

Practical Robust Formation Control for Nonlinear Multiagent Systems via Generative Adversarial Learning Framework: Theory and Experiment

Nuan Wen, Mir Feroskhan, *Member, IEEE*,

Abstract—Cyber attacks and disturbances greatly impair the performance of formation tasks in multiagent systems (MASs). To achieve robust formation control against these challenges, this article proposes a generative adversarial learning framework that is theoretically transparent and practically applicable. Rather than relying on an end-to-end deep neural networks (DNNs) architecture, our work leverage a double robust structure that combine the representation capabilities of DNNs with established, theoretically grounded linear control theory, ultimately achieving a practical, learning-based robust formation for MASs. Initially, generative adversarial networks (GANs) are used to linearize agent dynamics under false data injection (FDI) attacks and external disturbances. Subsequently, a proportional-integral (PI) protocol is employed to achieve overall robust formation. We present rigorous theoretical analyses of both stages, demonstrating the guaranteed convergence of GANs training and the closed-loop formation errors. Our approach is directly validated through a series of physical experiments involving multi-unmanned aerial vehicles (multi-UAVs), demonstrating robustness against attacks and disturbances during formation flights, without the sim-to-real gap commonly encountered in learning-based control frameworks.

Index Terms—Multiagent systems (MASs), generative adversarial networks (GANs), false data injection (FDI) attacks, disturbance.

I. INTRODUCTION

In recent years, formation execution in multi-agent systems (MASs) has garnered significant attention because of its wide-ranging applications in diverse engineering domains. These include coordinating ground vehicles for logistics [1], conducting reconnaissance and rescue missions with unmanned aerial vehicles (UAVs) [2], and enabling deep-space exploration via satellite formation flying [3]. However, ensuring desired formation performance in MASs poses two primary challenges [4], [5]. First, the nonlinear characteristics and unexpected disturbances in real-world environments complicate control development. Second, in practical scenarios, agents often interact only with their immediate neighbors, making the communication system vulnerable to continuous cyber attacks. Therefore, developing reliable, efficient, and trustworthy control methods to overcome these challenges in MAS formation task remains a substantial challenge.

Currently, numerous studies on cooperative MASs focus on group formation [6], formation maneuvering [7], event triggering [8], collision avoidance [9], and heterogeneous agents [10]. However, these studies often simplify agents as point masses modeled either by first-order linear integrators ($\dot{x}_i = u_i$) or second-order linear integrators ($\dot{x}_i = v_i$ and $\dot{v}_i = u_i$). This simplification overlooks the dynamic characteristics of the agents and disregards model uncertainties and external disturbances, which consequently degrades the formation's accuracy and stability. To address these limitations, many existing methods compensate for uncertainties and disturbances by employing observers [11], [12].

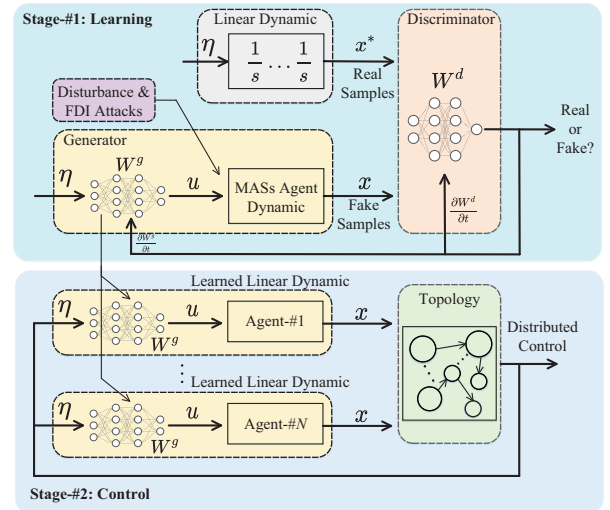


Fig. 1. Robust formation control for nonlinear MASs via generative adversarial framework. The proposed framework uses a two-stage approach instead of an end-to-end architecture. Stage-#1 applies GANs-based optimization to linearize the nonlinear dynamics of MASs while addressing system uncertainties. In Stage-#2, the generator from Stage-#1 provides input to the MAS agents, followed by classical linear control to achieve multi-agent formation.

Another essential concern in MASs is their susceptibility to malicious attacks, which primarily stem from two aspects [13]: breaching communication links and directly targeting individual agents. To mitigate these threats, efforts have been concentrated on developing fault-tolerant [14] and resilient control strategies [15]. Among the various types of attacks, one particularly damaging form is stochastic false data injection (FDI) [16], which disrupts MASs formations by injecting false data into measurements and control signals. FDI attacks on communication links pose a greater threat than sensor or

Manuscript received ****, *****; revised *****, ***. This Research is supported by the RIE2020/RIE2025 from MTC Young Individual Research Grants (YIRG) 2021 Grant Call (Grant No. M21K3c0121), administered by A*STAR. (Corresponding author: Mir Feroskhan.)

The authors are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798 (e-mail: nuan.wen@ntu.edu.sg; mir.feroskhan@ntu.edu.sg).

actuator faults, as the false data can spread across the network, leading to extensive damage. Consequently, MASs that rely on communication for formation control are particularly vulnerable to FDI attacks.

With advancements in data-driven approaches and deep neural networks, learning-based methods, especially reinforcement learning (RL), are becoming increasingly prevalent in the field of MASs formation task [17]. Two major directions have emerged for incorporating RL into the design of MASs formation control. The first approach employs traditional policy-iteration (PI) strategies, such as adaptive dynamic programming (ADP) [18], which offer stability but are limited by their use of single-layer neural networks. As a result, while this method updates controller parameters using neural networks, it remains within optimal control and does not fully utilize the representational capacity of neural networks. The second approach, utilizing deep reinforcement learning (DRL) [19], [20], begins by establishing a Markov Decision Process (MDP) followed by deep neural network training for continuous solution updates. This method faces challenges due to its lack of substantial theoretical backing, which hinders its practical application in engineering [21]. Additionally, applying DRL to physical MASs formation tasks requires extensive pre-training in virtual environments to reduce trial-and-error costs. However, simulations often fail to account for real-world complexities, resulting in controllers that struggle to generalize when deployed in real settings [22]. This sim-to-gap issue, combined with stability challenges during training, greatly limits the effectiveness of RL in MASs formation control, particularly in environments with disturbances and uncertainties [23], [24].

Our motivation, therefore, is to leverage learning-based methods, distinct from RL, to ensure robust formation in MASs under uncertainties caused by disturbances and FDI attacks. This motivation is driven by the potential of neural networks, especially their deep representation capabilities. Additionally, an essential aspect of practical learning-based methods that cannot be overlooked is optimizing data efficiency, reducing the need for extensive ground-truth data generation. Moreover, we seek to ensure theoretical transparency and provide convergence guarantees within this framework.

To address the performance challenges mentioned above, we turn to generative adversarial networks (GANs), which, in contrast to RL, have gained prominence as a semi-supervised learning model with proven convergence to Nash equilibrium [25]. GANs operate through the dynamic interaction of two independent neural networks: the generator (G) and the discriminator (D). The generator creates synthetic data, while the discriminator evaluates it against real data [26]. This adversarial interaction facilitates the creation of high-quality synthetic data. In essence, the networks (D and G) engage in a two-player minimax game with the following value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

where $p_z(z)$ denotes input prior variables and $p_{\text{data}}(x)$ repre-

sents the real data distribution.

In this article, as shown in Fig. 1, we move away from the end-to-end approach, where neural networks directly generate control signals for MASs formation control. Instead, we propose a more practical two-stage robust formation strategy. The first stage employs a generative adversarial learning framework, effectively utilizing the representational power of neural networks. This stage reshapes the nonlinear dynamics of MASs into a linear form while addressing uncertainties caused by external disturbances and FDI attacks. In the second stage, classical linear robust control methods are implemented to achieve guaranteed formation performance. This two-stage strategy offers two key advantages: it leverages neural networks to handle system uncertainties, and it seamlessly integrates classical control theory, enabling the direct application of well-established methods for closed-loop formation control. By combining learning-based techniques with traditional feedback control theory, this approach ensures efficiency and robustness. Furthermore, this learning-based MAS formation approach is highly flexible. After the first stage, where GANs linearize the dynamic of nonlinear agents, it can be easily extended to heterogeneous MASs formations [10], formations with switching topologies [27], and event-triggered formation scenarios [8]. Compared to existing literature, this article makes three significant contributions in terms of novelty and innovation:

- 1) **Data-efficient learning-based approach:** Although GANs are considered a semi-supervised learning method, they often rely on large amounts of ground-truth data to serve as real samples for the discriminator. In contrast to existing GANs-based control methods [28], [29], our framework avoids the inefficiency of pre-designing reference controllers. Instead, we utilize input-output data from linear dynamics as real samples, and through equivalent transformations, substitute these with matching input-output pairs. This strategy enhances data efficiency, providing a solution to the challenge of acquiring optimal decision-making data in learning-based methods.
- 2) **Theoretical stability and convergence:** Despite the theoretical potential of GANs for superior approximation through adversarial optimization, training challenges persist. The alternating optimization of two independent networks frequently results in saddle points or gradient explosions, leading to mode collapse and instability [30]. In contrast to original GANs [25], the adversarial loss function proposed in this study is theoretically assured to converge to a global optimum during optimization.
- 3) **Practical applicability:** While learning-based end-to-end approaches perform effectively in simulations, they often face challenges in real-world applications due to limited generalization and the presence of noise & disturbances. To address these limitations, our GANs learning framework employs a two-stage structure, integrating deep neural networks (DNNs) representation with traditional control theory's closed-loop mechanism. Each stage uses robust strategies to manage uncertainties in real-world MASs. Unlike many learning-based MASs formation control methods that are confined to simulations [9], [17], [23], we

introduce a workflow for real-world multi-UAVs formation, incorporating offline training, online adaptation, and final testing.

Notations: $\mathbb{R}^m$ represents the m -dimensional Euclidean space, and $\mathbb{R}^{m \times p}$ signifies the set of all $m \times p$ real matrices. The notation $\|\bullet\|$ is used for the Euclidean norm unless specified otherwise. Additionally, T indicates transpose, and $\langle \cdot \rangle$ represents the inner product of vectors. $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue of the matrix.

II. PRELIMINARIES AND PROBLEM STATEMENT

A. Basic Graph Theory

In graph theory, a directed network is commonly defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = 1, 2, \dots, N$ represents the nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ indicates the edges, and $\mathcal{A} = [a_{ij}] N \times N$ is the adjacency matrix, with $a_{ij} \geq 0$ for $i, j = 1, 2, \dots, N$. A directed edge $\mathcal{E}_{ij}$ connects nodes (i, j) , meaning that information flows from node i to node j , but not necessarily in the reverse direction. This edge exists only if $a_{ij} > 0$. The article focuses on directed networks without self-loops, so $(i, i) \notin \mathcal{E}$ for any $i \in \mathcal{V}$. The network $\mathcal{G}$ is considered strongly connected if there is a directed path between any two distinct nodes (i, j) . Furthermore, $\mathcal{G}$ is said to contain a spanning tree if one node, referred to as the root, has a directed path to all other nodes. The Laplacian matrix $\mathcal{L} = [\mathcal{L}_{ij}] \in \mathbb{R}^{N \times N}$ is defined by $\mathcal{L}_{ii} = \sum_{j=1}^N a_{ij}$ and $\mathcal{L}_{ij} = -a_{ij}$ for $i \neq j$. This guarantees that each row of $\mathcal{L}$ sums to zero, fulfilling the diffusion property.

B. Model Dynamics

Consider a dynamical MASs consisting of N agents, where the nonlinear dynamics of each agent are described as follows:

$$\dot{x}_i(t) = f(t, x_i) + g(t, x_i) u_i(t, x_i) + \hat{\varrho}_i(t) + w_i(t, x_i) \quad (2)$$

where $x_i(t) \in \mathbb{R}$ is the state. The functions $f(t, x_i)$ and $g(t, x_i)$ describe the true dynamics of the agent. The deviation caused by FDI attacks on the i -th agent is represented by $\hat{\varrho}_i(t)$. The variables $u_i(t)$ and $w_i(t, x_i)$ denote the control input and the total unknown uncertainty, which includes disturbances and model mismatches.

For the model expression $f(t, x_i)$, $g(t, x_i)$, uncertain term $w_i(t, x_i)$ and $\hat{\varrho}_i(t)$ in (2), the following assumptions hold:

Assumption 1: The nonlinear functions $f(\cdot)$ and $g(\cdot)$ are continuously differentiable and have locally Lipschitz derivatives, with the condition that $g(\cdot) > 0$. The unknown uncertainty w_i is constrained within a known compact set $W \subset \mathbb{R}$. Additionally, it holds that $\sup_{t \in [0, \infty)} |w_i| < \infty$ and $\sup_{t \in [0, \infty)} |\dot{w}_i| < \infty$.

Remark 1: Assumption 1 ensures that the nonlinearities in the model are Lipschitz continuous, a natural and widely accepted requirement in nonlinear control theory [31] and multi-agent systems [32]. This property guarantees the existence and uniqueness of the system's solution. The condition $g(\cdot) > 0$ further ensures system controllability [33]. The term $w_i(t, x_i)$ in (2) captures uncertainties such as bounded time-varying disturbances, bounded parameters perturbation, and nonlinear

terms with unknown parameters. The assumption that these uncertainties have finite energy is both practical and essential for ensuring the stability and boundedness of the closed loop system [34]. It is important to note that our GANs-based framework does not rely heavily on the true dynamics of the agent. Any unmodeled dynamics or unidentifiable components are incorporated within the uncertainty term $w_i(t, x_i)$. However, in practical implementation, we first perform system identification to estimate the model dynamics.

Assumption 2: This paper considers FDI attacks, modeled by

$$\hat{\varrho}_i(t) = \sum_{j \in N_i^*} \gamma_{ij}(t) \varrho_j(t) \quad (3)$$

where $\varrho_j(t) \in \mathbb{R}$ represents false data injected by an attacker, and $\gamma_{ij}(t)$ is a stochastic variable that follows a Bernoulli distribution:

$$\gamma_{ij}(t) = \begin{cases} 1, & \text{successful attack on communication} \\ & \text{channel } (j, i) \text{ at instant } t \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where the probability of a successful attack is described by

$$\begin{cases} \mathbb{P}\{\gamma_{ij}(t) = 1\} = \mathcal{P}_{ij} \\ \mathbb{P}\{\gamma_{ij}(t) = 0\} = 1 - \mathcal{P}_{ij} \end{cases} \quad (5)$$

Remark 2: Due to energy limitations, the attack signal $\varrho_j(t)$ must satisfy $\|\varrho\|^2 \leq \bar{\varrho}$, where $\varrho = [\varrho_1, \varrho_2, \dots, \varrho_N]^T$ and $\bar{\varrho}$ is a positive constant. A similar energy constraint is discussed in [35].

The directed topology among the agents in this study is assumed as follows:

Assumption 3: The directed network $\mathcal{G}$ contains a spanning tree, where the root node acts as the leader.

Remark 3: The Laplacian matrix of an undirected network $\mathcal{G}$ is symmetric, while in a directed network, it is usually asymmetric. Regardless of whether the network is directed or undirected, the Laplacian matrix remains semi-positive definite.

C. Control Objective with Key Research Challenges

We observe that the MASs described by (2) represent first-order nonlinear affine systems. Our objective is to develop a distributed control protocol $u_i(t) (\forall i \in N)$ that ensures the formation tracking error approaches zero, as shown in (6), despite the presence of unknown nonlinearities, uncertainties, and stochastic FDI in the agent dynamics.

$$\lim_{t \rightarrow \infty} (x_i(t) - x_j(t) - p_{ij}) = 0 \quad (6)$$

where $p_{ij} \in \mathbb{R}$ represents the desired distance between agents i and j .

The primary challenge in achieving distributed formation control lies in the system's inherently complex dynamics and significant model uncertainties. This study endeavors to surmount this challenge by leveraging a GANs framework to learn a linear dynamic system for each agent. To effectively implement this strategy, we must address several critical issues:

- 1) How can GANs be used to achieve robust formation control in MASs under uncertainties and attacks? Specifically, how should GANs be designed to effectively manage crucial aspects such as dataset composition, objective loss functions, and the structuring of the generator and discriminator?
- 2) Ensuring the stability and convergence of our proposed learning-based strategy is imperative. What measures can we implement to circumvent the model collapse or instability that typically arises during GANs training?
- 3) Additionally, based on theoretical proofs, what are the practical effects of this method?

The aforementioned questions will be explored in the subsequent sections.

III. ADVERSARIAL TRAINING WITH GANs FOR LINEARIZING DYNAMIC AGENTS UNDER FDI ATTACKS AND DISTURBANCES

This section presents a semi-supervised learning framework using GANs to linearize dynamic agents. The framework includes several essential elements: the neural network architecture for the controller, the dataset design, the definition of the adversarial loss function, and the gradient descent optimization for both the generator and discriminator.

A. DNN Parameterized Controller Architecture

For the nominal model of the agent:

$$\dot{x}_i(t) = f(t, x_i) + g(t, x_i) u_i(t) \quad (7)$$

If we design $u_i(t)$ in the following form:

$$u_i^*(x) = -\frac{f(t, x_i)}{g(t, x_i)} + \frac{1}{g(t, x_i)} \eta_i(t) \quad (8)$$

where $\eta_i(t) \in \mathbb{R}$ is the virtual input to the resulting approximately linearized system,

We can then generate a linearized input-output model:

$$\dot{x}_i(t) = \eta_i(t) \quad (9)$$

However, due to the presence of the stochastic term $\hat{\rho}_i(t)$ and the uncertainty term $w_i(t, x_i)$, (8) cannot completely eliminate the system's nonlinearities to form (9). To mitigate this challenge, it is essential to incorporate compensatory components into the feedback linearizing control, as detailed in (10):

$$u_i(t, x_i) = -\frac{f(t, x_i)}{g(t, x_i)} + \frac{1}{g(t, x_i)} \eta_i + \widetilde{\omega}_i(x_i) + \widehat{\omega}_i(x_i) \eta_i \quad (10)$$

Furthermore, to handle the continuous functions $\widetilde{\omega}_i(x_i)$ and $\widehat{\omega}_i(x_i)$ in (10), we have developed parameterized estimates as follows:

$$\begin{cases} \widetilde{\omega}_i(x_i) = \widetilde{\omega}_i(x_i; \widetilde{W}_i^g) \\ \widehat{\omega}_i(x_i) = \widehat{\omega}_i(x_i; \widehat{W}_i^g) \end{cases} \quad (11)$$

where $\widetilde{W}_i^g \in \mathbb{R}^{K_1}$ and $\widehat{W}_i^g \in \mathbb{R}^{K_2}$ represent parameters that are to be further refined through experimental research on the plant.

Collectively, for a given $(\widetilde{W}_i^g, \widehat{W}_i^g) \in W_i^g$, we formulate our controller estimate, designed to precisely linearize the plant, as follows:

$$u_i(x_i) = -\frac{f_i(x_i)}{g_i(x_i)} + \frac{1}{g_i(x_i)} \eta_i + \widetilde{\omega}_i(x_i; \widetilde{W}_i^g) + \widehat{\omega}_i(x_i; \widehat{W}_i^g) \eta_i \quad (12)$$

B. GANs-Based Optimization Framework

In the GANs-driven algorithm, it is essential to define real samples and fake samples and describe how they are generated.

Real samples are generated using the linear model in (9), which represents the input-output linear dependency. These samples are denoted as $\vartheta_i^* = [\eta_i, x_i^*]^T$. In practice, real samples are formulated as $\vartheta_i^* = [\eta_i, \dot{x}_i^* = \eta_i]$, where η_i is the virtual input, and $\dot{x}_i^*$ represents the linear output corresponding to η_i .

Fake samples, denoted as $\vartheta_i = [\eta_i, x_i]^T$, are generated through the combination of the DNN parameterized controller and the agent's nonlinear dynamics model. Specifically, the virtual input η_i is randomly sampled and integrated into (12), producing the control signal u_i . This signal is then applied to the nonlinear system (2), resulting in the actual output x_i . In practice, fake samples are expressed as $\vartheta_i = [\eta_i, \dot{x}_i]$. The generation of fake samples $\Upsilon = (\vartheta_1, \vartheta_2, \dots, \vartheta_M)$ in the generator relies on adjustable parameters $(\widetilde{W}_i^g, \widehat{W}_i^g) \in W_i^g$, as described in (11). The generator is designed to handle uncertainty and facilitate linear dynamics through the parameterized functions:

$$\begin{cases} \widetilde{\omega}_i(x_i; \widetilde{W}_i^g) = \langle \widetilde{W}_i^g, \widetilde{\sigma}_i(x_i) \rangle = \sum_{j=1}^{K_1} \widetilde{w}_{ij}^g \widetilde{\sigma}_{ij}(x_i) \\ \widehat{\omega}_i(x_i; \widehat{W}_i^g) = \langle \widehat{W}_i^g, \widehat{\sigma}_i(x_i) \rangle = \sum_{j=1}^{K_2} \widehat{w}_{ij}^g \widehat{\sigma}_{ij}(x_i) \end{cases} \quad (13)$$

where $\widetilde{w}_{ij}^g$ and $\widehat{w}_{ij}^g$ are the learning parameters with K_1 and K_2 denoting their respective dimensionalities. $\widetilde{\sigma}_{ij}$ and $\widehat{\sigma}_{ij}$ represent specific activation functions.

As illustrated in Fig.1, real samples are generated from the linearized input-output model (9), while fake samples are produced by the interaction of the DNN parameterized controller (10), (13) with the agent's nonlinear dynamics model (2). During GAN optimization, the generator continuously updates to produce fake samples, and the discriminator distinguishes between real and fake samples.

The original GANs [25] often suffer from mode collapse and can get stuck at saddle points. To overcome this, we refine the loss function into Relativistic Pairing GANs (RpGANs) [36], as outlined in (14). Here, $\mathcal{F}$ represents the generative function, and $\mathcal{H}$ serves as the discriminative function.

$$\min_{\vartheta} \max_{\mathcal{F}} \mathcal{H}(\mathcal{F}(\vartheta^*) - \mathcal{F}(\vartheta)) \quad (14)$$

In this study, (14) must be translated into a specific empirical loss form. To begin, the existence of M distinct real data points for agent- i is assumed as $\vartheta_{i1}^*, \vartheta_{i2}^* \dots \vartheta_{iM}^*$. Simultaneously, the corresponding fake data points are represented by $\Upsilon_i = (\vartheta_{i1}, \vartheta_{i2}, \dots, \vartheta_{iM})$. In terms of the relativistic pairing, $\mathcal{F}(\cdot)$ is defined as a linear function, specifically $\mathcal{F}(t) = W_d^T t$, where W_d represents the learning parameters of

the discriminator. Additionally, $\mathcal{H}$ is defined as a least square loss function, expressed as $\mathcal{H}(t) = (t - \varepsilon)^2$, with ε being the margin factor. To summarize, the formulated minimax objective function is presented as (15).

$$\Upsilon_i = \min_{(\vartheta_{i1}, \vartheta_{i2}, \dots, \vartheta_{iM})} \max_{W_i^d} \sum_{k=1}^M \left((W_i^d)^T (\vartheta_{ik}^* - \vartheta_{ik}) - \varepsilon_i \right)^2 \quad (15)$$

The next step involves addressing the minimax objective function (15). It is decomposed to derive the gradient update formulas for both the generator and the discriminator.

The generator creates new data instances $\Upsilon_i = (\vartheta_{i1}, \vartheta_{i2}, \dots, \vartheta_{iM})$, with the objective of emulating the real data distribution $\Upsilon_i^* = (\vartheta_{i1}^*, \vartheta_{i2}^*, \dots, \vartheta_{iM}^*)$. The associated loss function for the generator are shown in (16), denoted as $L_i^g(W_i^g; \Upsilon_i)$, aims to be minimized.

$$\begin{aligned} \Upsilon_i &= \min_{(\vartheta_{i1}, \vartheta_{i2}, \dots, \vartheta_{iM})} L_i^g(W_i^d; \Upsilon_i) \\ &\triangleq \frac{1}{M} \sum_{i=1}^M \left((W_i^d)^T (\vartheta_{ik}^* - \vartheta_{ik}) - \varepsilon_i \right)^2 \end{aligned} \quad (16)$$

Our objective is to optimize the discriminator, enhancing its proficiency in distinguishing whether an input pair is derived from the real samples or the generator. To achieve this, we focus on minimizing the least square loss between the real data set $\Upsilon_i^* = (\vartheta_{i1}^*, \vartheta_{i2}^*, \dots, \vartheta_{iM}^*)$ and the generated data set $\Upsilon_i = (\vartheta_{i1}, \vartheta_{i2}, \dots, \vartheta_{iM})$ by fine-tuning W_i^d . This optimization process culminates in the formulation of the discriminator's loss function, as expressed in (17) :

$$\begin{aligned} \min_{W_i^d \in \mathbb{R}^d} L_i^d(W_i^d; \Upsilon_i) \\ \triangleq \frac{1}{M} \sum_{k=1}^M \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \varepsilon_i \right)^2 \end{aligned} \quad (17)$$

To conclude, as shown in (18), the training process involves alternately minimizing the loss functions of the generator and discriminator to obtain their optimal learned parameters, denoted as W_i^{g*} and W_i^{d*} , respectively. Specifically, W_i^{g*} functions as the parameters for the learned linearization controller.

$$\begin{cases} W_i^g \leftarrow W_i^g - \alpha \nabla_{W_i^g} L_i^g(W_i^g; \Upsilon_i) \\ W_i^d \leftarrow W_i^d - \beta \nabla_{W_i^d} L_i^d(W_i^d; \Upsilon_i) \end{cases} \quad (18)$$

where α and β represent the learning rates.

C. Stability Analysis

To analyze the stability and convergence of the proposed GAN-based linearization under FDI attacks and disturbances, two additional assumptions must be established:

Assumption 4: Let $\widehat{W}_i^g \in \mathbb{R}^{K_1}$ and $\widehat{W}_i^d \in \mathbb{R}^{K_2}$ represent the generator parameters to be trained. We assume that both $\widehat{W}_i^g$ and $\widehat{W}_i^d$ are convex sets. Consequently, the product set $\widehat{W}_i^g \times \widehat{W}_i^d := \widehat{W}_i^g$ is also convex.

Remark 4: The convexity assumption in Assumption 4 is based on convex optimization principles, where convex feasible regions facilitate global convergence and simplify analysis [37]. This widely accepted assumption supports deriving

global solutions in optimization problems. In practice, ℓ_2 -regularization enforces convexity by constraining parameters within an ℓ_2 -norm ball [38], ensuring a bounded parameter space that stabilizes training and reduces overfitting.

Assumption 5: There exists the optimal discriminator W_i^{d*} , such that

$$c_i \triangleq \left(W_i^{d*} \right)^T \sum_{k=1}^M (\vartheta_{ik} - \vartheta_{ik}^*) \geq 0 \quad (19)$$

The optimal W_i^{d*} is regarded as the discriminator weight vector that, under ideal conditions, maximally separates real data from fake data.

Given the assumptions and parameters outlined earlier, we are now prepared to present our theoretical results.

Theorem-1 (Convex Optimality): Under Assumption 4 holds, for the controller synthesis defined in (12) and taking the form of (13), let the combination of $[\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}]$ and $[\hat{\sigma}_{i1}, \hat{\sigma}_{i2}, \dots, \hat{\sigma}_{iK_2}]$ in (13) be linearly independent. Then, a unique global minimizer is guaranteed to exist for the linearized dynamic loss.

Proof:

Substituting (12) into (2) results in the following outcome:

$$\begin{aligned} \dot{x}_i(t) &= \eta_i + \varpi_i(x_i) + \hat{\vartheta}_i(t) \\ &+ g_i(x_i) \left[\sum_{j=1}^{K_1} \tilde{w}_{ij}^g \tilde{\sigma}_{ij}(x_i) + \sum_{j=1}^{K_2} \hat{w}_{ij}^g \eta_i \hat{\sigma}_{ij}(x_i) \right] \end{aligned} \quad (20)$$

To extract the segments influenced by W_i^g , we can refine and simplify the expression in the following manner:

$$\dot{x}_i(t) = \Xi_i(x_i, \eta_i, \varpi_i) + \langle W_i^g, \hat{\Xi}_i(x_i, \eta_i) \rangle \quad (21)$$

where $\Xi_i(x_i, \eta_i, \varpi_i)$ is defined as $\Xi_i(x_i, \eta_i) = \eta_i + \varpi_i(x_i) + \hat{\vartheta}_i(t)$ and

$$\hat{\Xi}_i(x_i, \eta_i) = g(t, x_i) [\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}, \hat{\sigma}_{i1}\eta_i, \hat{\sigma}_{i2}\eta_i, \dots, \hat{\sigma}_{iK_2}\eta_i]$$

The linearized dynamic loss is characterized as:

$$\begin{aligned} L_i(x_i, \eta_i, W_i^g) &= \|\eta_i - \dot{x}_i(t)\|^2 \\ &= \left\| \eta_i - f - g \cdot u_i(W_i^g) - \varpi_i - \hat{\vartheta}_i \right\|^2 \end{aligned} \quad (22)$$

In (22), it provides an evaluation of the efficacy with which the learned controller $u_i(W_i^g)$ linearizes the plant at the state x_i , in the presence of the external disturbance ϖ_i and FDI attacks $\hat{\vartheta}_i$.

By defining $\Xi_i \triangleq \eta_i - \Xi_i(x_i, \eta_i, \varpi_i)$ and employing (21), (22) can be reformulated as:

$$\begin{aligned} L_i(W_i^g) &= \left\| \Xi_i - \langle W_i^g, \hat{\Xi}_i \rangle \right\|^2 \\ &= (\Xi_i - \langle W_i^g, \hat{\Xi}_i \rangle)^T (\Xi_i - \langle W_i^g, \hat{\Xi}_i \rangle) \\ &= W_i^g \hat{\Xi}_i^T \hat{\Xi}_i (W_i^g)^T - 2W_i^g \hat{\Xi}_i^T \Xi_i + \Xi_i^T \Xi_i \end{aligned} \quad (23)$$

Calculating the second derivative of (23) results in the Hessian matrix being derived as (24).

$$\frac{\partial^2}{\partial (W_i^g)^2} L(W_i^g) = \hat{\Xi}_i^T \hat{\Xi}_i \quad (24)$$

where $\widehat{\Xi}_i$ is shown in (21). In what follows, we employ a proof by contradiction to establish that the linear independence of the set $[\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}, \hat{\sigma}_{i1}, \hat{\sigma}_{i2}, \dots, \hat{\sigma}_{iK_2}]$ necessitates the linear independence of the extended set $[\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}, \hat{\sigma}_{i1}\eta_i, \hat{\sigma}_{i2}\eta_i, \dots, \hat{\sigma}_{iK_2}\eta_i]$.

Considering the confirmed linear independence of $[\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}, \hat{\sigma}_{i1}, \hat{\sigma}_{i2}, \dots, \hat{\sigma}_{iK_2}]$, let us, for the sake of contradiction, assume that the extended set $[\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}, \hat{\sigma}_{i1}\eta_i, \hat{\sigma}_{i2}\eta_i, \dots, \hat{\sigma}_{iK_2}\eta_i]$ is also linearly dependent. This presupposes the existence of coefficients $\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_{K_1}$ and $\hat{\alpha}_1\eta_i, \hat{\alpha}_2\eta_i, \dots, \hat{\alpha}_{K_2}\eta_i$, not all zero, satisfying (25).

$$\begin{aligned} & \tilde{\alpha}_1\tilde{\sigma}_{i1} + \tilde{\alpha}_2\tilde{\sigma}_{i2} + \dots + \tilde{\alpha}_{K_1}\tilde{\sigma}_{iK_1} \\ & + (\hat{\alpha}_1\eta_i)\hat{\sigma}_{i1} + (\hat{\alpha}_2\eta_i)\hat{\sigma}_{i2} + \dots + (\hat{\alpha}_{K_2}\eta_i)\hat{\sigma}_{iK_2} = 0 \end{aligned} \quad (25)$$

However, this would imply that $[\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}, \hat{\sigma}_{i1}, \hat{\sigma}_{i2}, \dots, \hat{\sigma}_{iK_2}]$ are linearly dependent, which contradicts our initial premise. Therefore, the extended set $[\tilde{\sigma}_{i1}, \tilde{\sigma}_{i2}, \dots, \tilde{\sigma}_{iK_1}, \hat{\sigma}_{i1}\eta_i, \hat{\sigma}_{i2}\eta_i, \dots, \hat{\sigma}_{iK_2}\eta_i]$ must indeed be linearly independent. This conclusion affirms that $\widehat{\Xi}_i^T \widehat{\Xi}_i$ is positive definite rather than semi-positive definite, thus ensuring that $L_i(W_i^g)$ is strictly convex, given that W_i^g is a convex set. This affirms the existence of a unique global minimizer for the linearized dynamic system as specified in (22). The proof of Theorem 1 is hereby completed. ■

In Theorem 1, we have proven that the conditions of linear independence on $\tilde{\sigma}_i$ and $\hat{\sigma}_i$ in (13) guarantee that the linearization loss $\|x_i^n - \eta_i\|$ is approached as a strictly convex optimization problem. Moving forward to Theorem 2, we focus on demonstrating the convergence of GANs dynamics, thereby proving the uniformly ultimately boundedness of sample error.

Theorem-2 (Convergent Dynamics): Considering real samples ϑ_{ik}^* and fake samples ϑ_{ik} generated by controller (12) and plant (2), and assuming Assumption 5 is valid, it follows that a certain threshold $\varepsilon_i^* > \|W_i^d\| \cdot \max\{\|\vartheta_{ik} - \vartheta_{ik}^*\|\}$ exists. For any chosen $\varepsilon_i \in (\varepsilon_i^*, +\infty)$, the convergence error, defined as $\tilde{\vartheta}_{ik} \triangleq \|\vartheta_{ik} - \vartheta_{ik}^*\|$, will be uniformly ultimately bounded.

Proof:

Taking into account the generator loss and discriminator loss provided in (16) and (17), we illustrate the evolution of the parameters $(W_i^d; \Upsilon_i)$ at the k -th iteration. This is achieved through simultaneous gradient descent, which can be represented as follows:

$$\begin{cases} W_i^d(j+1) = W_i^d(j) - \frac{\partial L_i^d(W_i^d; \Upsilon_i)}{\partial W_i^d} \Big|_{(W_i^d=W_i^d(j), \Upsilon_i=\Upsilon_i(j))} \alpha \Delta t \\ \vartheta_{ik}(j+1) = \vartheta_{ik}(j) - \frac{\partial L_i^g(W_i^d; \Upsilon_i)}{\partial \vartheta_{ik}} \Big|_{(W_i^d=W_i^d(j), \Upsilon_i=\Upsilon_i(j))} \beta \Delta t \end{cases} \quad (26)$$

where $\alpha\Delta t$ and $\beta\Delta t$ signify the learning rates for the discrete case scenario.

We investigate the behavior of arbitrary loss pairs under the influence of continuous dynamics governed by gradient

descent. In this scenario, both $W_i^d(t)$ and $\vartheta_{ik}(t)$ are functions explicitly dependent on continuous time. This leads to the expressions $W_i^d(j+1) = W_i^d(t_0 + j\Delta t)$ and $\vartheta_{ik}(j+1) = \vartheta_{ik}(t_0 + j\Delta t)$. As $\Delta t \rightarrow 0$, the process evolves into a dynamical system. This system is defined by the following ordinary differential equations, commonly referred to as infinitesimal gradient descent [39]:

$$\begin{aligned} \frac{dW_i^d}{dt} &= -\alpha_i \frac{\partial L_i^d(W_i^d; \Upsilon_i)}{\partial W_i^d} \\ \frac{d\vartheta_{ik}}{dt} &= -\beta_i \frac{\partial L_i^g(W_i^d; \Upsilon_i)}{\partial \vartheta_{ik}} \end{aligned} \quad (27)$$

Under the assumption that $\alpha_i = \beta_i = 1$, this leads to the following result:

$$\begin{cases} \frac{dW_i^d}{dt} = -\frac{\partial L_i^d(W_i^d; \Upsilon_i)}{\partial W_i^d} \\ \quad = \frac{2}{M} \sum_{k=1}^M (\vartheta_{ik}^* - \vartheta_{ik}) \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \varepsilon_i \right) \\ \frac{d\vartheta_{ik}}{dt} = -\frac{\partial L_i^g(W_i^d; \Upsilon_i)}{\partial \vartheta_{ik}} \\ \quad = \frac{2}{M} W_i^d \left((W_i^d)^T (\vartheta_{ik}^* - \vartheta_{ik}) - \varepsilon_i \right) \end{cases} \quad (28)$$

Let V_i be a proposed Lyapunov function candidate, characterized by:

$$V_i(W_i^d, \Upsilon_i) = \frac{1}{2} \sum_{k=1}^M \|\vartheta_{ik}^* - \vartheta_{ik}\|^2 + \frac{1}{2} \|W_i^d - W_i^{d*}\|^2 \quad (29)$$

From (29), it is derived that the derivative of V satisfies the following condition (30).

It is evident that $\dot{V}_i^1 \leq 0$ holds true. Therefore, in the subsequent analysis, our focus will be primarily on examining $\dot{V}_i^2$ in (31).

Given the precondition that $c_i \geq 0$ and $\varepsilon_i > \|W_i^d\| \cdot \max\{\|\vartheta_{ik} - \vartheta_{ik}^*\|\}$, it follows that $\dot{V}_i^2 \leq 0$. Consequently, this implies $\dot{V}_i \leq 0$. Furthermore, the convergence errors $\tilde{\vartheta}_{ik}$, trained through the GANs dynamics, are uniformly ultimately bounded. This conclusion effectively finalizes the proof of Theorem 2. ■

Remark 5: Early GANs [25] often suffer from mode collapse, where the generator repeatedly produces nearly identical outputs [36]. This occurs due to convergence to local Nash equilibria, making them susceptible to local minima and saddle points. To address these issues, Theorems 1 and 2 ensure global convergence and stability in our GAN design. Two key strategies are employed: 1) Structuring the generator's target function as strictly convex within the DNNs, ensuring a unique global minimum and stable convergence; and 2) Introducing adversarial loss from RpGANs to enforce global convergence. This approach ensures the discriminator W_i^d

achieves its optimal value W_i^{d*} , aligning generated data with real samples, as validated by the Lyapunov function (29).

$$\begin{aligned}
\dot{V}_i &= \sum_{k=1}^M (\vartheta_{ik} - \vartheta_{ik}^*)^T \frac{d\vartheta_{ik}}{dt} + \left(W_i^d - W_i^{*d} \right)^T \frac{dW_i^d}{dt} \\
&= \sum_{k=1}^M (\vartheta_{ik} - \vartheta_{ik}^*)^T \frac{2}{n} W_i^d \left((W_i^d)^T (\vartheta_{ik}^* - \vartheta_{ik}) - \varepsilon_i \right) \\
&\quad + \left(W_i^d - W_i^{*d} \right)^T \frac{2}{n} \sum_{i=1}^n (\vartheta_{ik}^* - \vartheta_{ik}) \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \varepsilon_i \right) \\
&= \frac{2}{M} \sum_{k=1}^M (\vartheta_{ik} - \vartheta_{ik}^*)^T W_i^d \left((W_i^d)^T (\vartheta_{ik}^* - \vartheta_{ik}) - \varepsilon_i \right) \\
&\quad + \frac{2}{M} \sum_{k=1}^M (W_i^d)^T (\vartheta_{ik}^* - \vartheta_{ik}) \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \varepsilon_i \right) \\
&\quad - \frac{2}{M} \sum_{k=1}^M \left(W_i^{*d} \right)^T (\vartheta_{ik}^* - \vartheta_{ik}) \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \varepsilon_i \right) \\
&= - \underbrace{\frac{4}{M} \sum_{k=1}^M \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) \right)^2}_{\dot{V}_i^1} \\
&\quad + \underbrace{\frac{2}{M} \sum_{i=1}^M \left(W_i^{*d} \right)^T (\vartheta_{ik}^* - \vartheta_{ik}) \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \varepsilon_i \right)}_{\dot{V}_i^2}
\end{aligned} \tag{30}$$

$$\begin{aligned}
\dot{V}_i^2 &= \sum_{k=1}^M \left(W_i^{*d} \right)^T (\vartheta_{ik} - \vartheta_{ik}^*) \left((W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \varepsilon_i \right) \\
&= \sum_{k=1}^M \left(W_i^{*d} \right)^T (\vartheta_{ik} - \vartheta_{ik}^*) (W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) - \left(W_i^{*d} \right)^T (\vartheta_{ik} - \vartheta_{ik}^*) \varepsilon_i \\
&\leq \sum_{k=1}^M \left\| \left(W_i^{*d} \right)^T (\vartheta_{ik} - \vartheta_{ik}^*) (W_i^d)^T (\vartheta_{ik} - \vartheta_{ik}^*) \right\| - \left(W_i^{*d} \right)^T (\vartheta_{ik} - \vartheta_{ik}^*) \varepsilon_i \\
&\leq \sum_{k=1}^M \left\| W_i^{*d} \right\| \|W_i^d\| \|(\vartheta_{ik} - \vartheta_{ik}^*)\|^2 - \left(W_i^{*d} \right)^T (\vartheta_{ik} - \vartheta_{ik}^*) \varepsilon_i \\
&= \sum_{k=1}^M \left\| W_i^{*d} \right\| \|W_i^d\| \|(\vartheta_{ik} - \vartheta_{ik}^*)\|^2 - \varepsilon_i \left(W_i^{*d} \right)^T \sum_{i=1}^n (\vartheta_{ik} - \vartheta_{ik}^*) \\
&= \sum_{k=1}^M \left\| W_i^{*d} \right\| \|W_i^d\| \|(\vartheta_{ik} - \vartheta_{ik}^*)\|^2 - \varepsilon_i c_i
\end{aligned} \tag{31}$$

IV. DESIGN OF DISTRIBUTED PI-BASED FORMATION PROTOCOL FOR LINEARIZED AGENT MODELS

A. Formation Control Laws

Through adversarial optimization with GANs, an equivalent linearized system (32) is obtained, where $\delta_i(t)$ represents the small approximation error in the linearization.

$$\dot{x}_i(t) = \eta_i(t) + \delta_i(t) \tag{32}$$

Since the leader in the directed network $\mathcal{G}$ does not receive information from its neighbors, the corresponding Laplacian

matrix $\mathcal{L}$ can be partitioned as follows:

$$\mathcal{L} = \begin{bmatrix} 0 & \Phi_{N-1}^T \\ \mathbb{1}_{N-1} & \hat{\mathcal{L}} \end{bmatrix} \tag{33}$$

of which $\hat{\mathcal{L}} \in \mathbb{R}^{(N-1) \times (N-1)}$.

To achieve formation control for dynamical MASs (32) in directed networks, this article proposes the design of PI protocols as follows:

$$\eta_i = \sum_{j=1}^N a_{ij} \left(\mathcal{K}_{\mathcal{P}} e_{ij}(t) + \mathcal{K}_{\mathcal{I}} \int_0^t e_{ij}(\tau) d\tau \right) \tag{34}$$

where $e_{ij}(t) = x_i(t) - x_j(t) - p_{ij}$, and a_{ij} represents the elements of the adjacency matrix $\mathcal{A}$.

Assuming $i = 1$ as the leader, then the formation error can be expressed as follows:

$$\hat{e}_i(t) = x_i(t) - x_1(t) - p_{i1}, \quad i = 2, \dots, N \quad (35)$$

where $p_{i1} \in \mathbb{R}$ is a constant, which means the desired distance between the leader and i th follower. It is hoped that $\hat{e}_i(t)$ will converge to a small neighborhood of zero. Based on (32) and assuming $\eta_1 = 0$, $\delta_1 = 0$ for the virtual leader input, the error dynamics model of formation system can be expressed as follows:

$$\begin{aligned} \dot{\hat{e}}_i &= \dot{x}_i - \dot{x}_1 \\ &= \delta_i(t) + \sum_{j=1}^N a_{ij} \left(\mathcal{K}_{\mathcal{P}} e_{ij}(t) + \mathcal{K}_{\mathcal{I}} \int_0^t e_{ij}(\tau) d\tau \right) \end{aligned} \quad (36)$$

According to the definition of the Laplace matrix, (36) can be rewritten as (37).

$$\dot{\hat{e}}_i = \delta_i(t) - \sum_{j=1}^N \hat{\mathcal{L}}_{ij} \left(\mathcal{K}_{\mathcal{P}} \hat{e}_i + \mathcal{K}_{\mathcal{I}} \int_0^t \hat{e}_i(\tau) d\tau \right) \quad (37)$$

Rewriting the error dynamics model in a compact vector form:

$$\dot{\hat{\mathcal{E}}}(t) = \Delta(t) - \left(\mathcal{K}_{\mathcal{P}} \hat{\mathcal{L}} \hat{\mathcal{E}}(t) + \mathcal{K}_{\mathcal{I}} \int_0^t \hat{\mathcal{L}} \hat{\mathcal{E}}(\tau) d\tau \right) \quad (38)$$

where $\hat{\mathcal{E}}(t) = [\hat{e}_2(t), \hat{e}_3(t), \hat{e}_4(t), \dots, \hat{e}_N(t)]^T$ and $\Delta(t) = [\delta_2(t), \delta_3(t), \delta_4(t), \dots, \delta_N(t)]^T$.

Let $\hat{\Sigma}(t) = \int_0^t \hat{\mathcal{E}}(\tau) d\tau$, one can rewrite:

$$\dot{\hat{\mathcal{E}}}(t) = \Delta(t) - \mathcal{K}_{\mathcal{P}} \hat{\mathcal{L}} \hat{\mathcal{E}}(t) - \mathcal{K}_{\mathcal{I}} \hat{\mathcal{L}} \hat{\Sigma}(t) \quad (39)$$

The matrix form reveals that:

$$\begin{bmatrix} \dot{\hat{\mathcal{E}}}(t) \\ \dot{\hat{\Sigma}}(t) \end{bmatrix} = \begin{bmatrix} -\mathcal{K}_{\mathcal{P}} \hat{\mathcal{L}} & -\mathcal{K}_{\mathcal{I}} \hat{\mathcal{L}} \\ I_{N-1} & 0_{N-1} \end{bmatrix} \begin{bmatrix} \hat{\mathcal{E}}(t) \\ \hat{\Sigma}(t) \end{bmatrix} + \begin{bmatrix} \Delta(t) \\ 0 \end{bmatrix} \quad (40)$$

B. Stability Analysis

The following lemmas and definition are introduced to facilitate the stability analysis.

Lemma 1 [40]: Assuming Assumption 3 holds, a positive vector $\psi = (\psi_1, \psi_2, \dots, \psi_{N-1})^T \in \mathbb{R}^{N-1}$ exists such that $\hat{\mathcal{L}}\psi = \mathbb{1}_{N-1}$. Moreover,

$$\Psi \hat{\mathcal{L}} + \hat{\mathcal{L}}^T \Psi \succ 0 \quad (41)$$

where $\Psi = \text{diag}\{\psi_1, \psi_2, \dots, \psi_{N-1}\}$ and $\hat{\mathcal{L}}$ is defined in (33).

Lemma 2 [41]: Let $\mathcal{L}$ denote the Laplacian matrix of a directed, strongly connected network $\mathcal{G}$. The vectors $\mathbb{1}_N = (1, 1, \dots, 1)^T$ and $\psi = (\psi_1, \psi_2, \dots, \psi_N)^T$ represent the right and left eigenvectors associated with the eigenvalue 0 of $\mathcal{L}$, satisfying $\mathcal{L}\mathbb{1}_N = 0$ and $\psi^T \mathcal{L} = 0$. Furthermore, a positive definite diagonal matrix $\Psi = \text{diag}\{\psi_1, \psi_2, \dots, \psi_N\}$ exists such that the matrix $\tilde{\mathcal{L}} = \frac{1}{2}(\Psi \mathcal{L} + \mathcal{L}^T \Psi)$ is symmetric, with both row and column sums equal to zero.

Definition 1 [42]: Consider $\mathcal{L}$ as the Laplacian matrix of a directed, strongly connected network $\mathcal{G}$. The general algebraic connectivity is given by:

$$a_{\psi}(\mathcal{L}) = \min_{x^T \theta=0, x \neq 0} \frac{x^T \tilde{\mathcal{L}} x}{x^T \Psi x}, \quad (42)$$

where $\tilde{\mathcal{L}} = \frac{1}{2}(\Psi \mathcal{L} + \mathcal{L}^T \Psi)$, $\Psi = \text{diag}(\psi_1, \psi_2, \dots, \psi_N)$, and $\psi = (\psi_1, \psi_2, \dots, \psi_N)^T$ with $\psi_i > 0$ and $\mathbb{1}_N^T \psi = 1$. It follows that $a_{\psi}(\mathcal{L}) > 0$.

With the above lemmas and definitions established, we are now ready to present the key findings of our MAS formation control study.

Theorem-3 (Stability): Consider the dynamical MASs (32) with distributed control protocols (34) in directed networks. Then, the formation tracking error (35) will converge to a small neighborhood of zero by designing the following parameters:

$$\begin{cases} \mathcal{K}_{\mathcal{P}} > 0 \\ 0 < \mathcal{K}_{\mathcal{I}} < \min \{ \mathcal{K}_{\mathcal{P}}^2 \lambda_{\min}(\hat{\mathcal{L}}), 2\mathcal{K}_{\mathcal{P}}^2 a_{\psi}(\hat{\mathcal{L}}) \} \end{cases} \quad (43)$$

Proof:

Choose a Lyapunov function candidate as:

$$\mathcal{V}(t) = \frac{1}{2} \begin{bmatrix} \hat{\mathcal{E}}(t) \\ \hat{\Sigma}(t) \end{bmatrix}^T \Omega \begin{bmatrix} \hat{\mathcal{E}}(t) \\ \hat{\Sigma}(t) \end{bmatrix} \quad (44)$$

where $\Omega = \begin{bmatrix} \Psi & \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi \\ \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi & 2\mathcal{K}_{\mathcal{I}} \tilde{\mathcal{L}} \end{bmatrix}$ and $\tilde{\mathcal{L}} = \frac{1}{2}(\Psi \mathcal{L} + \mathcal{L}^T \Psi)$.

First, we need to prove that $\mathcal{V}(t) \geq 0$ and $\mathcal{V}(t) = 0$ if and only if $\hat{\mathcal{E}}(t) = 0$ and $\hat{\Sigma}(t) = 0$, which demonstrates the positive definiteness of Ω .

The Lyapunov function (44) can be rewritten as:

$$\begin{aligned} \mathcal{V}(t) &= \frac{1}{2} \hat{\mathcal{E}}^T \Psi \hat{\mathcal{E}} + \frac{1}{2} \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \hat{\Sigma}^T \Psi \hat{\mathcal{E}} + \frac{1}{2} \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \hat{\mathcal{E}}^T \Psi \hat{\Sigma} + \hat{\Sigma}^T \mathcal{K}_{\mathcal{I}} \tilde{\mathcal{L}} \hat{\Sigma} \\ &= \frac{1}{2} \hat{\mathcal{E}}^T \Psi \hat{\mathcal{E}} + \frac{1}{2} \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \hat{\Sigma}^T \Psi \hat{\mathcal{E}} + \frac{1}{2} \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \hat{\mathcal{E}}^T \Psi \hat{\Sigma} \\ &\quad + \frac{1}{2} \hat{\Sigma}^T \mathcal{K}_{\mathcal{I}} (\Psi \mathcal{L} + \mathcal{L}^T \Psi) \hat{\Sigma} \end{aligned} \quad (45)$$

Given $a_{\psi}(\mathcal{L})$ in Definition 1, we can derive:

$$\begin{aligned} \mathcal{V}(t) &\geq \frac{1}{2} \hat{\mathcal{E}}^T \Psi \hat{\mathcal{E}} + \frac{1}{2} \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \hat{\Sigma}^T \Psi \hat{\mathcal{E}} + \frac{1}{2} \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \hat{\mathcal{E}}^T \Psi \hat{\Sigma} \\ &\quad + \mathcal{K}_{\mathcal{I}} a_{\psi}(\hat{\mathcal{L}}) \hat{\Sigma}^T \Psi \hat{\Sigma} \\ &= \frac{1}{2} \begin{bmatrix} \hat{\mathcal{E}}(t) \\ \hat{\Sigma}(t) \end{bmatrix}^T \begin{bmatrix} \hat{\Omega}_{11} & \hat{\Omega}_{12} \\ \hat{\Omega}_{21} & \hat{\Omega}_{22} \end{bmatrix} \begin{bmatrix} \hat{\mathcal{E}}(t) \\ \hat{\Sigma}(t) \end{bmatrix} \end{aligned} \quad (46)$$

where $\hat{\Omega}_{11} = \Psi$, $\hat{\Omega}_{12} = \hat{\Omega}_{21} = \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi$ and $\hat{\Omega}_{22} = 2\mathcal{K}_{\mathcal{I}} a_{\psi}(\hat{\mathcal{L}}) \Psi$.

We know from Lemma 1 that $\hat{\Omega}_{11} > 0$, and from (42), it follows that $\hat{\Omega}_{11} - \hat{\Omega}_{12} \hat{\Omega}_{22}^{-1} \hat{\Omega}_{12}^T > 0$. By the Schur complement [43], one can derive

$$\begin{bmatrix} \hat{\Omega}_{11} & \hat{\Omega}_{12} \\ \hat{\Omega}_{21} & \hat{\Omega}_{22} \end{bmatrix} > 0 \quad (47)$$

Therefore, the defined Lyapunov function (44) is positive definite.

It follows from (39) that the derivative of $\mathcal{V}(t)$ satisfies

$$\begin{aligned} \dot{\mathcal{V}}(t) &= \begin{bmatrix} \hat{\mathcal{E}} \\ \hat{\Sigma} \end{bmatrix}^T \begin{bmatrix} \Psi & \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi \\ \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi & 2\mathcal{K}_{\mathcal{I}} \hat{\mathcal{L}} \end{bmatrix} \begin{bmatrix} -\mathcal{K}_{\mathcal{P}} \hat{\mathcal{L}} \hat{\mathcal{E}} - \mathcal{K}_{\mathcal{I}} \hat{\mathcal{L}} \hat{\Sigma} + \Delta \\ \hat{\mathcal{E}} \end{bmatrix} \\ &= \hat{\mathcal{E}}^T \left(\frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi - \mathcal{K}_{\mathcal{P}} \Psi \hat{\mathcal{L}} \right) \hat{\mathcal{E}} + \hat{\Sigma}^T \left(-\frac{\mathcal{K}_{\mathcal{I}}^2}{\mathcal{K}_{\mathcal{P}}} \Psi \hat{\mathcal{L}} \right) \hat{\Sigma} \\ &\quad + \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \hat{\Sigma}^T \Psi \Delta + \hat{\mathcal{E}}^T \Psi \Delta \end{aligned} \quad (48)$$

It can be arrived that

$$\begin{aligned} \dot{\mathcal{V}}(t) &\leq \hat{\mathcal{E}}^T \left(\frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi - \mathcal{K}_{\mathcal{P}} \Psi \hat{\mathcal{L}} \right) \hat{\mathcal{E}} + \hat{\Sigma}^T \left(-\frac{\mathcal{K}_{\mathcal{I}}^2}{\mathcal{K}_{\mathcal{P}}} \Psi \hat{\mathcal{L}} \right) \hat{\Sigma} \\ &\quad + \frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \left\| \hat{\Sigma}^T \right\| \left\| \Psi \right\| \left\| \Delta \right\| + \left\| \hat{\mathcal{E}}^T \right\| \left\| \Psi \right\| \left\| \Delta \right\| \\ &= \hat{\mathcal{E}}^T \left(\frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi - \mathcal{K}_{\mathcal{P}} \Psi \hat{\mathcal{L}} \right) \hat{\mathcal{E}} + \hat{\Sigma}^T \left(-\frac{\mathcal{K}_{\mathcal{I}}^2}{\mathcal{K}_{\mathcal{P}}} \Psi \hat{\mathcal{L}} \right) \hat{\Sigma} \\ &\quad + \left(\frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \left\| \hat{\Sigma} \right\| + \left\| \hat{\mathcal{E}} \right\| \right) \left\| \Psi \right\| \left\| \Delta \right\| \end{aligned} \quad (49)$$

Based on (40), by selecting positive values for $\mathcal{K}_{\mathcal{P}}$ and $\mathcal{K}_{\mathcal{I}}$, and considering the nonnegative definiteness of the Laplacian matrix $\hat{\mathcal{L}}$ as outlined in Remark 3, the Schur complement theorem [42], [43] ensures that the coefficient matrix of the error dynamics in (40) is non-positive definite. As a result, $\|\hat{\mathcal{E}}(t)\|$ and $\|\hat{\Sigma}(t)\|$ are guaranteed to be bounded, i.e., $\|\hat{\mathcal{E}}(t)\| \neq \infty$ and $\|\hat{\Sigma}(t)\| \neq \infty$. Furthermore, as established by Theorem 2, the linearized error $\|\Delta(t)\| \rightarrow 0$ holds. Consequently, the term $\left(\frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \left\| \hat{\Sigma} \right\| + \left\| \hat{\mathcal{E}} \right\| \right) \left\| \Psi \right\| \left\| \Delta \right\|$ in (49) approaches zero. This leads to the following:

$$\dot{\mathcal{V}}(t) \leq \hat{\mathcal{E}}^T \left(\frac{\mathcal{K}_{\mathcal{I}}}{\mathcal{K}_{\mathcal{P}}} \Psi - \mathcal{K}_{\mathcal{P}} \Psi \hat{\mathcal{L}} \right) \hat{\mathcal{E}} + \hat{\Sigma}^T \left(-\frac{\mathcal{K}_{\mathcal{I}}^2}{\mathcal{K}_{\mathcal{P}}} \Psi \hat{\mathcal{L}} \right) \hat{\Sigma} \quad (50)$$

Given that the Laplace matrix $\hat{\mathcal{L}}$ is semi-definite and Ψ is a positive definite diagonal matrix, substituting (43) into (50) yields $\dot{\mathcal{V}}(t) < 0$. This implies that the formation tracking error $\hat{\mathcal{E}}$ converges to a small neighborhood of zero. This concludes the proof of Theorem 3. ■

Remark 6: Theorem 3 shows that the formation tracking error $\|\hat{\mathcal{E}}(t)\|$ will converge to a small neighborhood of zero under the proposed distributed PI-based formation control protocol (34). More precisely, this boundary is directly proportional to $\sup_{t \geq 0} \|\Delta(t)\|$, where the proportionality constant depends on $\mathcal{K}_{\mathcal{P}}$, $\mathcal{K}_{\mathcal{I}}$ and $\hat{\mathcal{L}}$. In other words, reducing $\sup_{t \geq 0} \|\Delta(t)\|$ can tighten the boundary of this neighborhood, leading to improved formation tracking performance.

V. ALGORITHM IMPLEMENTATION AND ANALYSIS

A. Framework Overview

The algorithmic procedure is formally presented in Fig. 2, segmented into three phases: initialization, adversarial training, and the subsequent formation control validation.

The adversarial training process follows the established methods described in the original GANs [25], where the discriminator is carefully pre-trained to differentiate between real and generated data prior to the generator's training. Furthermore, as explained in Theorem 2, the margin ε_i must be accurately set before each training iteration to guarantee the GANs' convergence.

Remark 7: Unlike end-to-end learning based control methods, this paper presents a two-stage approach, with robustness ensured at each stage. In the first stage, GANs handle uncertainties by transforming the agent's nonlinear dynamics, influenced by external disturbances and FDI attacks, into a linear system. Unlike observer-based methods that offer direct compensation, this approach addresses nonlinear characteristics. In the second stage, based on the linearized dynamics, the PI protocol is designed to ensure error convergence for MASs, as shown in (43). The PI parameters are adjusted to maintain a phase margin of $30^\circ \sim 40^\circ$ [44], providing robustness against all uncertainties.

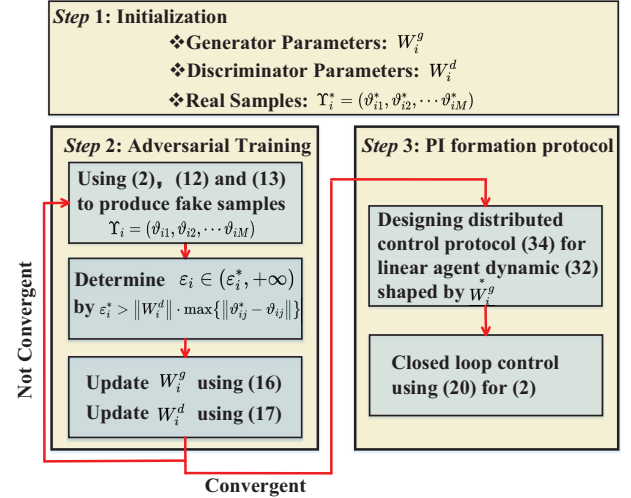


Fig. 2. Flowchart of the Algorithm Workflow. The control scheme can be divided into three stages: initialization, adversarial training, and subsequent validation of formation control.

B. Computational Requirements and Deployment

The GAN-based linearization in this study utilizes low-dimensional input-output data from nonlinear control systems. Unlike the high-dimensional data used in image, video, or language, this process requires significantly fewer computational resources. Moreover, the training procedure is conducted on a high-performance computer equipped with GPUs. Following training, the generator model parameters are exported using ONNX [45] and deployed to the agent's onboard computing units. Onboard computations are only limited to executing inference tasks for the generator-based linearization controller and running the PI-based formation protocol. By avoiding the need for training on limited onboard computation resource, the process significantly reduces debugging time and ensures efficient implementation.

VI. PHYSICAL EXPERIMENTS AND RESULTS

To validate our algorithm, we conduct physical multi-UAVs experiments in environments characterized by wind disturbances and FDI attacks, focusing on formation tasks with three quadrotors. The experimental videos can be available at: <https://youtu.be/5wdKSP2JXqA>.

A. Experimental Platform

The experimental platform is depicted in Fig. 3 (a). The OptiTrack motion capture system provides positional information for the UAVs through marker attachments placed on the drones. The UAV swarm consists of three DJI Tello quadrotor

UAVs, designated as UAV#1, UAV#2, and UAV#3, each with a maximum speed of 8 m/s. The WindShape device, a small indoor wind tunnel with wind-making capabilities, generates a sustained disturbance along the X_g with a wind speed of approximately 1.8 m/s. Additionally, Fan#1 and Fan#2 are rotatable fans that produce time-varying wind disturbances in the $X_g Y_g$ plane, with a maximum wind speed of 2 m/s.

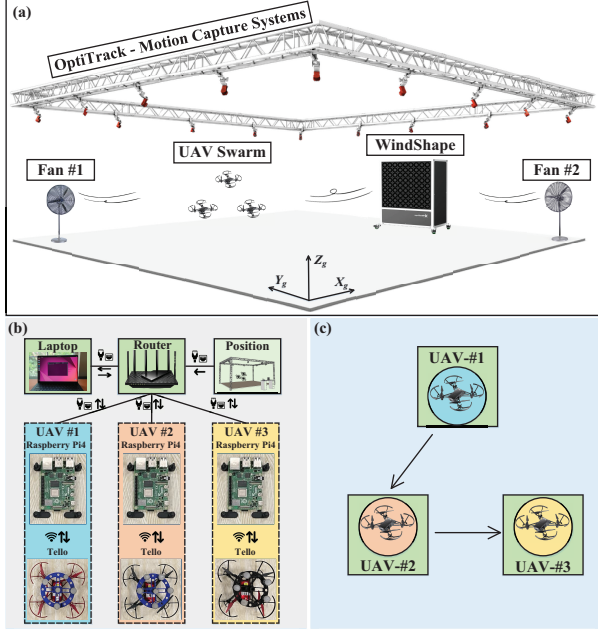


Fig. 3. Multi-UAV Experimental Platform: (a) Experimental scenario. (b) Composition and communication architecture of multi-UAVs. (c) Communication topology

Fig. 3 (b) illustrates the control and communication architecture of the multi-UAVs system. Each UAV control unit consists of a Tello drone, which is already capable of maintaining stability in the attitude channel, and a Raspberry Pi-4, which serves as the position controller where the formation algorithm is executed. The UAV connects to the Raspberry Pi-4 via a point-to-point wireless Wi-Fi connection, while each UAV control unit is connected to the router through a wired network. This setup assigns each UAV control unit a unique IP address, ranging from 192.168.1.201 to 192.168.1.203, allowing MAS communication within a defined topological network. Each UAV control unit receives positional data from OptiTrack via the router. Additionally, a laptop communicates with the multi-UAVs system to send commands for takeoff, landing, and flight operations. The communication mechanism for all system threads follows the Robot Operating System (ROS) publish-subscribe model.

B. Experimental Parameters

The DNNs parameters for the generator were initially configured based on the requirements of Theorem 1. We selected $K_1 = K_2 = 5$ and employed linearly independent activation functions $[\tilde{\sigma}_1, \tilde{\sigma}_2, \dots, \tilde{\sigma}_{K_1}, \hat{\sigma}_1, \hat{\sigma}_2, \dots, \hat{\sigma}_{K_2}]$, defined as Hermite polynomials: $H_0(x) = 1, H_1(x) = 2x, H_2(x) = 4x^2 - 2, H_3(x) = 8x^3 - 12x, \dots, H_{10}(x)$. This

configuration corresponds to a single-layer network with 10 neurons, using Hermite polynomials as activation functions. However, this architecture exhibited limited fitting capability for highly nonlinear dynamics. To address this, we considered increasing K_1 and K_2 , which would require higher-order Hermite polynomials. While this improved the network's capacity, it significantly increased computational complexity. According to Theorem 1, the linearization loss is quadratic with respect to the parameters, and $\hat{\Xi}_i^T \hat{\Xi}_i$ in (24) is guaranteed to be semi-positive definite, ensuring convexity. The single-layer design with linearly independent activation functions is chosen to ensure theoretical strict convexity. To balance theoretical strict convexity and practical generalization ability, we adopt a trade-off approach. The generator network is redesigned with 2 hidden layers, each containing 50 neurons, and ReLU activation functions are used. This enhanced the generator's ability to model nonlinear uncertainties while maintaining computational efficiency. The discriminator, in contrast, is configured with two hidden layers of 15 neurons each. As its primary role is binary classification (distinguishing real from fake samples), it does not require the complexity needed for fitting nonlinear uncertainties. Both the generator and discriminator include an ℓ_2 -regularization factor ($C = 0.01$) to prevent overfitting and to ensure that the feasible parameter set remains convex, as discussed in Remark 4. Additionally, spectral normalization [46] is applied to the linear layers, where the weight matrix is rescaled by its spectral norm.

The communication topology is shown in Fig. 3(c). Therefore, $a_\psi(\mathcal{L}) = 0.2247$ and $\lambda_{\min}(\hat{L}) = 2.0$. The desired distances in ground-fixed coordinate $X_e Y_e Z_e$ are set as follows: in the X_e direction, $p_{12}^x = p_{13}^x = 1.5$; in the Y_e direction, $p_{12}^y = 1.0, p_{13}^y = -1.0$; in the Z_e direction, $p_{12}^z = 0.0, p_{13}^z = 0.0$. The false data for the FDI attacks, defined in (3), are generated by $\varrho(t)_1^{xyz} = 1.1 + 2 \sin(t), \varrho(t)_2^{xyz} = 2.1 - \cos(t)$, and $\varrho(t)_3^{xyz} = 1.5 + 0.5 \sin(t)$. The probabilities of successful FDI attacks in different communication channels, specified in (5), are set as $\mathcal{P}_{12}^{xyz} = 0.01$ and $\mathcal{P}_{23}^{xyz} = 0.02$. Then the uncertainties caused by FDI attacks $\hat{\varrho}_1(t), \hat{\varrho}_2(t)$, and $\hat{\varrho}_3(t)$ are applied as input disturbances to the controller inputs of UAVs #1, #2, and #3 during the formation flight task.

Remark 8: During the training and parameter tuning process, several practical tricks and techniques are applied. The initial weights of the neural networks, particularly the generator, significantly influence the ability to compensate for nonlinear uncertainties. To address this, we adopt a pre-training and fine-tuning approach. First, GAN-based linearization is conducted offline under disturbance-free conditions to obtain stable initial parameters. These parameters are then adapted online under various disturbance scenarios. For tuning the closed-loop formation gain parameters $(\mathcal{K}_P, \mathcal{K}_I)$, we follow a specific procedure: Initially, under conditions without disturbances or attacks, we analyze the static formation error of UAVs. By evaluating the closed-loop step response, the system's phase margin is adjusted to approximately $30^\circ \sim 40^\circ$, ensuring sufficient robustness for the closed-loop system.

C. System identification, Training offline & Online adaption

In this section, we describe the process of training GANs to linearize the UAV-agent's dynamic model in the presence

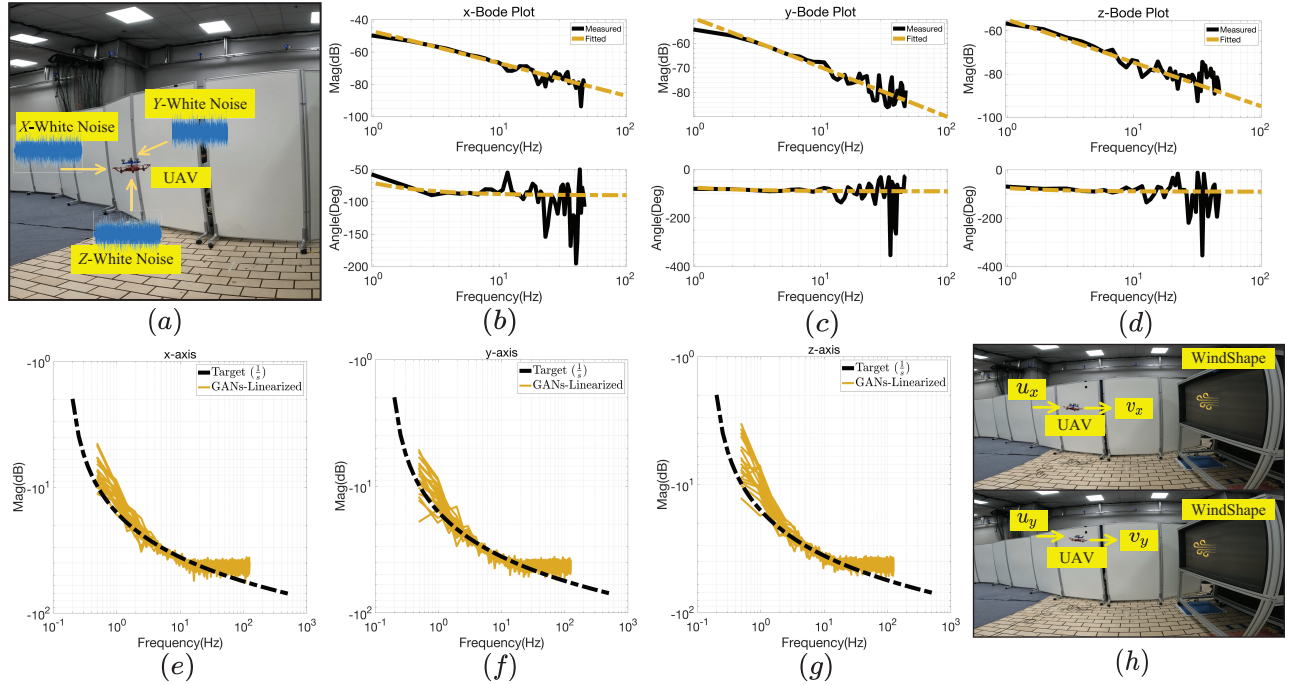


Fig. 4. Results of learning: (a) Open-loop system identification of UAV dynamics. The nominal dynamic model along the X_b , Y_b , and Z_b axes is identified using white noise excitation as input, with position output recorded in the absence of disturbances. (b-d) Identification fitting: UAV X_b , Y_b , and Z_b axes. Spectral analysis is performed to determine the open-loop frequency characteristics, followed by amplitude curve fitting in the low-to-mid frequency range on the Bode plot. (e-g) GANs linearization: UAV X_b , Y_b , and Z_b axes. This stage is implemented during offline training, equivalent to a pre-training phase to obtain the optimal parameters of the GANs under the nominal model. (h) Online adaptation: UAV X_b and Y_b axes under disturbance. The pre-trained DNN weights are fine-tuned to respond to WindShape disturbances, ensuring stable linearized input-output dynamics in the X_b and Y_b directions.

of uncertainties, prior to the real formation flight test. The initial step is to identify the nominal dynamic model of the UAV along the X_b , Y_b , and Z_b axes in the absence of disturbances. White noise is first applied as input, and the resulting UAV position output is recorded, as depicted in Fig.4 (a). This excitation signal is chosen for its ability to efficiently identify broadband system frequency characteristics in a single run, covering the entire frequency spectrum under analysis [47]. Additionally, the relatively low amplitude of white noise minimizes potential structural damage to the UAV and ensures the stability of markers used for position and velocity measurement. Spectral analysis of the input-output sequence is then performed to determine the open-loop frequency characteristics of the system. The amplitude curve in the low-to-mid frequency range is subsequently fitted on the Bode plot, as shown in Fig.4 (b-d), to derive the system's nominal dynamic model, excluding uncertainties, as outlined in (2). The resulting input-output transfer functions for the three axes are:

$$\begin{cases} \frac{Y_x(s)}{U_x(s)} = \frac{0.0131}{0.457s + 1} \\ \frac{Y_y(s)}{U_y(s)} = \frac{0.01211}{0.589s + 1} \\ \frac{Y_z(s)}{U_z(s)} = \frac{0.006999}{0.601s + 1} \end{cases} \quad (51)$$

The next step involves performing offline linearization training on the nominal UAV dynamic model from (51) using GANs, as detailed in Fig.2. We conducted 20 experiments

with varying weight initializations, setting the final epoch to 2000 and applying a uniform learning rate of 0.005. The results, presented in Fig.4 (e-g), indicate that the GAN-based linearization closely approximates the target linear model in the low-to-mid frequency range of $1 \sim 10$ Hz.

After completing the offline training, the initial network weights of G and D are obtained. Random disturbances generated by WindShape are then applied for online adaptation, as illustrated in Fig.4 (h). Under varying disturbance conditions, we manually input pulse commands, and OptiTrack is used to capture the actual velocity response. A segment of these input-output data is stored and treated as fake samples, while Step 2 in Fig. 2 is conducted to update the network parameters of the G and D during online training. This process ensures that the command input and velocity output in the X_b and Y_b directions remain consistent, even under disturbance, thereby preserving linear input-output dynamics. Since WindShape only generates wind in the horizontal direction, no uncertainties are applied in the Z_b direction during the online adaptation.

D. Case #1: 3D Formation Flight Experiments

In this section, we validate the overall formation performance of the multi-UAVs system in 3D space. According to Theorem 3, the control parameters are set as follows: $K_p^x = 5.33$, $K_i^x = 2.32$, $K_p^y = 5.17$, $K_i^y = 2.12$, $K_p^z = 10.51$, and $K_i^z = 5.29$.

Flight formation experiments are conducted in two scenarios: static formation (Fig.5 (a)) and dynamic formation (constant speed flight in the X_b direction, Fig.5 (d)). Performance is evaluated under three conditions: no wind and

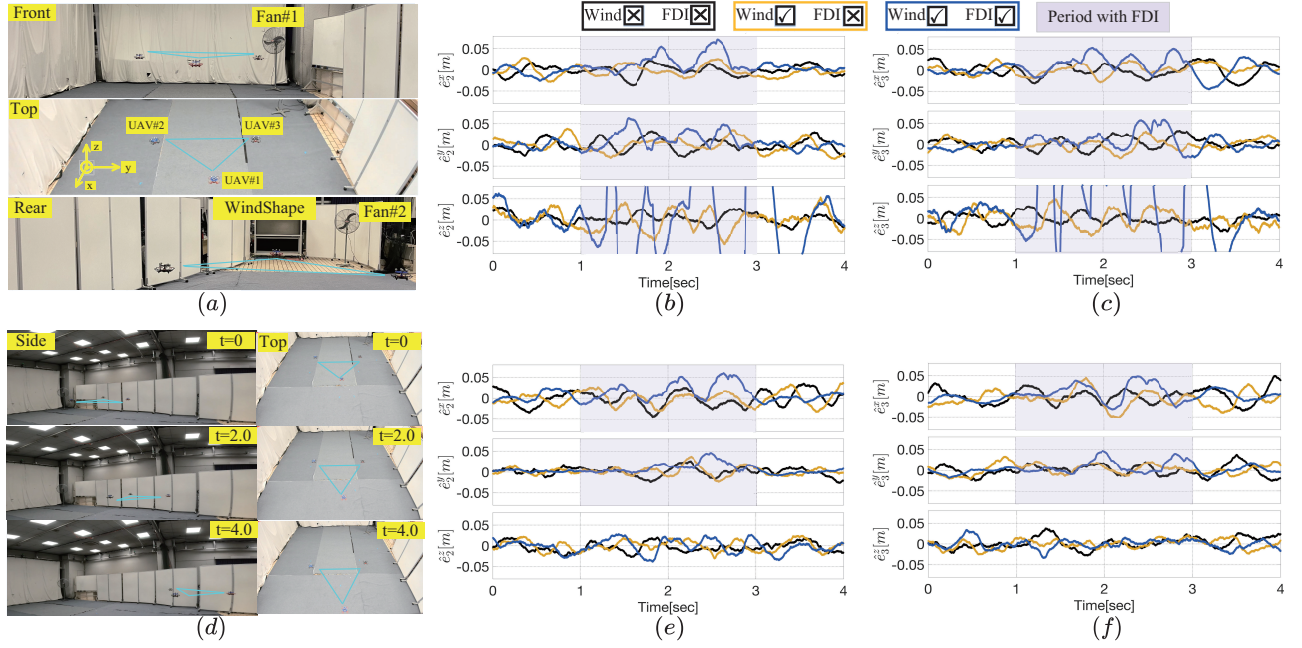


Fig. 5. Results of formation flight experiments: (a) Static formation view. Under wind disturbances and FDI attacks, UAVs #1 to #3 maintain their formation while remaining stationary. (b) The static formation error $\hat{e}_2(t)$, as defined in (35), is used to evaluate the tracking performance between UAV#1 and UAV#2. (c) Static formation error $\hat{e}_3(t)$, evaluating the tracking performance between UAV#1 and UAV#3. (d) Dynamic formation view. Under wind disturbances and FDI attacks, UAVs #1 to #3 fly along the X_g axis while maintaining the desired formation. (e) Dynamic formation error $\hat{e}_2(t)$, evaluating the tracking performance between UAV#1 and UAV#2. (f) Dynamic formation error $\hat{e}_3(t)$, evaluating the tracking performance between UAV#1 and UAV#3.

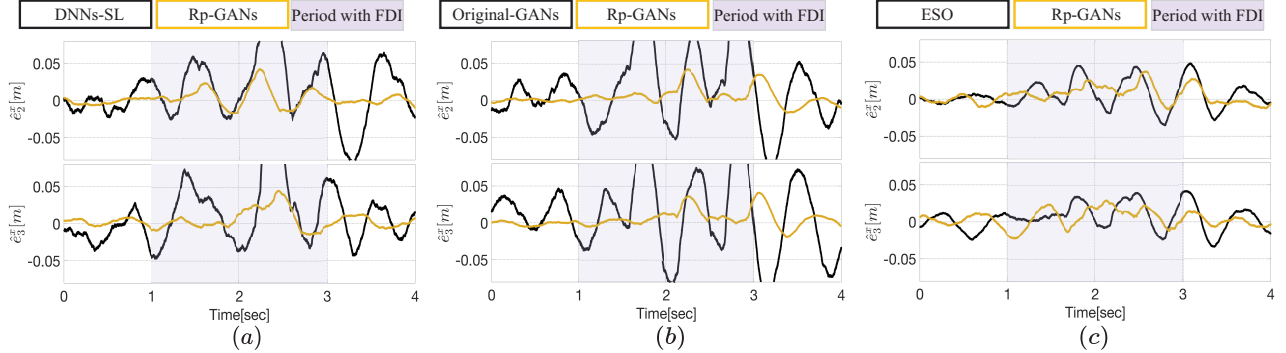


Fig. 6. Results of comparative experiments: (a) Supervised Learning (SL) with DNNs versus Rp-GANs. (b) Original GANs versus Rp-GANs. (c) ESO versus Rp-GANs.

no FDI, wind without FDI, and wind with FDI. In the wind disturbance condition, the disturbance lasts for the entire 4-second formation process. For the FDI attack condition, random attacks occur between $t = 1$ sec and $t = 3$ sec.

In the static formation results (Fig. 5 (b-c)), the GANs-based formation method maintains a formation error within 0.05 m in both the X_b and Y_b directions, despite wind disturbance and FDI attacks. However, in the Z_b direction, where no online adaptation for disturbances is applied, significant error fluctuations are observed under FDI attacks.

E. Case #2: Comparative Validation Experiments

This section examines the role of adversarial optimization and Rp-Loss in (15) within the GANs-based formation control framework, comparing its performance with classical observer-based robust control methods. Experiments are conducted in a static formation scenario (Fig. 5 (a)), focusing on the X_b

direction, which experiences the strongest wind disturbances. PI parameters remain constant across all experiments.

As a semi-supervised method, GANs employ game-theoretic adversarial optimization. Fig. 6 (a) compares Rp-GANs with supervised learning DNNs (DNNs-SL) in robust formation tasks. Both use identical network parameters, but Rp-GANs leverage adversarial training, enabling better linearization and smaller errors under uncertainties, including FDI attacks. Fig. 6 (b) highlights Rp-GANs' advantages over original-GANs, which suffer from mode collapse and instability. While both use similar network architectures, Rp-GANs achieve lower errors and enhanced stability. Under FDI attacks, original-GANs fail to maintain formation, validating Rp-GANs' robustness. Finally, Fig. 6 (c) compares Rp-GANs with the Extended State Observer (ESO) [48]. Both perform similarly without attacks, but Rp-GANs handle sudden FDI

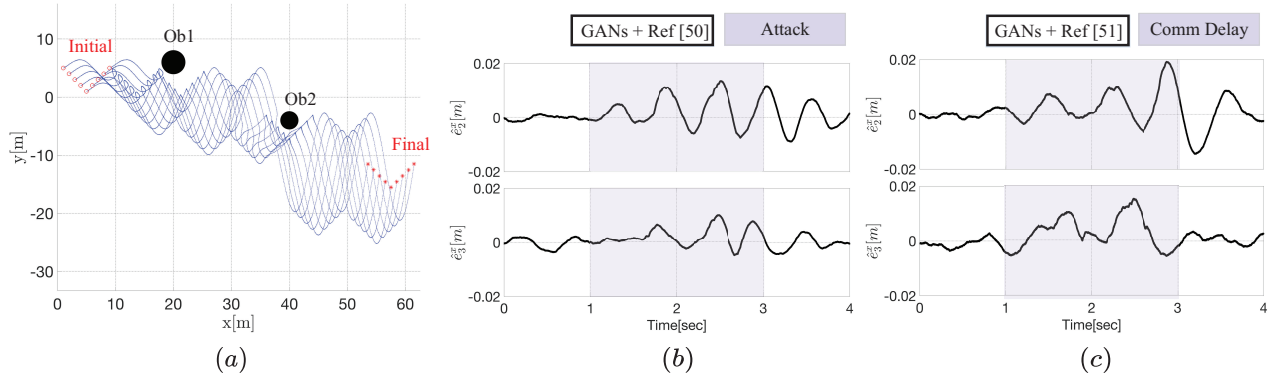


Fig. 7. Results of back-end expansion experiments: (a) Scalability Experiment with 9 UAVs in V-Formation and Obstacle Avoidance. (b) Under Complex FDI Attacks. (c) Under Communication Delays.

attacks more effectively, as ESO is better suited for continuous disturbances. The GANs-based linearization achieves robust performance without explicitly estimating uncertainties.

P. Simulation Experiments for Back-End Expansion

The proposed framework operates in two stages: the front-end uses a GANs-based linearization controller to transform nonlinear agent dynamics into well-shaped linear models, while the back-end focuses on developing robust formation control methods for these linearized agents. The following experiments extend the back-end distributed formation protocol to address various tasks using the GANs-based linearization controller as the foundation. Fig. 7 (a) demonstrates the results of expanding the number of agents and validating obstacle avoidance, employing the repulsion-based method from [49]. Fig. 7 (b) presents the validation under complex sensor attacks, where the attack models and observer configurations are designed following [50]. These methods successfully mitigate the impact of sensor-based disturbances. Fig. 7 (c) addresses the issue of communication delays using a distributed control protocol based on the method described in [51], ensuring robustness against delays in multi-agent systems. All above results highlight the flexibility of the GANs-based linearization controller, which integrates seamlessly with existing distributed formation control methods designed for linear systems. This adaptability enables the framework to effectively address common challenges in multi-agent systems, such as obstacle avoidance, complex network attacks, and communication delays.

VII. CONCLUSION

In this study, we addressed the robust formation control of MASs with nonlinear dynamics, external disturbances, and FDI attacks by developing a GAN-based controller to manage total uncertainties and facilitate linear dynamics. Our methodology incorporates adversarial training and demonstrates that formation maintenance is achievable with the application of our proposed PI control protocol. Theoretical analysis confirmed stability and convergence, and physical validations demonstrated effectiveness against mode collapse and instability issues typically problematic in GANs training.

REFERENCES

[1] I. Maza, F. Caballero, J. Capitán, J. R. Martínez-de Dios, and A. Ollero, "Experimental Results in Multi-UAV Coordination for Disaster Man-

agement and Civil Security Applications," *J Intell Robot Syst*, vol. 61, no. 1-4, pp. 563–585, Jan. 2011.

[2] S. Saripalli, J. Montgomery, and G. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Trans. Robot. Automat.*, vol. 19, no. 3, pp. 371–380, Jun. 2003.

[3] R. S. Smith and F. Y. Hadaegh, "Control of Deep-Space Formation-Flying Spacecraft; Relative Sensing and Switched Information," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 106–114, Jan. 2005.

[4] Q. Wei, X. Wang, X. Zhong, and N. Wu, "Consensus Control of Leader-Following Multi-Agent Systems in Directed Topology With Heterogeneous Disturbances," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 423–431, Feb. 2021.

[5] Y. Yang, X. Xiao, and T. Li, "Attacks on Formation Control for Multiagent Systems," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12 805–12 817, Dec. 2022.

[6] Z. Pan, Z. Sun, H. Deng, and D. Li, "A Multilayer Graph for Multiagent Formation and Trajectory Tracking Control Based on MPC Algorithm," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13 586–13 597, Dec. 2022.

[7] L. Chen, J. Xiao, R. C. H. Lin, and M. Feroskhan, "Angle-Constrained Formation Maneuvering of Unmanned Aerial Vehicles," *IEEE Trans. Contr. Syst. Technol.*, vol. 31, no. 4, pp. 1733–1746, Jul. 2023.

[8] X. Ge, Q.-L. Han, L. Ding, Y.-L. Wang, and X.-M. Zhang, "Dynamic Event-Triggered Distributed Coordination Control and its Applications: A Survey of Trends and Techniques," *IEEE Trans. Syst. Man Cybern, Syst.*, vol. 50, no. 9, pp. 3112–3125, Sep. 2020.

[9] C. Mu and J. Peng, "Learning-Based Cooperative Multiagent Formation Control With Collision Avoidance," *IEEE Trans. Syst. Man Cybern, Syst.*, vol. 52, no. 12, pp. 7341–7352, Dec. 2022.

[10] H. Liu, P. Weng, X. Tian, and Q. Mai, "Distributed adaptive fixed-time formation control for UAV-USV heterogeneous multi-agent systems," *Ocean Engineering*, vol. 267, p. 113240, Jan. 2023.

[11] J. Lan, Y.-J. Liu, D. Yu, G. Wen, S. Tong, and L. Liu, "Time-Varying Optimal Formation Control for Second-Order Multiagent Systems Based on Neural Network Observer and Reinforcement Learning," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 35, no. 3, pp. 3144–3155, Mar. 2024.

[12] Z. Liu, Y. Li, F. Wang, and Z. Chen, "Reduced-Order Observer-Based Leader-Following Formation Control for Discrete-Time Linear Multi-Agent Systems," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1715–1723, Oct. 2021.

[13] W. He, W. Xu, X. Ge, Q.-L. Han, W. Du, and F. Qian, "Secure Control of Multiagent Systems Against Malicious Attacks: A Brief Survey," *IEEE Trans. Ind. Inf.*, vol. 18, no. 6, pp. 3595–3608, Jun. 2022.

[14] G. Li, X. Wang, Z. Zuo, Y. Wu, and J. Lü, "Fault-Tolerant Formation Control for Leader-Follower Flight Vehicles Under Malicious Attacks," *IEEE Trans. Intell. Veh.*, pp. 1–15, 2024.

[15] G. Franze, F. Tedesco, and D. Famularo, "Resilience Against Replay Attacks: A Distributed Model Predictive Control Scheme for Networked Multi-Agent Systems," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 3, pp. 628–640, Mar. 2021.

[16] J. Xiao and M. Feroskhan, "Cyber Attack Detection and Isolation for a Quadrotor UAV With Modified Sliding Innovation Sequences," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7202–7214, Jul. 2022.

- [17] C. Mu, J. Peng, and C. Sun, "Hierarchical Multiagent Formation Control Scheme via Actor-Critic Learning," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 34, no. 11, pp. 8764–8777, Nov. 2023.
- [18] H. Zhang, J. H. Park, D. Yue, and W. Zhao, "Nearly Optimal Integral Sliding-Mode Consensus Control for Multiagent Systems With Disturbances," *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 51, no. 8, pp. 4741–4750, Aug. 2021.
- [19] C. Yan, X. Xiang, and C. Wang, "Fixed-Wing UAVs flocking in continuous spaces: A deep reinforcement learning approach," *Robotics and Autonomous Systems*, vol. 131, p. 103594, Sep. 2020.
- [20] J. Xiao, P. Pisutsin, and M. Feroskhan, "Collaborative Target Search With a Visual Drone Swarm: An Adaptive Curriculum Embedded Multistage Reinforcement Learning Approach," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 36, no. 1, pp. 313–327, Jan. 2025.
- [21] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems," *IEEE Trans. Automat. Contr.*, vol. 64, no. 7, pp. 2737–2752, Jul. 2019.
- [22] K. Muhammad, A. Ullah, J. Lloret, J. D. Ser, and V. H. C. De Albuquerque, "Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions," *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 7, pp. 4316–4336, Jul. 2021.
- [23] D. Wang, T. Fan, T. Han, and J. Pan, "A Two-Stage Reinforcement Learning Approach for Multi-UAV Collision Avoidance Under Imperfect Sensing," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3098–3105, Apr. 2020.
- [24] M. Wei, L. Zheng, Y. Wu, H. Liu, and H. Cheng, "Safe Learning-Based Control for Multiple UAVs Under Uncertain Disturbances," *IEEE Trans. Automat. Sci. Eng.*, pp. 1–14, 2023.
- [25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.
- [26] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.
- [27] R. Zuo, Y. Li, M. Lv, and Z. Liu, "Distributed asynchronous consensus control of nonlinear multi-agent systems under directed switching topologies," *Automatica*, vol. 152, p. 110952, Jun. 2023.
- [28] H. Ren and P. Ben-Tzvi, "Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks," *Robotics and Autonomous Systems*, vol. 124, p. 103386, Feb. 2020.
- [29] K.-F. Chu, A. Y. S. Lam, C. Fan, and V. O. K. Li, "Disturbance-Aware Neuro-Optimal System Control Using Generative Adversarial Control Networks," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 32, no. 10, pp. 4565–4576, Oct. 2021.
- [30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [31] L. B. Freidovich and H. K. Khalil, "Performance Recovery of Feedback-Linearization-Based Designs," *IEEE Trans. Automat. Contr.*, vol. 53, no. 10, pp. 2324–2334, Nov. 2008.
- [32] C. Xian, Y. Zhao, Z.-G. Wu, G. Wen, and J.-A. Pan, "Event-Triggered Distributed Average Tracking Control for Lipschitz-Type Nonlinear Multiagent Systems," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 779–792, Feb. 2023.
- [33] J. Umlauf and S. Hirche, "Feedback Linearization Based on Gaussian Processes With Event-Triggered Online Learning," *IEEE Trans. Automat. Contr.*, vol. 65, no. 10, pp. 4154–4169, Oct. 2020.
- [34] Y.-X. Li, "Command Filter Adaptive Asymptotic Tracking of Uncertain Nonlinear Systems With Time-Varying Parameters and Disturbances," *IEEE Trans. Automat. Contr.*, vol. 67, no. 6, pp. 2973–2980, Jun. 2022.
- [35] X.-G. Guo, P.-M. Liu, Z.-G. Wu, D. Zhang, and C. K. Ahn, "Hybrid Event-Triggered Group Consensus Control for Heterogeneous Multiagent Systems With TVNUD Faults and Stochastic FDI Attacks," *IEEE Trans. Automat. Contr.*, vol. 68, no. 12, pp. 8013–8020, Dec. 2023.
- [36] A. Jolicoeur-Martineau, "On relativistic f-divergences," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 13–18 Jul 2020, pp. 4931–4939.
- [37] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [38] I. Selesnick, "Sparse Regularization via Convex Analysis," *IEEE Trans. Signal Process.*, vol. 65, no. 17, pp. 4481–4494, Sep. 2017.
- [39] J. Cohen, S. Kaur, Y. Li, J. Z. Kolter, and A. Talwalkar, "Gradient descent on neural networks typically occurs at the edge of stability," in *International Conference on Learning Representations*, 2021.
- [40] G. Wen, Z. Duan, G. Chen, and W. Yu, "Consensus Tracking of Multi-Agent Systems With Lipschitz-Type Node Dynamics and Switching Topologies," *IEEE Trans. Circuits Syst. I*, vol. 61, no. 2, pp. 499–511, Feb. 2014.
- [41] Wenwu Yu, Guanrong Chen, Ming Cao, and J. Kurths, "Second-Order Consensus for Multiagent Systems With Directed Topologies and Nonlinear Dynamics," *IEEE Trans. Syst., Man, Cybern. B*, vol. 40, no. 3, pp. 881–891, Jun. 2010.
- [42] H. Gu, P. Liu, J. Lu, and Z. Lin, "PID Control for Synchronization of Complex Dynamical Networks With Directed Topologies," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1334–1346, Mar. 2021.
- [43] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.
- [44] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 13th ed. Upper Saddle River, NJ: Pearson, 2016.
- [45] ONNX Community, "Open neural network exchange," <https://github.com/onnx/onnx>, 2017.
- [46] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [47] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*, 2nd ed. Hoboken, NJ: Wiley-IEEE Press, 2012.
- [48] K. Guo, J. Jia, X. Yu, L. Guo, and L. Xie, "Multiple observers based anti-disturbance control for a quadrotor UAV against payload and wind disturbances," *Control Engineering Practice*, vol. 102, p. 104560, Sep. 2020.
- [49] X. Ge, Q.-L. Han, J. Wang, and X.-M. Zhang, "A Scalable Adaptive Approach to Multi-Vehicle Formation Control with Obstacle Avoidance," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 6, pp. 990–1004, Jun. 2022.
- [50] A. Mustafa, H. Modares, and R. Moghadam, "Resilient synchronization of distributed multi-agent systems under attacks," *Automatica*, vol. 115, p. 108869, May 2020.
- [51] J. Cheng, Y. Kang, B. Xin, Q. Zhang, K. Mao, and S. Zhou, "Time-Varying Trajectory Tracking Formation H_∞ Control for Multiagent Systems With Communication Delays and External Disturbances," *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 52, no. 7, pp. 4311–4323, Jul. 2022.



focused on learning-based control and drone applications.



Nuan Wen obtained his B.E. degree from Hebei University of Technology, Tianjin, China, in 2011, and later his M.S. and Ph.D. degrees from Beihang University, Beijing, China, in 2014 and 2018, respectively. From 2021 to 2023, he served as a Postdoctoral Researcher at the School of Automation Science and Electrical Engineering, Beihang University. In 2024, he transitioned to a position as a Research Fellow at the School of Mechanical & Aerospace Engineering, Nanyang Technological University. His research areas are now primarily

Mir Feroskhan (Member, IEEE) received the B.E. degree (Hons.) in aerospace engineering from Nanyang Technological University, Singapore, in 2011, and the Ph.D. degree in aerospace engineering from the Florida Institute of Technology, Melbourne, FL, USA, in 2016. He is currently an Assistant Professor with the School of Mechanical and Aerospace Engineering, NTU. His research interests include intelligent systems, multi-agent systems, flight dynamics and control, and aerial robotics.