Perceptron-Learning for Scalable and Transparent Dynamic Formation in Swarm-on-Swarm Shepherding

Tung Nguyen, Jing Liu, Hung Nguyen, Kathryn Kasmarik, Sreenatha Anavatti, Matthew Garratt, and Hussein Abbass
School of Engineering and Information Technology
University of New South Wales - Canberra
Canberra, Australia
{tung.nguyen,jing.liu5,hung.nguyen}@student.adfa.edu.au; {kathryn.kasmarik,agsrenat,m.garratt,h.abbass}@adfa.edu.au

***Abstract*—Swarm guidance, such as the case of guiding a group of sheep away from a field, is a challenging task. As the swarm size increases, it becomes necessary that multiple control points, or sheepdogs, are needed to guide the swarm. In this paper, a swarm of unmanned aerial vehicles (UAVs) acts as a moving safety network (aka a formation) that not only guides the sheep swarm, but also prevents them from dispersing or reversing to the other side of the field. We investigate two types of formations. The first type acts as a baseline, maintains fixed distances from the sheep swarm, and relies on fixed predefined angular structure relative to the sheep's global centre of mass (GCM). The second type is dynamic, where the force vector to control the UAV and the individual distance of each UAV from the sheep's GCM are controlled by a Perceptron, with the weights optimized by a particle swarm optimization algorithm. We evolve five Perceptrons to specialize in relative positions in the formation, which fixes the space cost for the optimization algorithm, while allowing the size of the swarm of UAVs to scale up. We demonstrate that the use of Perceptron-networks for dynamic control scheme reduces the total distance travelled by the UAVs, is transparent when interpreted with Hinton diagrams, and transferable to a larger number of UAVs.**

***Index Terms*—Multi-Agent Systems, Formation Control, Shepherding, Particle Swarm Optimization**

I. Introduction

Bio-inspired swarm control problems synthesize novel perspectives from a variety of fields including biology, control theory, and artificial intelligence [1]. Studies in multi-agent systems and swarm robotics attempt to propose cost-effective algorithms to create swarm behaviour of simple agents/robots [2], [3].

Swarm control approaches can be divided into two categories: rule-based and learning-based algorithms [4]. The former algorithms use a set of fixed rules or predefined equations to compute the dynamics of the system based on local or global state information [5], [6]. While they are simple in design and scalable, they often do not generalize well to different contexts.

The latter approach is based on machine learning, offers flexibility, and eliminates the need for a large amount of knowledge to pre-exist. Various methods applying reinforcement learning or deep reinforcement learning combined with team communication or a shared mental model have been proposed to learn decentralized policies for swarm control [7]–[9]. Nevertheless, these approaches do not scale up well with an increasing number of swarm members due to the significant increase in the computational resources required to train multiple agents simultaneously.

The shepherding problem is inspired by sheep-herding using a single or multiple sheepdogs in agriculture. Shepherding is a flocking behaviour in which one or multiple external agents/sheepdogs, guide a swarm of autonomous agents, called flocking or sheep agents towards a predefined target. The basic idea has been transferred to the context of swarm robotics and multi-agent systems [10]. A wide range of practical applications [10], [11] of shepherding problem within robotics include herding cattle or other free-living animals, guiding a group of sheep away from a field area, assisting in human crowd control activities, cleaning environmental hazards like oil-spills, or driving cells to repair tissue in internal medicine [12].

Strömbom et al. [10], [13] develop a heuristic rule-based shepherding model to explain the interaction between one intelligent agent and a swarm of autonomous agents using two basic behaviours: collecting and driving. The former behaviour collects astray sheep, while the latter drives a flock towards a goal.

However, the use of a single shepherding agent to control an entire swarm is ineffective when the number of entities to be guided increases and their complex behaviours, such as random flocking or dispersing, challenge the capability of a single sheepdog to complete the task successfully [14], [15]. The use of multiple shepherding agents is a promising way to address this issue. Some authors [14], [15] use rules for multiple agents in order to create fixed sheepdogs formations such as a line or an arc. In nature, though, the sheepdogs do not abide to a strict geometric shape or formation. This begs a fundamental question: how do sheepdogs decide on their proximity to the sheep while organizing themselves in a flexible formation?

In this paper, we propose a low-cost dynamic formation learning approach for multiple UAVs to guide a swarm of

sheep. In the proposed learning approach, five hyperbolic tangent (tanh) Perceptrons are used to adapt the positions of multiple sheepdogs in the controlling area behind the flock and towards the target. This approach is designed to be independent of the number of UAVs used for shepherding; that is, the five tanh networks can be used by more than five agents. In order to optimize the weights of the five networks, Particle Swarm Optimization (PSO) [16], [17] is used. This algorithm is able to produce solutions in a vectorized form, allowing efficient runs on graphical processing units (GPUs). Other advantages of PSO are the ease of implementation and a relatively low number of parameters compared to other algorithms. This proposed learning approach utilizes the advantages of PSO to achieve better performance of dynamic formation. It also provides scalability when the size of the flock and the number of UAVs increase.

II. Related work

The flocking behaviour can be seen widely in animal herds [18], for example, ants, birds, and fishes. A large number of agents interact and collaborate with one another in order to achieve different objectives such as finding paths or seeking food. Understanding these swarm behaviours helps to not only design effectively distributed and coordinated control methods in multi-agent systems but also inspire swarm intelligence optimization algorithms such as Particle Swarm Optimization (PSO) [16], [17].

In an early research on shepherding [19], the authors attempt to learn a set of rules for a shepherding agent to guide a sheep swarm to a desired target by using Genetic Algorithms. In another study, Lien et al. [20] attempt to simulate four behaviours: herding, covering, patrolling, and collecting. The combination of those exhibits effective shepherding strategies. However, both approaches above are more appropriate for guiding a small flock size (less than 40) [21].

Aiming to guide a large number of agents or a swarm of agents (more than 40), Strömbom et al. [10], [13] develop a heuristic model for shepherding using two basic behaviours: collecting and driving. The model is promising and is able to herd up to 300 sheep, although the success rate declines unless multiple shepherds are in use.

In the context of using multiple shepherding agents, Lien et al. [14], [15] propose the use of a group of shepherds in order to control a swarm of sheep, demonstrating multiple shepherds perform better than a single shepherd. Their control method creates different fixed formations for multiple shepherds like a line or an arc. However, those multi-shepherd frameworks are limited to a small flock size.

Learning dynamic formation control of multiple shepherds using Strömbom's model, which is well-known for its ability to control large flocks, is an unexplored area. A dynamic formation adapts the formation to the flock, which we hypothesize to improve efficiency.

In this paper, we propose a dynamic formation production model, whose parameters are optimized by PSO [16], [17], for multiple UAVs to guide a large number of sheep towards a desired target.

III. PSO-based dynamic formation learning

An arc formation similar to [14] is created by a set of fixed rules. Each shepherd specializes in a fixed position in the arc formation. We adopt the model introduced by Strömbom et al. [13]. The environment is a $L \times L$ square and obstacle-free paddock. Given M shepherds and N sheep in the environment, the notations for the shepherds and the sheep are $B = \{\beta_1, \ldots, \beta_j, \ldots, \beta_M\}$ and $\Pi = \{\pi_1, \ldots, \pi_i, \ldots, \pi_N\}$, respectively. The behaviours of the sheep are identical to that in Strömbom's model. Each sheep π_i has four behaviours at a time step t as follows:

1) *Escaping*: This behaviour is triggered when there is a repulsive force $F^t_{\pi_i\beta_j}$ between sheep π_i and shepherd β_j. The position of π_i and that of β_j at time step t are denoted as $P^t_{\pi_i}$ and $P^t_{\beta_j}$. The force exists when the distance between them is no more than a predefined distance $R_{\pi\beta}$; that is,
$$\|P^t_{\pi_i} - P^t_{\beta_j}\| \leq R_{\pi\beta} \tag{1}$$
2) *Collision avoidance*: This behaviour is triggered when there is a repulsive force between sheep π_i and other sheep $\pi_{k\neq i}$. The force is able to exist if the distance between them is less than or equal to a threshold $R_{\pi\pi}$; that is,
$$\exists k \neq i,\ such\ that\ \|P^t_{\pi_i} - P^t_{\pi_k}\| \leq R_{\pi\pi} \tag{2}$$
$F^t_{\pi_i\pi_{-i}}$ is denoted as the summed force vector on sheep π_i influenced by all others within the threshold distance.
3) *Grouping*: This behaviour represents an attraction force $F^t_{\pi_i\Lambda^t_{\pi_i}}$ of sheep π_i to the centre of mass of its neighbors $\Lambda^t_{\pi_i}$.
4) *Jittering*: To avoid impasse, a random noise vector, sampled from the normal distribution, is added to sheep π_i. This noise is presented as $F^t_{\pi_i\epsilon}$ with a weight $W_{e\pi_i}$ and summed into the total force.

Then, each sheep π_i has a total force $F^t_{\pi_i}$ representing a weighted sum of force vectors $F^t_{\pi_i\beta_j}$, $F^t_{\pi_i\pi_{-i}}$, $F^t_{\pi_i\Lambda^t_{\pi_i}}$, $F^t_{\pi_i\epsilon}$, and the previous total force $F^{t-1}_{\pi_i}$ at time step $t-1$; that is,

$$F^t_{\pi_i} = W_{\pi_\upsilon}F^{t-1}_{\pi_i} + W_{\pi\Lambda}F^t_{\pi_i\Lambda^t_{\pi_i}} + W_{\pi\beta}F^t_{\pi_i\beta_j} + W_{\pi\pi}F^t_{\pi_i\pi_{-i}} + W_{e\pi_i}F^t_{\pi_i\epsilon} \tag{3}$$

The shepherds include two key behaviours: reaching to and maintaining their fixed positions in the arc formation. Each shepherd observes the state of the environment and produces a directional vector $F^t_{\beta_j}$ that leads it to an appropriate position to guide the sheep. The current positions of the shepherds and the sheep are updated according to Equations 4 and 5 given $S^t_{\beta_j}$ and $S^t_{\pi_i}$ be the speed of β_j and the speed of π_i at time t, respectively. However, in Strömbom model, the speeds of the agents are fixed.

$$P^{t+1}_{\beta_j} = P^t_{\beta_j} + S^t_{\beta_j}F^t_{\beta_j} \tag{4}$$

$$P_{\pi_i}^{t+1} = P_{\pi_i}^{t} + S_{\pi_i}^{t} F_{\pi_i}^{t} \quad (5)$$

The dynamic spatial distribution of individuals in the sheep swarm, which a shepherd does not fully aware of, contributes to the task complexity.

A. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a well-known swarm intelligence optimization algorithm and has been successfully applied to many realistic problems such as neural network optimization due to its ability to search globally with low computational cost [16]. In PSO, each particle, which is regarded as a point in the D-dimensional search space, represents a candidate solution to the optimization problem. Particles in the swarm adjust their flying velocities based on both their individual experience and the swarm experience in order to arrive at better solutions. The position of the i^{th} particle is denoted as $X_i = \{X_i^d,\ d = 1, ..., D\}$ and its velocity is represented as $V_i = \{V_i^d,\ d = 1, ..., D\}$. D is the dimension of the search space. The velocity and position of the i^{th} particle are manipulated as follows:

$$V_i^d \leftarrow V_i^d + c_1 \cdot r_1^d \cdot (Pbest_i^d - X_i^d) + c_2 \cdot r_2^d \cdot (Gbest^d - X_i^d) \quad (6)$$

$$X_i^d \leftarrow X_i^d + V_i^d,\ i = 1, 2, ..., N_p \quad (7)$$

where N_p is the population size, c_1, c_2 are two acceleration constants, and r_1, r_2 are two random numbers in the range [0, 1]. $Pbest_i$ is the personal best historical position of i^{th} particle, representing individual's own experience, while $Gbest$ is the global best position of the swarm, representing the swarm experience.

Algorithm 1 The pseudo code of PSO-w algorithm

Require: variables $X^d, d = 1, ..., D$, the evaluation function $F(X)$

1: **Initialize** the algorithms parameters N_p, c_1, c_2, the population positions $X_i, i = 1, ..., N_p$ and velocity $V_i, i = 1, ..., N_p$
2: **while** termination conditions not met **do**
3: **for** $i = 1, ..., N_p$ **do**
4: Update the velocity V_i according to (8) and (9)
5: Update the position X_i according to (7)
6: Evaluate the fitness value $F(X_i)$ of particle
7: **end for**
8: Update $Pbest, Gbest$
9: **end while**
10: $X^* = Gbest, f(X^*) = F(Gbest)$

Ensure: the best solution X^*, the best objective value $F(X^*)$

During the initialization phase, N_p particles, which represent N_p potential solutions, are randomly generated in the D-dimensional space. The objective function of the optimization problem is used to evaluate the fitness value of each particle in a specific position, based on which $Pbest$ and $Gbest$ are selected. Then particles update their positions according to Equation 6 and Equation 7. If the particle finds a better position in the solution space compared to its previous $Pbest$, its $Pbest$ will be updated with that position. $Gbest$ will be updated only if the swarm finds a better global solution compared to previous $Gbest$. Particles keep moving in the search space until the stopping conditions are met. Hence, $Gbest$ is considered the optimal solution for the problem.

In PSO, the update of particle's flying velocity consists of three parts: the "previous velocity" part, the "cognition" part related to $Pbest$ and the "social" part related to $Gbest$. The first "previous velocity" part benefits the global search while the last two parts facilitate the local search [22]. To better balance the global and local search, Shi and Eberhart [22] added an inertia weight w into Equation 6 to control the effect of previous velocity. This standard PSO is known as (PSO-w). A large w allows particles explore more areas while a small w favors exploitation in local areas of the search space. The velocity in PSO-w with linearly decreasing w is updated according to the following equation:

$$V_i^d \leftarrow w \cdot V_i^d + c_1 \cdot r_1^d \cdot (Pbest_i^d - X_i^d) + c_2 \cdot r_2^d \cdot (Gbest^d - X_i^d) \quad (8)$$

$$w = w_{max} - (w_{max} - w_{min}) \cdot k / Max_{gen} \quad (9)$$

where w_{max} is the maximum initial w, w_{min} is the minimum w, k is the current number of generations and Max_{gen} is the maximum number of generations. In this way, the inertial weight is linearly decreased from w_{max} to w_{min} to balance exploitation and exploration. The pseudo code of the PSO-w algorithm is shown in Algorithm 1.

B. A PSO-based Dynamic Formation Learning for swarm-on-swarm guidance

In this paper, a PSO-based dynamic formation learning method is proposed for optimizing the formation of multiple UAVs to guide the sheep swarm towards a target position. The formation of the UAVs is determined by a set of subgoal points whose positions are generated based on the sheep's positions. These subgoals are produced by Perceptron-networks optimized by PSO. Each UAV is assigned a subgoal as its navigation destination to deploy the formation.

Figure 1 illustrates a formation of UAVs to herd the sheep swarm, consisting of multiple subgoal points which are defined according to two factors: the angular position θ and the distance δ. Five Perceptron-networks are evolved as subgoal production models to specialize in relative positions in the formation for multiple UAVs. Tactical subgoals are generated at different locations within 180 degree arc in each formation for guiding the sheep swarm. The Perceptron-networks are trained to specialize in locations for UAVs acting as left fielder (LF), left midfielder (LMF), midfielder (MF), right midfielder (RMF), and right fielder (RF) respectively. LF, MF and RF UAVs specialize in the most left, the middle and the most right positions in the formation respectively. The UAVs which specialize in a position between two UAVs on left or right-hand side are called LMF and RMF UAVs. This type of formation generation method allows the size of the swarm of UAVs to scale up without an increase in the number of networks to be optimized.

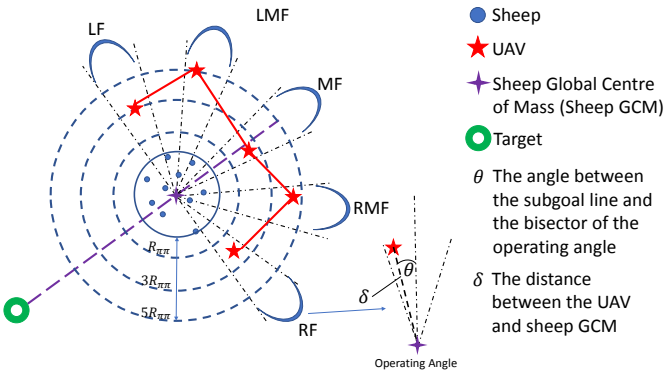


Fig. 1: A dynamic formation learning problem for swarm-on-swarm guidance based on shepherding mode of control. Each UAV is assigned an operating angle. The subgoal of the UAV is defined by angle θ and the distance δ.

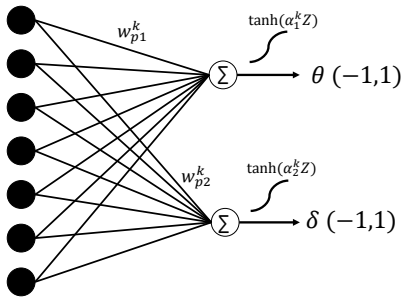


Fig. 2: Subgoal production model with two outputs.

As shown in Figure 2, each Perceptron-network consists of 2 Perceptrons which have N_I inputs and 2 output values with no hidden layer between them. The weights are denoted as $w_{pj}^k,\ k = 1, ..., 5,\ p = 1, ..., N_I,\ j = 1, 2$ where k is the index of five Perceptron-networks, p is the index of N_I inputs and j is the index of two Perceptrons. The number of inputs is N_I and is relative to the specific problem scenario. The outputs of the network are activated by hyperbolic tangent activation functions ($tanh(\alpha Z)$) which result in outputs ranging between -1 and 1. $tanh(\alpha_1^k Z)$ and $tanh(\alpha_2^k Z)$ are used for two outputs of the k^{th} network respectively, which are then scaled to:

- *Angular Position* $\theta \in [-\vartheta, \vartheta]$
- *Distance Factor* δ

Each subgoal is defined as a position that has a distance of δ from the sheep's GCM, and the line connecting the subgoal and the sheep's GCM forms with the bisector of the operating angle assigned to a corresponding UAV an angle of θ. Based on the above, the parameters involved in each subgoal production model consist of $N_I \times 2$ weights of the networks and 2 α values of the $tanh(\alpha X)$ activation functions for the corresponding outputs. As we have 5 subgoal production models based on locations, the maximum number of parameters that can be searched for in the dynamic formation learning problem is $5 \times (N_I \times 2 + 2)$. PSO is applied to these models to obtain higher guidance performance by optimizing the entire parameter set or only some specific parts of it.

TABLE I: Environmental parameters in the simulation.

Parameter	Meaning	Value
L	Length and Width of Environment	150
N	Number of Sheep	{50,100,200}
M	Number of Shepherds/UAVs	{5,7,9}
$R_{\pi\beta}$	Sensing range of a sheep for a UAV	65
$R_{\pi\pi}$	Sensing range of a sheep for another sheep	2
$W_{\pi\pi}$	Sheep repulsion strength from other sheep	2
$W_{\pi\beta}$	Sheep repulsion strength from UAVs	1
$W_{\pi\Lambda}$	Sheep attraction strength to sheep centre of mass	1.05
W_{π_υ}	Inertial strength of sheep previous direction	0.5
$W_{e\pi_i}$	Strength of sheep movement noise	0.3
$W_{e\beta_j}$	Strength of UAV movement noise	0.3
$\lvert\Omega_{\pi_i\pi}\rvert$	Number of sheep (neighborhood) a sheep can sense	25
S_π	Maximum speed of sheep	1
S_β	Maximum speed of a UAV	2
$\mathbb{D}$	Minimum distance between the sheep's global centre of mass and the target for successful mission	5

IV. Experiments

The swarm-on-swarm guidance with dynamic formation control is simulated using the model presented in Section III. The parameters regarding environment initialization and the interaction between agents for the simulation are listed in Table I. In each simulation, a predefined number of sheep are randomly initialized at the centre of the paddock with their coordinates in the range of between 1/4 and 3/4 of the length/width of the environment. The UAVs are randomly spawned at the lower left corner, with their coordinates not exceeding 1/10 of the length/width of the environment, near the target position (at $(0, 0)$). The mission is for all UAVs to collect and herd the sheep swarm towards and until they reach the target. Each simulation run ends if the mission is successful, i.e. all sheep are collected and herded to the target, or the number of time steps reaches a limit of 2000.

A. States and Actions

Each UAV initially uses axis-parallel movements to travel to the back of the swarm of sheep until all sheep are in the third quadrant of the unit circle with the location of the UAV as the origin. The UAVs then deploy the formation using five production networks mentioned in Section III-B.

In this experiment, the number of inputs N_I is set to seven, where each subgoal production network takes a 7-component vector $(R_S, x_{ct}, y_{ct}, x_{sc}, y_{sc}, x_{scs}, y_{scs})$ as input states, which consists of both common and individual agent's state information:

- Common information:
  - *Radius of Sheep Flock* (R_S) (m): the distance between the sheep global centre of mass and the furthest sheep
  - *Direction from Sheep Global Centre of Mass to Target* (x_{ct}, y_{ct})

- Individual information:
  - *Direction from the Shepherd to Sheep Global Centre of Mass* (x_{sc}, y_{sc})
  - *Direction from the Shepherd to Global Centre of Mass of other Shepherds* (x_{scs}, y_{scs})

All states are divided by the Paddock Length L to scale the range of input to $[0, 1)$.

Each UAV uses one of the production network to generate a subgoal consisting of angular position θ and distance δ relative to the sheep swarm based on the inputs. The agent then computes its normalized force vector $F^t_{\beta_j sg}$ towards that assigned position, then the movement of the UAV is guided by:

$$F^t_{\beta_j} = F^t_{\beta_j sg} + W_{e\beta_j} F^t_{\beta_j \epsilon} \tag{10}$$

where $F^t_{\beta_j}$ is the total force exerted on the UAV β_j, $F^t_{\beta_j \epsilon}$ is the movement noise of the UAV and $W_{e\beta_j}$ is the strength of the noise.

Each UAV is assigned a $\pi/9$ (20 degree) operating angle relative to the sheep swarm at the LF, LMF, MF, RMF, or RF positions as demonstrated in Figure 1. The first output of the production network (in a range of $[-1, 1]$) is mapped to an adjusted angle in the range of $[-\pi/18, \pi/18]$. Within the assigned operating angle, the subgoal point lies on a subgoal line that forms with the bisector, at the middle of the corresponding section, an angle $\theta \in [-\pi/18, \pi/18]$, where a positive θ indicates the subgoal line is rotated an angle of $|\theta|$ counter-clockwise, and a negative θ indicates the subgoal line is rotated an angle of $|\theta|$ clockwise. The location of the subgoal on the subgoal line has the distance in the range $\delta \in [R_S + R_{\pi\pi}, R_S + 5R_{\pi\pi}]$ from the sheep global centre of mass.

B. Experimental Setups

The description of every configuration is shown in Table II, including the fixed formation baseline model and five dynamic formation learning models with different parameters alternatively fixed or optimized. The parameters to optimize include the weights w^k_{pj} and scaling factors α^k_j, $k = 1, ..., 5$, $p = 1, ..., N_I$, $j = 1, 2$, and the outputs for each Perceptron network are angular position θ and distance δ. For example, in the model named α, all weights w of Perceptron-Network are preset. Then for five Perceptron networks, five α values used in tangent activation function are the parameters required to be optimized with PSO algorithm. In this case, one output θ is computed while the distance δ is fixed for each network.

The parameters of PSO which are shown in Table III are selected based on the reference [22] and test experiments. The fitness function is the average distance each UAV travels divided by the size of the environment. Each particle of PSO generates a set of parameters whose fitness value is the average fitness over ten randomly initialized simulations. In total, it is required $20 \times 10 = 200$ simulations to compute the fitness values of all particles within a generation of size 20.

The best set of parameters found by the PSO in each configuration is tested on 100 randomly generated test scenarios. The performance of these models, introduced in Section IV-C, are then compared against the baseline fixed formation model.

The scalability of the learning algorithm is also tested with 7 and 9 shepherding agents with only five production networks. The models learned in cases of 50 sheep are then transferred to scenarios with 100 and 200 sheep to validate the sensitivity of the proposed method to an increase in the size of sheep swarm.

C. Evaluation Metrics

There are three assessment metrics for the proposed algorithm as below:

- **Number of steps**: the number of time steps for the sheep to be herded to the target location.
- **UAV's mean travel distance**: the distance that one UAV moves in the environment on average.
- **Radius of sheep swarm**. The radius of the sheep swarm can be displayed over time along with the formation of the UAVs to see how the formation created by the UAVs affects the collection and guidance of the sheep. The smaller the radius of the sheep swarm the better the formation. It is evaluated through visual inspection of the trajectory figures.

V. Results and Discussion

In this section, we analyze the performance of different training configurations when taking into account the different number of UAVs. The formation and sheep's movement of the baseline method and the best dynamic method obtained through learning are then investigated further to see how their proposed formations affect the trajectory and behaviour of the sheep. Finally, the sensitivity analysis of the dynamic method is conducted with an increase of sheep swarm size from 50 to 100 and 200 in order to evaluate the scalability of our proposed method.

A. Performance Analysis of Different Training Configurations

In all experiments of 50 sheep, all methods exhibit similar number of time steps on average. Table IV illustrates the number of steps as well as the mean travel distance of each UAV in the mission with five, seven and nine UAVs. Among dynamic formation learning methods, only the ones with α-δ (10 α with fixed weights, 2-output networks) or w-α-δ (80 parameters including 70 weights, 10 α with 2-output network) learned by the PSO algorithm achieve lower travel distance per UAV on average than the baseline method.

In particular, the w-α-δ configuration can be considered the best for all cases with five, seven, and nine UAVs operating on 50 sheep with lowest travel distance (ANOVA and 2-sample t-test at the significance level of either 0.05 or 0.01). Due to the fact that the travel distance is proportional to the power consumption of the UAV, our proposed algorithm with optimized values of both weights w and scaling factor α to adjust both the dynamic angle and distance from each UAV relative to the sheep's GCM is more efficient than the baseline fixed formation.

TABLE II: Configurations of experiments.

Method ID	#Outputs per network	Weights	Activation	Distance	#Optimized parameters in total
baseline	N/A	N/A	N/A	$\forall k\ \delta^k = R_S + 3R_{\pi\pi}$	0
α	1 (θ)	$\forall(k,p)\ w_p^k = \frac{1}{7}$	$\alpha_1^k \in [-10, 10]$	$\forall k\ \delta^k = R_S + 3R_{\pi\pi}$	5 (α_1^k)
w	1 (θ)	$w_{p1}^k \in [-1, 1]$	$\forall k\ \alpha^k = 1$	$\forall k\ \delta^k = R_S + 3R_{\pi\pi}$	35 (w_{p1}^k)
α-δ	2 (θ,δ)	$\forall(k,p)\ w_{p1}^k = w_{p2}^k = \frac{1}{7}$	$\alpha_1^k, \alpha_2^k \in [-10, 10]$	$\delta^k \in [R_S+R_{\pi\pi}, R_S+5R_{\pi\pi}]$	10 (α_1^k, α_2^k)
w-δ	2 (θ,δ)	$w_{p1}^k, w_{p2}^k \in [-1, 1]$	$\forall k\ \alpha_1^k = \alpha_2^k = 1$	$\delta^k \in [R_S+R_{\pi\pi}, R_S+5R_{\pi\pi}]$	70 (w_{p1}^k, w_{p2}^k)
w-α-δ	2 (θ,δ)	$w_{p1}^k, w_{p2}^k \in [-1, 1]$	$\alpha_1^k, \alpha_2^k \in [-10, 10]$	$\delta^k \in [R_S+R_{\pi\pi}, R_S+5R_{\pi\pi}]$	80 (α_1^k, α_2^k, w_{p1}^k, w_{p2}^k)

where $k = 1, ..., 5;\ p = 1, ..., 7$

TABLE III: The parameters in PSO.

Parameter	Meaning	Value
N_p	The population size	20
Max_{gen}	The maximum number of generations	100
c_1, c_2	Two acceleration constants	2
w_{max}	The maximum/initial inertial weight	0.9
w_{min}	The minimum/final inertial weight	0.4

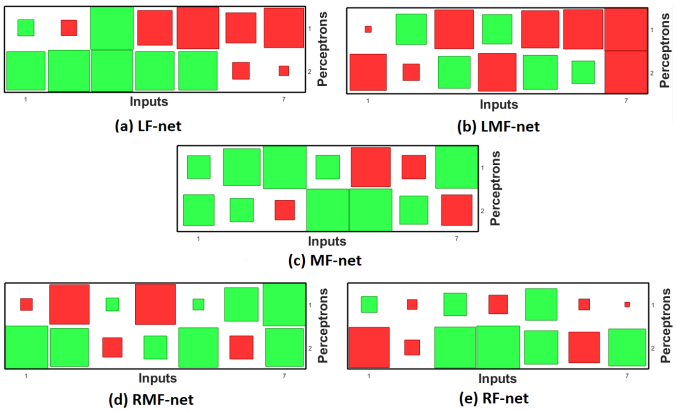


Fig. 3: The Hinton diagrams of weights matrices of five w-α-δ perceptron-networks with 7 inputs (columns) and 2 outputs (rows) in the experiment involving 5 UAVs and 50 sheep. The green and red squares represent the positive and negative values respectively. The larger the square the larger the absolute value of the weight.

On the other hand, when increasing the number of UAVs, the mean increase in the travel distance per UAV with baseline method is approximately 4 to 5 meters. The travel distances of UAVs using dynamic methods also increase along with the increase in the number of UAVs. Most are equivalent or even worse than the baseline method, except the w-α-δ model with less than two meters for each UAV on average. Thus, the scalability of the dynamic formation learning method with w-α-δ configuration is higher than that of the baseline method. The insignificant changes of the number of time steps and the travel distance per UAV suggest that the model learned with our proposed method can be applied to a larger number of UAVs.

B. Learned Weights of Specialized Networks

We further investigate different weight matrices of five specialized w-α-δ networks to understand how the networks are tailored to produce adaptive behaviour for their own operating areas. Figure 3 shows the weights between seven inputs and two outputs of every network. Positive weights (green colors) indicate positive reinforcement of the corresponding inputs and their positive correlation to the output, while negative weights indicate negative reinforcement.

Each network learns different weight profiles toward its operating area in the formation. For example, given an increase in radius of the sheep swarm (first input), the LF-net, MF-net, and RF-net tend to direct the UAVs slightly to counter-clockwise side while the rest to clock-wise side. In this case, the output distances (second outputs) of LF, MF, RMF UAVs increase significantly while the distances of LMF and RF ones decrease. Some UAVs on both left and right sides change their distance to effectively collect back the sheep while others shrink closer to maintain the driving force to the whole sheep swarm. In another instance, given the increase of the y-coordinate of the vector between a UAV and the local centre of mass of other UAVs (7th input), i.e. the UAV of interest tends to move north of the other UAVs' neighborhood, the subgoals of LF and LMF UAVs (located at the north relative to others) are likely to move clockwise toward the south to maintain the formation. On the other hand, the MF, RMF, and RF UAVs, which stay at the south of the neighbourhood, lightly move south or stay at the same angle.

Our reliance on a simple Peceptron enabled the interpretation above to ensure transparency of the formation model, while allowing formation to adapt in real-time.

C. The Influence of Dynamic Formation on Swarm's Movement

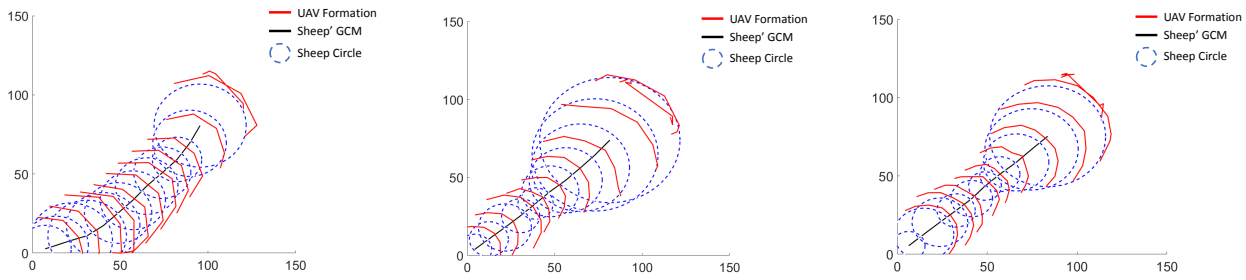
The fixed and dynamic formations have different effects on the collecting and herding ability of the UAV swarm. Figure 4 and 5 show the formations of UAV swarm and the movement of sheep swarm under influence from a fixed formation (baseline) and a dynamic formation produced from Perceptron network optimized by PSO with the w-α-δ configuration.

The movement of the sheep swarm, reflected by the trajectory of the sheep's GCM, in 5 UAV case is less optimal than the one produced by the dynamic formation. In all experiments, the rates of decrease of the sheep's radius are similar, i.e. both methods can collect the sheep effectively. However, while the baseline cannot showcase the stabilization

TABLE IV: Number of steps and travel distance per UAV in 6 different configurations with 5, 7 and 9 UAVs operating on 50 sheep.

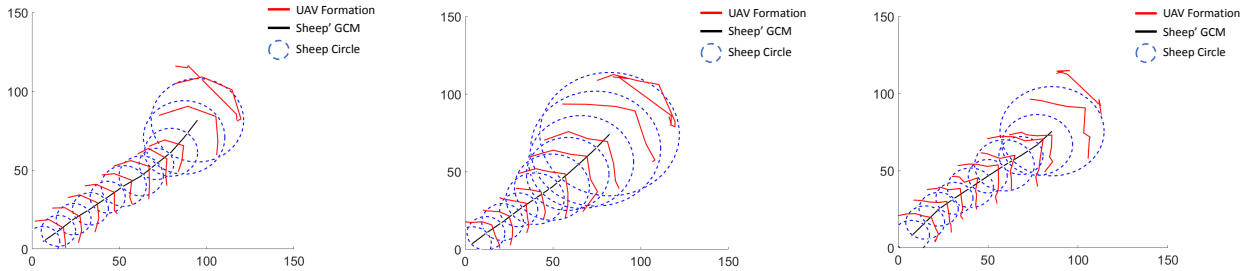
Method	Number of UAVs					
	5		7		9	
	#steps	Travel distance/UAV	#steps	Travel distance/UAV	#steps	Travel distance/UAV
baseline	245.70±13.52	431.93±16.77	246.42±10.79	435.61±15.48	246.23±11.41	436.85±16.81
α	245.55±13.37	430.92±17.15	245.87±10.83	435.42±15.64	245.89±8.73	436.68±13.13
w	245.73±11.50	432.48±15.06	245.28±13.73	433.72±17.24	247.46±14.17	438.70±20.57
α-δ	247.90±10.39	425.48±13.17	247.28±11.22	430.69±13.30	248.74±13.32	436.44±17.54
w-δ	245.64±13.14	431.18±18.59	249.14±11.99	438.81±16.70	248.47±18.00	438.60±22.08
w-α-δ	247.65±17.77	**418.87±21.35**	248.66±12.95	**420.55±16.51***	245.68±12.24	**421.73±15.63***

The figures in **bold** are better than their counterparts at significant level of 0.05.
The figures with * are better than their counterparts at significant level of 0.01.



(a) 5 UAVs (b) 7 UAVs (c) 9 UAVs

Fig. 4: Formations of UAVs (red) and the circles containing the whole sheep swarm (blue) of size 50 with baseline method.



(a) 5 UAVs (b) 7 UAVs (c) 9 UAVs

Fig. 5: Formations of UAVs and the circles containing the whole sheep swarm (blue) of size 50 with dynamic method using PSO to optimize w-α-δ configuration.

of the sheep swarm's radius when driving them, our proposed method can maintain a small sheep radius until the end of the mission. When the sheep's circle expands as a result of instability of baseline method, the UAVs can be forced to hover or fly backward to maintain a certain distance with the sheep's GCM, which raises the travel distance of the UAVs.

On the other hand, the dynamic formation created by our optimized Perceptrons can adapt to the complex movement of the sheep within the swarm, approach and modify the force exerted on sheep to maintain the geometry and steady speed of the sheep swarm. Hence, the travel distance of each UAV can be reduced effectively by our proposed method.

D. Transfer Learning in Dynamic Formations

We conduct further studies in which we transfer the model trained on a scenario where the size of sheep swarm is 50 to cases of larger sizes of sheep swarm. The performance of the model for 100 and 200 sheep are still equivalent to the performance of the baseline fixed model method (Table V). The travel distance of each UAV on average for the specific case of nine UAVs and 200 sheep is even significantly lower than that of the baseline method (two-sample t-test at significant level of 0.05). It can be concluded that even without the retraining, the dynamic property of the formation produced by the proposed method is elastic and is able to adapt to the change in the size of sheep swarm, which provides equivalent or even better performance than the baseline method.

VI. Conclusion and Future Work

In this paper, we introduce a swarm-on-swarm guidance with the use of UAVs as shepherds to influence a large size of sheep swarm to guarantee a safe operation on a simulation environment. Our proposed framework, producing forces for UAVs to control their formation, uses five Perceptron model whose outputs are the angles and the distances of the UAVs relative to the sheep swarm. Each Perceptron-network is

TABLE V: The number of steps and travel distance per UAV on average achieved by different number of UAVs with an increase in the number of sheep.

#sheep	Method	Number of UAVs					
		5		7		9	
		#steps	Travel distance per UAV	#steps	Travel distance per UAV	#steps	Travel distance per UAV
100	baseline	323.28±29.11	509.35±29.58	336.74±32.90	529.99±35.44	333.32±34.02	529.29±36.65
	w-α-δ	327.39±32.22	507.31±30.75	343.72±46.03	524.71±46.79	340.82±43.48	525.28±43.95
200	baseline	348.72±32.34	535.79±33.44	366.29±22.01	560.10±24.45	369.96±27.77	567.83±29.26
	w-α-δ	355.88±36.79	538.36±37.80	368.46±32.30	553.89±30.01	367.06±36.09	**553.48±37.58**

The figures in **bold** are better than their counterparts at significant level of 0.05.

evolved using PSO algorithm to specialize in different relative positions in the formation, which significantly reduces the search space for optimization, while still provide scalability to different UAVs' fleet sizes.

Compared to the fixed formation produced by the baseline method, our proposed model, with both learned weights and scaling factors of the activation functions for outputting subgoals with changing angles and distances, achieves better performance in terms of travel distance by the UAVs which can reduce the power usage. When testing the models with more than five UAVs, the model scales up better than the other methods.

Moreover, the investigation on the shape of the formation and the movement of sheep swarm over time confirms that the dynamic formation learning model produces more flexible formations of UAVs and reduces fluctuation of the sheep swarm. It is also concluded from the results that our proposed method has potential to transfer to the environment with larger number of sheep.

Our next step is to transfer the model to the Gazebo simulation environment before implementing the model in our UAV testing facilities. Currently, the model works on a level of abstraction that is transferable to our UAV environment with autonomy level 3. If the autonomy level is lower, the level of abstraction of the current study may not be appropriate and may not transfer easily.

REFERENCES

[1] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.

[2] R. Carelli, C. De la Cruz, and F. Roberti, "Centralized formation control of non-holonomic mobile robots," *Latin American applied research*, vol. 36, no. 2, pp. 63–69, 2006.

[3] H. Oh, A. R. Shirazi, C. Sun, and Y. Jin, "Bio-inspired self-organising multi-robot pattern formation: A review," *Robotics and Autonomous Systems*, vol. 91, pp. 83–100, 2017.

[4] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[5] A. Guillet, R. Lenain, B. Thuilot, and V. Rousseau, "Formation control of agricultural mobile robots: A bidirectional weighted constraints approach," *Journal of Field Robotics*, 2017.

[6] D. Xu, X. Zhang, Z. Zhu, C. Chen, and P. Yang, "Behavior-based formation control of swarm robots," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[7] T. Nguyen, H. Nguyen, E. Debie, K. Kasmarik, M. Garratt, and H. Abbass, "Swarm Q-learning with knowledge sharing within environments for formation control," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[8] G. Palmer, K. Tuyls, D. Bloembergen, and R. Savani, "Lenient multi-agent deep reinforcement learning," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 443–451.

[9] C. Speck and D. J. Bucci, "Distributed UAV swarm formation control via object-focused, multi-objective sarsa," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 6596–6601.

[10] D. Strömbom and A. J. King, "Robot collection and transport of objects: A biomimetic process," *Frontiers in Robotics and AI*, vol. 5, p. 48, 2018.

[11] P. Nalepka, R. W. Kallen, A. Chemero, E. Saltzman, and M. J. Richardson, "Practical applications of multiagent shepherding for human-machine interaction," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2019, pp. 168–179.

[12] D. J. Cohen, W. J. Nelson, and M. M. Maharbiz, "Galvanotactic control of collective cell migration in epithelial monolayers," *Nature materials*, vol. 13, no. 4, p. 409, 2014.

[13] D. Strömbom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. Sumpter, and A. J. King, "Solving the shepherding problem: heuristics for herding autonomous, interacting agents," *Journal of the royal society interface*, vol. 11, no. 100, p. 20140719, 2014.

[14] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3402–3407.

[15] W. Lee and D. Kim, "Autonomous shepherding behaviors of multiple target steering robots," *Sensors*, vol. 17, no. 12, p. 2729, 2017.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[17] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. IEEE, 1995, pp. 39–43.

[18] C. W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*. ACM, 1987, vol. 21, no. 4.

[19] A. Schultz, J. J. Grefenstette, and W. Adams, "Roboshepherd: Learning a complex behavior," *Robotics and Manufacturing: Recent Trends in Research and Applications*, vol. 6, pp. 763–768, 1996.

[20] J.-M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 4159–4164.

[21] B. Bennett and M. Trafankowski, "A comparative investigation of herding algorithms," in *Proc. Symp. on Understanding and Modelling Collective Phenomena (UMoCoP)*, 2012, pp. 33–38.

[22] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. IEEE, 1998, pp. 69–73.