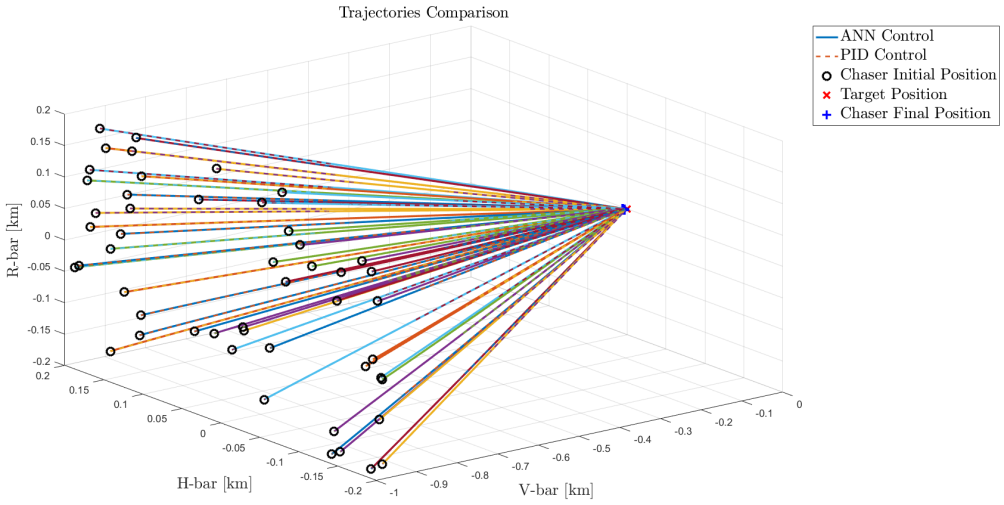


Figure 18: PID and ANN Trajectory comparison

The time histories of the translational state variables and control are shown in Figures 19 - 21.

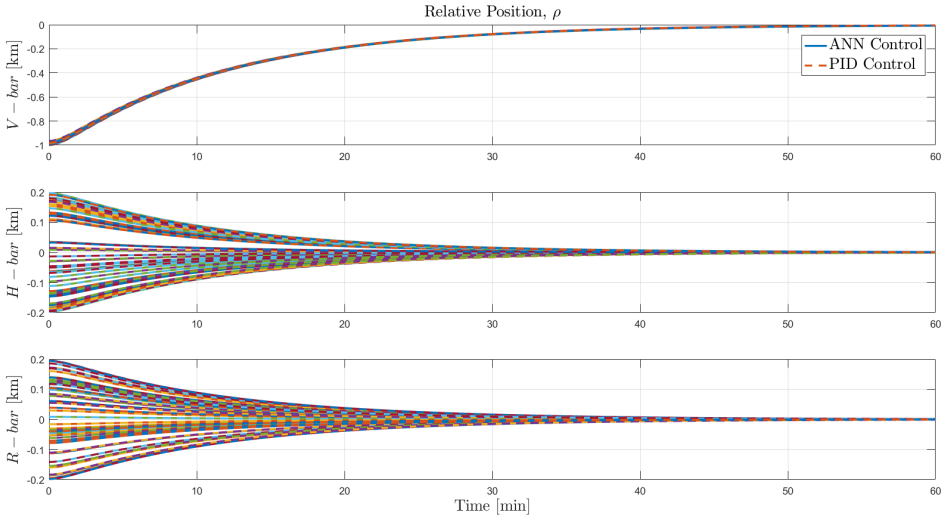


Figure 19: ANN(PID) relative position

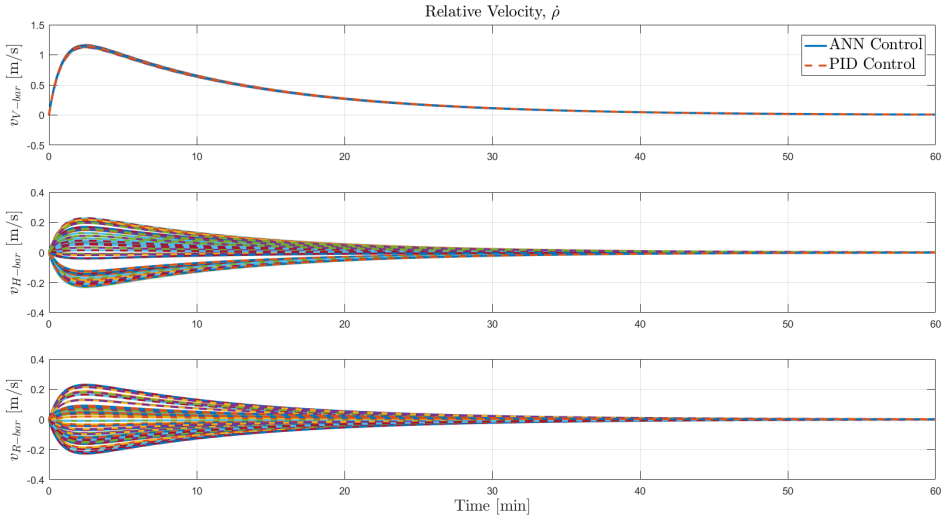


Figure 20: ANN(PID) relative velocity

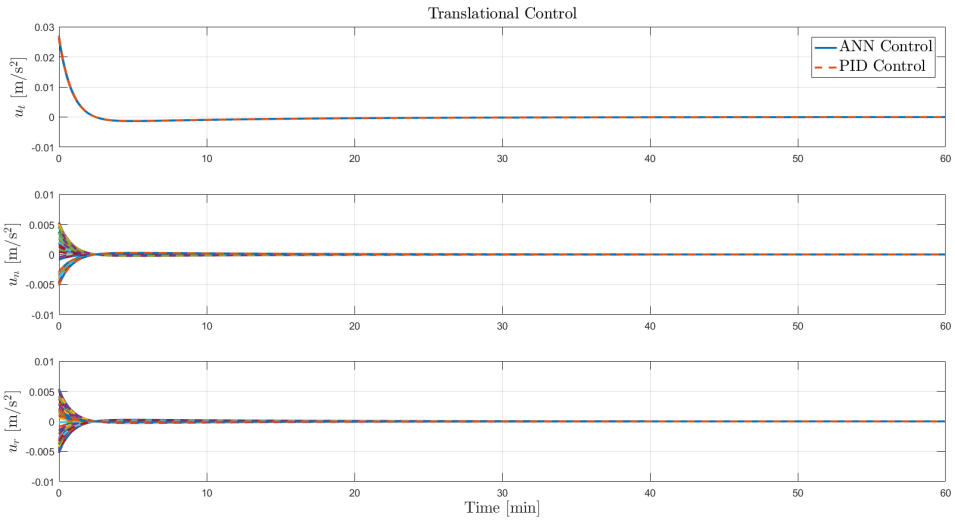


Figure 21: ANN(PID) translational control effort

The attitude behavior is shown in Figures 22 and 23. Comparing these results with those relative to the SDRE ANN is clear the sensitivity of the database to the bandwidth of the system.

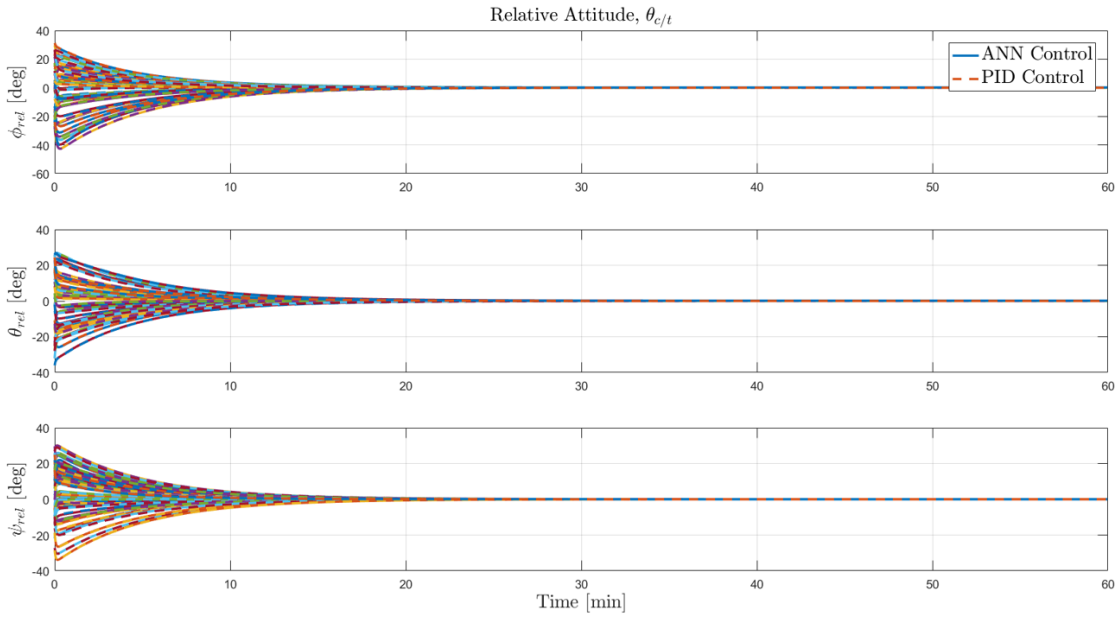


Figure 22: ANN(PID) relative attitude time history

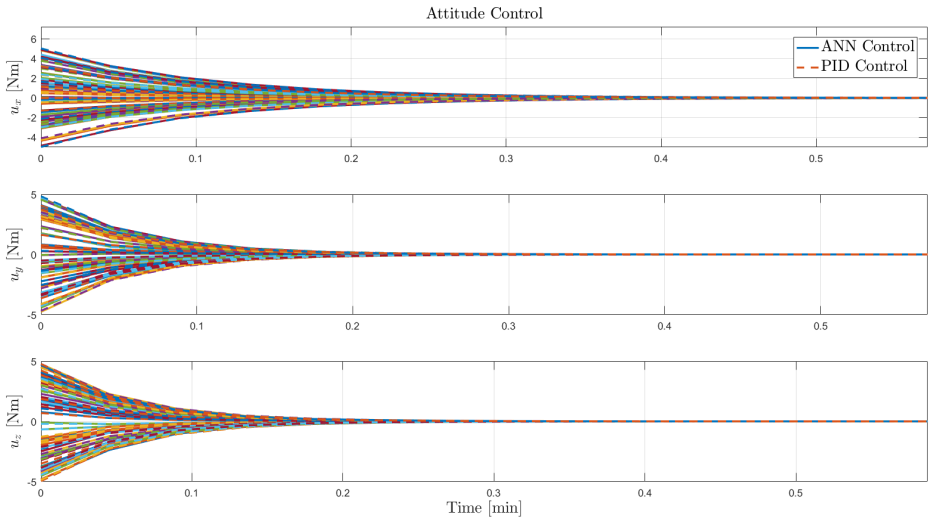


Figure 23: ANN(PID) attitude control effort

The performance parameters between the classical controller and its ANN counterpart are summarized in the next table. They also indicate the successful behavior of the network.

Variable	PID Value	ANN_{PID} Value
T	1 Hour	1 Hour
$\bar{e}_{\boldsymbol{\rho}}$	~ 5.6 m	~ 7.65 m
$\sigma_{\boldsymbol{\rho}}$	0.03	0.62
$\bar{e}_{\dot{\boldsymbol{\rho}}}$	~ 0.015 m/s	~ 0.014 m/s
$\sigma_{\dot{\boldsymbol{\rho}}}$	$\sim 8.65 \times 10^{-4}$	$\sim 8 \times 10^{-4}$
$\Delta\bar{V}$	2.3 m/s	2.3 m/s
$\sigma_{\Delta V}$	$\sim 6.4 \times 10^{-3}$	$\sim 6.8 \times 10^{-3}$
$\Delta\bar{W}$	3.0723 deg/s	3.0679 deg/s
$\sigma_{\Delta W}$	0.7595	0.7579

Table 5: ANN(PID) vs PID Trajectories Statistical Analysis

CONCLUSIONS

The present paper presents the preliminary development of a control strategy, based on neural networks, to be used in the final phase of a rendezvous maneuver between two space vehicles. The challenge is to evaluate their performance in a dynamic environment characterized by a three-body model necessary to study future missions in the Earth-Moon system. The main objective was the

capabilities of training and evaluating a set of ANNs to replicate closed loop performance obtained with the simulator *Rossonero* developed earlier for the European Space Agency[17].[5]
The datasets were generated using two controllers: the first, taken from,[5] is a high frequency optimal controller that uses the state dependent Riccati equation structure, the second is a classical PID controller developed "ad hoc" for this work. The rationale for these selections was to compare high frequency, high bandwidth control, requirements with a a "softer one" in terms of database size and training constraints. The main conclusions, taking into account the limited resources of a standard laboratory computer were:

- The neural network, trained on the basis of SDRE data, was not able to learn the behavior of the SDRE translational control. The reason is due to the fact that the translational control signals have high frequency content, and are of impulsive nature, which led to the necessity of la large amount of data for efficient training (for instance each simulation contained ~ 7500 samples and the overall size of the dataset required to train the ANN was around $1.5GB$).

- The neural networks, trained on the basis of the SDRE attitude data and on the PID data, were able to learn their behaviour and to perform a correct control on the test set, with the amount of data provided.

ACKNOWLEDGMENT

This work was partially supported by ESA under contract No. 000121575/17/NL/hh. The view expressed herein can in no way be taken to reflect the official opinion of the European Space Agency.

REFERENCES

[1] NASA, "Genesis: search for origins," tech. rep., https://https://www.nasa.gov/specials/artemis/, 2021.

[2] K. Laurini, J. Piedboeuf, B. Schade, K. Matsumoto, F. Spiero, and A. Lorenzoni, "The Global Exploration Roadmap," *IAC-11 B*, Vol. 3, 2018.

[3] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, *Dynamical Systems, the Three-Body Problem and Space Mission Design*. Marsden Books, 2011, 10.1142/97898127926170222.

[4] K. Howell, "Families of orbits in the vicinity of the collinear libration points," *The Journal of the Astronautical Sciences*, Vol. 49, No. 1, 2001, pp. 107–125, 10.1007/BF03546339.

[5] G. Bucchioni, "Rendezvous in Cis-lunar Space Near Rectilinear Halo Orbit:Dynamics and Control Issues," Vol. 68, No. 3, 2021, pp. 1–38, 110.3390/aerospace8030068.

[6] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*. New York: Cambridge University Press, 2003.

[7] F. Ankersen, *Guidance, Navigation, Control and Relative Dynamics for Spacecraft Proximity Maneuvers*. PhD thesis, Aalborg University, Department of Electronic Systems, 2011.

[8] W. Clohessy and R. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.

[9] Y. Luo, J. Zhang, and G. Tang, "Survey of orbital dynamics and control of space rendezvous," *Chinese Journal of Aeronautics*, Vol. 27, No. 1, 2014, pp. 1–11.

[10] L. Breger and J. P. How, "Safe Trajectories for Autonomous Rendezvous of Spacecraft," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1478–1489, 10.2514/6.2006-6584.

[11] Y. Lian, Y. Meng, G. Tang, and L. Liu, "Constant-thrust glideslope guidance algorithm for time-fixed rendezvous in real halo orbit," *Acta Astronautica*, Vol. 79, 2012, pp. 241–252, 10.1016/j.actaastro.2012.04.049.

[12] Y. Lian and G. Tang, "Libration point orbit rendezvous using PWPF modulated terminal sliding mode control," *Advances in Space Research*, Vol. 52, No. 12, 2013, pp. 2156–2167, 10.1016/j.asr.2013.08.034.

[13] Y. Lian, L. Liu, Y. Meng, G. Tang, and K. Chen, "Learning the optimal state-feedback using deep networks," pp. 232–236, 10.1109/RAST.2011.5966831.

[14] C. Sánchez-Sánchez, D. Izzo, and D. Hennes, "Learning the optimal state-feedback using deep networks," *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2016, pp. 1–8.

[15] D. Tailor and D. Izzo, "Learning the optimal state-feedback via supervised imitation learning," *Astrodynamics*, Vol. 3, No. 4, 2019, pp. 361–374.

[16] D. Izzo, M. Märtens, and B. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," *Astrodynamics*, Vol. 3, No. 4, 2019, pp. 287–299.

[17] G. Franzini, G. Bucchioni, M. Innocenti, and M. Casasco, "Rossonero: a tool for preliminary rendezvous mission design in the restricted three-body problem," *7th International Conference on Astrodynamics Tools and Techniques*, Vol. 1, NLD, 2018, pp. 1–7.

[18] P. Gurfil, M. Idan, and N. J. Kasdin, "Adaptive neural control of deep-space formation flying," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 3, 2003, pp. 491–501.

[19] G. Franzini and M. Innocenti, "Relative Motion Dynamics in Restricted Three-Body Scenarios," *Journal of Spacecraft and Rockets*, Vol. 56, No. 5, 2019, pp. 1322–1337.

[20] G. Bucchioni and M. Innocenti, "Dynamical issues in rendezvous operations with third body perturbation," *Advances in the Astronautical Sciences*, Vol. 171, 2019, pp. 549–564.

[21] F. D'Onofrio, G. Bucchioni, and M. Innocenti, "ROSSONERO: new features in the simulation tool for realistic rendezvous mission design in the restricted three-body problem," *Proceedings of 8th ICATT International Conference on Astrodynamics Tools and Techniques*, European Space Agency, 2021, pp. 1–15.

[22] T. Çimen, "State-dependent Riccati equation (SDRE) control: a survey," *IFAC Proceedings Volumes*, Vol. 41, No. 2, 2008, pp. 3761–3775.

[23] J. Cloutier and J. Cockburn, "The state-dependent nonlinear regulator with state constraints," *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, Vol. 1, 2001, pp. 390–395 vol.1, 10.1109/ACC.2001.945577.

[24] H. Dong, Q. Hu, and M. R. Akella, "Dual-quaternion-based spacecraft autonomous rendezvous and docking under six-degree-of-freedom motion constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 5, 2018, pp. 1150–1162.

[25] C. Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: study on landing problems," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 5, 2018, pp. 1122–1135.

[26] C. I. Sprague, D. Izzo, and P. Ögren, "Learning Dynamic-Objective Policies from a Class of Optimal Trajectories," *2020 59th IEEE Conference on Decision and Control (CDC)*, IEEE, 2020, pp. 597–602.

[27] S. Li, E. Öztürk, C. De Wagter, G. C. d. Croon, and D. Izzo, "Aggressive online control of a quadrotor via deep network representations of optimality principles," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 6282–6287.

[28] R. Furfaro, I. Bloise, M. Orlandelli, P. Di Lizia, F. Topputo, R. Linares, *et al.*, "Deep learning for autonomous lunar landing," *2018 AAS/AIAA Astrodynamics Specialist Conference*, Vol. 167, Univelt, 2018, pp. 3285–3306.

[29] J. Broida and R. Linares, "Spacecraft rendezvous guidance in cluttered environments via reinforcement learning," *29th AAS/AIAA Space Flight Mechanics Meeting*, American Astronautical Society Ka'anapali, Hawaii, 2019, pp. 1–15.

[30] C. E. Oestreich, R. Linares, and R. Gondhalekar, "Autonomous Six-Degree-of-Freedom Spacecraft Docking Maneuvers via Reinforcement Learning," *arXiv preprint arXiv:2008.03215*, 2020.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.