Formation control with connectivity assurance for missile swarm: a natural co-evolutionary strategy approach

Chen Junda

*School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China

Abstract

Formation control problem is one of the most concerned topics within the realm of swarm intelligence, which is usually solved by conventional mathematical approaches. In this paper, however, we present 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 heterogeneous reference target, and exponential error function is made to be the objective function such that the swarm converge to optimal equilibrium states satisfying certain formation requirements. Focusing on the issue of local optimum and unstable evolution, we incorporate a novel model-based policy constraint and a population adaptation strategies that greatly alleviates the performance degradation. With application of the Molloy-Reed criterion in the field of network communication, we developed an adaptive topology method that assure the connectivity under node failure and its effectiveness are validated both theoretically and experimentally. Experimental results valid the effectiveness of the proposed formation control approach. More significantly, we showed that it is feasible to treat generic formation control problem as Markov Decision Process (MDP) and solve it through iterative learning.

Keywords: Formation control, natural co-evolutionary strategy, connectivity, multi-agent system

1. Introduction

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

In this paper, we focus on the formation control of missile swarm. Similar to common formation control problems, the formation control for missile swarm covers formation initialization, contraction, expansion and reconfiguration [4], corresponding with formation producing and formation tracking problems according to [5]. Based on the sensing capability and interaction level, the current dominant research in [6] classifies formation control methods into position-, displacement-, and distance-based control. The main difference between the different control methods is the ability to sense relative or absolute state information. For this reason, we propose to classify angle-based control as a special 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 huge realistic application value outside of cyclic pursuit problem [7].

In practice, when dealing with large-scale swarm systems, control methods that the system can adopt are usually subject to limited communication bandwidth, communication quality and other interference issues, so more flexible communication means that are tolerant of communication failures are necessary for achieving robust information transmission. However, traditional communication methods based on fixed communication topology are difficult to cope with unexpected situations and provide continuous and reliable communication. For developing more robust intelligent swarm communication algorithm, two newly developed adaptive communication methods in the field of networked control system are known as ad-hoc based network and cluster based network [8]. The author in [9] 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 algorithm is further employed in flying ad-hoc network (FANET), which is a distributed and self-organizing communication framework. Under the same ad-hoc framework, a new approach is developed utilizing received signal strength instead of localization facilitates to measure individual distances [10]. Considering that distance is usually closely related to communication quality, and that the perception of distance does not require an additional communication bandwidth, it is highly feasible to regard distance as a major factor in configuring the communication topology. In this work, we develop an adaptive topology communication method based on minimizing the communication distance and overcome the problem of head failure in cluster-based communication method to ensure the 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 hardly 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 are introduced as more flexible alternatives for operating in unstructured or dynamic environments surrounded with 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 but not limited to formation optimal configuration, motion planning, etc. A comprehensive survey on the development of such algorithms for aircraft motion planning problem is presented in [12]. In [13] Liu investigated a PSO-based algorithm in the optimal design of missile formation configuration. In [14], Seung-Mok propose a cooperative co-evolution PSO algorithm based on the traditional PSO-based MPC, which optimize in a distributed way and improves the speed and performance of the original algorithm. Another population-based metaheuristic known as the genetic algorithm (GA) is employed to evolve positioning strategy of a formation controlled multi-robot systems [15]. A gradient-descent-based reinforcement learning method utilizing actor-critic framework is proposed for optimal consensus control of multi-agent systems. Existing research on consensus control provides another perspective on solving the formation problem, since 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 to search 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 weak computational performance and low cost despite of the majority of the computational cost spent at 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 SOM-based neural network approach for motion planning in the formation control problem of a multi-AUV system. By the way, the SOM is a sort of unsupervised neural network that usually serves for specialized purpose, like, competitive learning and task assignment [18]. An an iterative algorithm to find the most energy-sufficient motion position for each AUV. The algorithm is an iterative algorithm that is used to search for the most energy-efficient motion path of the agent. Since the algorithm takes all individuals’ states as input and the target path as output, it can be regarded as centralized method which is not desirable for the practical deployment requirement.

The general idea of the heuristic algorithm using neural network controller is similar to other algorithms such as PSO, ACO, etc., which are designed to find the optimal solution of the cost function under the constraints by iterative search, but the most essential difference between the former and the other algorithms is that the former optimizes the neural network weights and the final optimized controller can be deployed directly, while the latter optimizes directly in the solution space by single-step. Both methods have their advantages and disadvantages. Since the latter does not need pre-training, it relies too much on single-step computation and it is difficult to guarantee the speed requirement for real-time operation. Therefore neural network controller as an adaptive and learnable controller is considered as a promising means of intelligent control in the future [19].

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 [20], a modified Grossberg neural network (GNN) is used to generate the shortest path to avoid the obstacle and reach the target point. Lan adopts an RBFNN to estimate the system disturbance in [21], which leads to enhanced robustness and adaptability of swarm formation control based on the artificial potential field method. Additional researches that implement adaptive neural networks to estimate the uncertain and nonlinear dynamics of the system can be viewed in [22][23][24]. Further, in [25], Lan use reinforcement learning theory to train a neural network controller that can be applied to the distributed control of swarm system 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 inferior to traditional methods, and are relatively more flexible and applicable.

To investigate the applicability of neural network controllers within the framework of formation control problem, in this paper we propose to adopt a multi-layer perceptron (MLP) neural network controller using a cooperative NES-based approach and confirm that the formation control problem can also be solved as a multi-agent collaborative task by the neural network controller.

The foremost motivation of this paper is to develop a metaheuristic evolutionary computational approach to solve the formation control problem for MASs, in the meanwhile exploring the usage of neural networks (NNs) controller. We use a recently proposed NCES algorithm [26] to realize such of a vision and apply it to the control of a second-order multi-missile system. We incorporate a policy constraint approach to enhance the stability of the algorithm so as to optimize the objective function to find the Nash equilibrium strategy. The proposed algorithm is more flexible compared with conventional approaches [4][27][28][29]. In addition, an adaptive topology scheme is designed to solve the common node failure problem in formation control, and this method can achieve stable communication connections at relative low communication cost. To further improve the performance of the algorithm and mitigate the local optimum issue, we also propose a stable population adaptation method. Emulation experiments show that the proposed formation control algorithm can handle tasks such as formation maintenance, reference trajectory tracking and formation switching in the face of obstacles with high accuracy, and is immune to disturbances.
such as random initial position and node failure.

The remainder of this work continues as follows. In Section 2, the non-linear multi-missile system is modeled and the displacement-based formation control problem is formulated, including the specification of formation patterns, while the background of NCES algorithm is also introduced in this section. In Section 3, an distributed NCES-based formation control algorithm using a neural network controller is proposed, and in Section 4, experiments are conducted for a variety of scenarios and results are derived. Finally, conclusions are presented in Section 5.

2. Preliminaries and Problem Formulation

2.1. System modeling of swarm of cruise missiles

This paper focuses on the formation control in two-dimensional space, i.e., the OXY planar space of the inertial coordinate frame. As shown in Fig. 1, the swarm consists of multiple missiles, each missile can be treated as a point mass. In the simplified dynamics model, we do not consider aerodynamic factors and the effect of the missile’s own inertia tensor. Consider a total of N missiles, for missile $i$ denoted by $M_i$, $V_m$, $\alpha_m$, $x_m$ and $y_m$ are used to represent the velocity and heading angle, and its coordinates in the global coordinate frame, respectively. Let $X_i = [x_m, y_m, \alpha_m]^T$ represents the state vector of the missile, which are not necessarily required hereinafter, and $X_r = [x_r, y_r, \alpha_r]^T$ represents the state of the reference target. For brevity, let $\dot{X}_i = [X_i^T, V_m]^T$ be the extended state vector with speed inserted. As shown in the partial enlargement of the actuation decomposition, the missiles have two independent inputs $a_{vi}, a_{li}$ which are the acceleration commands both along and perpendicular to the direction of the velocity, respectively. The second-order system dynamic of the $i$th missile can be expressed as

$$\begin{align*}
\dot{X}_i &= \Psi(V_m, \alpha_m) + \Phi(V_m)u_i \\
X_i &= AX_i,
\end{align*}$$

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

$$\begin{align*}
\Psi(V_m, \alpha_m) &= \begin{bmatrix} V_m \cos \alpha_m \\ V_m \sin \alpha_m \end{bmatrix}, \\
\Phi(V_m) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.
\end{align*}$$

(2)

$A = [I_3, 0_{3,1}]$.

where $I_n, 0_{m,n}$ denote the identity and zero matrix, and $j$ represents the neighboring missile. Note that system Eq. (1) is similar to the system discussed in [30], which is often called unicycle-like vehicle. In this paper, the missile is a nonholonomic system with controllable speed and capable of maneuvering with limited actuation and dynamic constraints. We assume that under certain communication relationship, the relative state $\xi_{ij}$ can be sensed by missile $i$. In addition we make the following assumptions about the system:

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

Assumption 2. It is assumed that missile controller applies the Cascade control [31] structure, such that its altitude is automatically controlled by the inner-loop control, which is a stable closed-loop. We focus on the design for the missiles’ trust and lateral acceleration of the outer control loop. This divide and conquer approach is widely used in many of the literature [32] to simplify the problem in order to verify the effectiveness of the control method, and is also valid for three dimensional dynamic models.

2.2. Formation control under displacement-based framework

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 are expressed in the coordinate system, such that the center of the formation is the origin which should satisfies

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

(3)

Let $N_i$ denotes the set of neighboring missiles of the $i$th missile. In the displacement-based framework, each missile has to align its own coordinate system with the global coordinate system and to be able to sense the relative positions and orientations of its neighbors, whereas its position in the global coordinate system is not necessarily required. Suppose that the reference target should be kept in the formation center, then the tracking error of the $i$th missile is

$$e_i = \sum_{j \in N_i} [(P_i - P_j) - (X_i - X_j)] + \zeta(X_r + P_i - X_i),$$

(4)
where $P_t = [\lambda_t, \lambda_0, 0]^T$ and $z_t = 1$ if the state of the target is available to the $j^{th}$ missile, else $z_t = 0$, $e_t \in \mathbb{R}^3$. 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 individual missile’s heading angles on the error vector to the relative coordinate system that are aligned with the missile’s orientation, we define the rotated error vector as $e_{ri} = R_3(-\alpha) e_t$, where $R_3(-\alpha) \in \mathbb{R}^{3 \times 3}$ is the three dimensional rotation matrix defined as

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

Let’s $e_{ri} = [e_{rix}, e_{riy}, e_{riw}]^T$. Considering system Eq. (1), by differentiating $e_{ri}$ with respect to time and considering the discretization of system loop, we obtain the following error dynamic

$$e_{ri} = G_i u_t + \sum_{j \in N_i} F_{ij} u_j + D_i u_t + H_i, \tag{6}$$

where

$$G_i = \begin{bmatrix}
-\tau(L_i + \xi) & e_{ri}/v_i & 0 \\
0 & -e_{ri}/v_i & \xi \\
0 & -(L_i + \xi)/v_i & 0
\end{bmatrix}$$

$$F_{ij} = \begin{bmatrix}
\tau \cos \theta_{ji} & 0 \\
\tau \sin \theta_{ji} & 0 \\
0 & 1/v_j
\end{bmatrix},$$

$$D_i = \begin{bmatrix}
\xi \cos \hat{\theta}_i & 0 \\
\xi \sin \hat{\theta}_i & 0 \\
0 & \xi
\end{bmatrix},$$

$$H_i = \begin{bmatrix}
-(L_i + \xi)/v_i \sum_{j \in N_i} \cos \theta_{ji} v_j \\
0
\end{bmatrix} - \frac{\sum_{j \in N_i} \sin \theta_{ji} v_j}{2} \sum_{j \in N_i} \cos \theta_{ji} v_j$$

$L_i$ is the number of missiles belonging to the set of neighbors $N_i$, and $\tau$ is the simulation time step size. Furthermore, let $u_t = [a_t, a_0]^T$ and $u_t = [v_t, w_t]^T$ denote the vector of control command, and $\tau$ be the control loop constant which is selected by the controller.

Generally, the objective metric at time $t$ can be defined by function of the error vector, which is

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

where $K_C = \text{diag}[k_1, k_3, k_3]$ is the symmetric positive defined matrix with positive weights. By choosing $K_C$ properly, the swarm balances between keeping the formation shape or tracking the reference target as a priority. The optimal design of the formation controller $u_t$ can be formulated as a nonlinear optimization problem

$$u_t = \arg \max_{u_t} \int_0^T J_t(e_{ri}(t))dt, \tag{9}$$

subjected to

$$u_{\min} \leq u_t \leq u_{\max},$$

$$v_{\min} \leq v_t \leq v_{\max}. \tag{10}$$

in which $T$ denotes the total flight time, and $u_{\min}, u_{\max}, v_{\min}, v_{\max}$ represents the system constraints.

In this paper we discuss the parametric definition of the formation geometry, which is rarely mentioned in the other works. We define two formation patterns, the regular polygon and the straight-line formation which are shown in Fig. [2] and denote them by $\lambda_{(\alpha_p, l_p)}^P$ and $\lambda_{(\alpha_p, l_p)}^L$, $l_p$ and $\alpha_p$ are the parameters that control the size and rotation of the formation, specifically we have

$$\lambda_{(\alpha_p, l_p)}^P = \left[ R_3(\alpha_p)[l_p \cos \left(\frac{2\pi}{N}(i - 1)\right), l_p \sin \left(\frac{2\pi}{N}(i - 1)\right)]^T, i \in [1, 2, ..., N]\right],$$

$$\lambda_{(\alpha_p, l_p)}^L = \left[ R_2(\alpha_p)[l_p \frac{N - 2i + 1}{2}, 0]^T : i \in [1, 2, ..., N]\right].$$

and $R_2(\alpha)$ is a two-dimensional rotation matrix similar to Eq. (5), which is

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

**Figure 2:** Two types of formation patterns

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, so that the sum of its $x$ and $y$ coordinates is 0, which satisfies Eq. (3). Similarly when the number of nodes of a regular polygon is even, we can also get this conclusion. Still when the number of nodes is odd, we
can prove that it still satisfies this prerequisite, and the following is a simple proof.

Suppose that the regular polygon formation defined in Eq. (11) consists of \( N \in [2n + 1 : n \in \mathbb{Z}] \) nodes, the Cartesian coordinates of each node in the formation are denoted by \( P_i = [\lambda_i, \lambda_j]^T \), so that the sum of all coordinates is

\[
\sum_{i=1}^{N} P_i = \begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix},
\]

Suppose the angle between the vector from the coordinate origin to an arbitrary node and the positive direction of the \( Y \)-axis is \( \rho \). We can obtain the following sum of coordinates by multiplying with the rotation matrix to rotate node one to the \( Y \)-axis

\[
R_2(\rho) \sum_{i=1}^{N} P_i = \begin{bmatrix} \xi_i \cos \rho - \eta_i \sin \rho \\ \xi_i \sin \rho + \eta_i \cos \rho \end{bmatrix}.
\]

At this configuration, since the formation is symmetric about the \( Y \)-axis, the sum of its x-coordinates can be obtained as 0, so we have

\[
\xi_i \cos \rho - \eta_i \sin \rho = 0,
\]

thus \( \xi_i = \xi_i = 0 \), Eq. (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, Square patterns [1]. Moreover, as discussed in the following Section 3, this formation paradigm can also generate some asymmetric patterns, such as Wedge, Crescent patterns [21], etc., by deleting nodes appropriately.

2.3. Natural Co-evolutionary strategy for MAS

With limited sensing capability, the MAS system can be modeled as multi-agent partial observable Markov decision process (POMDP), which is an extension on the basis of the fundamental Markov decision process (MDP). Such process can be expressed as

\[
P(S \times A) \rightarrow S',
\]

with \( S \) and \( S' \) representing the system state before and after transition, and \( A = [a_1, ..., a_n] \) is the actions of all agents in the system. \( P(\cdot) \) is the transition probability function. It is said to be deterministic optimization problem if \( P(\cdot) = 1 \) for all time, otherwise it is stochastic optimization problem.

Considering of the simplified system model neglecting uncertainties, the formation control problem discussed in general can be solved using optimization algorithms under multi-agent POMDP. Assuming that the fitness function measuring the performance of agent \( i \) is \( f(\theta_i, \theta_{N}) \), in which \( \theta_i \) is the controller(policy) parameter representing agent \( i \), and \( \theta_{N} = \{ \theta_j : j \in N_i \} \) is set of neighboring parameters of agent \( i \). The objectives of optimal control is then to find the optimal control strategy \( \theta^*_{i}, i \in (1, ..., N) \), such that

\[
F((f(\theta^*_{i}, \theta_{N_i}) : i \in (1, ..., N))) \geq F((f(\theta_i \neq \theta^*_{i}, \theta_{N_i}) : i \in (1, ..., N))),
\]

where \( F(\cdot) \) denote the overall performance of all agents in the MAS. Such 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 difficult to find a Nash equilibrium strategy while satisfying the system constraints Eq. (10), and also limited by the nonlinear continuous system and the cost function with coupled parameters [33, 14]. For the formation control problem which requires sufficient cooperation among neighboring agents, \( F(\cdot) \) can be regarded as a simple summation of all individual \( f(\cdot) \), and the formation control problem can be viewed as a constrained dynamic optimization problem with the objective of minimizing the total cost of formation error. Thus the Nash equilibrium strategy can be obtained using a co-evolutionary algorithm that is designed for evolving simultaneous to reach the overall optimum fitness. In our previous work [26], we improved on the natural evolutionary strategy (NES) and proposed an natural co-evolutionary strategy (NCES) algorithm that seeks global optimality for constrained multi-objective optimization problem in multi-agent system. In brief, NCES is a bio-inspired population-based algorithm that is capable of optimizing high-dimensional parameters, such as the weights of neural networks, towards the direction of higher fitness. The NCES algorithm usually proceeds as follows: first, the parameters \( \theta_i \) for \( i \in (1, ..., N) \) 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. Then, in the second step, iterative optimization is performed, and new perturbations \( \epsilon \) 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, and their fitness values are then evaluated in the system in a distributed way, so that each population obtains the following corresponding gradient information

\[
g_{\theta_i} = \frac{1}{m \sigma^2} \sum_{i=1}^{m} f(\theta'_i, \theta_{N_i}) \prod_{c \in N_i} p(\epsilon_c),
\]

and updates its parameters in a gradient ascent manner

\[
\theta_i = \theta_i + \eta_a g_{\theta_i}.
\]

Finally, loop the second step until convergence or the Nash equilibrium strategy is found. For more detailed procedure of the algorithm, please refer to the original research.

3. Applying Natural Co-evolutionary Strategy to formation control via neural networks

3.1. Distributed co-evolutionary strategy optimizing neural networks controller

Motivated by previous works which implement NNs in MAS, we use a multi-layer perceptron (MLP) NNs with a single hidden layer of 16 nodes as the distributed controller. The schematic of the neural networks controller is depicted in Fig. 8. The weighting matrix for each layer are represented by \( W_1 \) and \( W_2 \), with \( \phi(\cdot) \) and \( \psi(\cdot) \) as their activation function, respectively. Specifically, \( \phi(\cdot) \) is the Sigmoid function and \( \psi(\cdot) \) is
we propose a series of supplementary techniques to enhance the convergence speed and accuracy of the algorithm. Another advantage of the neural network controller is shown to effectively impose restrictions on the output to satisfy the system constraints. Another advantage of the neural network controller is shown 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

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

Note that the dashed network input in the schematic represents the transform signal (TS), 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 firstly determined that the parameters of the controller are \( \theta_i = [W_{1i}, W_{2i}] \). The fitness function is the objective function Eq. (8), i.e., \( f(\theta_i, \theta_{A_i}) = J_i \), which is non-linearly coupled with the states of neighboring agents and can be only evaluated through the interaction feedback within the system. In order to apply NCES to the training of the missile formation controller in this paper, the plain NCES algorithm is not sufficient to guarantee the global stability and the convergence speed of the algorithm. To further improve the performance of the algorithm and to 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.

3.2. Population adaptation technique

In [26] we point out the importance of learning rate on 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 here a novel population size adaptation algorithm is considered, which can be used to adjust the evolutionary process adaptively. Based on the previous works [34, 35], we can learn that the trend of the gradient is related to the complexity of the objective function, usually for the more complex optimization region such as multi-modal or noised function the estimated gradient is gentle, while in the flatter region such as the 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 \( p_\theta \) is introduced to detect the resistance in the evolutionary process. The population size is adapted based on the length of the evolutionary path following an empirical common sense that larger population size will lead to higher accuracy of the estimated gradient. Considering of the parameters of the \( i^{th} \) agent as the vector consisting of the weighting matrices \( W_{1i}, W_{2i} \), that is \( \theta_i \in \mathbb{R}^2 \), and \( s \) is the total number of weights. The evolution path in iteration \( t \) is calculated by accumulated the square of the Mahalanobis distance of the parameter movement of all agents as

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

where \( \Sigma \) 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:

\[
\eta_p(t) = \eta_p(t-1)(\beta + (1-\beta)\rho_\theta(t-1) / \rho_\theta(t)),
\eta_p(t) = \text{clip}(\text{max}(\eta_p(t), \eta_p(t-1)), \eta_p^{\text{min}}, \eta_p^{\text{max}}), \tag{22}\]

where \( \beta \) is a constant factor that determines the growth rate of the population size and \( \eta_p^{\text{min}}, \eta_p^{\text{max}} \) are the minimum and maximum population size which are sent to the clip function clip() in case of undesirable adapted values. Note that since the total optimization complexity tends to increase as evolution progresses, we adopt 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 setup in [36], which should be a good candidate, and the boundaries are determined as

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

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

3.3. Cluster-based adaptive topology

The inter-agent connectivity in the filed of multi-agent system have been primarily modeled and characterized by means of graph theory [37, 5], which in this paper 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 \( G \) is defined as \((V, \epsilon)\), where \( V = \{v_1, v_2, ..., v_n\} \) represents the set of nodes, and \( \epsilon \subseteq V \times V \) represents the set of edges composed of directed connections.
of different nodes. The neighbor set of node $i$ is defined as $N_i = \{j \in V : (i, j) \in e\}$. $j \in V$ is said to be connected to $i \in V$ in a directed way if $(i, j) \in e$ and $(j, i) \notin e$, otherwise $j$ is said to be connected to $i$ in an undirected way if both $(i, j)$ and $(j, i)$ belong to $e$. A directed path of $G$ is a series of adjacent edges of the form $(v_1, v_2), (v_2, v_3), ..., (v_j, v_k)$, and a graph is said to have a spanning tree if there exist a directed path from one node to any other nodes, and a graph is said to be connected in directed (undirected) manner if there is a directed (undirected) path between any pair of distinct nodes. We use the adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ to represent the above node connectivity, in which

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

The associated Laplacian matrix is $L = D - A$. $D = \text{diag}(a_i)$ 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 symmetric semi-positive definite matrix.

One of the foremost goal of designing communication method is to ensure highly reliable and low latency communication for all nodes in the swarm. The communication quality of missile swarm is easily affected by various conditions especially when performing in hostile environment. In another word, swarm of missiles are constantly subject to threats from enemy’s air defense system during flight, and some missiles may get intercepted and lost communication with the rest of the swarm. Therefore, dynamic communication approaches needs to ensure that when some nodes fail, the swarm can still maintain effective communication to complete the mission. Based on the previous work in network topology and swarm communication, we developed a novel adaptive cluster-based network, which adaptively reconfigure the communication topology to achieve robust and fault-tolerant communication. Under the guidance of the well known Molloy-Reed criterion [38], which is

$$k = \frac{k^2}{k} > 2,$$

$k$ 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 be at least connected to two other nodes. What is also known is that for displacement-based formation, the formation is persistent only if there exists at least a spanning-tree in the communication network. When establishing a communication connection, the formation error tends to grow as the communication distance between nodes increases, so we favour the connection method which minimizes the length of the communication chains [39]. With the minimal constraint of satisfying the above conditions, we propose to set one node(usually node 1) of the network as the cluster head and the other nodes choose another communication node according to the inter-node distances that are derived from the formation definition. For achieving this, we define the inter-agent distance matrix as

$$D_g = [d_{ij}] \in \mathbb{R}^{n \times n}, \quad d_{ij} = ||a_{pi} - A_{pj}||,$$

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

$$a_{ij} = \begin{cases} 1, & \text{if } i \neq h, \text{ and } j \in \{\min |k = \arg \min_k d_{ih} : \quad k \in \mathbb{N}, 1 \leq k \leq n, k \neq h, h\}; \\ 0, & \text{otherwise}; \end{cases}\quad (27)$$

In the proposed adaptive topology, the cluster head undertakes the task of broadcasting its own state to other nodes in the network, or 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 follower nodes.

![Figure 4: Schematic diagram of the adaptive topology network with 5 nodes](image)

Using the above communication configuration method for a network with 5 nodes as an example, the obtained communication topology is shown in Fig. 4a and its adjacent matrix is

$$A^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$ \quad (28)$$

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

$$A^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$ \quad (29)$$

Using the proposed adaptive communication method, the network connectedness can be ensured no matter how many nodes fail. The proof is that graph $G$ is uniformly connected since for any $t \geq t_0$, there exists a node $h \in V$ such that $h$ is the root of a spanning tree [40].

3.4. 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 [41], 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 is also known as safety exploration [42] has been increasingly applied to simulation and training of realistic robotic controllers, by which not only can personal safety be ensured during the training process, but also global convergence can be accelerated to some extent. Based on the above facts, we propose a model-based constrained policy strategy. The non-linear error dynamic of the second-order system has been obtained in Eq. (6), which is a function of the system input. To apply the method, we assume that the control input of the communication object can be obtained by the agent, so that at time \( t \), the predicted formation error at next step can be calculated as follows:

\[
\hat{e}_p(t) = e_{p}(t) + \hat{e}_p(t)\tau,
\]

where \( \tau \) is the time step, and the error deviation \( \Delta e_p(t) \in \mathbb{R}^3 \) is

\[
\Delta e_p(t) = |\hat{e}_p(t)| - |e_p(t)|,
\]

then the aggregate matrix is defined as \( \Delta E(t) = [..., \Delta e_p(t),...] \in \mathbb{R}^{3na} \), for \( i \in \{1, 2, ..., n\} \). A termination indicator \( ST \) is assigned to the system so that if it is a nonzero value, the system terminates and restart the training algorithm, and it is defined as

\[
ST(t) = \begin{cases} 1, & \text{if } \min(\Delta E(t)) > \delta_s; \\ 0, & \text{otherwise,} \end{cases}
\]

and \( \delta_s \) is the threshold that measures the maximum error increment that the algorithm can tolerate. It satisfies \( \delta_s < (2I + 1)\eta V_{\text{max}} \), where \( I \) is the number of neighbors for the agent with the most neighbors in the MAS. With a reasonable choice of \( \delta_s \), 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 detailed pseudo-code of the proposed algorithm that is used to train the missile formation controllers of interest is shown in [41]. 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.

4. Simulation and result analysis

In this section various experiments are implemented to demonstrate the effectiveness of the proposed formation control algorithm. In order to simulate actual physical environment, the construction of the experimental scenario and the modeling of the

Algorithm 1 The distributed NCES based formation control algorithm

**Input**: agent number \( N \in \mathbb{N}^+ \), population size \( \eta_p \), standard deviation \( \sigma \), \( \beta \), evolution path \( \rho_b \), number of parameters \( m \), iteration \( i \)

**Initialize**

for each agent \( i \in \{1, ..., N\} \) do
  initialize parameter \( \theta_i^{(0)} \)
  \( t \leftarrow 0 \)
  \( \theta_i(0) \leftarrow \theta_i^{(0)} \)
  \( \eta_p(0) \leftarrow 10 + 5 \ln(m) \)
  \( \eta_p^{\min} \leftarrow \eta_p(0) \)
  \( \eta_p^{\max} \leftarrow 4\eta_p(0) \)
end for

while stopping criterion not met do
  for \( k = 1, ..., m \) do
    for each agent \( i \in \{1, ..., N\} \) do
      sample \( \epsilon^k_i \sim N(0, \sigma^2 I) \)
      \( \theta_i^{(0)} \leftarrow \theta_i + \epsilon^k_i \)
    end for
    evaluate fitness \( f(\theta_1^i, \theta_N^i) \), for \( i \in \{1, ..., N\} \)
  end for
  for each agent \( i \in \{1, ..., N\} \) do
    calculate natural gradient:
    \( g_h \leftarrow \frac{1}{m\sigma^2} \sum_{k=1}^{m} f(\theta_1^k, \theta_N^k) \epsilon^k_i \prod_{i \in N} p(\epsilon^k_i) \)
    \( \theta_i(t) \leftarrow \theta_i(t-1) + \eta_p \cdot g_h \)
  end for
  append evolution path:
  \( \rho_b(t) \leftarrow \sum_{i=1}^{N} [\theta_i(t) - \theta_i(t-1)]^T \Sigma^{-1} [\theta_i(t) - \theta_i(t-1)] \)
end while

Algorithm 2 AdaptPopulationSize(\( \rho_b \))

if length(\( \rho_b \)) > 1 then
  \( \eta_p(t) \leftarrow \eta_p(t-1)(1 + (1 - \beta) \frac{\rho_b(t-1) \max}{\min} \eta_p(t-1), \eta_p^{\min}, \eta_p^{\max}) \)
else
  \( \eta_p(t) \leftarrow \eta_p(0) \)
end if
multi-agent system were carried out in PyBullet [43]. We simulated the operation of the missile swarm in three-dimensional space at a fixed altitude, and for convenience the trajectories was plotted as two-dimensional planar graphs. Some complex aerodynamic parameters such as air resistance are removed, while collision detection is preserved. The simulation timestep $\tau$ is set to 0.1 for the following experiments.

### 4.1. Basic formation control

First of all the basic formation control tasks are implement to examine the validity of the algorithm under basic situations, the missile swarm of five missiles entails tracking the reference target moving along diagonal or spiral trajectory while keeping the formation geometric. 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 all time, that is $\int_{t_0}^{t_f} |e_i(t)| dt \to 0$, for $i \in \{1, \ldots, N\}$. The hyper-parameters of the algorithm is listed in Table 1 and it is noted that this parameter setting applies to all of the following experiments. Additionally, the system constraints are shown in Table 2, the missiles as well as the reference target are subjected to saturated control input and limited states. For the linear trajectory, the target is driven by constant control input, while for the spiral trajectory, the control input of the target is $v_r = 0.65 - 0.01t$ (km/s) and $\omega_r = 0.1 + 0.01t$ (rad/s).

For convenience, we use $|e_i|$ to represent the resultant error of each agent. The trajectories of the two situations are shown in Fig. 5 and Fig. 6 and the corresponding analytical results are presented in Fig. 7 and Fig. 8. It can be observed that the resultant error is maintained in a small interval (within 0.1) in both cases and that a good synchronization of the velocity and heading angles among the missiles is achieved, despite the fact that velocity information is not provided.
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 are separated from the reference target with their positions initialized randomly in an area of 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 moves along the y-axis with a constant speed of 0.5km. Fig. 9 shows the motion trajectory of the swarm formation, and Fig. 10 shows the result of the policy network as depicted in Fig. 3. It’s assumed that whenever the obstacle is detected by the leader missile, or more literally the cluster head, the missile then sends a signal to all connected missiles to perform formation switch $\lambda^R \rightarrow \lambda^L$. Although formation geometries are predefined by the controller, the shapes can be controlled by the parameter $I_f$ and $\beta$, which is changeable during the flight.

In this scenario, two walls are placed in the trajectory of the formation flight to create narrow spaces, and the missiles need to negotiate the obstacles by changing the shape or size of the formation. We implement a event-based formation switching strategy, in which nodes in the formation are able to detect obstacles within a certain range $d_e$ and send a formation switching signal to all other nodes if an obstacle is detected. Similarly, after crossing the obstacle and reaching a safe distance, a formation recovery signal will be sent to restore the original formation.

To cross the obstacle, we consider both changing the formation pattern and formation size, and the time-varying definition of the formation in both cases are as follows:

$$\lambda^l(t) = \begin{cases} \lambda_{(0,0.5)}^R & \text{if } d_e \leq n \cdot I_f \\ \lambda_{(\pi/4,0.5)}^R & \text{otherwise} \end{cases},$$

$$\lambda^2(t) = \begin{cases} \lambda_{(0,0.5)}^R & \text{if } d_e \leq n \cdot I_f \\ \lambda_{(0.2)}^R & \text{otherwise} \end{cases}.$$  

(33)

The result trajectories are shown in Fig. 11 and Fig. 12, it can be observed that when obstacles are detected, the swarm can swiftly adjust by changing the formation pattern or size, and then recover the formation quickly after going through.

4.3. Switching formations

Further, we discuss the case in which the missile swarm are supposed to switch among formation patterns in order to avoid obstacles and go through narrow spaces. To achieve this transformation, an additional node $TS$ is appended to the input layer of the policy network as depicted in Fig. 4. It’s assumed that whenever the obstacle is detected by the leader missile, or more literally the cluster head, the missile then sends a signal to all connected missiles to perform formation switch $\lambda^R \rightarrow \lambda^L$. Although formation geometries are predefined by the controller, the shapes can be controlled by the parameter $I_f$ and $\beta$, which is changeable during the flight.

In this scenario, two walls are placed in the trajectory of the formation flight to create narrow spaces, and the missiles need to negotiate the obstacles by changing the shape or size of the formation. We implement a event-based formation switching strategy, in which nodes in the formation are able to detect obstacles within a certain range $d_e$ and send a formation switching signal to all other nodes if an obstacle is detected. Similarly, after crossing the obstacle and reaching a safe distance, a formation recovery signal will be sent to restore the original formation.

To cross the obstacle, we consider both changing the formation pattern and formation size, and the time-varying definition of the formation in both cases are as follows:

$$\lambda^l(t) = \begin{cases} \lambda_{(0,0.5)}^R & \text{if } d_e \leq n \cdot I_f \\ \lambda_{(\pi/4,0.5)}^R & \text{otherwise} \end{cases},$$

$$\lambda^2(t) = \begin{cases} \lambda_{(0,0.5)}^R & \text{if } d_e \leq n \cdot I_f \\ \lambda_{(0.2)}^R & \text{otherwise} \end{cases}.$$  

(33)

The result trajectories are shown in Fig. 11 and Fig. 12, it can be observed that when obstacles are detected, the swarm can swiftly adjust by changing the formation pattern or size, and then recover the formation quickly after going through.

4.4. Formation control under node failure

To verify the effectiveness of the proposed algorithm under node failure. In this experimental scenario, a swarm with six nodes are designed to pursue the reference target in a regular polygon formation, and the reference target moves in a sinusoidal fashion. At $t = 20$ s during the pursuit, the cluster head node 1 and the cluster member node 4 suffer attacks and will disconnect from the other nodes in the swarm, and the remaining nodes of the swarm need to maintain their original formation and complete the formation task.

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

5. Conclusion

In this paper, a novel distributed NCES based formation control algorithm for a second-order multi-missile system using neural network controller has been proposed. The algorithm minimizes the formation shape error and tracking error by training the optimal network controller through iterative learning, combining with a policy constraint approach to enhance the stability of the algorithm. Additionally, we have designed an adaptive topology scheme for the node failure situation, which can achieve stable communication connections at a low communication cost. We have also proposed a stable population adaptation method base on evolution path, in order to further improve the performance of the algorithm and alleviate local optimum issue. Extensive experiments demonstrated that the proposed formation control algorithm is capable of accomplishing tasks such as formation maintenance, tracking of reference trajectories and formation transformation in face of obstacles with high accuracy, and with certain robustness to cope with situations such as random initial positions and node failures. Future work may include the investigation of online evolution-ary algorithms to solve the problem of unknown or stochastic system models, building on existing work.
Figure 7: Analytical results of the linear trajectory case

Figure 8: Analytical results of the spiral trajectory case
Figure 9: Trajectories of moving into formation case

Figure 10: Resultant error curves of moving into formation case

Figure 11: Trajectories of switching formation type case

Figure 12: Trajectories of switching formation size case
Figure 13: Trajectories of the node failure case

Figure 14: Resultant error curves of the node failure case
Reinforcement Learning 25.

[43] E. Coumans, Y. Bai, Pybullet, a python module for physics simulation for games, robotics and machine learning (2016).