https://doi.org/10.3311/PPee.40919
Creative Commons Attribution ⓘ

Periodica Polytechnica Electrical Engineering and Computer Science, 69(3), pp. 280–290, 2025

Enhanced Fixed-wing Leader-followers UAV Formation Control Integrating Pre-tuned TD3 Reinforcement Learning and Consensus-based Control Methods

Huda Naji Al-Sudany[1*], Béla Lantos[1]

[1] Department of Control Engineering and Information Technology, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[*] Corresponding author, e-mail: alsudany@iit.bme.hu

Received: 07 May 2025, Accepted: 03 July 2025, Published online: 16 July 2025

Abstract

This paper addresses a critical challenge in the field of attitude control of fixed-wing Unmanned Aerial Vehicles (UAVs), focusing on the leader and multi-followers' problem. It introduces Twin Delayed Deep Deterministic Policy Gradient (TD3) based on Cascade-forward ANN networks approach to control the leader and multiple followers in autonomous navigation and path tracking. The TD3 component is initially trained off-line and then applied in real-time. The control system incorporates a novel adaptive Laplacian consensus protocol using an undirected communication graph model that adjusts inter-UAV connection weights in real time based on relative positions. This approach was implemented on a leader-follower formation consisting of one leader and three follower UAVs. The paper includes a stability analysis of the proposed method, demonstrating the system's overall stability. The effectiveness of this approach is validated through a MATLAB simulation which demonstrates that the TD3 based on ANN (Cascade-forward networks), which is used to train the actor and the twin critics networks, has superior performance with low tracking error, good formation keeping during aggressive maneuvers, and reduced control surface oscillations. The application exhibited improved adherence to the prescribed distance during sharp turns, with fewer formation deviations at the trajectory end point. The findings result verify that combining strategies leads to the formation integrity. The TD3 based on ANN which uses Cascade-forward networks implementation offers also significant improvement in stability, accuracy, and control efficiency for practical UAV formation control applications.

Keywords

unmanned aerial vehicles (UAVs), attitude control, artificial neural network (ANN), reinforcement learning (RL), twin delayed deep deterministic policy gradient (TD3), Laplacian consensus algorithm

1 Introduction

Recently, long distance fixed-wing unmanned aerial vehicles (UAVs) have gained widespread use due to their simple design, compact size, and high versatility. In military applications, UAVs are employed for tasks such as reconnaissance and ground attacks. In civilian contexts, they are used for activities including surveillance, pesticide application, and project inspections. These UAVs are fundamentally nonlinear systems characterized by multiple highly coupled variables, as well as numerous inputs and outputs. During trajectory tracking missions, the flight process is subject to various disturbances, including unmodeled dynamics, internal aerodynamic fluctuations and external environmental factors. These challenges complicate the design of trajectory tracking control systems. Thus, developing a high-precision trajectory tracking controller is crucial for the effectiveness of the UAV's flight control system [1]. With the increasing complexity of aerial missions in ambiguous environments, the incorporation of advanced machine learning techniques into classical control frameworks has become a driving force. Unlike other neural networks, ANN networks possess strong self-learning capabilities, enabling them to derive system control relationships through online learning. This enhances the system's resistance to interference. Furthermore, ANN are characterized by their robustness and fault tolerance, allowing them to adjust and adapt to changes within the control system effectively. This adaptability ensures that the control system meets the desired requirements as outlined in this study [2].

Recently many articles indicated that deep reinforcement learning (DRL) is well capable of managing the uncertainties

Cite this article as: Al-Sudany, H. N., Lantos, B. "Enhanced Fixed-wing Leader-followers UAV Formation Control Integrating Pre-tuned TD3 Reinforcement Learning and Consensus-based Control Methods", Periodica Polytechnica Electrical Engineering and Computer Science, 69(3), pp. 280–290, 2025. https://doi.org/10.3311/PPee.40919

and nonlinearities in multi-UAV systems. Particularly, the Twin Delayed Deep Deterministic Policy Gradient (TD3) method has been established as a robust solution for robotics continuous control tasks, by being immune to overestimation bias through its twin-critic design [3, 4].

The UAV formation control is further complicated by the need for accurate relative positioning and vehicle-to-vehicle communication, leading to research in consensus-based control methods. Adaptive Laplacian consensus approaches that adaptively change connectivity weights based on inter-UAV distances offer a sound solution to resilient formation keeping in aggressive maneuvers [5]. More recent studies have integrated such adaptive consensus algorithms with neural network controllers to further improve tracking performance and efficiency in control. In this study, Sekwa model of UAVs was utilized in [6]. The fixed-wing UAV control and parameter estimation details are provided in [7].

The main task of this paper is the evaluation of using TD3 together with ANN-based Cascade-forward networks methodology to improve the training of TD3, which was performed off-line, and uploaded as Simulink model to later use in control UAV formation. In our approach, an adaptive Laplacian consensus algorithm is employed, which updates inter-vehicle coupling in real time using position error as inputs and ensures the formation to maintain its integrity despite sudden maneuvers. Our hybrid strategy does not only suppress all spatial-directional tracking errors but also avoids oscillations of control surfaces that diminishes actuators' wear and increases system-wide energy efficiency. Finally, through the integration of DRL with adaptive consensus and neural network techniques, our work addresses the challenge of precise formation control in multi-UAV systems. The remaining sections of the paper are arranged as follows: Section 2 summarizes research methodology, Section 3 deals with stability analysis framework, Section 4 presents findings and discussion, Section 5 contains conclusion and further research.

2 Formation control strategy with TD3 and ANN

2.1 Control strategy

The central problem of the UAV control, both for a single UAV or for their ensemble, is the attitude (orientation) control. If the attitude is well set, then the longitudinal motion (position) can usually be established separately.

The components of the followers may have similar HW/SW structure. Their kinematic and dynamic model can be assumed equal with similar parameters and conventional sensors (3D IMU, 3D magnetometer and GPS). Such a set of sensors is now always relatively cheap and reliable.

However, in formation control the leader has a distinguished role. Only the leader has the "brain" chip that can make decisions intelligently, the followers only have the chips that can receive the control command action and send the state signals.

The goal of the state is to reach the target area with the formation as soon as possible. If the leader enters the target area, the mission is completed. During the motion (fixed or moving) obstacles may be present and collision of the formation with the obstacles and between the formation components themselves must be avoided which needs special sensors and decisions making possibility of the leader. Consequently, path design and other intelligent problems are the task of the leader.

Communication has the main tasks to collect state information of the followers for the leader supporting path design, and send control commands for the followers, furthermore, exchange state and control information among the connected followers. It will be assumed that the communication is defined by an undirected graph.

The formation control generates the set of control signals for the different low level (single) UAVs which play the role of command signals of the low-level actuators. The control of a single UAV will not be discussed here but it is well known that the best methods use quaternion logarithm-based attitude (orientation) control and nonlinear inversion plus PID-based position control [7, 8]. The position control uses intensively the nonlinear dynamic model of the single UAV so that its identification must be solved before [9].

For formation control at High-level it is usually assumed that the UAVs can be approximated by the point-mass model in the Flat-Earth (Quasi) Inertial NED System. This simplifies the kinematic model, but because of the presence of acceleration, the dynamic model remains nonlinear [10]. It is typical to assume that the followers have equal orientation with the leader. Hence, for attitude control two problems are dominant nowadays:

1. The leader develops different paths with common orientation for the UAVs. The distances between the formation UAVs are prescribed and the formation should move with common velocity after a transient. The present paper deals with this problem.
2. No distances are prescribed, but the components must follow the path of the leader without collision [11].

2.2 System description

Inner Loop (Attitude Control): Inputs: Desired angles (ϕ_d, θ_d, ψ_d), and current angles (ϕ, θ, ψ) plus angular velocities (p, q, r). Outputs: Control commands δ_e: elevator, δ_a: aileron, δ_r: rudder.

Outer Loop (Path Tracking): Inputs: Desired position (X_d, Y_d, Z_d), and current position (X, Y, Z). Outputs: Desired angles (ϕ_d, θ_d, ψ_d).

The proposed formation control system adopts a leader-followers architecture comprising one leader UAV and three follower UAVs, all operating within a three-dimensional environment. The followers are tasked with tracking the leader's trajectory while preserving a prescribed formation geometry. The control system is structured hierarchically and is comprised of three primary components:

1. a path tracking controller,
2. a formation maintenance controller, and
3. an attitude stabilization controller.

2.3 Components of the combination

It will be assumed that the reader is familiar with RL [12].

2.3.1 Cascade-forward networks

Cascade-forward networks (MATLAB tool) [13] were used to train both actor and twin critics networks. One hidden layer with 50 neurons was used to create actor and another hidden layer with 18 neurons to create network for each critic.

2.3.2 TD3-based reinforcement learning control

TD3 is a model-free reinforcement learning algorithm used to learn optimal action policies based on a reward signal. The TD3will be trained to map states (input data: angles) to corresponding actions (command data).

Reinforcement learning involves the agent interacting with its environment and learning by trial and error (in the absence of labeled data). As a measure of performance, the learning agent is presented with a sequential decision problem and receives feedback. This interaction is represented as the feedback structure. To further enhance the attitude control system, a Twin Delayed Deep Deterministic Policy Gradient (TD3) reinforcement learning algorithm is incorporated. This algorithm employs an actor-critic architecture comprising [14]:

1. Actor Network: A deep neural network mapping the state $\boldsymbol{s}_t$ to the control action $\boldsymbol{a}_t$, featuring hidden layers with 50 neurons.
2. Critic Networks: Two separate networks estimate the Q-value for state-action pairs, thus providing robust value estimates through double Q-learning.

It addresses the overestimation bias inherent in traditional Deep Deterministic Policy Gradient (DDPG) methods through three key innovations:

1. Twin Critic Networks: Dual Q-networks to decouple bias and variance in value estimation.
2. Delayed Policy Updates: Reduced actor update frequency to stabilize training.
3. Target Policy Smoothing: Addition of noise to target actions to mitigate value function overfitting.

In the TD3 approach, two sets of Critic networks are utilized to compute distinct Q. To prevent continuous overestimation, the desired Q is determined by choosing the minimum Q of the two networks. Actor network $\boldsymbol{\pi}_\varphi$ is used to create action, similarly to DDPG. The Actor network is updated following many updates to the Critic network [12].

Parameter updates are delayed thus the Actor can decide what to do once the critical network was not overloaded.

The chosen action is exploration with noise:

$$\boldsymbol{a} \sim \pi_\varphi(\boldsymbol{s}) + \varepsilon, \text{ where } \varepsilon \sim N(0,\sigma). \tag{1}$$

The target Q calculation formula is:

$$Q'(\boldsymbol{s}_{t+1},\boldsymbol{a}_t) = \min\left(Q_1'(\boldsymbol{s}_{t+1},\boldsymbol{a}_t), Q_2'(\boldsymbol{s}_{t+1},\boldsymbol{a}_t)\right). \tag{2}$$

The loss function of the Critic network is computed in order to update the network of the Critic.

$$\begin{aligned} \text{Loss}_1 &= \frac{1}{N}\sum_i \left(\left(\boldsymbol{r}_i + \gamma Q'(\boldsymbol{s}_{t+1},\boldsymbol{a}_t)\right) - Q_1(\boldsymbol{s}_t,\boldsymbol{a}_t)\right)^2 \\ \text{Loss}_2 &= \frac{1}{N}\sum_i \left(\left(\boldsymbol{r}_i + \gamma Q'(\boldsymbol{s}_{t+1},\boldsymbol{a}_t)\right) - Q_2(\boldsymbol{s}_t,\boldsymbol{a}_t)\right)^2 \end{aligned} \tag{3}$$

The actor network is updated by policy gradient:

$$\nabla_\varphi J(\varphi) = N^{-1}\sum \nabla_a Q(\boldsymbol{s},\boldsymbol{a})\big|_{a=\pi_\varphi(s)} \nabla_\varphi \pi_\varphi(\boldsymbol{s}). \tag{4}$$

The target network is updated by the following rules:

$$\begin{aligned} \theta_i' &\leftarrow \tau\theta_i + (1-\tau)\theta_i' \\ \varphi_i' &\leftarrow \tau\varphi_i + (1-\tau)\varphi_i'. \end{aligned} \tag{5}$$

The TD3 algorithm integrates several measures, including regularization (with coefficients from 0.01 to 0.015 for the actor and critics), and early stopping based on validation.

The actor network is represented by

$$\boldsymbol{a}_t = \mu_\theta(\boldsymbol{s}_t) \tag{6}$$

while the critic networks are defined as

$$Q_{\varphi_i}(\boldsymbol{s}_t,\boldsymbol{a}_t),\ i \in \{1,2\} \tag{7}$$

with θ and φ_i being the respective parameters. Its topology differs in output layer: 1 neuron (linear activation). Normal distributions have a variance parameter σ, reward value is $\boldsymbol{r}$, and γ is the discount factor. The soft update attenuation coefficient is denoted by $\tau(\tau \leq 1)$. By adding noise into the target actions, the smooth regularization enables TD3 to effectively smooth the target strategy and avoid the policy network from over-fitting.

2.4 Incorporating UAV Interactions with undirected graph

In multi-agent systems like UAV formations, interactions between agents are critical. By integrating the undirected graph factor the approach accounts for mutual influence among UAVs, ensuring coordinated behavior in a decentralized setting. The path tracking controller ensures that each follower UAV maintains a defined offset relative to the leader's path, while the formation maintenance controller regulates the relative positions among followers. Concurrently, the attitude stabilization controller is responsible for maintaining the desired orientation of each UAV. The dynamics of each UAV are characterized using a six-degrees-of-freedom (6-DOF) model that encompasses both translational and rotational motions.

The dynamics of a single UAV is modeled by the differential equation

$$\dot{\boldsymbol{x}}_i = f\left(\boldsymbol{x}_i, \boldsymbol{u}_i\right), \tag{8}$$

where $\boldsymbol{x}_i = \left[p^T, v^T, \phi, \theta, \psi, P, Q, R\right]_i^T$ or

$\boldsymbol{x}_i = \left[p^T, v^T, q^T, P, Q, R\right]_i^T$.

The $q = \left[q_0, q_1, q_2, q_3\right]^T = [s, \overline{w}^T]^T$ is the unit quaternion, and the control input vector is given by

$$\boldsymbol{u}_i = \left[F_T, \delta_a, \delta_e, \delta_r\right]_i^T \tag{9}$$

with F_T representing the thrust force and δ_e, δ_a, δ_r corresponding to the elevator, aileron, and rudder deflections, respectively. Aerodynamic forces are initially modeled in the wind frame and then transformed to the body frame through the transformation matrix from wind-axis to body axis frame based on the angle of attack α and sideslip angle β. The transformation is given by $\mathrm{Rot}\left(y, -\propto\right)\mathrm{Rot}\left(z, \beta\right) = \boldsymbol{S}_T$, where

$$\boldsymbol{S}_T = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix}, \quad r_b = \boldsymbol{S}_T r_w \tag{10}$$

with the corresponding transpose $\boldsymbol{S} = \boldsymbol{S}_T^T$ used for the inverse transformation. The full state and full control are $\boldsymbol{x} = \left[x_1^T, x_2^T, \ldots, x_N^T\right]^T$ and $\boldsymbol{u} = \left[u_1^T, u_2^T, \ldots, x_N^T\right]^T$ respectively for followers, while $\boldsymbol{x}_l$ and $\boldsymbol{u}_l$ for leader.

The communication of the formation is given by a fully connected undirected graph where all the followers are mutually connected to each other and also to the leader, see Chapters 2 and 3 of [15], and Chapter 3 of [16]. The properties of the graph indicate similar influence between any two UAVs which results in a well-tuned control framework. For formation control, an enhanced Laplacian consensus algorithm is implemented. In contrast to conventional consensus methods with fixed weights, this approach employs an adaptive Laplacian matrix whose weights are dynamically adjusted based on the relative position errors. The Laplacian matrix is

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}. \tag{11}$$

Where $\boldsymbol{D}$ = diag($d_1 \ldots d_N$) is a diagonal matrix with elements $d_i = \sum_{j=1}^{N} a_{ij}$ where N is the number of followers (in our experiments $N = 3$), and $\boldsymbol{A} = [a_{ij}]$ is the $N \times N$ type adjacency matrix, whose elements are

$$a_{ij} = \begin{cases} w_{ij}, & \text{if UAVs } i \text{ and } j \text{ are connected} \\ 0, & \text{otherwise} \end{cases}.$$

The weight w_{ij} is defined by

$$w_{ij} = 0.2 * \left(\frac{2}{1 + e^{-0.5\min\left(100 * \left|d_{ij} - d_{ij}^*\right|\right)}} - 1 \right) \tag{12}$$

with d_{ij} representing the actual distance between UAVs i and j, with d_{ij}^* being the desired inter-UAV distance. This sigmoid-based weighting function effectively bounds the weights between −0.2 and 0.2, ensuring smooth transitions and mitigating the risk of instability due to abrupt control actions. The error is formulated as

$$e_i\left(t\right) = \alpha_c \left[\sum_{j=1}^{N} \boldsymbol{L} * \left(x_i\left(t\right) - x_j\left(t\right)\right) \right] + \beta_f \left[x_i\left(t\right) - x_l\left(t\right)\right]. \tag{13}$$

The gains α_c (for consensus) and β_f (for formations) are selected (0.03 and 0.015) respectively.

Next, the following formula was used for the leader-follower consensus-controllers (CC):

$$CC_i\left(t\right) = -c * e_i + \left[\delta_a, \delta_e, \delta_r\right]_i^T \tag{14}$$

where $i = 1, \ldots, N$ and c is positive constant.

2.5 The stages of implementation

Stage 0: Off-line tuning of TD3: The desired paths of the followers are coming from the leader playing the role of the "brain" chip. The desired position and orientation of the body frames of the followers with respect to the leader are given in the NED frame, which is a quasi-inertia frame.

Moreover, the desired positions of the followers are known through the fixed formation configuration, hence the leader can easily produce the path for each follower. On the other hand, the orientation (attitude) of each follower is equal to the leader's orientation by assumption, hence the off-line TD3 tuning is similar for each follower, which simplifies the tuning. The input of the tuning contains the desired Euler angles of the reference path together with their first and second derivatives of the reference path, the previous value of the reference path, the Euler angles, and the angular velocities, called the extended state. The outputs are represented by control commands δ_e, δ_a, δ_r which indicate the action. The actor network has 50 neurons, while the twin critic networks have 18 neurons each. The neural network architecture uses the cascade forward net tool in Matlab [13] to train both actor and critic networks. Training, validation, and test are divided by 80%, 10%, and 10% proportion of total data. Fig. 1 presents the performance of the TD3 training. A leader UAV is followed by three followers UAVs in the online phase and the orientation is the same. The trained model is loaded, and real-time operation starts during the online phase. Table 1 shows the parameters related to the performance of TD3.

Stage 1: Storage reserve for data logging and setting the initial conditions: To speed up SW, the storage space is reserved for the total time length of signals. For the known constant desired speed of follower UAVs the controller sampling time Tc = 10 ms was chosen. For simplicity, in the sequel Euler method is used for integration of the state equations, although RK4 can be used similarly. This stage includes the initial state, an output place holder, and parameters to control the simulation loop where the notation $t = kTc$ is used. The number of followers was chosen as $N = 3$.

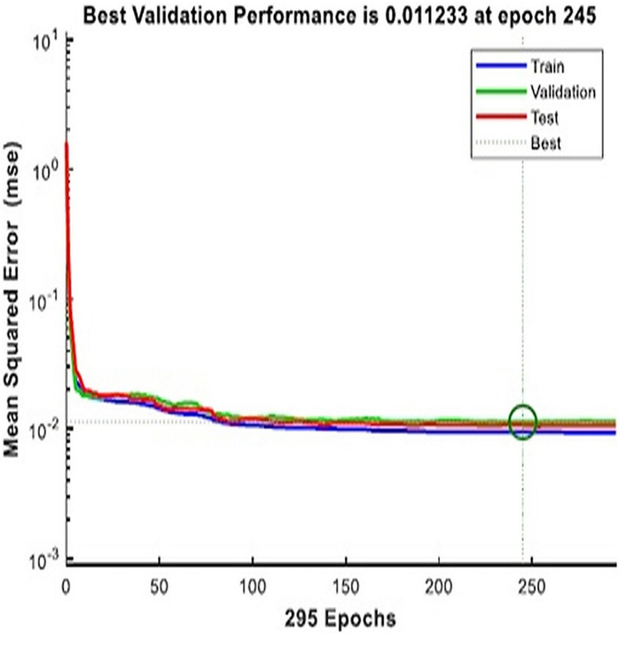

Fig. 1 Performance of TD3 training

Table 1 The parameters that influence the performance of TD3

Parameter	Value
Action clip (Maximum control action magnitude (rad)	0.5
Noise scale (Exploration noise during operation)	0.05
Target update rate (Soft update rate for target networks (τ))	0.005
Policy delay (Update policy every d critic updates)	2
Actor network training parameter goal	1e-4
Actor network performs parameter regularization	0.01
Number of episodes	5000
Discount factor	0.99
Batch size	256

Stage 2: Time loop and embedded follower's loop: The time loop considers the interval $[t = kTc, t + 1 = (k + 1)Tc]$. For every UAV, the differential equations are operating independently. As a result, a control cycle can be applied to every time sample and each follower. The states and the controls are known for the left time in the interval and the control will be determined for the next time and the states will be integrated for the next time in the interval. The undirected communication graph determines the state and control exchange. The leader UAV gives the reference signals to the followers and is pre-designed to follow a particular trajectory. Using the specified communication architecture, the leader continually communicates its current state information to the followers at each time step, including position and attitude. The followers are able to calculate their own control signals in relation to the leader's state, which guarantees coordinated behavior and formation maintenance. The components, $\boldsymbol{A}$ which indicates the adjacency matrix and diagonal matrix $\boldsymbol{D}$ and the Laplacian graph ($\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$), can be used in the cycle based on Eq. (11). Then a sequence of computations begins for each follower. Since each UAV's attitude is known (at the left side of the interval), the error to the reference signal can be computed. Next, nonlinear inversion and PID control (for position tracking) are used to calculate the thrust force, which is the first component of the control input vector in Eq. (9). Following the determination of the thrust, the differential equations for the absolute value of the velicity (vT), α, and β can be solved. After calculating the dynamic error using Eq. (13), the consensus controller (CC) in Eq. (14), is used to determine the new actuator commands δ_e, δ_a, and δ_r for time t + 1. Finally, the new control inputs are used to update the system states. The flow chart is shown in Algorithm 1.

Algorithm 1 Flow chart of the proposed control method

```
Saved TD3 network from training
Begin
    For sample $k$ in number of samples % $t = kTc$
Apply Eqs. (11), (12) to compute Laplacian matrix and the weight $w_{ij}$
    For follower $i$ in number of followers
Compute the dynamic errors Eq. (13)
Calculate Eq. (14) to obtain the consensus controllers ($CC$) based on
Eqs. (11)–(13)
    Calculate $U_{UG,i}$ from undirected graph based on Eq. (14)
    Calculate $U_{\mathrm{TD3},i}$ from trained TD3 networks $U_{\mathrm{TD3},i} = [\delta_e, \delta_a, \delta_r]$
        Apply Eq. (15)
        Apply UAV model equations to get thrust force $F_T$
        Extend $U_i$ in Eq. (15) with $F_T$
        Apply to UAV model to update the state of the system based
        on the new inputs (Eq. (9))
        end for $i$
    end for $k$
end
```

3 Stability analysis

Because of the stochastic character, the original Lyapunov theory is less suitable for stability investigation. The stability analysis of the proposed UAV control framework, based on the TD3 algorithm, assures stability in stochastic sense. The next investigation shows that the fusion with consensus-based extension, is also in harmony with the deterministic bounded input, bounded output (BIBO) stability.

The training process utilizes neural networks for actor and critic functions. While this implementation incorporates a deeper network architecture and regularization techniques to promote robust learning, it is important to note that certain aspects of the standard TD3 algorithm, such as target networks and target policy smoothing, are simplified in this approach.

The actor network is trained to produce control commands for the UAV. The output layer of the actor network employs a bounded activation function, then the control output $U_{\mathrm{TD3},i}$ will inherently be bounded within a specific range, this means there is a constant M_1 that applies

$$\left|U_{\mathrm{TD3},i}\right| < M_1 < \infty .$$

The main goal is to keep followers close to leader considering errors between followers. The command output $U_{\mathrm{TD3},i}$ of TD3 is weighted as with normalized weights $K_{\mathrm{TD3}} = 1$ and then control command $U_{UG,i}$ based on undirected graph is added also with a weight $K_{UG} < 1$:

$$U_i\left(t\right) = K_{\mathrm{TD3}} U_{\mathrm{TD3},i} + K_{UG} U_{UG,i} . \tag{15}$$

$U_{UG,i}$ is also bounded by a constant $\left|U_{UG,i}\right| < M_2 < \infty$. $U_{\mathrm{TD3},i}(t)$ represents control commands obtained from trained TD3 networks.

Denote $S(k)$ the state of the follower i at sample k or time instant t:

$$S\left(k\right) = x_i\left(k\right). \tag{16}$$

3.1 Proving BIBO stability under control U_i

It has to be shown that the followers remain stable (i.e., bounded) when applying the control law

$$U_i\left(t\right) = K_{\mathrm{TD3}} U_{\mathrm{TD3},i} + K_{UG} U_{UG,i} .$$

The TD3 control output is bounded within small ranges based on training on stable data. The TD3 control output U_{TD3} is bounded by the actor's tanh output layer:

$$\left\|U_{\mathrm{TD3}}\right\| \le U_{\mathrm{TD3,max}} \le M_1 < \infty . \tag{17}$$

$U_{UG,i}$ is the weighted mixture of bounded $U_{UG,i}$ components by the adjacency matrix between followers and at the end is multiplied by a small constant which can be ensured to be less than $U_{\mathrm{TD3},i}$. This ensures:

$$U_{UG,i} \ll \left\|U_{\mathrm{TD3}}\right\| \le U_{\mathrm{TD3,max}} \le M_1 < \infty .$$

Since $|U_{\mathrm{TD3}}|$ and $U_{UG,i}$ are bounded, U_i remains bounded.

The time derivative of the state $S(k)$ in Eq. (16), denoted $\dot{S}(k)$, it describes how the state vector changes over time:

$$\dot{S}_k = \dot{x}_i = f_k\left(x_i, U_i\right). \tag{18}$$

Assuming that function Eq. (8) $f_k(x_i, U_i)$, at sample k for follower i is Lipschitz continuous, then finite inputs produce finite state changes.

The training data used for the actor network plays a crucial role in the stability of the learned policy. If the training data encompasses stable and well-behaved control actions, the actor network is more likely to learn a policy that generates stable commands. The L2 regularization applied during training further helps to prevent the network weights from growing excessively, which can contribute to more stable and generalized outputs.

Define $e(k) = S(k) - S_{des}(k)$, because the error dynamics under the hybrid control law is $\dot{e}(k) = \dot{S}(k) - \dot{S}_{des}(k)$ where $S_{des}(k)$ is the desired state, hence

$$\dot{e}\left(k\right) = \left[f_k\left(x_i, U_i\right)\right] - \dot{S}_{des}\left(k\right). \tag{19}$$

Since $f(x, U_i)$ is Lipschitz, and U_i is bounded, the error dynamics remain finite and controllable. Notice that BIBO stability is equivalent to control signals convergence. The evolution of this error over time is crucial for stability analysis. The final result is that the control output generated by actor network is bounded and the system dynamics are Lipschitz continuous; this boundedness of the input can lead to the boundedness of the state and thus BIBO stability.

3.2 Justification of stability using TD3 components

Architectural and algorithmic improvements in the TD3 framework guarantee the reliability of the suggested TD3-based control system. There are many factors which can influence the stability.

Delays in actor updates prevent policy oscillation, target policy smoothing guarantees resistance to noise and clipped double Q-learning reduces overestimation bias. These elements are working together and enforce stable learning dynamics, limited tracking errors, and smooth control instructions. Simulation findings verify the approach's stability by confirming that the TD3 agent consistently exhibits leader-follower behavior across a range of environments. Based on plots, TD3 implementation provides good stability with effective damping of high-frequency oscillations, appropriate control magnitudes during maneuvers, quick settling after disturbances, good power spectrum roll-off indicating smooth control. The overall smooth control with limited high-frequency content suggests TD3 implementation is successfully improving the stability of the UAV formation. Stability with thrust force which extend U_i based on Eq. (20) that can compute the thrust force

$$F_T = \frac{1}{g_0}\left(\dot{v}_{ref} - f_0 + a_1 e_v\right), \tag{20}$$

where:

- g_0 represents a gain or input-to-output of control loop;
- $\dot{v}_{ref}$ is the desired acceleration in quasi-inertia NED frame;
- f_0 is the current (aerodynamic and gravity) deceleration/per unit mass;
- e_v is the speed tracking error, and
- a_1 is the gain.

With next conditions:

1. All gains are chosen in a stable state.
2. Other control commands are proved to be stable.
3. UAV parameters are nominal, and dynamics of UAV are well modeled.
4. F_T is also limited due to dynamics limitations and ensures stability in the worst and unexpected cases.

With these conditions, the full vector of control commands in Eq. (9) is applied and its results are stable, which is approved by simulations results in Section 4.

4 Simulation results

The implementation of the control method TD3 based ANN was successful in formation flight control with the UAVs maintaining their relative positions as they followed the prescribed follower path. Let's first consider Cascade TD3 tracking of the trajectory. Figs. 2–4 indicate the good

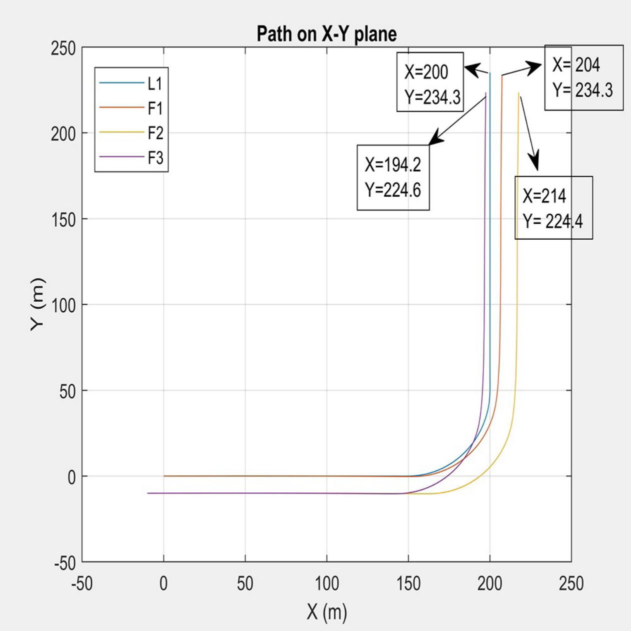

Fig. 2 2D tracking of the trajectory of Cascade TD3

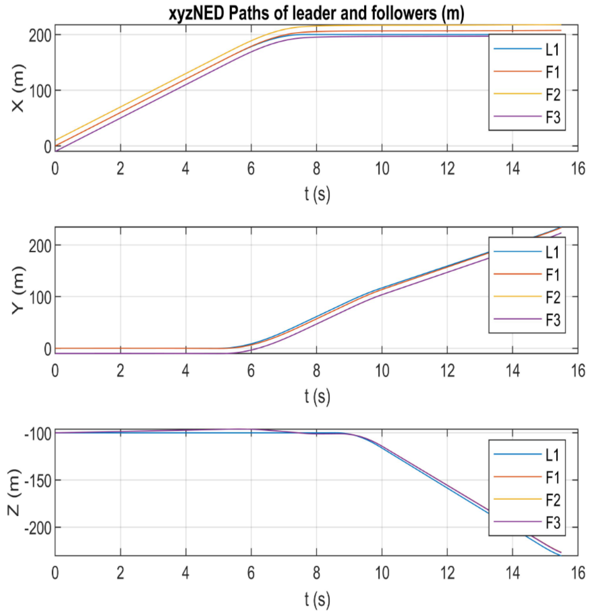

Fig. 3 Paths of leader and followers of Cascade TD3

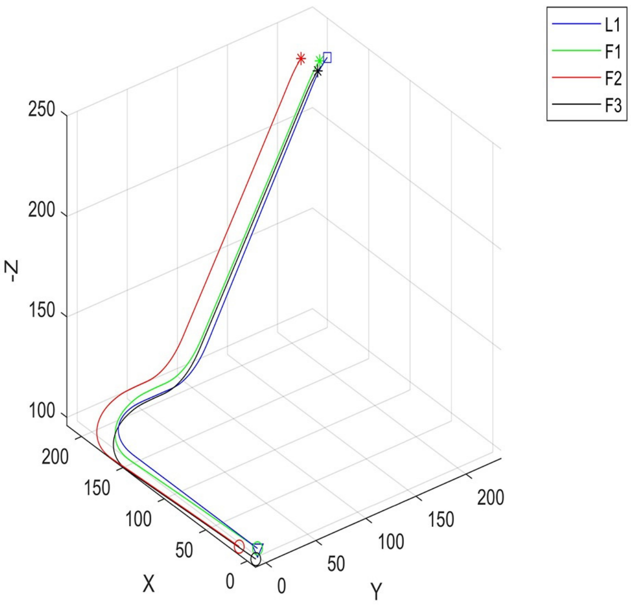

Fig. 4 3D tracking of the trajectory of Cascade TD3

position coordinates and tracking properties of the leader ($L1$) and followers $F1$, $F2$, $F3$ over time and in 3D. All followers maintained acceptable relative positions that did not change behind the leader, and the formation integrity was preserved along the trajectory.

4.1 Control response analysis

The control surface deflections and the angular rates provide valuable indications of the performance characteristics.

Figs. 5–7 show angular rate p, q, r while Figs. 8–10 present the elevator (δ_e), aileron (δ_a) and rudder (δ_r), responses of Cascade TD3 respectively.

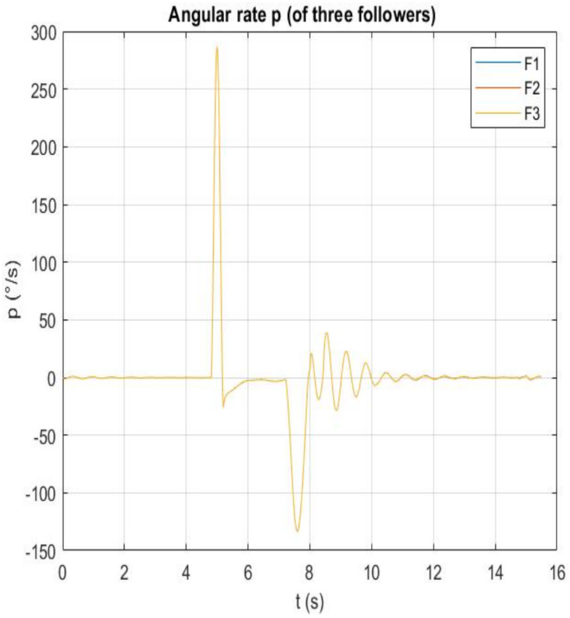

Fig. 5 Angular rate p (of three followers) of Cascade TD3

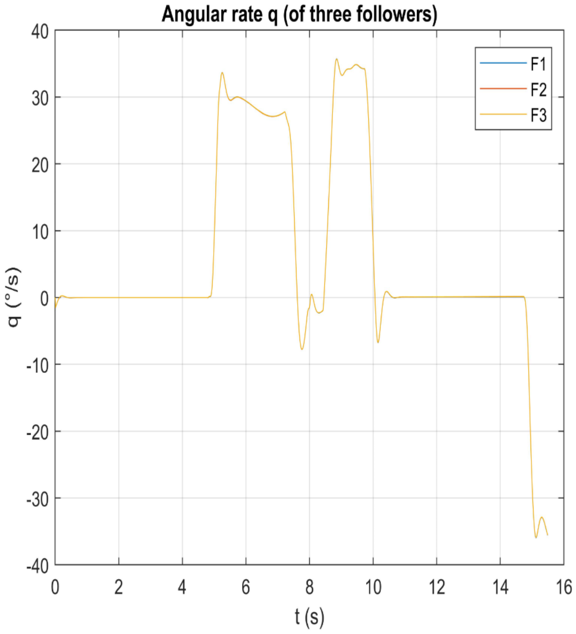

Fig. 6 Angular rate q (of three followers) of Cascade TD3

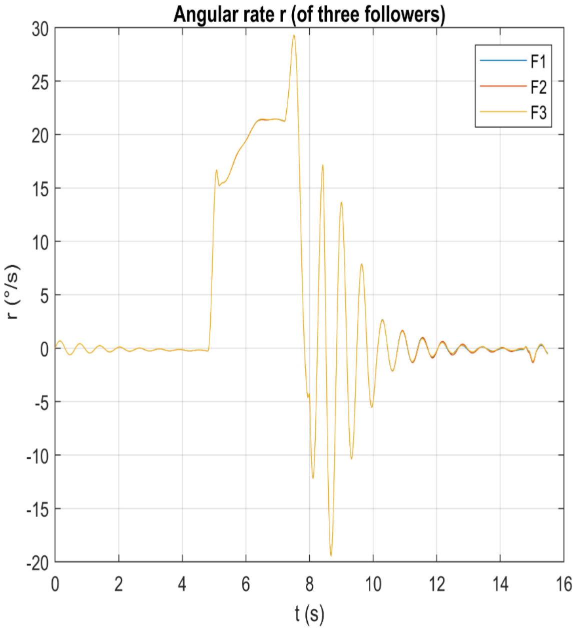

Fig. 7 Angular rate r (of three followers) of Cascade TD3

Linear velocity components in Fig. 11 and Thrust force in Fig. 12 of all followers are also shown. The absolute value of the velocity controls is 30 m/s with small errors.

The superior performance of the TD3 based on cascade forward network approach can be attributed to several factors. TD3 reinforcement learning algorithm based on ANN demonstrates better generalization capability for the complex, nonlinear dynamics of UAV formation flight.

This is particularly evident in the oscillatory behavior in the control surface deflections and angular rates.

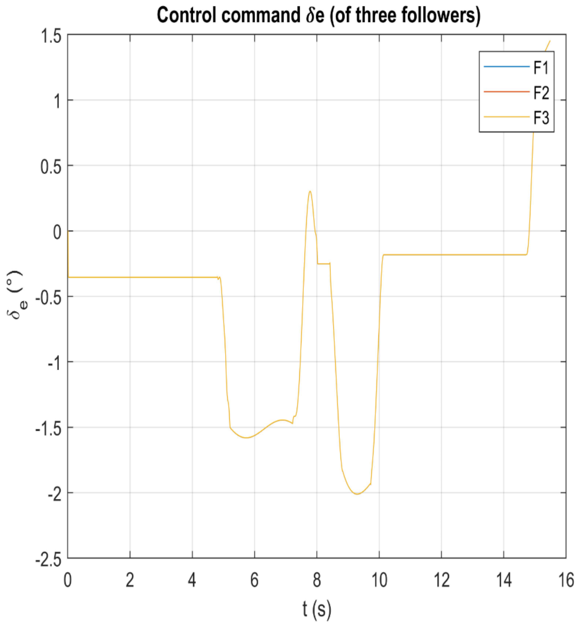

Fig. 8 Control command δ_e (of three followers) of Cascade TD3

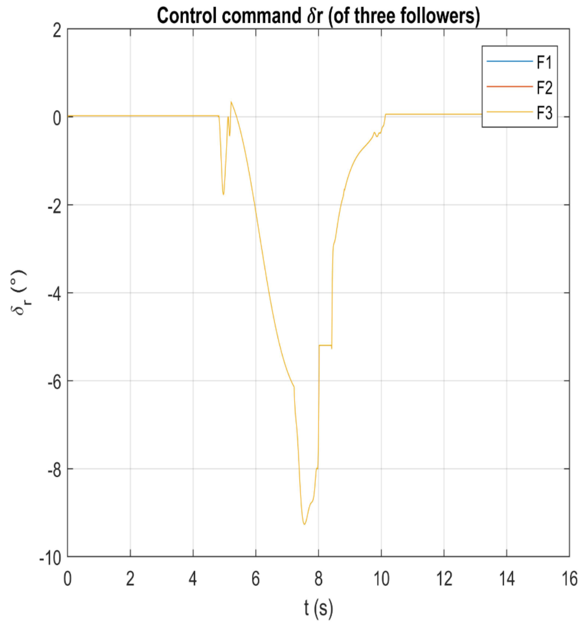

Fig. 10 Control command δ_r (of three followers) of Cascade TD3

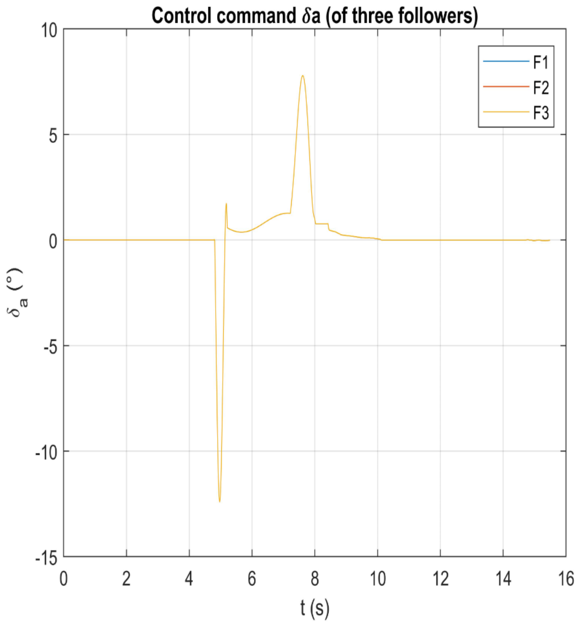

Fig. 9 Control command δ_a (of three followers) of Cascade TD3

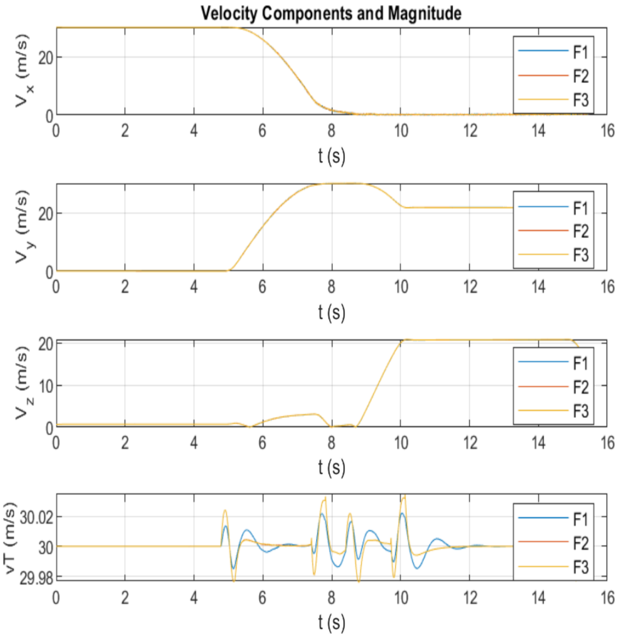

Fig. 11 Velocity component and magnitude of all followers of Cascade TD3

5 Conclusion and further research

The paper proposed a novel UAV formation control system that integrates TD3 based on ANN using cascade forward network to train both actor and critic networks together with an adaptive Laplacian consensus protocol.

This hybrid strategy addresses the complex nonlinear dynamics and inter-vehicle coordination challenges involved in multi-UAV formations. Simulation results indicate that the TD3 based on ANN solution enjoys significant performance. The MSE of Actor network is 4.1941e-06 and MSE of Critic networks 1 and 2 are: 0.00057657, 0.02234 respectively.

Moreover, the adaptive consensus strategy generated tighter formation hold and smoother control surface motions that are critical in reducing actuator stress and increasing overall flight stability, particularly with hard maneuvers. These findings support the integration of current deep reinforcement learning with adaptive consensus methods that can considerably improve UAV system formation stability, tracking accuracy, and control performance.

Another State of the Art approach can be found in [11], however there only the path of the leader is prescribed, and the followers have to follow it without collision.

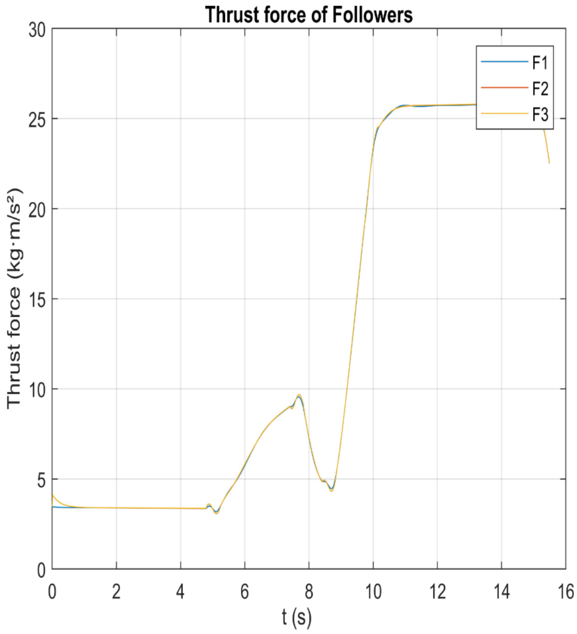

Fig. 12 Thrust forces of all followers of Cascade TD3

Future work will focus on transferring these simulation results to actual flight tests and exploring further optimizations in neural network architectures and consensus adaptation algorithms to robustly handle dynamic and uncertain conditions.

References

[1] Mohsan, S. A. H., Khan, M. A., Noor, F., Ullah, I., Alsharif, M. H. "Towards the unmanned aerial vehicles (UAVs): A comprehensive review", Drones 6(6), 147, 2022. https://doi.org/10.3390/drones6060147

[2] Thandar, A. M., Khaing, M. K. "Radial basis function (RBF) neural network classification based on consistency evaluation measure", International Journal of Computer Applications, 54(15), pp. 20–23, 2012. https://doi.org/10.5120/8642-2463

[3] Azar, A. T., Koubaa, A., Ali Mohamed, N., Ibrahim, H. A., Ibrahim, Z. F., Kazim, M., …, Casalino, G. "Drone deep reinforcement learning: A review", Electronics, 10(9), 999, 2021. https://doi.org/10.3390/electronics10090999

[4] Mosali, N. A., Shamsudin, S. S., Alfandi, O., Omar, R., Al-Fadhali, N. "Twin delayed deep deterministic policy gradient-based target tracking for unmanned aerial vehicle with achievement rewarding and multistage training", IEEE Access, 10, pp. 23545–23559, 2022. https://doi.org/10.1109/ACCESS.2022.3154388

[5] Li, K., Zhao, K., Song, Y. “Adaptive consensus of uncertain multi-agent systems with unified prescribed performance", IEEE/CAA Journal of Automatica Sinica, 11(5), pp. 1310–1312, 2024. https://doi.org/10.1109/JAS.2023.123723

[6] Blaauw, D. "Flight control system for a variable stability blended-wing-body unmanned aerial vehicle", MSc Thesis, University of Stellenbosch, 2009.

[7] Bodó, Z., Lantos, B. "Nonlinear control of maneuvering fixed wing UAVs using quaternion logarithm", In: 2020 23rd International Symposium on Measurement and Control in Robotics (ISMCR), Budapest, Hungary, 2020, pp. 1–6. ISBN 978-1-6654-0480-8 https://doi.org/10.1109/ISMCR51255.2020.9263760

[8] Parwana, H., Patrikar, J. S., Kothari, M. "A novel fully quaternion based nonlinear attitude and position controller", In: 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 2018, AIAA 2018-1587. https://doi.org/10.2514/6.2018-1587

[9] Al-sudany, H. N., Lantos, B. "Extended Linear Regression and Interior Point Optimization for Identification of Model Parameters of Fixed Wing UAVs", Acta Polytechnica Hungarica, 21(6), pp. 69–88, 2024. https://doi.org/10.12700/APH.21.6.2024.6.4

[10] Yu, Z., Li, J., Xu, Y., Zhang, Y., Jiang, B., Su, C.-Y. "Reinforcement learning-based fractional-order adaptive fault-tolerant formation control of networked fixed-wing UAVs with prescribed performance", IEEE Transactions on Neural Networks and Learning Systems, 35(3), pp. 3365–3379, 2024. https://doi.org/10.1109/TNNLS.2023.3281403

[11] Xu, D., Guo, Y., Yu, Z., Wang, Z., Lan, R., Zhao, R., Xie, X., Long, H. "PPO-Exp: Keeping fixed-wing UAV formation with deep reinforcement learning", Drones, 7(1), 28, 2023. https://doi.org/10.3390/drones7010028

[12] Kaelbling, L. P., Littman, M. L., Moore, A. W. "Reinforcement learning: A survey", Journal of Artificial Intelligence Research, 4, pp. 237–285, 1996. https://doi.org/10.1613/jair.301

[13] MathWorks "Cascadeforwardnet (R2025a)", [computer program] Available at: https://www.mathworks.com/help/deeplearning/ref/cascadeforwardnet.html [Accessed: 06 May 2025]

[14] Bai, S., Song, S., Liang, S., Wang, J., Li, B., Neretin, E. "UAV maneuvering decision-making algorithm based on twin delayed deep deterministic policy gradient algorithm", Journal of Artificial Intelligence and Technology, 2(1), pp. 16–22, 2022. https://doi.org/10.37965/jait.2021.12003

[15] Yu, Z., Zhang, Y., Jiang, B., Su, C.-Y. "Fault-Tolerant Cooperative Control of Unmanned Aerial Vehicles", Springer, 2024. ISBN 978-981-99-7663-8
https://doi.org/10.1007/978-981-99-7661-4

[16] Lantos, B., Márton, L. "Nonlinear Control of Vehicles and Robots", Springer, 2011. ISBN 978-1-84996-121-9
https://doi.org/10.1007/978-1-84996-122-6

Appendix

The developed SW, is based on MATLAB R2025a. The main functions used from MATLAB are train cascadeforwardnet [13] and some others used by the latter.

Beside them a high level novel own function was developed:

```
function [actor_net, critic_net] = TrainTD3 (input_data, otput_data)
which calls cascadeforwardnet, and train

actor_net = cascadeforwardnet (50,"trainbr");
actor_net.train Param.goal = 0;
actor_net.train Param.max_fail = 2000;
actor_net.train Param.epochs = 1000;
actor_net.train Param.show window = true;
actor_net.divideParam.train ratio = 0.80;
actor_net.divide Param.test ratio = 0.10;
actor_net.divide Param.val ratio = 0.10
[actor_net, actor_tr] = train (actor_net, input_data', output_data');

critic networks can be tuned in a similar way.
```

In on-line; applications the already trained TD3 can be used for attitude (orientation) control. The thrust-force control is based on nonlinear inversion and PID control technique.