Neural Network-based Imitation Learning for Optimal Satellite Formation Control

Oleksandr Dyblenko, Caterina Santoro, Luciano Blasi, Egidio D'Amato, Immacolata Notaro

***Abstract*—Satellite Formation Control (SFC) with on-off impulsive thrusters represents a significant challenge in space missions. Classic and optimal control approaches with control action quantization provide possible strategies, but limited in performance due to system linearity assumptions and model accuracy requirements. This paper investigates the use of a neural network controller synthesized via imitation learning to replicate the behavior of an expert Linear Quadratic Regulator (LQR) controller with quantized actions. A Multi-Layer Perceptron (MLP) neural network is trained on data generated by a LQR controller designed for 6U CubeSats formation with on-off thrusters, learning the state-action mapping that implicitly includes quantization effects. The goal is to assess whether a neural controller can achieve comparable performance to the expert while offering potential advantages in generalization and adaptability.**

I. Introduction

Satellite Formation Control (SFC) represents a strategic technology for a wide range of modern space missions, including high-resolution Earth observation, astrophysical scientific research, environmental monitoring, space defense missions [1]. Unlike single satellites, formations enable distributed sensing capabilities, offering operational flexibility and system redundancy. However, formation control introduces several challenges due to the interaction between satellites, the distribution of tasks and the optimization of fuel consumption.

The use of small satellites and CubeSats for formation flying has gained significant interest in recent years due to their cost-effectiveness and rapid deployment capabilities [2], [3]. However, the limitations imposed by onboard resources, particularly computational power and propulsion capacity, need the development of efficient and robust control strategies. Small satellites often rely on low-thrust impulsive on-off thrusters for attitude and orbit control given their simplicity and reliability. However, the discrete nature of on-off thrusters introduces significant non-linearities and discontinuities in the control system dynamics.

The design of controllers for systems with on-off actuators is a well-studied problem in control theory, traditionally addressed using several approaches including optimal control with input quantization, pulse-width pulse-frequency modulation, model predictive control, and switching control strategies [4], [5].

Optimal control based techniques, like Linear Quadratic Regulator (LQR), are widely used in SFC [6], [7]. However, the discrete nature of thrusters makes the solution sub-optimal and does not guarantee reliability and robutstness. Model Predictive Control (MPC) based solutions have been proposed for SFC with on-off thrusters [8], [9], but their real-time implementation is often limited by computational constraints of small satellites. Moreover, MPC formulations for discrete on-off thrusters lead to mixed-integer optimization problems that are computationally intensive to solve [9].

Pulse-Width Pulse-Frequency (PWPF) modulation techniques [10], [11] provide an alternative approach by modulating both the duration and frequency of thruster pulses to approximate continuous control commands, offering good performance with reduced vibration excitation and fuel consumption. However, PWPF modulators require careful parameter tuning for different mission phases and spacecraft configurations and their performance can degrade under large disturbances or rapid maneuvers.

Switching control strategies, including variable structure control and sliding mode control [12], [13], offer robustness to uncertainties and disturbances by switching between different control laws. While these approaches provide strong theoretical stability guarantees, they typically suffer from chattering effects due to high-frequency switching of thrusters, leading to increased fuel consumption and potential structural fatigue.

Recently, machine learning techniques have emerged for the design of spacecraft control systems [14], [15]. In general, Neural Networks (NN) have been employed both for the identification of the dynamical systems and the synthesis of adaptive controllers [16], [17]. In particular, Multi-Layer Perceptron (MLP) architectures have shown effectiveness in non-linear control approximation [18].

In the context of SFC, several works have addressed the use of neural networks and machine learning [19], [20], distinguishing between offline pre-trained controllers as surrogates of optimal or model-based policies and online/adaptive schemes that update the network in flight [21], [22].

This work represents a preliminary study proposing the use of Imitation Learning (IL) to synthesize satellite formation controllers. Two NN-based controllers are presented, in order to balance network complexity and dataset size requirements.

In recent years, IL has emerged as a promising approach for the synthesis of NN-based control policies directly from

Oleksandr Dyblenko, Caterina Santoro, Luciano Blasi, and Immacolata Notaro are with the Department of Engineering, Universitá degli Studi della Campania "L. Vanvitelli", 81031, Aversa (CE), Italy; e-mail: oleksandr.dyblenko@studenti.unicampania.it (OD), caterina.santoro@unicampania.it (CS), luciano.blasi@unicampania.it (LB), immacolata.notaro@unicampania.it (IN)

Egidio D'Amato are with the Department of Science and Technology, Universitá degli Studi di Napoli "Parthenope", Napoli, Italy; e-mail: egidio.damato@uniparthenope.it (ED)

data generated by expert controllers [23], [24]. The resulting NN controller replicates the behavior of the expert controller and it is ready for improvements that other approaches cannot receive due to their limitations. This technique offers several advantages: ability to learn complex control strategies directly from data, without requiring an explicit formulation of the optimization problem; potential for generalization to operating conditions different from those seen during training; flexibility to further incorporate non-linear dynamics and implicit constraints.

To effectively integrate the controller into satellite formations, a leader-follower architecture was considered, where one or more chief satellites define the reference trajectory and the deputy satellites maintain relative positions. The expert controller is designed using a LQR with on-off thruster quantization. The controller was designed using a linear Hill-Clohessy-Wiltshire (HCW) model. It generates a dataset of state-action pairs for training the MLP network. The trained NN controller is then evaluated in simulation to validate the effectiveness of the proposed strategy.

The article is structured as follows: Section II describes the control problem, with details on the expert controller formulation; Section III illustrates two MLP neural network architectures; Section IV compares the NN-based controllers with the expert by numerical simulation of a trailing CubeSats formation; finally, Section V resumes conclusions with future research directions.

II. Problem statement

The scope of this work is the development of a NN-based controller for the management and reconfiguration of satellite formations in Low Earth Orbit (LEO).

More specifically, we consider N_s 6U CubeSats, each of them featuring a cuboid-shaped central body, equipped with a cold gas propulsion system composed of six actuator modules, mounted on the six faces of the CubeSat. Each module contains four identical on–off thrusters, each capable of providing a fixed thrust level of $f = 25mN$. As a result, for each semi-axis, a maximum force of $4f = 100mN$ can be generated by simultaneously activating all four thrusters associated with that direction. The actuators are purely binary, and no thrust modulation is available at the single-thruster level.

The satellite relative dynamics is described using the Hill-Clohessy-Wiltshire (HCW) model, where one or more satellites act as leaders, defining the reference orbit, while the follower satellites maintain predefined relative positions with respect to the leader.

The objectives of the flight control system include:

- Maintaining assigned relative positioning between satellites in the formation, ensuring adherence to mission-specific configurations;
- Minimizing fuel consumption to extend mission duration and reduce operational costs;
- Ensuring robustness against orbital perturbations and model uncertainties;
- Implementing control strategies compatible with on-off thruster actuation constraints.

The NN-based controller is synthesized using imitation learning, where a Multi-Layer Perceptron (MLP) network is trained to clone the behavior of an expert controller based on LQR with on-off thruster quantization. The training dataset is generated through simulations of the expert controller under various conditions.

A. LQR Controller with On-Off Thrusters

The expert policy used for imitation learning is based on a Linear Quadratic Regulator (LQR) designed to minimize the tracking error with respect to a time-varying reference trajectory, followed by a deterministic thrust quantization strategy.

The LQR was designed based on the linearized dynamics defined by the HCW equations, which provide a simplified representation of the relative motion of two spacecraft in unperturbed near-circular orbits. These equations are formulated in a Local Vertical Local Horizontal (LVLH) reference frame, centered in the leader's center of mass, with:

- axis x_{HCW} in the radial direction, from the center of the Earth towards the satellite;
- axis y_{HCW} in the velocity direction;
- axis z_{HCW} completing the right-handed frame.

In this frame, the state vector $\boldsymbol{\xi}(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$ contains the follower's position $\mathbf{p}(t) = [x(t), y(t), z(t)]^T$ and velocity $\mathbf{v}(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$ with respect to its leader, while $\overline{\boldsymbol{v}}(t) = [a_x(t), a_y(t), a_z(t)]^T$ represents the control accelerations. The HCW equations of motion can be written in state-space form as:

$$\dot{\boldsymbol{\xi}}(t) = \mathbf{A}\boldsymbol{\xi}(t) + \mathbf{B}\overline{\boldsymbol{v}}(t) \tag{1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\overline{\omega}^2 & 0 & 0 & 0 & 2\overline{\omega} & 0 \\ 0 & 0 & 0 & -2\overline{\omega} & 0 & 0 \\ 0 & 0 & -\overline{\omega}^2 & 0 & 0 & 0 \end{bmatrix} \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The mean motion $\overline{\omega} = \sqrt{\frac{GM_\oplus}{\|\mathbf{r}^{ECI}(t)\|_2^3}}$ represents the leader's orbital angular velocity, which depends on the distance from the Earth center $\|\mathbf{r}^{ECI}\|_2$, the gravitational constant G and the mass of the Earth $M_\oplus$. In this context, the tracking error $\boldsymbol{e_\xi}(t)$is defined as the deviation from the desired reference trajectory $\boldsymbol{\xi_{ref}}(t)$:

$$\boldsymbol{e_\xi}(t) = \boldsymbol{\xi}(t) - \boldsymbol{\xi_{ref}}(t) \tag{3}$$

The LQR control law is obtained by minimizing the infinite-horizon quadratic cost function

$$J = \int_0^\infty \left(\boldsymbol{e}_{\boldsymbol{\xi}}^T(t)\mathbf{Q}\boldsymbol{e_\xi}(t) + \overline{\boldsymbol{v}}^T(t)\mathbf{R}\overline{\boldsymbol{v}}(t)\right) dt \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{6\times6}$ is a positive semi-definite weight matrix on the states (penalizing position and velocity deviations) and $\mathbf{R} \in \mathbb{R}^{3\times3}$ is a positive definite weight matrix on the inputs (penalizing fuel consumption). The LQR solution is:

$$\boldsymbol{v}(t) = -\mathbf{K}\boldsymbol{e_\xi}(t) \tag{5}$$

where $\mathbf{K}$ is the feedback gain obtained by solving the associated Riccati equation.

The optimal continuous control $\boldsymbol{v}$ is then translated in a discrete on-off command for the 24 thrusters through a predefined allocation logic.

Denote with $m(t)$ the satellite mass at time t, the continuous force generated by LQR is $m(t)\boldsymbol{v}(t)$. The desired force along the axis a, $m(t)v_a(t)$, is mapped to a discrete-valued force through a quantization and saturation process defined as

$$n_a(t) = \min\left(4, \max\left(0, \left\lfloor \frac{|m(t)v_a(t)|}{f} + \frac{1}{2} \right\rfloor\right)\right), \tag{6}$$

where $n_a \in \{0,1,2,3,4\}$ represents the number of thrusters to be activated along axis a, and $\lfloor \frac{|m(t)v_a(t)}{f} + \frac{1}{2} \rfloor$ rounds the $m(t)v_a(t)/f$ to nearest integer.

Consequently, the resulting force applied along axis a is given by

$$F_a(t) = s_a(t)n_a(t)f. \tag{7}$$

where $s_a(t) = \text{sign}\,(m(t)v_a(t))$ defines the direction by preserving the sign of the force.

Thruster allocation is performed using a fixed and ordered activation sequence. For each axis a, at time t the activation class $\pi_a(t)$ can assume only 9 values $\pi_a \in \{-4,-3,-2,-1,0,1,2,3,4\}$:

$$\pi_a(t) = s_a(t)n_a(t) \tag{8}$$

The on-off command associated with the i-th thruster aligned with the positive semi-axis a is defined as

$$\delta_{a^+,i}(t) = \begin{cases} 1, & i \le n_a(t) \text{ and } s_a(t) = 1 \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1,\dots,4 \tag{9}$$

Similarly, we define $\delta_{a^-,i}(t)$ the on-off command associated with the i-th thruster aligned with the negative semi-axis a:

$$\delta_{a^-,i}(t) = \begin{cases} 1, & i \le n_a(t) \text{ and } s_a(t) = -1 \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1,\dots,4 \tag{10}$$

This deterministic allocation strategy guarantees a unique mapping between the continuous control action and the corresponding thruster activation pattern, defined by a 24-dimensional vector $\boldsymbol{\delta}(t) \in \{0,1\}^{24}$.

III. NN-based Controller and Imitation Learning

In this work, we propose two architectures for the NN-based controller: a monolithic approach and a modular scheme that separates the continuous regulation problem from the discrete quantization problem.

The former architecture Φ_1 is a single feed-forward MLP neural network that directly replaces both the LQR controller and the force quantization and allocation block, learning an end-to-end mapping from tracking errors to on–off thruster commands.

The neural controller Φ_1 has the following structure:

- **Input layer:** 6 neurons, corresponding to the components of the tracking error state $\mathbf{e}_\xi(t) \in \mathbb{R}^6$.
- **Hidden layers:** 3 fully connected layers, each composed of 256 neurons, with leaky ReLU activation functions (denoted as σ).
- **Output layer:** 24 neurons with linear activation, each corresponding to one thruster command. The network outputs $\hat{\boldsymbol{\delta}}^{\Phi_1}_{NN} \in [0,1]^{24}$ are subsequently mapped to binary on–off signals $\boldsymbol{\delta}^{\Phi_1}_{NN} \in \{0,1\}^{24}$ via a fixed thresholding operation.

The forward pass of the network corresponds to the evaluation of a static nonlinear state-feedback law mapping the input $\boldsymbol{e_\xi}(t)$ to the control output $\hat{\boldsymbol{\delta}}^{\Phi_1}_{NN}$. Let $L^{\Phi_1} = 4$ denote the total number of network layers excluding the input layer. The forward propagation is defined as

$$\mathbf{h}^{\Phi_1}_0 = \mathbf{e}_\xi(t) \tag{11}$$

$$\mathbf{h}^{\Phi_1}_l = \sigma\left(\mathbf{W}^{\Phi_1}_l \mathbf{h}^{\Phi_1}_{l-1} + \mathbf{b}^{\Phi_1}_l\right), \qquad l = 1,2,3 \tag{12}$$

$$\hat{\boldsymbol{\delta}}^{\Phi_1}_{NN} = \mathbf{W}^{\Phi_1}_4 \mathbf{h}^{\Phi_1}_3 + \mathbf{b}^{\Phi_1}_4 \tag{13}$$

Here, $\mathbf{W}^{\Phi_1}_l$ and $\mathbf{b}^{\Phi_1}_l$ denote the weight matrix and bias vector of the l-th layer, respectively.

The monolithic network is trained via supervised imitation learning by minimizing a mean-squared error loss computed over a batch sampled from the training data:

$$\mathcal{L}_{\Phi_1} = \frac{1}{N_{\Phi_1}} \sum_{k=1}^{N_{\Phi_1}} \left\| \hat{\boldsymbol{\delta}}^{\Phi_1}_{NN}(\mathbf{e}_{\boldsymbol{\xi}_k}(t)) - \boldsymbol{\delta}^{LQR}_k(t) \right\|^2_2 \tag{14}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, k indexes is the training samples, $\boldsymbol{\delta}^{LQR}_k$ represents the on-off control provided by the expert, and N_{Φ_1} the size of the random batch extracted from the training dataset $\mathcal{D}^{\Phi_1}$.

A critical issue of this approach is the size of the dataset $\mathcal{D}^{\Phi_1}$, required to adequately cover the switching surfaces induced by the quantization and saturation of thrust. In addition, some thruster activation patterns occur rarely, resulting in an imbalanced dataset and potentially poor generalization near decision boundaries. For this purpose, each component of the six-dimensional tracking-error state was discretized using 19 values. Specifically, two values were selected around each quantization threshold of the expert controller in order to capture the switching behavior induced by thrust rounding. An additional value corresponding to the maximum admissible tracking error was included to represent the saturated control regime. This construction yields a total of 19^6 grid points. Although the dataset was carefully built, the monolithic architecture still needs a large amount of data.

Furthermore, the large neuron count and the choice of more than 2 layers are justified by the high dimensionality and

complexity of the function to be approximated, which is non-linear and non-continuous due to the quantization ad allocation operations.

To overcome these issues, we consider a modular architecture that separates the continuous regulation problem from the discrete quantization problem. The key insight is that the LQR output $\boldsymbol{\upsilon}^{LQR}$ is a continuous signal, while the quantization is applied independently to each axis.

The proposed architecture, Φ_2, is composed of two neural networks: a network Φ_2^C that imitates the continuous LQR control law, and a network Φ_2^Q that imitates the axis-wise thrust quantizer. The quantizer network is replicated for each translational axis $a \in \{x, y, z\}$.

The network Φ_2^C consists of:

- **Input layer:** 6 neurons, corresponding to the components of the tracking-error state $\boldsymbol{e_\xi}(t)$.
- **Hidden layers:** 2 fully connected layers, each with 32 neurons and ReLU activations.
- **Output layer:** 3 neurons with linear activation, providing the continuous acceleration command $\boldsymbol{\upsilon}_{NN}^{\Phi_2^C}(t) \approx \boldsymbol{\upsilon}^{LQR}(t)$.

Let $L_2^{LQR} = 3$ denote the total number of layers excluding the input layer (three hidden layers plus the output layer). The forward propagation is defined as

$$\mathbf{h}_0^{\Phi_2^C} = \boldsymbol{e_\xi}(t), \tag{15}$$

$$\mathbf{h}_l^{\Phi_2^C} = \sigma\left(\mathbf{W}_l^{\Phi_2^C}\mathbf{h}_{l-1}^{\Phi_2^C} + \mathbf{b}_l^{\Phi_2^C}\right), \qquad l = 1, 2 \tag{16}$$

$$\boldsymbol{\upsilon}_{NN}^{\Phi_2^C} = \mathbf{W}_3^{\Phi_2^C}\mathbf{h}_2^{\Phi_2^C} + \mathbf{b}_3^{\Phi_2^C}, \tag{17}$$

where $\sigma(\cdot)$ denotes a ReLU activation function.

The network Φ_2^C is trained via supervised learning using demonstrations from the LQR controller. The loss function is

$$\mathcal{L}_{\Phi_2^C} = \frac{1}{N_{\Phi_2^C}} \sum_{k=1}^{N_{\Phi_2^C}} \left\| \boldsymbol{\upsilon}_{NN}^{\Phi_2^C}\left(\mathbf{e}_{\boldsymbol{\xi}_k}(t)\right) - \boldsymbol{\upsilon}_k^{LQR}(t) \right\|_2^2 \tag{18}$$

where $\boldsymbol{\upsilon}_k^{LQR}(t)$ is the continuous acceleration provided by the expert (LQR controller), and $N_{\Phi_2^C}$ the size of the batch sampling the training dataset $\mathcal{D}^{\Phi_2^C}$.

The second network Φ_2^Q is used to imitate the quantization logic along a single axis a. The network Φ_2^Q is composed by:

- **Input layer:** 1 neuron, corresponding to the continuos acceleration on axis a, $\upsilon_a(t)$.
- **Hidden layers:** 2 fully connected layers of 64 neurons with ReLU activations.
- **Output layer:** 9 neurons, representing the probabilities $\hat{\boldsymbol{\zeta}}_{a,NN}^{\Phi_2^Q} \in [0,1]^9$ of activating each class in the set $\{-4, -3, -2, -1, 0, 1\,2, 3, 4\}$ along the axis a. The network outputs are subsequently mapped to binary on-off signals $\{0, 1\}$ by selecting the class with the maximum probability value.

The forward pass is defined as

$$\mathbf{g}_0^{\Phi_2^Q} = \upsilon_a(t), \tag{19}$$

$$\mathbf{g}_l^{\Phi_2^Q} = \sigma\left(\mathbf{W}_l^{\Phi_2^Q}\mathbf{g}_{l-1}^{\Phi_2^Q} + \mathbf{b}_l^{\Phi_2^Q}\right), \qquad l = 1, 2, \tag{20}$$

$$\mathbf{g}_3^{\Phi_2^Q} = \mathbf{W}_3^{\Phi_2^Q}\mathbf{g}_2^{\Phi_2^Q} + \mathbf{b}_3^{\Phi_2^Q} \tag{21}$$

$$\hat{\boldsymbol{\zeta}}_{a,NN}^{\Phi_2^Q} = \frac{\exp(\mathbf{g}_3{}^{\Phi_2^Q})}{\left\|\exp(\mathbf{g}_3{}^{\Phi_2^Q})\right\|_1} \tag{22}$$

where $\|\cdot\|_1$ represents the sum-norm and eq. (22) corresponds to element-wise softmax operation over the output layer.

The network Φ_2^Q is trained to minimize the cross-entropy (CE) loss defined as

$$\mathcal{L}_{\Phi_2^Q} = -\frac{1}{N_{\Phi_2^Q}} \sum_{k=1}^{N_{\Phi_2^Q}} \boldsymbol{\zeta}_{a_k}^Q(t) \cdot \log\left(\hat{\boldsymbol{\zeta}}_{a,NN}^{\Phi_2^Q}\left(\upsilon_{a_k}^{LQR}(t)\right)\right) \tag{23}$$

where $\boldsymbol{\alpha} \cdot \boldsymbol{\beta}$ indicates the scalar product between two vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and $N_{\Phi_2^Q}$ is the batch size. $\boldsymbol{\zeta}_{a_k}^Q \in \{0,1\}^9$ is the one-hot encoding label associated to the input training data, where a unit element is placed to represent the class to which it belongs.

The final controller is obtained by composing the two networks. First, the LQR imitator, Φ_2^C generates the continuous control action:

$$\boldsymbol{\upsilon}_{NN}^{\Phi_2^C}(t) = \Phi_2^C(\boldsymbol{e_\xi}(t)) \tag{24}$$

Then, a dedicated quantizer network is applied to the acceleration required on each axis a. The network Φ_2^Q processes the continuous acceleration $\upsilon_{a,NN}^{\Phi_2^C}(t)$ to determine the activation class:

$$\pi_a^{\Phi_2^Q} = \text{argmax}\ \boldsymbol{\zeta}_{a,NN}(t) \tag{25}$$

Unlike discrete quantizers with sharp discontinuities and zero gradients, the neural network-based quantizer Φ_2^Q is differentiable. This enables a future fine-tuning phase where the entire composite controller can be optimized end-to-end on the nonlinear simulator via backpropagation. Furthermore, this will allow to avoid a fixed quantization rule, but a policy that can take into account nonlinearities, perturbations, and model mismatch observed in the full simulator.

The modular approach significantly reduced data requirements through functional decomposition:

- The LQR imitator Φ_2^C was trained on 117000 different state-action pairs;
- The quantizer network Φ_2^Q was trained on only 6000 samples, corresponding to 1D grid sampling of the quantization function.

This represents a reduction in total training data from 19^6 to approximately 123000 samples— a factor of $\approx 382\times$.

For clarity, we provide schematic diagrams of both approaches using a simplified notation, in Figures 1 and 2.

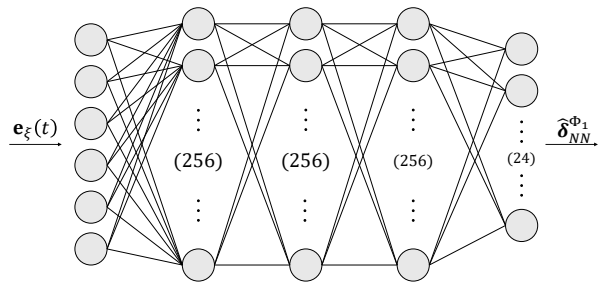

Fig. 1: Architecture for neural controller Φ_1.

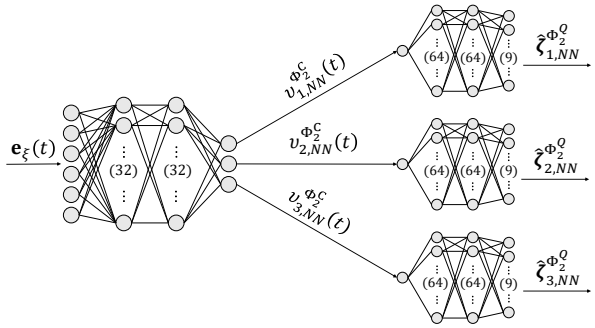

Fig. 2: Architecture for neural controller Φ_2.

IV. Results

Simulations were carried out on a nonlinear 3-DOF orbital dynamics simulator representative of LEO operations, assuming the body frame aligned with the orbital frame (null roll, pitch, and yaw). The model integrates translational motion, accounts for variable mass due to propulsion and includes the dominant perturbations at low altitude: atmospheric drag (using the USA Standard Atmosphere density model) and Earth's oblateness (J_2).

The simulation scenario involves three CubeSats in trailing formation orbiting in a Sun-Synchronous LEO orbit, with a required mutual separation of about 150 m. Table I summarizes initial conditions, formation configuration details and desired relative trajectories of each spacecraft are summarized. As reported in the table, an initial position error was introduced to evaluate the ability to restore the desired configuration. Sat #0 denotes the virtual leader of the formation, which represents the reference trajectory for Sat #1.

TABLE I: Initial conditions and formation requirements.

	Sat #0	Sat #1	Sat #2	Sat #3
Semi-major axis [km]	7001.388	7001.388	7001.398	7001.388
Eccentricity	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Inclination [deg]	97	97	97	97
RAAN [deg]	0	0	0	0
Argument of periapsis [deg]	0	0	0	0
Initial true anomaly [deg]	0	$4 \cdot 10^{-6}$	$-9 \cdot 10^{-4}$	$-1.8 \cdot 10^{-3}$
Leader	-	Sat #0	Sat #1	Sat #2
Desired anomaly w.r.t. the leader [deg]	-	0	$-9 \cdot 10^{-4}$	$-9 \cdot 10^{-4}$
Initial position error [m]	-	0.5	10	10

Tracking performance for each satellite j was compared in terms of:

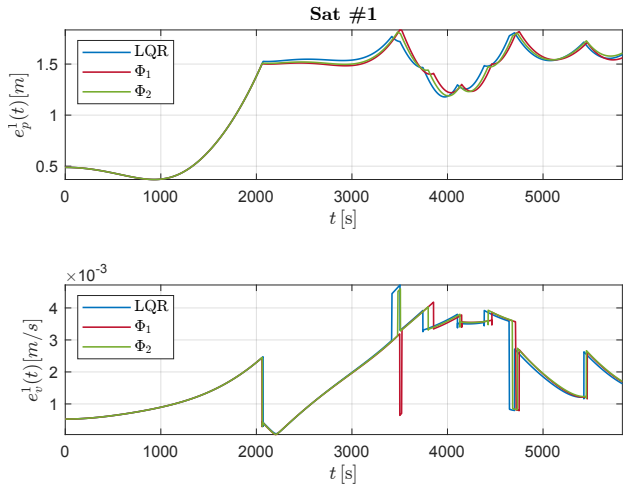

Fig. 3: Tracking error in relative position and velocity between Sat #1 and Sat #0 using LQR controller and NN-based controllers.

- position and velocity errors, computed as:

$$e_{\mathbf{p}}^j(t) = \|\mathbf{p}_j(t) - \overline{\mathbf{p}}_j(t)\|_2 \quad e_{\mathbf{v}}^j(t) = \|\mathbf{v}_j(t) - \overline{\mathbf{v}}_j(t)\|_2 \tag{26}$$

- fuel consumption over one orbital period ($T \approx 5800s$), estimated as follows:

$$\mathcal{J}_{tot}^j = \sum_a \int_{t=0}^{T} |F_a(t)|\, dt \qquad a = x, y, z \tag{27}$$

Figure 3 shows the comparison of the results obtained by using both the NN schemes against the expert controller in terms of tracking error in relative position and velocity between Sat #1 and Sat #0, its reference trajectory. Both NN-based controllers are able to clone the behavior of the expert with only some negligible differences. After the initial reconfiguration manoeuvre, similar steady state results are achieved, as shown in Figures 4 and 5, with the tracking error of followers with respect to their leaders (Sat #1 for Sat #2 and Sat #2 for Sat #3). Table II shows that the differences in terms of fuel consumption are negligible, with the modular scheme being the best expert clone.

TABLE II: Performance comparison in terms of total impulse.

	Total Impulse $[N \cdot s]$		
	LQR	Φ_1	Φ_2
Sat #1	0.225	0.225	0.225
Sat #2	3.100	3.150	3.100
Sat #3	2.695	2.625	2.750

V. Conclusions

This paper presented an NN-based controller with imitation-learning approach for a formation of CubeSats with on-off thrusters. Two architectures were compared: a monolithic MLP and a modular scheme that separates continuous regulation from discrete quantization. The numerical results show that the

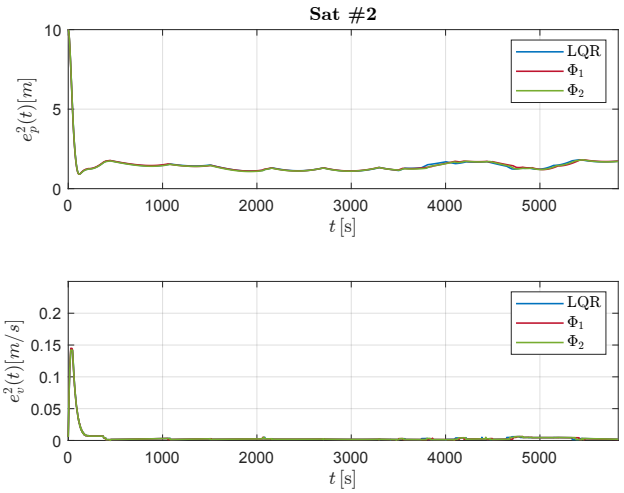

Fig. 4: Tracking error on the relative position and velocity between Sat #2 and Sat #1 using LQR controller and NN-based controllers.

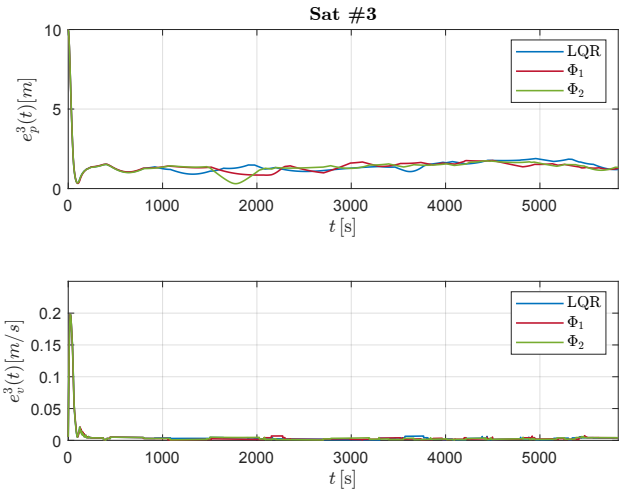

Fig. 5: Tracking error on the relative position and velocity between Sat #3 and Sat#2 using LQR controller and NN-based controllers.

modular controller achieves tracking performance comparable to the expert LQR while significantly reducing training data requirements. Future work will focus on end-to-end fine-tuning of the composite network involving the nonlinear simulator to further improve robustness and performance under realistic perturbations.

REFERENCES

[1] G. P. Liu and S. Zhang, "A survey on formation control of small satellites," *Proceedings of the IEEE*, vol. 106, no. 9, pp. 1860–1885, 2018.

[2] S. Bandyopadhyay, R. Foust, G. P. Subramanian, S. J. Chung, and F. Y. Hadaegh, "Review of formation flying and constellation missions using nanosatellites," *Journal of Spacecraft and Rockets*, vol. 53, no. 3, pp. 567–578, 2016.

[3] G. Di Mauro, M. Lawn, and R. Bevilacqua, "Survey on guidance navigation and control requirements for spacecraft formation-flying missions," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 3, pp. 581–602, 2018.

[4] S. Kukreti, A. Walker, P. Putman, and K. Cohen, "Genetic algorithm based lqr for attitude control of a magnetically actuated cubesat," 2015.

[5] M. Leomanni, A. Garulli, A. Giannitrapani, and F. Scortecci, "An mpc-based attitude control system for all-electric spacecraft with on/off actuators," in *52nd IEEE Conference on Decision and Control*, pp. 4502–4507, 2013.

[6] Y. Yang, "Analytic lqr design for spacecraft control system based on quaternion model," *Journal of Aerospace Engineering*, vol. 25, no. 2, pp. 196–204, 2012.

[7] S. R. Starin and R. K. Yedavalli, "Design of a lqr controller of reduced inputs for multiple spacecraft formation flying," pp. 1234–1239, 2001.

[8] R. Vazquez, F. Gavilan, and E. F. Camacho, "Model predictive control for spacecraft rendezvous in elliptical orbits with on/off thrusters," *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 251–256, 2015.

[9] S. Di Cairano, H. Park, and I. Kolmanovsky, "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 12, pp. 1398–1427, 2012.

[10] G. Song, N. V. Buck, and B. N. Agrawal, "Spacecraft vibration reduction using pulse-width pulse-frequency modulated input shaper," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 3, pp. 366–372, 1999.

[11] M. Navabi and H. Rangraz, "Comparing optimum operation of pulse width-pulse frequency and pseudo-rate modulators in spacecraft attitude control subsystem employing thruster," in *6th International Conference on Recent Advances in Space Technologies*, pp. 119–124, 2013.

[12] A. Heydari and S. N. Balakrishnan, "Optimal orbit transfer with on-off actuators using a closed form optimal switching scheme," in *AIAA Guidance, Navigation, and Control Conference*, p. 4635, 2013.

[13] H. Rouzegar, A. Khosravi, and P. Sarhadi, "Vibration suppression and attitude control for the formation flight of flexible satellites by optimally tuned on-off state-dependent riccati equation approach," *Transactions of the Institute of Measurement and Control*, vol. 42, no. 16, pp. 3210–3228, 2020.

[14] S. Silvestrini and M. Lavagna, "Deep learning and artificial neural networks for spacecraft dynamics, navigation and control," *Drones*, vol. 6, no. 10, p. 270, 2022.

[15] M. Shirobokov, S. Trofimov, and M. Ovchinnikov, "Survey of machine learning techniques in spacecraft control design," *Acta Astronautica*, vol. 179, pp. 51–66, 2021.

[16] X. Cao, P. Shi, Z. Li, and M. Liu, "Neural-network-based adaptive backstepping control with application to spacecraft attitude regulation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 9, pp. 2078–2088, 2017.

[17] A. M. Zou, K. D. Kumar, and Z. G. Hou, "Quaternion-based adaptive output feedback attitude control of spacecraft using chebyshev neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 9, pp. 1457–1471, 2010.

[18] L. Cheng, Z. Wang, F. Jiang, and Y. Gao, "Real-time optimal control for spacecraft orbit transfer via multiscale deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3794, 2018.

[19] R. Sun, J. Wang, D. Zhang, and X. Shao, "Neural-network-based sliding-mode adaptive control for spacecraft formation using aerodynamic forces," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 7, pp. 1583–1598, 2018.

[20] S. Silvestrini and M. Lavagna, "Neural-based predictive control for safe autonomous spacecraft relative maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 1, pp. 126–143, 2021.

[21] H. Yu, L. Dou, X. Zhang, and J. Li, "Safe reinforcement learning: Optimal formation control with collision avoidance of multiple satellite systems," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[22] J. G. Elkins, R. Sood, and C. Rumpf, "Bridging reinforcement learning and online learning for spacecraft attitude control," *Journal of Aerospace Information Systems*, 2022.

[23] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: a survey of learning methods," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–35, 2017.

[24] T. Osa, J. Pajarinen, G. Neumann, and J. A. Bagnell, "An algorithmic perspective on imitation learning," *Foundations and Trends in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.