mathematics

MDPI

Article

Formation Control with Connectivity Assurance for Missile Swarms by a Natural Co-Evolutionary Strategy

Junda Chen [1], Xuejing Lan [1,*], Ye Zhou [2] and Jiaqiao Liang [1]

[1] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China
[2] School of Aerospace Engineering, Engineering Campus, Universiti Sains Malaysia,
Nibong Tebal 14300, Malaysia
[*] Correspondence: lanxj@gzhu.edu.cn

Abstract: Formation control is one of the most concerning topics within the realm of swarm intelligence. This paper presents a metaheuristic approach that leverages a natural co-evolutionary strategy to solve the formation control problem for a swarm of missiles. The missile swarm is modeled by a second-order system with a heterogeneous reference target, and the exponential of the resultant error is accumulated to be the objective function such that the swarm converges to optimal equilibrium states satisfying specific formation requirements. Focusing on the issue of the local optimum and unstable evolution, we incorporate a novel model-based policy constraint and a population adaptation strategy that significantly alleviates the performance degradation of the existing natural co-evolutionary strategy in terms of slow training and instability of convergence. With application of the Molloy–Reed criterion in the field of network communication, we developed an adaptive topology method that assures connectivity under node failure, and its effectiveness is validated theoretically and experimentally. The experimental results demonstrate that the accuracy of formation flight achieved by this method is competitive with that of conventional control methods and is much more adaptable. More significantly, we show that it is feasible to treat the generic formation control problem as an optimal control problem for finding a Nash equilibrium strategy and solving it through iterative learning.

Keywords: multi-agent system; formation control; natural co-evolutionary strategy; connectivity

MSC: 93A16; 68W50

check for updates

Citation: Chen, J.; Lan, X.; Zhou, Y.; Liang, J. Formation Control with Connectivity Assurance for Missile Swarms by a Natural Co-Evolutionary Strategy. *Mathematics* **2022**, *10*, 4244. https://doi.org/10.3390/math10224244

Academic Editor: Jakub Nalepa

Received: 16 September 2022
Accepted: 10 November 2022
Published: 13 November 2022

1. Introduction

Intelligent control of swarm systems has been widely used in tracking, rescue and delivery [1,2]. Through the interactive cooperation of multiple agents, the swarm can exhibit complex intelligent behaviors, such as cohesion, separation and alignment, known as the Reynolds rules [3]. Among the various research directions, the formation control problem has been widely studied on a variety of system models for its theoretical and practical values.

In this paper, we focus on the formation control of missile swarms. Similar to common formation control problems, formation control for a missile swarm covers formation initialization, contraction, expansion and reconfiguration [4], corresponding with formation producing and formation tracking problems [5]. Based on the sensing capability and interaction level, the current dominant research [6] classifies formation control methods into position-, displacement- and distance-based control. The main difference between these control methods is the ability to sense relative or absolute state information. Thus, we propose classifying angle-based control as a particular case of displacement-based control, in which the relative distance constraint from the relative position is removed. In this paper, we focus on the displacement-based control framework not only for its simplicity and stability but also for its vast realistic application value [7].

In practice, when dealing with large-scale swarm systems, control methods that the system can adopt are usually subjected to limited communication bandwidth, communication quality and other interference issues. Hence, more flexible communication means that tolerance to communication failures is necessary for achieving robust information transmission. However, traditional communication methods based on a fixed communication topology have difficulty coping with unexpected situations and providing continuous and reliable communication. For developing more robust intelligent swarm communication algorithms, two newly developed adaptive communication methods in the field of network control system are known as ad hoc-based networks and cluster-based networks [8]. A recent study utilized the received signal strength instead of localization facilities to measure individual distances [9]. Another approach under the same ad-hoc framework [10] developed a more robust intelligent swarm communication algorithm based on the Molloy–Reed criterion and grey prediction, improving the robustness of the drone swarm network and the reliability of data transmission. Such an algorithm is further employed in the flying ad-hoc network (FANET), which is a distributed and self-organizing communication framework. Considering that distance is usually closely related to communication quality, and the perception of distance does not require additional communication bandwidth, it is highly feasible to treat distance as a major factor in configuring the communication topology. In this paper, we develop an adaptive topology communication method based on minimizing the communication distance and overcome the problem of head failure in the cluster-based communication method to ensure connection stability for formation control.

Formation control methods based on traditional control theory and dynamic systems can provide more robust control for single control objectives and motion patterns [11], but it is hard to achieve high-precision coordination through conventional control methods with multiple objectives or in environments that are too complex to be modeled. That is why intelligent control approaches have been introduced as more flexible alternatives for operating in unstructured or dynamic environments surrounded by multiple uncertainties.

A number of metaheuristic algorithms have been shown to cope well with multi-objective, complex constrained optimization problems and have been widely used in all aspects of formation control, including the formation of optimal configurations and motion planning. A comprehensive survey on the development of such algorithms for aircraft motion planning problems is presented in [12]. In [13], Liu investigated a particle swarm optimization (PSO)-based algorithm in the optimal design of missile formation configuration. In [14], Seung-Mok proposed a cooperative co-evolution PSO algorithm based on the traditional PSO-based model predictive control, which optimizes in a distributed way and improves the speed and performance of the original algorithm. Another population-based metaheuristic algorithm, known as the genetic algorithm (GA), was employed to evolve the positioning strategy of formation-controlled multi-robot systems [15]. A gradient descent-based reinforcement learning method utilizing an actor-critic framework was proposed for optimal consensus control of multi-agent systems. Existing research on consensus control provides another perspective on solving the formation problem, since the general formation control problem is a special kind of consensus problem which requires that certain errors are maintained between the states of neighboring robots rather than identical states. Although existing metaheuristic-based algorithms have been applied to formation control in searches for single-step optimal solutions in real time, these methods are usually slow when running and do not have the ability to migrate and learn. In contrast, neural network-based controllers can be trained to achieve the same performance through iterative learning and can be easily deployed to compact mobile units with low computational performance and cost, although the majority of the computational cost is spent in the training stage.

There is a dearth of existing research using neural network controllers, despite the fact that they are commonly used under the reinforcement learning (RL) paradigms [16] and trained to solve specific control or decision-making problems. In [17], Liu developed a self-organizing map (SOM)-based neural network approach for motion planning in the formation control problem of a multiple autonomous underwater vehicle (AUV) system, where the SOM is an unsupervised neural network serving specific purposes, such as competitive learning and task assignment [18]. This is an iterative algorithm used to search for the most energy-efficient motion path for the agent. Since it takes all individuals' states as input and the target path as the output, it can be regarded as a centralized method which is not desirable for the practical deployment requirement.

The general idea of the heuristic algorithm using a neural network controller is similar to other algorithms, such as PSO and the ant colony optimization (ACO) algorithm, which are designed to find the optimal solution of the cost function under certain constraints by iterative searching. The essential difference between the former as well as other algorithms is that the former optimizes the neural network weights, and the final optimized controller can be deployed directly. In contrast, the latter optimizes directly in the solution space in a single step. Both methods have their advantages and disadvantages. Since the latter does not need pretraining, it relies too much on single-step computation, and it is difficult to guarantee the speed requirement for real-time operation. Therefore, a neural network controller, as an adaptive and learnable controller, is considered a promising means of intelligent control in the future [19]. Evolutionary strategy can effectively and stably optimize the neural network structure and the network weights [20,21], thus benefiting any possible application scenario that requires observation of large amounts of historical data.

Apart from acting as a controller to generate control commands, neural networks are widely used in control systems for sensing, decision making, trajectory planning and many other purposes. In [22], a modified Grossberg neural network (GNN) was used to generate the shortest path to avoid obstacles and reach a target point. Similar research on optimal path or trajectory design is also available in [23,24]. Lan [25] adopted the radial basis function to estimate the system disturbance, which led to enhanced robustness and adaptability of swarm formation control based on the artificial potential field method. Additional research that implements adaptive neural networks to estimate the uncertain and nonlinear dynamics of the system can be found in [26–28]. Furthermore, in [29], Lan used reinforcement learning theory to train a neural network controller that could be applied to the distributed control of swarm systems in an unknown dynamic environment, where the agents in the swarm can perform basic intelligent behaviors such as tracking and obstacle avoidance. Likewise, extensive research has shown that neural networks have certain robustness in many scenarios without being inferior to traditional methods and are relatively more flexible and applicable.

The foremost motivation of this paper is to develop a metaheuristic evolutionary computational approach to solve the formation control problem for multi-agent systems (MASs) while exploring the usage of neural network (NN) controllers. We use a recently proposed natural co-evolutionary strategy (NCES) algorithm [30] to realize such a vision and apply it to the control of a second-order multi-missile system. The contributions of this paper are manifold. First, we incorporate a policy constraint approach to enhance the stability of the algorithm in order to optimize the objective function and find the Nash equilibrium strategy. The proposed algorithm is more flexible for adapting to varying control objectives compared with conventional approaches [4,31–33]. Then, an adaptive topology scheme is designed to address the common node failure problem in formation control, and this method can achieve stable communication connections at a relatively low communication cost. Finally, a stable population adaptation method is proposed to further improve the performance of the algorithm and mitigate the local optimum issue. Emulation experiments show that the proposed formation control algorithm can handle tasks such as formation maintenance, reference trajectory tracking and formation switching with high accuracy in the presence of obstacles, and it is immune to disturbances such as randomized initial positions and node failure. The implementation code (https://github.com/GEYOUR/Swarm-Missile-Formation-Control accessed on 9 September 2022) of the parallelized algorithm as well as the experimental platform are available online to encourage further research on the constrained swarm system.

The remainder of this work is constructed as follows. In Section 2, the nonlinear multi-missile system is modeled, and the displacement-based formation control problem is formulated, including the specification of formation patterns. In Section 3, the distributed NCES-based formation control algorithm is proposed based on a neural network controller. In Section 4, numerical experiments are conducted for a variety of scenarios, and the results are presented. Finally, the conclusions and recommendations are presented in Section 5.

2. Preliminaries and Problem Formulation

2.1. System Modeling of a Swarm of Cruise Missiles

This paper focuses on formation control in a two-dimensional space (i.e., the OXY planar space of the inertial coordinate frame). As shown in Figure 1, the swarm consists of multiple missiles, and each missile can be treated as a point mass. In the simplified dynamics model, we do not consider aerodynamic factors or the effect of the missile's own inertia tensor. Considering a total of N missiles, the ith missile is denoted by M_i, while V_{mi}, α_{mi}, x_{mi} and y_{mi} are used to represent the speed, heading angle and coordinates of M_i in the global coordinate frame. Let $\mathbf{x}_i = [x_{mi}, y_{mi}, \alpha_{mi}]^T$ represent the measurement vector of the missile, which is not necessarily required hereinafter, and $\mathbf{x}_r = [x_r, y_r, \alpha_r]^T$ represent the measurement vector of the reference target. For brevity, let $\hat{\mathbf{x}}_i = [\mathbf{x}_i^T, V_{mi}]^T$ be the state vector with speed inserted. Here, a_{vi} and a_{li} are defined as the acceleration commands along the direction of the velocity and perpendicular to the direction of the velocity, respectively. As shown in the partial enlargement of the actuation decomposition, the two acceleration commands are independent and perpendicular to each other. The second-order system dynamic of the ith missile can be expressed as

$$\begin{cases} \dot{\hat{\mathbf{x}}}_i = \Psi(\hat{\mathbf{x}}_i) + \Phi(\hat{\mathbf{x}}_i)u_i \\ \mathbf{x}_i = C\hat{\mathbf{x}}_i \end{cases}, \tag{1}$$

where $u_i = [a_{vi}, a_{li}]^T$ is the input for M_i and

$$\begin{aligned} \Psi(\hat{\mathbf{x}}_i) &= \begin{bmatrix} V_{mi}\cos\alpha_{mi} \\ V_{mi}\sin\alpha_{mi} \\ 0 \\ 0 \end{bmatrix}, \\ \Phi(\hat{\mathbf{x}}_i) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1/V_{mi} \\ 1 & 0 \end{bmatrix}, \\ C &= \left[I_3, 0_{3,1}\right], \end{aligned} \tag{2}$$

where I_n and $0_{m,n}$ denote the identity and zero matrices. Note that the system in Equation (1) is similar to the system discussed in [34], which is often called the unicycle-like vehicle. In this paper, the missile is a nonholonomic system with controllable speed that is capable of maneuvering with limited actuation and dynamic constraints. We assume that under a certain communication relationship, the relative state $z_{ji} = \mathbf{x}_j - \mathbf{x}_i$ can be sensed by missile i, where j represents the neighboring missile. In addition, we make the following assumptions about the system:

Assumption 1. *We assume that the reference target or leader is a first-order model driven by an unknown changeable speed and angular velocity commands, which means that its vertical acceleration is not available, but its lateral acceleration is subject to the same constraints as the follower missiles. Hence, the heterogeneous system may bring in more complexity for mathematical analysis and stability issues.*

Assumption 2. *It is assumed that the missile controller applies the cascade control [35] structure such that its altitude is automatically controlled by the inner control loop, which is a stable closed loop. We focus on the design for the missiles' thrust and lateral acceleration of the outer control loop. This divide-and-conquer approach is widely used in many studies [32,36] to simplify the problem in order to verify the effectiveness of the control method, and it is also valid for three-dimensional dynamic models.*

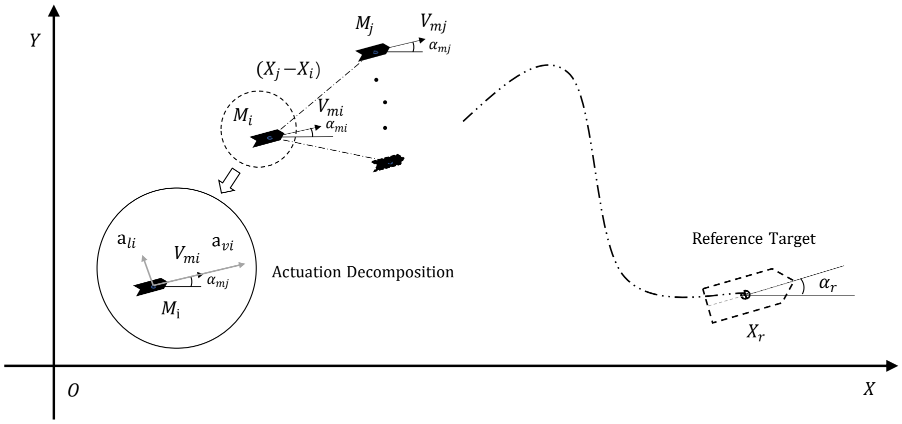


Figure 1. Engagement scenario.

2.2. Formation Control under Displacement-Based Framework

Let us denote the coordinate of the ith node of the formation by λ_{pi}. The formation pattern is defined by $\lambda = \{\lambda_{pi} = [\lambda_{xi}, \lambda_{yi}]^T : i = 1, 2, \ldots, N\}$, which is the set of Cartesian coordinates of the origin of each node that is expressed in the coordinate system such that the center of the formation is the origin, which should satisfy

$$\sum_{i=1}^{N} \lambda_{xi} = 0, \sum_{i=1}^{N} \lambda_{yi} = 0. \tag{3}$$

In the displacement-based framework, each missile has to align its own coordinate system with the global coordinate system and be able to sense the relative positions and orientations of its neighbors, whereas its position in the global coordinate system is not necessarily required. Let $\mathcal{N}_i$ denote the set of neighboring missiles of the ith missile. Suppose that the reference target should be kept in the formation center. Then, the tracking error of the ith missile is

$$e_i = \sum_{j \in \mathcal{N}_i} [(P_i - P_j) - (\mathbf{x}_i - \mathbf{x}_j)] + \zeta_i(\mathbf{x}_r + P_i - \mathbf{x}_i), \tag{4}$$

where $e_i \in \mathbb{R}^3$ and $P_i = [\lambda_{xi}, \lambda_{yi}, 0]^T$. If the target state is available to the ith missile, then $\zeta_i = 1$; otherwise, $\zeta_i = 0$. It is worth noting that the first term represents the formation maintenance error, while the second term represents the target tracking error when it is able to obtain the target information. However, the error vector is defined in the global coordinate system. In order to standardize the effect of an individual missile's heading angles on the error vector to the relative coordinate system that is aligned with the missile's orientation, we define the rotated error vector as $e_{ri} = R_3(-\alpha_i)e_i$, where $R_3(-\alpha) \in \mathbb{R}^{3\times3}$ is the three-dimensional rotation matrix defined as

$$R_3(-\alpha_i) = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i & 0 \\ -\sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

Additionally, there exists $e_{ri} = \begin{bmatrix} e_{ix} & e_{iy} & e_{i\alpha} \end{bmatrix}^T$, where the subscript r represents the rotational transformation. Considering the system in Equation (1) and the discretization of the system control loop, we can obtain the following error dynamics:

$$\dot{e_{ri}} = G_i u_i + \sum_{j \in \mathcal{N}_i} F_{ij} u_j + D_i u_r + H_i, \tag{6}$$

where

$$\begin{aligned} G_i &= \begin{bmatrix} -\tau(L_i + \xi_i) & e_{iy}/v_i \\ 0 & -e_{ix}/v_i \\ 0 & -(L_i + \xi_i)/v_i \end{bmatrix}, \\ F_{ij} &= \begin{bmatrix} \tau \cos \alpha_{ji} & 0 \\ \tau \sin \alpha_{ji} & 0 \\ 0 & 1/v_j \end{bmatrix}, \\ D_i &= \begin{bmatrix} \xi_i \cos \hat{\alpha_i} & 0 \\ \xi_i \sin \hat{\alpha_i} & 0 \\ 0 & \xi_i \end{bmatrix}, \\ H_i &= \begin{bmatrix} -(L_i + \xi_i) v_i + \sum_{j \in \mathcal{N}_i} \cos \alpha_{ji} v_j \\ \sum_{j \in \mathcal{N}_i} \sin \alpha_{ji} v_j \\ 0 \end{bmatrix}. \end{aligned} \tag{7}$$

Here, L_i is the number of missiles belonging to the set of neighbors $\mathcal{N}_i$, and τ is the simulation step size, where $\alpha_{ji} = \alpha_j - \alpha_i$ and $\hat{\alpha_i} = \alpha_r - \alpha_i$. Furthermore, $u_i = [a_{vi}, a_{li}]^T$and $u_r = [v_r, w_r]^T$ denote the vector of the control commands of missile i and the reference target, respectively.

Generally, the objective at time t can be defined by a function of the error vector:

$$J_i(t, e_{ri}(t)) = \exp(-e_{ri}^T(t) K_C e_{ri}(t)), \tag{8}$$

where $K_C = \text{diag}[\text{k}_1, \text{k}_2, \text{k}_3]$ is the symmetric positive definite matrix. K_C balances the shape and consistent direction of movement of the formation. The optimal design of the formation controller u_i can be formulated as a nonlinear optimization problem:

$$u_i^* = \arg\max_{u_i} \int_0^T J_i(t, e_{ri}(t)) \mathrm{d}t, \tag{9}$$

This is subjected to

$$\begin{aligned} u_{min} &\leq u_i \leq u_{max}, \\ V_{min} &\leq V_{mi} \leq V_{max}, \end{aligned} \tag{10}$$

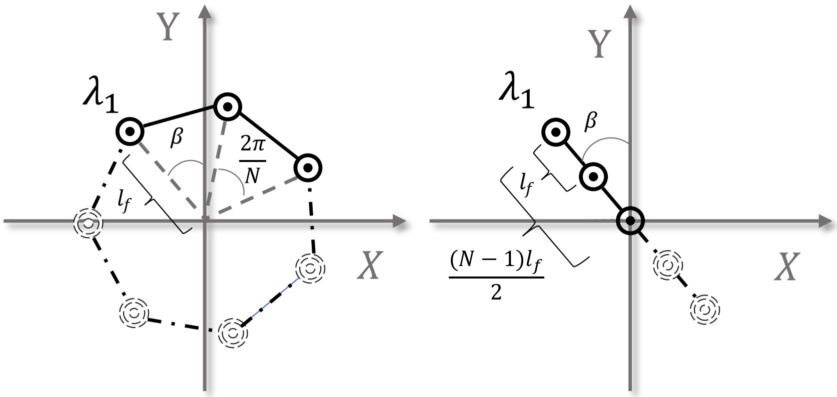
in which T denotes the total flight time and $u_{min}, u_{max}, V_{min}$ and V_{max} represent the system constraints.

In this paper, we define two formation patterns—the regular polygon and the straight line formation—which are shown in Figure 2 and denote each formation pattern by $\lambda^P_{(\beta, l_f)}$ and $\lambda^L_{(\beta, l_f)}$, respectively. In this figure, each dot represents a node of the formation, and the nodes are connected to each other under the constraint relationship to form specific formation shapes. l_f and β are the parameters that control the size and rotation of the formation, respectively, and we have

$$\begin{aligned} \lambda^P_{(\beta, l_f)} &= \{R_2(\beta)[l_f \sin(\frac{2\pi}{N}(i-1)), l_f \cos(\frac{2\pi}{N}(i-1))]^T : \\ &\quad i \in \{1, 2, \ldots, N\}\}, \\ \lambda^L_{(\beta, l_f)} &= \{R_2(\beta)[0, l_f \frac{N - 2i + 1}{2}]^T : i \in \{1, 2, \ldots, N\}\}, \end{aligned} \tag{11}$$

In addition, $R_2(\beta)$ is a two-dimensional rotation matrix similar to that in Equation (5), which is

$$R_2(\beta) = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}, \tag{12}$$



(**a**) Regular polygon formation (**b**) Straight line formation

Figure 2. Two types of formation patterns.

Examples of a variety of formation patterns can be found in [1,37,38], but specific geometric definitions are represented only in this paper.

From the above definition, it can be intuitively observed that the straight line formation is symmetric with respect to the origin of the coordinate system that defines it such that the sum of its x and y coordinates is zero, which satisfies Equation (3). Similarly, when the number of nodes of a regular polygon is even, we can also get this conclusion. Therefore, we only need to prove that when the number of nodes is odd, it still satisfies this prerequisite as follows.

Suppose that the regular polygon formation defined in Equation (11) consists of $N \in \{2n+1 : n \in \mathbb{Z}\}$ nodes. The Cartesian coordinates of each node in the formation are denoted by $P_i = [\lambda_{xi}, \lambda_{yi}]^T$ such that the sum of all coordinates is

$$\sum_{i=1}^{N} P_i = \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix}. \tag{13}$$

Suppose node 1 is located at the y axis such that $\xi_x = 0$. After rotating the formation by ρ, we can obtain the following sum of coordinates by multiplication with the rotation matrix:

$$R_2(\rho) \sum_{i=1}^{N} P_i = \begin{bmatrix} \xi_x \cos\rho - \xi_y \sin\rho \\ \xi_x \sin\rho + \xi_y \cos\rho \end{bmatrix}. \tag{14}$$

The angle between the vector from the coordinate origin to node k and the corresponding direction of the Y axis is $2(k-1)\pi/N$. If node k is rotated to be on the y coordinate axis, since the formation is again symmetric about the Y axis, the summation of the x coordinates is

$$\xi_x \cos(2(k-1)\pi/N) - \xi_y \sin(2(k-1)\pi/N) = 0, \forall k = 1, 2, \ldots, N. \tag{15}$$

Since N is an odd number, if k is not equal to 1 or $(N+1)/2$, then $2(k-1)\pi/N$ is not equal to 0 or π, and $\sin(2(k-1)\pi/N)$ is not constantly equal to 0, and therefore, $\xi_y = 0$. Thus, Equation (14) is a zero matrix for an arbitrary ρ value, and Equation (3) holds. It is shown that many other formation patterns can be considered as variants of the above two formation patterns, such as *row, column and square* patterns [1]. Moreover, as discussed in the following section, this formation paradigm can also generate some asymmetric patterns, such as *wedge and crescent* patterns [25], by deleting nodes appropriately.

3. Applying Natural Co-Evolutionary Strategy to Formation Control via Neural Networks

3.1. Natural Co-Evolutionary Strategy for MASs

With limited sensing capability, an MAS can be modeled as a multi-agent partially observable Markov decision process (POMDP), which is an extension based on the Markov decision process (MDP). Such a process can be expressed as

$$P(S \times A_u \to S'), \tag{16}$$

with S and S' representing the system state before and after transitions, respectively, and $A_u = [u_1, \ldots, u_n]$ being the action vector for all agents in the system. $P(\cdot)$ is the transition probability function. It is said to be a deterministic problem if $P(\cdot)$ is either one or zero; otherwise, it is a stochastic problem.

Considering the simplified system model neglecting uncertainties, the formation control problem in general can be solved using optimization algorithms under a multi-agent POMDP. Assuming that the fitness function measuring the performance of agent i is $f_i(\theta_i, \theta_{\mathcal{N}_i})$, in which θ_i is its controller (policy) parameter, and $\theta_{\mathcal{N}_i} = \{\theta_j : j \in \mathcal{N}_i\}$ is the set of parameters for its neighboring agents, then the objective of optimal control is to find the optimal control strategy $\theta_i^*, i \in (1, \ldots, N)$ such that

$$\begin{gathered} F(\{f_i(\theta_i^*, \theta_{\mathcal{N}_i}) : i \in (1, \ldots, N)\}) \geq \\ F(\{f_i(\theta_i \neq \theta_i^*, \theta_{\mathcal{N}_i}) : i \in (1, \ldots, N)\}), \end{gathered} \tag{17}$$

where $F(\cdot)$ denotes the overall performance of all agents in the MAS. Such a solution is often referred to as the Nash equilibrium strategy, meaning that no further improvement can be made to individual solutions without deteriorating the performance. It has been found to be very difficult to find a Nash equilibrium strategy for the nonlinear continuous agent system whose cost function is coupled with neighboring states while satisfying the system constraints in Equation (10) [14,39]. For the formation control problem, which requires sufficient cooperation among neighboring agents, $F(\cdot)$ can be regarded as a simple summation of all individuals $f_i(\cdot)$, and the formation control problem can be viewed as a constrained dynamic optimization problem with the objective of minimizing the total cost of the formation error. Thus, the Nash equilibrium strategy can be obtained using a co-evolutionary algorithm that is designed for evolving simultaneously to reach the overall optimum fitness.

In a previous work [30], we improved the natural evolutionary strategy (NES) and proposed an NCES algorithm that seeks global optimality for the constrained multi-objective optimization problem in multi-agent systems. In brief, the NCES algorithm is a bio-inspired, population-based algorithm capable of optimizing high-dimensional parameters, such as neural network weights, toward the direction of higher fitness. The NCES algorithm usually proceeds as follows. First, the parameters θ_i for $i \in (1, \ldots, N)$ are initialized, and the optimization objective is determined for a specific control problem. The fitness function $f(\cdot)$ which is embedded within the system model is thus obtained. In the second step, iterative optimization is performed, and new perturbations ϵ_i are sampled m times at the beginning of each iteration step, which obey a probability distribution $p(\cdot)$ to obtain the perturbed population $\theta_i' = \theta_i + \epsilon_i$ corresponding to each agent. Then, their fitness values are evaluated in the system in a distributed way so that each population obtains the following corresponding gradient information:

$$g_{\theta_i} = \frac{1}{m\sigma^2} \sum_{i=1}^{m} f(\theta_i', \theta_{\mathcal{N}_i}) \epsilon_i \prod_{c \in \mathcal{N}_i} p(\epsilon_c), \tag{18}$$

The parameters are also updated in a gradient ascent manner:

$$\theta_i = \theta_i + \eta_\alpha g_{\theta_i}. \tag{19}$$

Finally, the second step is looped until convergence or the Nash equilibrium strategy is found. Compared with the conventional NES algorithm, the NCES algorithm provides a more accurate estimation of gradients in the presence of multiple interactive agents. More details of the algorithm can be found in [30].

3.2. Distributed Co-Evolutionary Strategy Optimizing a Neural Network Controller

To investigate the applicability of neural network controllers within the framework of formation control problems, this paper proposes adopting a multi-layer perceptron (MLP) neural network controller with a cooperative NES-based training approach. The MLP NN has a single hidden layer of 16 nodes, and its schematic is depicted in Figure 3.

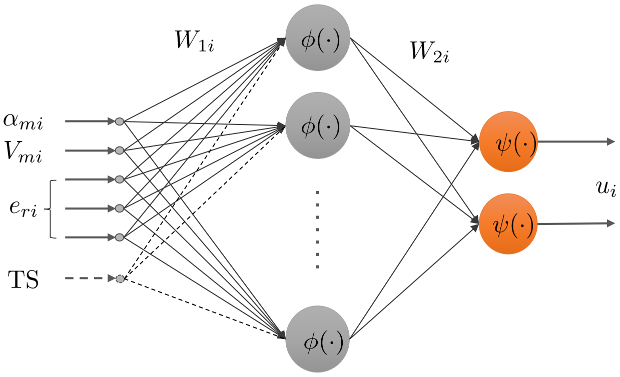


Figure 3. Neural network controller schematic.

The weighting matrices for each layer are represented by W_{1i} and W_{2i}. $\phi(\cdot)$ and $\psi(\cdot)$ are the activation functions for the hidden layer and output layer. Specifically, $\phi(\cdot)$ is the *sigmoid* function, and $\psi(\cdot)$ is selected as the hyperbolic tangent function $(Tanh)$ in order to impose restrictions on the output and satisfy the system constraints. Another advantage of the neural network controller is that saturated control can be achieved by a reasonable choice of activation function without restricting the solution space. With $z = [\alpha_{mi}, V_{mi}, e_{ri}]^T$ denoting the input, the output of the neural network controller is

$$u_i = \psi(W_{2i} \cdot \phi(W_{1i} \cdot z)). \tag{20}$$

Note that the dashed network input TS in Figure 3 represents the transform signal, which is included in the input only when needed, as described in Section 4. The same network configuration will be used in the following experiments for all agents. Applying the previously mentioned NCES algorithm to train this controller, it is first determined that the parameters of the controller are $\theta_i = [W_{1i}, W_{2i}]$. The fitness function is the objective function in Equation (8) (i.e., $f(\theta_i, \theta_{\mathcal{N}_i}) = J_i$), which is nonlinearly coupled with the states of neighboring agents and can only be evaluated through the interaction feedback within the system. In order to apply the NCES algorithm to the training of the missile formation controller in this paper, the plain NCES algorithm is insufficient to guarantee the global stability and the convergence speed of the algorithm. To further improve the performance of the algorithm and perform specific optimization for the formation problem, we propose a series of supplementary techniques to effectively enhance the convergence speed and accuracy of the algorithm.

3.3. Population Adaptation Technique

In [30], we pointed out the importance of the learning rate for the accuracy of the algorithm and proposed an algorithm for learning rate adaptation. However, adjusting the learning rate often consumes a lot of time. In order to ensure the speed of the algorithm, a novel population size adaptation algorithm is used to adjust the evolutionary process adaptatively. Based on the previous works [40,41], we can learn that the trend of the gradient is related to the complexity of the objective function. Usually, for a more complex optimization region, such as in a multi-modal or noised function, the estimated gradient is gentle, while in a flatter region, such as a spherical space, the gradient is relatively steep. To estimate the accuracy of the estimated natural gradient under the movement of the parameter distribution, the evolution path ρ_θ is introduced to detect the resistance in the evolutionary process. The population size is adapted based on the length of the evolutionary path following empirical common sense that a larger population size will lead to higher accuracy for the estimated gradient. The weight matrix W_{1i}, W_{2i} of agent i is considered its parameter vector θ_i, for which $\theta_i \in \mathbb{R}^s$ and s is the total number of weights. The evolution path in iteration t is calculated by accumulating the square of the Mahalanobis distance of the parameter movement of all agents as

$$\rho_\theta(t) = \sum_{i=1}^{N} [\theta_i(t) - \theta_i(t-1)]^T \Sigma^{-1} [\theta_i(t) - \theta_i(t-1)]. \tag{21}$$

Note that Σ is the covariance of the probability distribution from which the new populations are sampled, since the evolution path should not depend on the parameterization of the probability distribution. The population size $\eta_p(t)$ is then adjusted according to the evolutionary path as follows:

$$\begin{aligned} \eta_p(t) &= \eta_p(t-1)(\beta + (1-\beta)\frac{\rho_\theta(t-1)}{\rho_\theta(t)}), \\ \eta_p(t) &= \text{clip}(\max(\eta_p(t), \eta_p(t-1)), \eta_p^{\min}, \eta_p^{\max}), \end{aligned} \tag{22}$$

where β is a constant factor that determines the growth rate of the population size and $\eta_p^{\min}$ and $\eta_p^{\max}$ are the minimum and maximum population size which are sent to the clip function $\text{clip}(\cdot)$ in case of undesirable adapted values, respectively. Note that since the total optimization complexity tends to increase as evolution progresses, we adopted a non-decreasing strategy to adjust the population size to ensure stability. The initial population size is set to be $\eta_p(0) = 10 + 5\ln(s)$, referring to the default set-up in [42], which should be a good candidate, and the boundaries are determined as

$$\eta_p^{\min} = \eta_p(0), \eta_p^{\max} = 4\eta_p(0). \tag{23}$$

Note that s is the number of parameters of one agent. The above configuration was found to be appropriate according to the experimental results. Through empirical observation, we came to an intuitive conclusion that the increment of the population size does not necessarily lead to an improvement in evolutionary quality and sometimes even leads to difficulty in convergence or falling into the local optimum. This is probably because a large population size will greatly average the contributions of individuals and thus reduce the exploratory nature of each individual, so it is wise to adjust the size appropriately rather than just thinking that the larger, the better.

3.4. Cluster-Based Adaptive Topology

The inter-agent connectivity in the field of multi-agent system has been primarily modeled and characterized by means of graph theory [6,43], which will be utilized to identify the observable neighbors of the missiles in the formation. In this subsection, we review some of the basics of graph theory. Suppose there are n agents in the MAS (agents can be represented by nodes), and a graphs $\mathcal{G}$ is defined as $(\mathcal{V}, \epsilon)$, where $\mathcal{V} = (v_1, v_2, \ldots, v_n)$ represents the set of nodes and $\epsilon \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges composed of directed connections of different nodes. The neighbor set of the node i is defined as $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \epsilon\}$, and $j \in \mathcal{V}$ is said to be connected to $i \in \mathcal{V}$ in a directed way if $(i,j) \in \epsilon$ and $(j,i) \notin \epsilon$, while j is said to be connected to i in an undirected way if both (i,j) and (j,i) belong to ϵ. A directed path of $\mathcal{G}$ is a series of adjacent edges of the form $(v_1, v_2), (v_2, v_3), \ldots, (v_i, v_j), (v_j, v_k)$, and a graph is said to have a spanning tree if there exists a directed path from one node to any other nodes, and a graph is said to be connected in a directed (undirected) manner if there is a directed (undirected) path between any pair of distinct nodes. We use the adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ to represent the above node connectivity, in which

$$a_{ij} = \begin{cases} 0, & \text{if i=j or } (i,j) \notin \epsilon \\ 1, & \text{otherwise} \end{cases}. \tag{24}$$

The associated Laplacian matrix is $L = D - A$. $D = diag(a_i)$ is the diagonal matrix of vertex degrees where a_i is the degree of vertex i, and $a_i = \sum_{j=1}^{n} a_{ij}$. The Laplacian matrix L is a symmetric semi-positive definite matrix.

One of the foremost goals of designing a communication method is to ensure highly reliable and low-latency communication for all nodes in the swarm. The communication quality of the missile swarm is easily affected by various conditions, especially when performing in a hostile environment. In other words, the swarm of missiles is constantly subject to threats from the enemy's air defense system during flight, and some missiles may get intercepted and lose communication with the rest of the swarm. Therefore, dynamic communication approaches need to ensure that when some nodes fail, the swarm can still maintain effective communication to complete a mission. Based on the previous work in network topology and swarm communication, we developed a novel adaptive cluster-based network which adaptively reconfigures the communication topology to achieve robust and fault-tolerant communication. Under the guidance of the well-known Molloy–Reed criterion [44], which is

$$k = \frac{< k^2 >}{< k >} > 2, \tag{25}$$

where k denotes the average degree of an arbitrary node which can be considered somehow as the node degrees a_i, it can be inferred that each node in the communication network should at least be connected to two other nodes. For displacement-based formation, the formation is persistent only if there exists at least a spanning tree in the communication topology. When establishing a communication connection, the formation error tends to grow as the communication distance between nodes increases, so we favor the connection method which minimizes the length of the communication chains [37]. With the minimal constraint of satisfying the above conditions, we propose setting one node (usually node 1) of the network as the cluster head while the other nodes choose another communication node according to the inter-node distances that are derived from the formation definition. To achieve this, we define the inter-agent distance matrix as

$$D_g = [d_{ij}] \in \mathbb{R}^{n\times n},\ d_{ij} = ||\lambda_{pi} - \lambda_{pj}||, \tag{26}$$

where d_{ij} is the Euclidean distance between nodes that can be calculated from the definition of the formation in Section 2 and D_g is a symmetric positive definite matrix. Given that node h is selected as the cluster head, the element of the adjacency matrix is determined as

$$a_{ij} = \begin{cases} 1, & \text{if } i \neq h, \text{and } j \in \{\gamma^*, h\}; \\ 0, & \text{otherwise;} \end{cases} \tag{27}$$

where, $\gamma^* = \arg\min_{\gamma} d_{i\gamma}$, $\gamma \in \mathbb{N}, 1 \leq \gamma \leq n$ and $\gamma \neq h$. In case there are multiple γ that minimize $d_{i\gamma}$, the one with the highest order is taken. In the proposed adaptive topology, the cluster head undertakes the task of broadcasting its own state to the other nodes in the network, or the follower nodes need to follow the cluster head. In either case, the cluster head is not constrained by nodes other than the reference target. It is also important to note that only the cluster head has access to the reference target information, which greatly reduces the communication or detection burden of the follower nodes.

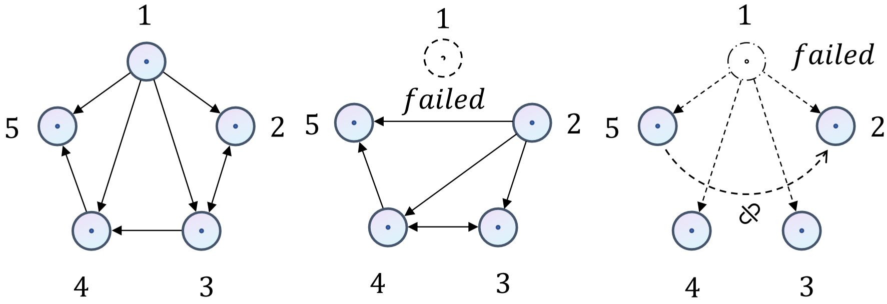
Using the above communication configuration method for a network with five nodes as an example, the obtained communication topology is shown in Figure 4a, and its adjacent matrix is

$$A^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{28}$$

Node 1 is the cluster head by default. When node 1 fails, node 2 successively inherits the head position as shown in Figure 4b, and the adjacent matrix becomes

$$A^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{29}$$

As a comparison, the traditional leader-follower topology in the case of leader node failure can be seen in Figure 4c. Due to the lack of adjacent links, effective connections among the nodes cannot be reformed. Using the proposed adaptive communication method, network connectedness can be ensured. The proof is that graph $\mathcal{G}$ is uniformly connected since, for any $t \geq t_0$, there exists a node $h \in \mathcal{V}$ such that h is the root of a spanning tree [45].



(**a**) All nodes are valid (**b**) Node 1 failed (**c**) Node 1 failed

Figure 4. Schematic diagram of topology network with five nodes: (**a**,**b**) the adaptive topology networks and (**c**) the traditional leader-follower topology.

3.5. Model-Based Constrained Policy

The NCES algorithm explores the optimal policy that maximizes fitness by continuously interacting with the environment. Consequently, the NCES algorithm and the reinforcement learning algorithm are prone to fail to achieve global convergence of the control policy, in many cases due to excessive cumulative time or inherent defects in the design of the objective function. In much of the previous literature, control policies were allowed to freely explore regions of the environment under system constraints. In recent years, the idea of constrained policy has gradually emerged [46], which states that control policies should be executed under constraints that can be imposed either by human-developed rules or by the feedback state of the system. Constrained policy, also known as safety exploration [47], has been increasingly applied to the simulation and training of realistic robotic controllers by which not only can personal safety be ensured during the training process, but global convergence can also be accelerated to some extent. Based on the above facts, we propose a model-based constrained policy. The nonlinear error dynamics of the second-order system were obtained in (6), which were a function of the system input. To apply this method, we assume that the control input of the communication recipient can be obtained by the agent. Thus, at time t, the predicted formation error at the next step can be calculated as follows:

$$\hat{e}_{ri}(t) = e_{ri}(t) + \dot{e}_{ri}(t)\tau, \tag{30}$$

where τ is the time step and $\dot{e_{ri}}$ is the derivative of the resultant error as described in Equation (6). The error deviation $\triangle e_{ri}(t) \in \mathbb{R}^3$ is

$$\triangle e_{ri}(t) = |\hat{e}_{ri}(t)| - |e_{ri}(t)|, \tag{31}$$

which is derived from the absolute values of the two error vectors. Then, the aggregate matrix is defined as $\triangle E(t) = [\ldots \triangle e_{ri}(t) \ldots] \in \mathbb{R}^{3\times n}$ for $i \in \{1,2,\ldots,n\}$. A termination indicator ST is assigned to the system so that if it is a nonzero value, the system terminates and restarts the training algorithm, and it is defined as

$$ST(t) = \begin{cases} 1, & \text{if } \min(\triangle E(t)) > \delta_s; \\ 0, & \text{otherwise,} \end{cases} \tag{32}$$

where δ_s is the threshold that measures the maximum error increment the algorithm can tolerate. It satisfies $\delta_s < (2l+1)\tau V_{max}$, where l is the number of neighbors for the agent with the maximum number of neighbors in the MAS. With a reasonable choice of δ_s, the algorithm is expected to exclude control strategies that deviate too much from the desired trajectory at an early stage. Experimental evidence shows that without such a constrained policy, the convergence speed and global optimality of the algorithm will be degraded. Applying the above adaptation techniques, the pseudo-code of the proposed algorithm used to train the missile formation controllers is shown in Algorithms 1 and 2. Since the network topology adaptation and constrained policy strategies are embedded in the process of fitness evaluation, they are not exhibited in the pseudo-code.

Algorithm 1 The distributed NCES based formation control algorithm.

```
Input: agent number $N$, population size $\eta_p$, standard deviation $\sigma$, rotation angle $\beta$, evolution path $\rho_\theta$, number of parameters $m$, iteration $t$
  Initialize
    for each agent $i = 1,\ldots,N$ do
      initialize parameter $\theta_i^{init}$
      $t \leftarrow 0$
      $\theta_i(0) \leftarrow \theta_i^{init}$
      $\eta_p(0) \leftarrow 10 + 5\ln(m)$
      $\eta_p^{min} \leftarrow \eta_p(0)$
      $\eta_p^{max} \leftarrow 4\eta_p(0)$
    end for
  while stopping criterion not met do
    $t \leftarrow t+1$
    for $k = 1,\ldots,m$ do
      for each agent $i = 1,\ldots,N$ do
        sample $\epsilon_i^k \sim N(0,\sigma^2 I)$
        $\theta_i^k \leftarrow \theta_i + \epsilon_i^k$
      end for
      evaluate fitness $f(\theta_i^k, \theta_{\mathcal{N}_i}^k)$, for $i = 1,\ldots,N$
    end for
    for each agent $i = 1,\ldots,N$ do
      calculate natural gradient:
      $g_{\theta_i} \leftarrow \frac{1}{m\sigma^2}\sum_{k=1}^{m} f(\theta_i^k, \theta_{\mathcal{N}_i}^k)\epsilon_i^k \prod_{c\in\mathcal{N}_i} p(\epsilon_c^k)$
      $\theta_i(t) \leftarrow \theta_i(t-1) + \eta_\alpha \cdot g_{\theta_i}$
    end for
    append evolution path:
    $\rho_\theta(t) \leftarrow \sum_{i=1}^{N}[\theta_i(t) - \theta_i(t-1)]^T\Sigma^{-1}[\theta_i(t) - \theta_i(t-1)]$
    AdaptPopulationSize($\rho_\theta$)
  end while
```

Algorithm 2 Adapt population size (ρ_θ).

```
if length($\rho_\theta$) $> 1$ then
    $\eta_p(t) \leftarrow \eta_p(t-1)(\beta + (1-\beta)\frac{\rho_\theta(t-1)}{\rho_\theta(t)})$
    $\eta_p(t) \leftarrow \mathrm{clip}(\max(\eta_p(t), \eta_p(t-1)), \eta_p^{\min}, \eta_p^{\max})$
else
    $\eta_p(t) \leftarrow \eta_p(0)$
end if
```

4. Simulation and Result Analysis

In this section, numerical experiments are conducted to demonstrate the effectiveness of the proposed formation control algorithm. In order to simulate the actual physical environment, the construction of the experimental scenario and the modeling of the multi-agent system were carried out in PyBullet [48]. We simulated the operation of the missile swarm in three-dimensional space at a fixed altitude, and for convenience, the trajectories were plotted as two-dimensional planar graphs. Some complex aerodynamic parameters, such as air resistance, were removed, while collision detection was preserved. The simulation time step τ was set to be 0.1 for the following experiments.

4.1. Basic Formation Control

First, the basic formation control tasks were implemented to examine the validity of the algorithm under basic situations. A swarm of five missiles entails tracking the reference target moving along a diagonal or spiral trajectory while keeping the formation geometry. The reference target or virtual leader is set to be at the center of the formation in order to achieve error-free formation control. The objective is to achieve zero tracking error as well as zero formation maintenance error for the whole time (i.e., $\int_t |e_{ri}(t)|\mathrm{d}t \to 0_3$ for $i \in \{1, \dots, N\}$). The hyperparameters of the algorithm are listed in Table 1, and it is noted that this parameter setting applied to all of the following experiments. Additionally, the system constraints are shown in Table 2. The missiles, as well as the reference target, were subjected to saturated control inputs and limited states. For the linear trajectory, the target was driven by a constant control input, while for the spiral trajectory, the control inputs of the target were $v_r = 0.65 - 0.01t$ (km/s) and $\omega_r = 0.1 + 0.01t$ (rad/s).

Table 1. Hyperparameters of the proposed algorithm.

Symbol	Description	Value
η_α	Learning rate	0.02
τ	Time step	0.1
σ	Standard deviation	0.2
β	Population size adaptation factor	0.84
K_c	Cost weight matrix	$[0.15, 0.15, 0.1]^T$

Table 2. System constraints.

Symbol	Description	Value
V_{max}	Maximum speed of both missile and reference target	0.8 km/s
V_{min}	Minimum speed of both missile and reference target	0.3 km/s
a_{lmax}	Maximum lateral acceleration	40 g
a_{vmax}	Maximum speed acceleration	30 g

For convenience, we used $|e_{ri}|$ to represent the resultant error of each agent. The trajectories of the two situations are shown in Figures 5 and 6, and the corresponding analytical results are presented in Figures 7 and 8. It can be observed that the convergent resultant error was maintained in a small interval (within 0.05) in both cases, and the results were of comparable if not better accuracy than the comparison results in [33], which kept the position error within 50 m. In addition, good synchronization of the speed and heading angles among the missiles was achieved, although the neighboring speed information was not provided.

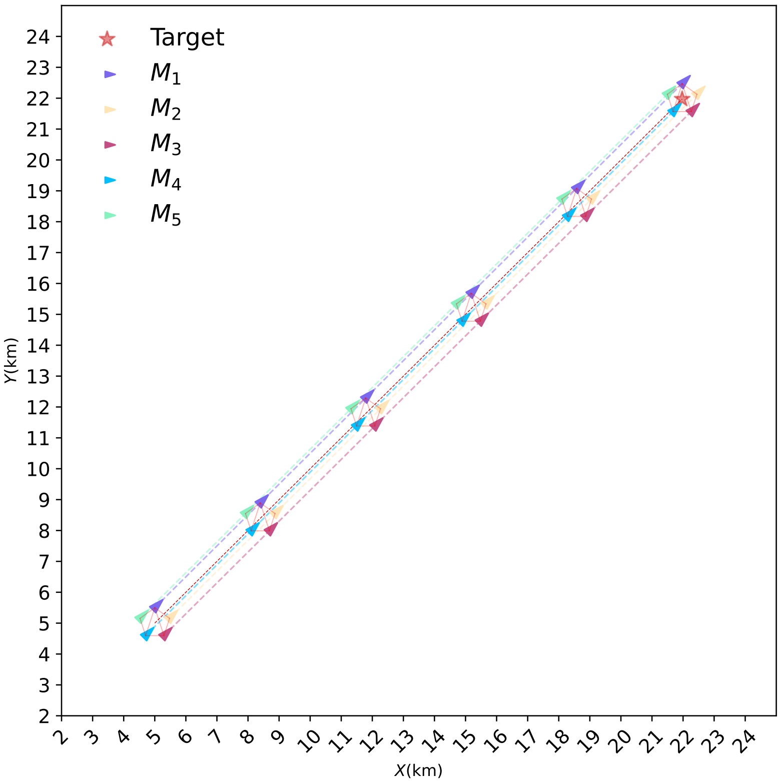


Figure 5. Trajectories of basic formation control along linear trajectory.

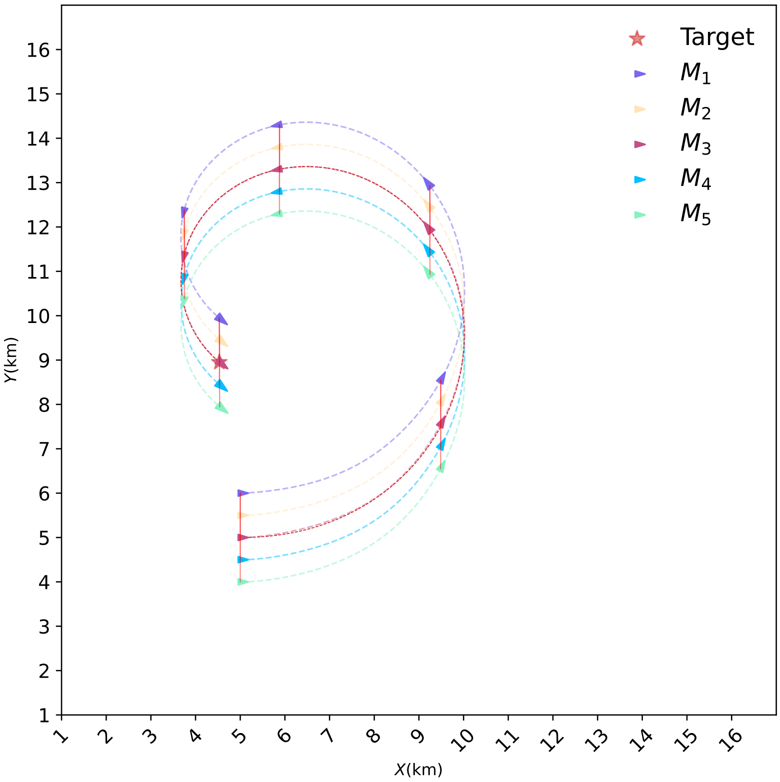
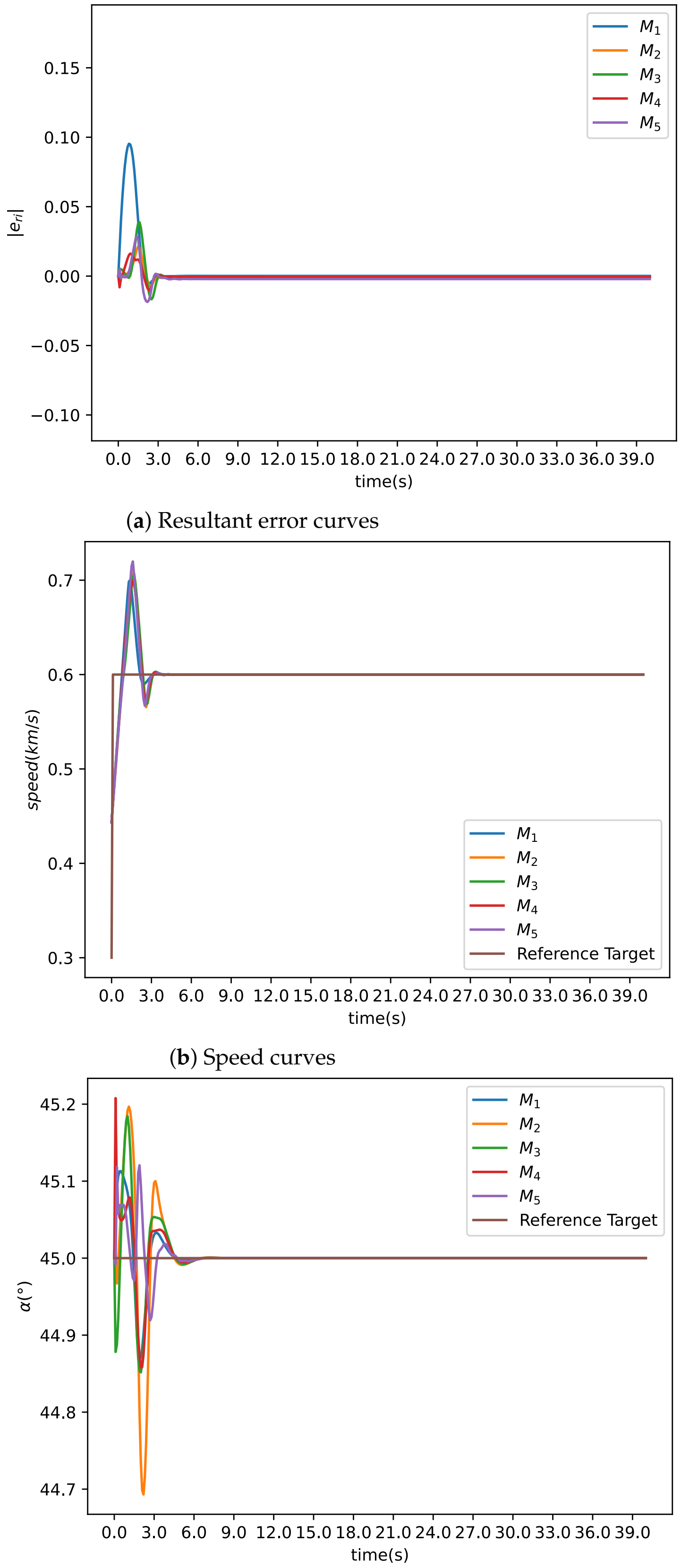


Figure 6. Trajectories of basic formation control along spiral trajectory.

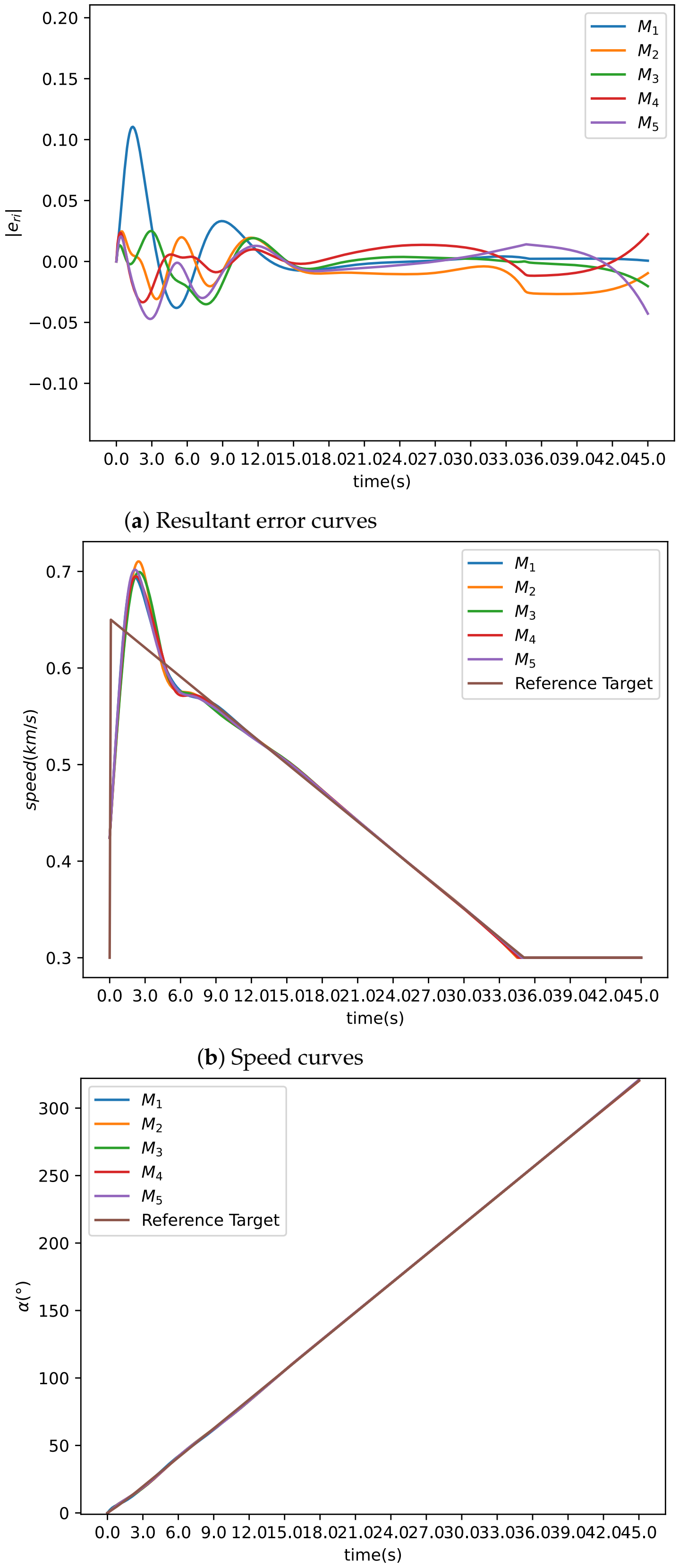


(**a**) Resultant error curves

(**b**) Speed curves

(**c**) Heading angle curves

Figure 7. Analytical results of the linear trajectory case.



(**a**) Resultant error curves

(**b**) Speed curves

(**c**) Heading angle curves

Figure 8. Analytical results of the spiral trajectory case.

4.2. Moving into Formation

Moving into formation is, however, different from cases where the formation is in the ideal geometry in the initial state. The missiles were separated from the reference target, with their positions initialized randomly in an area 4 km wide and 3 km long. The missiles need to first move to the designated formation shape and then track the reference target in a consistent motion direction, and in this case, the target moved along the y axis with a constant speed of 0.5 km. Figure 9 shows the motion trajectory of the swarm formation, and Figure 10 shows the resultant error during the flight. The results indicate that the formation was able to adjust to the desired formation shape rapidly (mostly within 10 s) and achieved high accuracy in tracking and maintaining the formation in the case of random initial distribution.

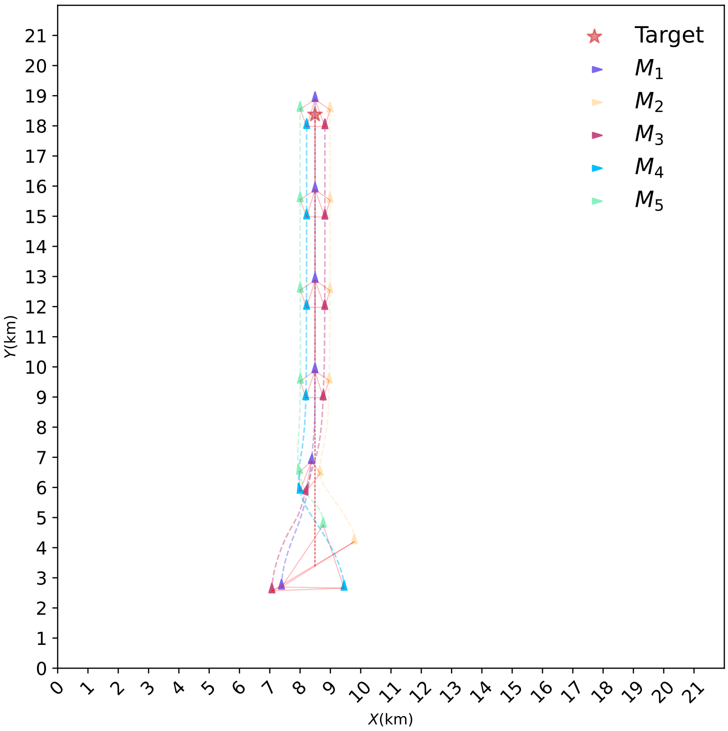


Figure 9. Trajectories in case of moving into formation.

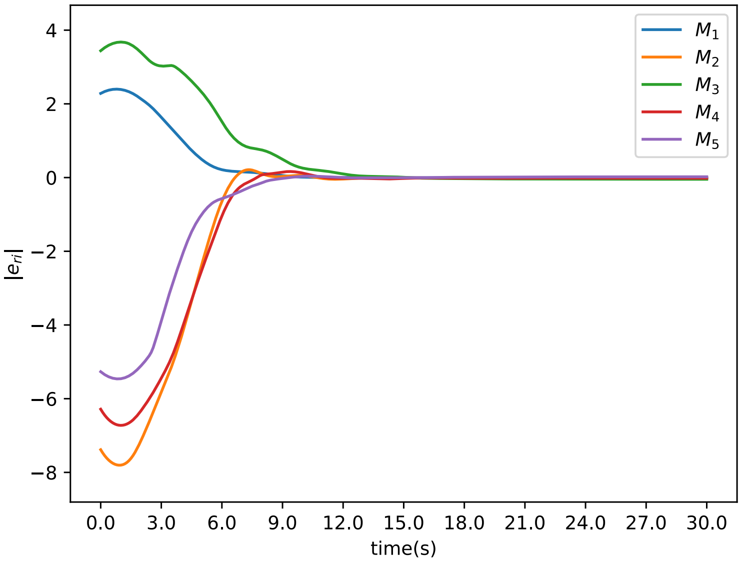


Figure 10. Resultant error curves for case of moving into formation.

4.3. Switching Formations

Furthermore, we discuss the case in which the missile swarm is supposed to switch among formation patterns in order to avoid obstacles and go through narrow spaces. To achieve this transformation, an additional input node TS was appended to the input layer of the policy network as depicted in Figure 3. It was assumed that whenever the obstacle was detected by the leader missile or the literal cluster head, the missile then sendt a signal to all connected missiles to perform a formation switch of $\lambda^P \to \lambda^L$. Although formation geometries are predefined by the controller, the shapes can be controlled by parameters l_f and β, which can be changed during flight.

In this scenario, two walls were placed in the trajectory of the formation flight to create narrow spaces, and the missiles needed to pass through the obstacles by changing the shape or size of the formation. We implemented an event-based formation-switching strategy, in which the nodes in the formation were able to detect obstacles within a certain range d_c and send a formation-switching signal to all other nodes if an obstacle were to be detected. Similarly, after crossing the obstacle and reaching a safe distance, a formation recovery signal would be sent to restore the original formation.

To pass through the obstacles, we considered changing both the formation pattern and formation size, and the time-varying definitions of the formation in both cases were as follows.

The resulting trajectories are shown in Figures 11 and 12. It can be observed that when obstacles were detected, the swarm could swiftly adjust by changing the formation pattern or size and then recover the formation quickly after going through the narrow space:

$$\begin{aligned} \lambda^1(t) &= \begin{cases} \lambda^P_{(0,0.5)}, & \text{if } d_c \leqslant n \cdot l_f \\ \lambda^L_{(-\pi/4,0.5)}, & \text{otherwise.} \end{cases}, \\ \lambda^2(t) &= \begin{cases} \lambda^P_{(0,0.5)}, & \text{if } d_c \leqslant n \cdot l_f \\ \lambda^P_{(0,0.2)}, & \text{otherwise.} \end{cases}, \end{aligned} \tag{33}$$

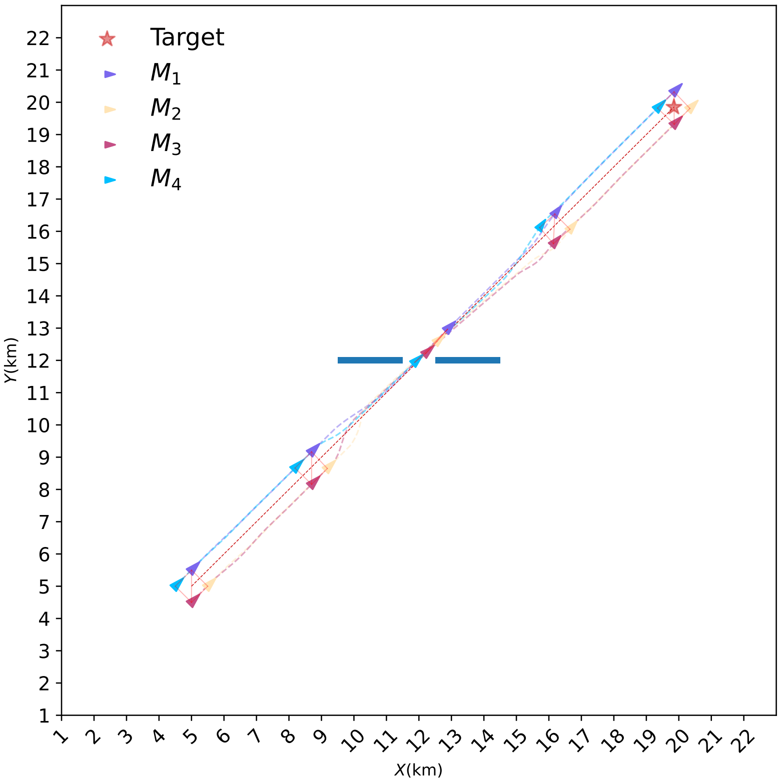


Figure 11. Trajectories for case of switching formation type.

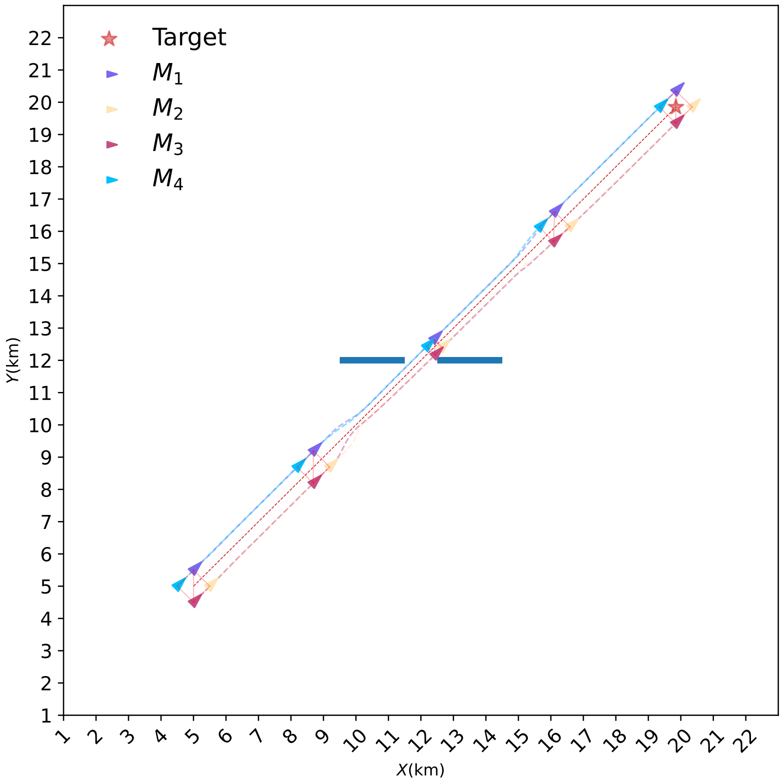


Figure 12. Trajectories for case of switching formation size.

4.4. Formation Control under Node Failure

An experiment was conducted in this scenario to verify the effectiveness of the proposed algorithm under node failure, where a swarm with six nodes was designed to pursue the reference target in a regular polygon formation and the reference target moved in a sinusoidal fashion. At $t = 20$ s during pursuit, cluster head node 1 and cluster member node 4 suffered attacks and would disconnect from the other nodes in the swarm, and the remaining nodes of the swarm needed to maintain their original formation and complete the formation task.

From the results in Figures 13 and 14, it can be observed that the swarm selected node 2 as the cluster head to reorganize the communication topology after node failure and successfully maintained the original formation shape after a short fluctuation. Finally, the robustness of the proposed formation control algorithm against node failure was validated. To investigate the effect of policy constraints on the control performance, we compared the results of the NCES-based formation control method that imposed policy constraints with the one without policy constraints. In some cases, such as node failure and switching formation, there was a certain chance that the algorithm would not converge. Moreover, in all cases, the convergence rate was generally improved by more than 20% with policy constraint rather than without it. Therefore, policy constraint is essential for the training of the NCES-based neural network controller.

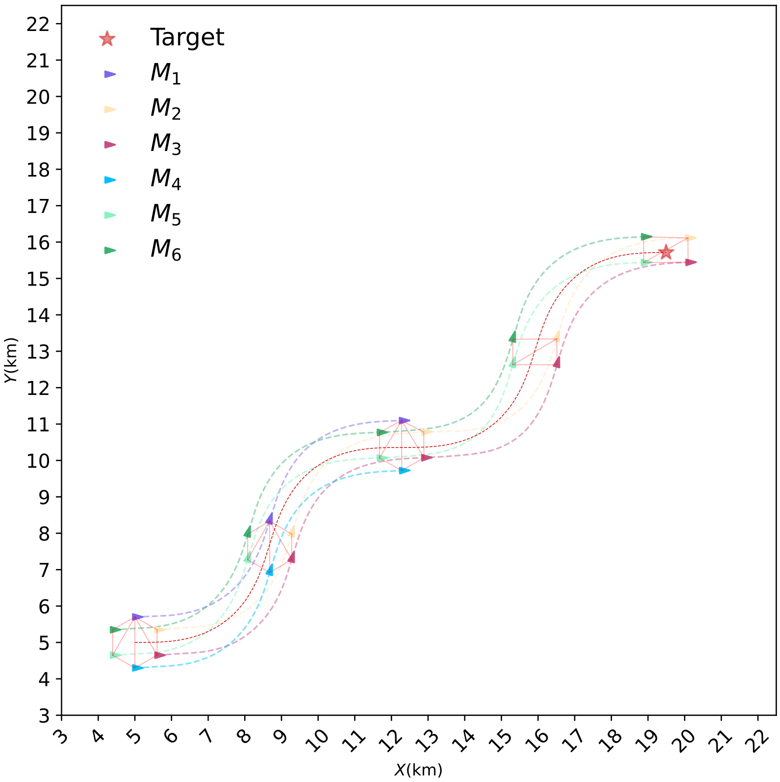


Figure 13. Trajectories of the node failure case.

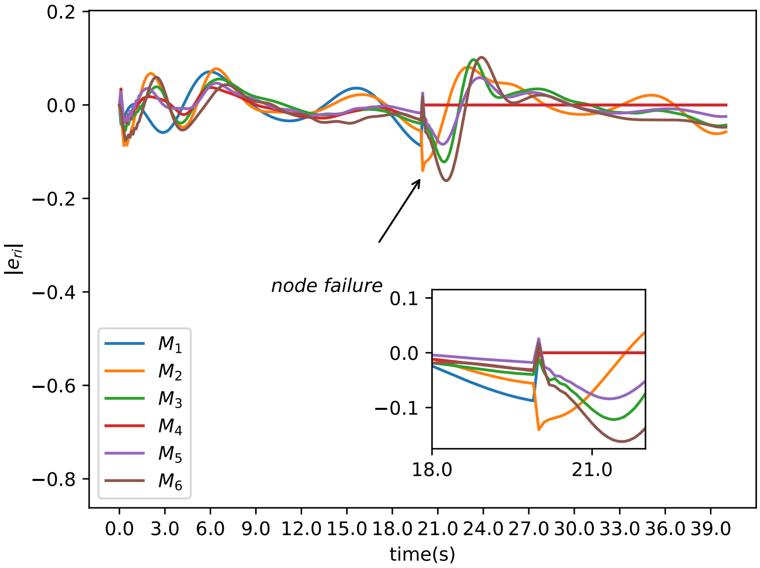


Figure 14. Resultant error curves of the node failure case.

5. Conclusions

This paper proposes a novel distributed NCES-based formation control algorithm for a second-order multi-missile system using neural networks. The algorithm minimizes the formation shape error and tracking error by training the optimal network controller through iterative learning, and it was combined with a policy constraint approach to enhance the stability of the algorithm. Additionally, we designed an adaptive topology scheme for the node failure situation which can achieve stable connections at a low communication cost. We also proposed a stable population adaptation method based on the evolution path, which further improved the performance of the algorithm and alleviated local optimum issues. The numerical experiments demonstrated that the proposed formation control algorithm is capable of accomplishing tasks, such as formation maintenance, tracking of reference trajectories, formation transformation and obstacle avoidance. This indicates that the proposed algorithm has a high level of accuracy and robustness to cope with situations such as random initial positions and node failures. This paper also discussed the parametric definition of the formation geometry as a background supplement to the field.

Due to the characteristics of constrained co-evolution, the algorithm is expected to be applied to predict the amount of available renewable energy by environmental indicators, such as gas emissions or wind, estimate the water absorption by crops in controlled agriculture and even provide new ways to solve quantum many-body problems [49–51]. Future works are recommended to investigate online evolutionary algorithms to solve the problem of unknown or stochastic system models and to apply the proposed algorithm to real-world experiments.

Author Contributions: Conceptualization, J.C. and X.L.; methodology, J.C.; software, J.C. and J.L.; formal analysis, J.C., X.L. and J.L.; writing—original draft preparation, J.C., X.L. and Y.Z.; writing—review and editing, J.C., X.L., Y.Z. and J.L.; supervision, X.L. and Y.Z.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Guangzhou Science and Technology Project (Grant No. 202102010403).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lim, H.; Yeonsik, K.; Kim, J.; Kim, C. Formation Control of Leader Following Unmanned Ground Vehicles Using Nonlinear Model Predictive Control. In Proceedings of the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, 14–17 July 2009; pp. 945–950. [CrossRef]
2. Shi, P.; Yan, B. A Survey on Intelligent Control for Multiagent Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 161–175. [CrossRef]
3. Reynolds, C.W. Flocks, Herds, and Schools: A Distributed Behavioral Model. In *SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*; Association for Computing Machinery: New York, NY, USA, 1987; p. 10.
4. Cui, N.; Wei, C.; Guo, J.; Zhao, B. Research on Missile Formation Control System. In Proceedings of the 2009 International Conference on Mechatronics and Automation, Changchun, China, 9–12 August 2009; pp. 4197–4202. [CrossRef]
5. Ren, W.; Cao, Y. *Distributed Coordination of Multi-Agent Networks*; Communications and Control Engineering; Springer: London, UK, 2011. [CrossRef]
6. Oh, K.K.; Park, M.C.; Ahn, H.S. A Survey of Multi-Agent Formation Control. *Automatica* **2015**, *53*, 424–440. [CrossRef]
7. Marshall, J.; Broucke, M.; Francis, B. Formations of Vehicles in Cyclic Pursuit. *IEEE Trans. Autom. Control* **2004**, *49*, 1963–1974. [CrossRef]
8. Asaamoning, G.; Mendes, P.; Rosário, D.; Cerqueira, E. Drone Swarms as Networked Control Systems by Integration of Networking and Computing. *Sensors* **2021**, *21*, 2642. [CrossRef] [PubMed]
9. Shrit, O.; Martin, S.; Alagha, K.; Pujolle, G. A New Approach to Realize Drone Swarm Using Ad-Hoc Network. In Proceedings of the 2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Budva, Montenegro, 28–30 June 2017; pp. 1–5. [CrossRef]
10. Chen, W.; Liu, J.; Guo, H.; Kato, N. Toward Robust and Intelligent Drone Swarm: Challenges and Future Directions. *IEEE Netw.* **2020**, *34*, 278–283. [CrossRef]
11. Slotine, J.J.E.; Li, W. *Applied Nonlinear Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1991.
12. Wu, Y. A Survey on Population-Based Meta-Heuristic Algorithms for Motion Planning of Aircraft. *Swarm Evol. Comput.* **2021**, *62*, 100844. [CrossRef]
13. Liu, S.; Huang, F.; Yan, B.; Zhang, T.; Liu, R.; Liu, W. Optimal Design of Multimissile Formation Based on an Adaptive SA-PSO Algorithm. *Aerospace* **2021**, *9*, 21. [CrossRef]
14. Lee, S.M.; Kim, H.; Myung, H.; Yao, X. Cooperative Coevolutionary Algorithm-Based Model Predictive Control Guaranteeing Stability of Multirobot Formation. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 37–51. [CrossRef]
15. Pessin, G.; Osório, F.; Hata, A.Y.; Wolf, D.F. Intelligent Control and Evolutionary Strategies Applied to Multirobotic Systems. In Proceedings of the 2010 IEEE International Conference on Industrial Technology, Via del Mar, Chile, 14–17 March 2010; pp. 1427–1432. [CrossRef]
16. Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef]

17. Li, X.; Zhu, D. An Adaptive SOM Neural Network Method for Distributed Formation Control of a Group of AUVs. *IEEE Trans. Ind. Electron.* **2018**, *65*, 8260–8270. [CrossRef]
18. Barreto, G.; Araujo, A. Identification and Control of Dynamical Systems Using the Self-Organizing Map. *IEEE Trans. Neural Networks* **2004**, *15*, 1244–1259. [CrossRef] [PubMed]
19. Zhang, H.; Jiang, H.; Luo, Y.; Xiao, G. Data-Driven Optimal Consensus Control for Discrete-Time Multi-Agent Systems With Unknown Dynamics Using Reinforcement Learning Method. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4091–4100. [CrossRef]
20. Tanaka, T.; Moriya, T.; Shinozaki, T.; Watanabe, S.; Hori, T.; Duh, K. Automated Structure Discovery and Parameter Tuning of Neural Network Language Model Based on Evolution Strategy. In Proceedings of the 2016 IEEE Spoken Language Technology Workshop (SLT), San Diego, CA, USA, 13–16 December 2016; pp. 665–671. [CrossRef]
21. Vidnerová, P.; Neruda, R. Evolution Strategies for Deep Neural Network Models Design. *Ceur Workshop Proc.* **2017**, *1885*, 159–166.
22. Wang, X.; Yadav, V.; Balakrishnan, S.N. Cooperative UAV Formation Flying With Obstacle/Collision Avoidance. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 672–679. [CrossRef]
23. Vasile, M.; Minisci, E.; Locatelli, M. Analysis of Some Global Optimization Algorithms for Space Trajectory Design. *J. Spacecr. Rocket.* **2010**, *47*, 334–344. [CrossRef]
24. Hughes, E. Multi-Objective Evolutionary Guidance for Swarms. In Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02 (Cat. No.02TH8600), Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1127–1132. [CrossRef]
25. Lan, X.; Wu, Z.; Xu, W.; Liu, G. Adaptive-Neural-Network-Based Shape Control for a Swarm of Robots. *Complexity* **2018**, *2018*, 8382702. [CrossRef]
26. Fei, Y.; Shi, P.; Lim, C.C. Neural Network Adaptive Dynamic Sliding Mode Formation Control of Multi-Agent Systems. *Int. J. Syst. Sci.* **2020**, *51*, 2025–2040. [CrossRef]
27. Ni, J.; Shi, P. Adaptive Neural Network Fixed-Time Leader–Follower Consensus for Multiagent Systems With Constraints and Disturbances. *IEEE Trans. Cybern.* **2021**, *51*, 1835–1848. [CrossRef]
28. Yang, S.; Bai, W.; Li, T.; Shi, Q.; Yang, Y.; Wu, Y.; Chen, C.L.P. Neural-Network-Based Formation Control with Collision, Obstacle Avoidance and Connectivity Maintenance for a Class of Second-Order Nonlinear Multi-Agent Systems. *Neurocomputing* **2021**, *439*, 243–255. [CrossRef]
29. Lan, X.; Liu, Y.; Zhao, Z. Cooperative Control for Swarming Systems Based on Reinforcement Learning in Unknown Dynamic Environment. *Neurocomputing* **2020**, *410*, 410–418. [CrossRef]
30. Chen, J.; Lan, X.; Zhao, Z.; Zou, T. Cooperative Guidance of Multiple Missiles: A Hybrid Co-Evolutionary Approach. *arXiv* **2022**, arXiv:2208.07156.
31. Xingguang, X.; Zhenyan, W.; Zhang, R.; Shusheng, L. Time-Varying Fault-Tolerant Formation Tracking Based Cooperative Control and Guidance for Multiple Cruise Missile Systems under Actuator Failures and Directed Topologies. *J. Syst. Eng. Electron.* **2019**, *30*, 587–600. [CrossRef]
32. Wei, C.; Shen, Y.; Ma, X.; Guo, J.; Cui, N. Optimal Formation Keeping Control in Missile Cooperative Engagement. *Aircr. Eng. Aerosp. Technol.* **2012**, *84*, 376–389. [CrossRef]
33. Zhang, Z.; Zhang, K.; Han, Z. A Novel Cooperative Control System of Multi-Missile Formation Under Uncontrollable Speed. *IEEE Access* **2021**, *9*, 9753–9770. [CrossRef]
34. Aicardi, M.; Casalino, G.; Bicchi, A.; Balestrino, A. Closed Loop Steering of Unicycle like Vehicles via Lyapunov Techniques. *IEEE Robot. Autom. Mag.* **1995**, *2*, 27–35. [CrossRef]
35. Dinesh, K.; Vijaychandra, J.; SeshaSai, B.; Vedaprakash, K.; Srinivasa, R.K. A Review on Cascaded Linear Quadratic Regulator Control of Roll Autopilot Missile. 2021. Available online: https://doi.org/10.2139/ssrn.3768344 (accessed on 1 January 2020).
36. Ren, W. Consensus Strategies for Cooperative Control of Vehicle Formations. *IET Control Theory Appl.* **2007**, *1*, 505–512. [CrossRef]
37. Das, A.; Fierro, R.; Kumar, V.; Ostrowski, J.; Spletzer, J.; Taylor, C. A Vision-Based Formation Control Framework. *IEEE Trans. Robot. Autom.* **2002**, *18*, 813–825. [CrossRef]
38. Lewis, M.A.; Tan, K.H. High Precision Formation Control of Mobile Robots Using Virtual Structures. *Auton. Robot.* **1997**, *4*, 387–403. [CrossRef]
39. Sefrioui, M.; Perlaux, J. Nash Genetic Algorithms: Examples and Applications. In Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), La Jolla, CA, USA, 16–19 July 2000; Volume 1, pp. 509–516. [CrossRef]
40. Nishida, K.; Akimoto, Y. PSA-CMA-ES: CMA-ES with Population Size Adaptation. In Proceedings of the Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; pp. 865–872. [CrossRef]
41. Nomura, M.; Ono, I. Towards a Principled Learning Rate Adaptation for Natural Evolution Strategies. In *Applications of Evolutionary Computation, Proceedings of the 25th European Conference, EvoApplications 2022, Held as Part of EvoStar 2022, Madrid, Spain, 20–22 April 2022*; Springer: Cham, Switzerland, 2022.
42. Glasmachers, T.; Schaul, T.; Yi, S.; Wierstra, D.; Schmidhuber, J. Exponential Natural Evolution Strategies. In Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO '10), Portland, OR, USA, 7–11 July 2010; p. 393. [CrossRef]
43. Desai, J.; Ostrowski, J.; Kumar, V. Controlling Formations of Multiple Mobile Robots. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Leuven, Belgium, 20 May 1998; Volume 4, pp. 2864–2869. [CrossRef]

44. Barabási, A.L. Network Science Network Robustness. p. 54. Available online: http://networksciencebook.com/chapter/8 (accessed on 1 January 2020).
45. Lin, Z.; Francis, B.; Maggiore, M. State Agreement for Continuous-Time Coupled Nonlinear Systems. *SIAM J. Control Optim.* **2007**, *46*, 288–307. [CrossRef]
46. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained Policy Optimization. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 22–31.
47. Ray, A.; Achiam, J.; Amodei, D. Benchmarking Safe Exploration in Deep Reinforcement Learning. *arXiv* **2019**, arXiv:1910.01708.
48. Coumans, E.; Bai, Y. Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning. 2016–2019. Available online: http://pybullet.org (accessed on 1 January 2020).
49. Mason, K.; Duggan, J.; Howley, E. Forecasting Energy Demand, Wind Generation and Carbon Dioxide Emissions in Ireland Using Evolutionary Neural Networks. *Energy* **2018**, *155*, 705–720. [CrossRef]
50. Concepcion II, R.; Lauguico, S.; Almero, V.J.; Dadios, E.; Bandala, A.; Sybingco, E. Lettuce Leaf Water Stress Estimation Based on Thermo-Visible Signatures Using Recurrent Neural Network Optimized by Evolutionary Strategy. In Proceedings of the 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC), Kuching, Malaysia, 1–3 December 2020; pp. 1–6. [CrossRef]
51. Chen, A.; Choo, K.; Astrakhantsev, N.; Neupert, T. Neural Network Evolution Strategy for Solving Quantum Sign Structures. *Phys. Rev. Res.* **2022**, *4*, L022026. [CrossRef]