The University of Manchester Research

Bio-inspired Collision Avoidance in Swarm Systems via Deep Reinforcement Learning

DOI:
10.1109/TVT.2022.3145346

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):
Na, S., Niu, H., Lennox, B., & Arvin, F. (2022). Bio-inspired Collision Avoidance in Swarm Systems via Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology*, *71*(3), 2511-2526. https://doi.org/10.1109/TVT.2022.3145346

Published in:
IEEE Transactions on Vehicular Technology

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

OPEN ACCESS

Download date:28. Sept. 2023

Bio-inspired Collision Avoidance in Swarm Systems via Deep Reinforcement Learning

Seongin Na, *Student Member, IEEE*, Hanlin Niu, *Member, IEEE*, Barry Lennox, *Senior Member, IEEE*, and Farshad Arvin, *Senior Member, IEEE*

***Abstract*—Autonomous vehicles have been highlighted as a major growth area for future transportation systems and the deployment of large numbers of these vehicles is expected when safety and legal challenges are overcome. To meet the necessary safety standards, effective collision avoidance technologies are required to ensure that the number of accidents are kept to a minimum. As large numbers of autonomous vehicles, operating together on roads, can be regarded as a swarm system, we propose a bio-inspired collision avoidance strategy using virtual pheromones; an approach that has evolved effectively in nature over many millions of years. Previous research using virtual pheromones showed the potential of pheromone-based systems to maneuver a swarm of robots; however, designing an individual controller to maximise the performance of the entire swarm is a major challenge. In this paper, we propose a novel deep reinforcement learning (DRL) based approach that is able to train a controller that introduces collision avoidance behaviour. To accelerate training, we propose a novel sampling strategy called *Highlight Experience Replay* and integrate it with a Deep Deterministic Policy Gradient algorithm with noise added to the weights and biases of the artificial neural network to improve exploration. To evaluate the performance of the proposed DRL-based controller, we applied it to navigation and collision avoidance tasks in three different traffic scenarios. The experimental results showed that the proposed DRL-based controller outperformed the manually-tuned controller in terms of stability, effectiveness, robustness and ease of tuning process. Furthermore, the proposed Highlight Experience Replay method outperformed than the popular Prioritized Experience Replay sampling strategy by taking 27% of training time average over three stages.**

***Index Terms*—Collision Avoidance, Autonomous Vehicles, Multi-agent systems, Deep Reinforcement Learning, Swarm Robotics.**

I. Introduction

WITH the emergence of machine learning and other new technologies, the development of autonomous vehicles has been drawing greater attention as a possible method of future transportation [1]. The wide deployment of autonomous vehicles is expected to bring invaluable benefits to society across diverse aspects including reduced traffic congestion, savings in fuel / battery usage and most importantly, increased safety. However, if autonomous vehicles are to receive widespread acceptance for use on roads then robust collision avoidance mechanisms in general operating environments are essential [2], [3].

Researchers in the field of autonomous vehicles have proposed a wide range of collision avoidance mechanisms, which have been based on numerous techniques including control theory [4], potential fields [5], trajectory planning [6], adaptive control [7], [8], velocity-based approaches [9]–[11] and biologically-inspired method [12]. More recently, deep learning based approaches have been proposed that utilise the power of deep learning to derive optimal controllers from large amounts of data that minimise collisions involving autonomous vehicles [13], [14]. For example, in [15], DRL was used by multi-agents to avoid collisions and improve navigation tasks.

Large numbers of autonomous vehicles operating together on a network of roads can be regarded as a swarm, where a swarm is a system that consists of a large number of agents, controlled in a decentralised manner using local communication between individual agents or agents and the environment, that work towards a common goal [16]. Swarms are frequently observed in nature where they perform complex tasks using large numbers of agents, all with limited capabilities, such as the colonies of ants, bees and termites. Since autonomous vehicles can be controlled independently and use local communication to achieve a common goal, e.g. minimising collisions, whilst optimising other driving factors, such as congestion and travel times, they can be regarded as a swarm system.

One of the effective communication strategies that swarm systems utilise in nature is pheromone-based communication [17]. A pheromone is a chemical substance that is released by an individual to trigger immediate behavioural or developmental changes in other individuals who sense it [18]. One of the great advantages of using pheromones to communicate is that it allows scalable collective behaviours without requiring direct communication between the agents. For example, both a single ant and a large number of ants operating in a collective can perform foraging using the same pheromone trail without the need to increase communication costs [19].

Furthermore, pheromone communication allows the optimisation of collective performance utilising feedback mechanisms that use combinations of pheromones that can each introduce different behaviours [20]. These benefits mean that

This work was supported by EU H2020-FET-OPEN RoboRoyale project [grant number 964492] and the Engineering and Physical Sciences Research Council (EPSRC) RAIN and RNE projects [grant numbers EP/R026084/1 and EP/P01366X/1]. The first author would like to express special thanks to the University of Manchester for supporting his PhD study with the President's Doctoral Scholar Award.

All authors are with the Swarm & Computational Intelligence Lab (SwaCIL) at the Department of Electrical and Electronic Engineering, School of Engineering, University of Manchester, Manchester M13 9PL, U.K. (e-mail: {seongin.na, hanlin.niu, barry.lennox, farshad.arvin}@manchester.ac.uk)

when compared to traditional methods of communication, pheromone-based techniques can be more suitable when scalability and adaptability are required with large numbers of agents in a swarm. Pheromone-based communications are therefore suitable in various real-world applications. For example, in [21], the effectiveness of pheromone-based communications was demonstrated in an urban waste management application where the approach was shown to outperform other more traditional waste management techniques.

A wide range of domains including optimisation [22], vehicle routing [23] and robotics [24]–[27] have recently adopted pheromone-based communication systems to address particular challenges. Despite the effectiveness of pheromone based communications, significant challenges remain if this technique is to be adopted more widely. One particular challenge is the design of controllers for the individual agents within a swarm, that are able to maximise the performance of the entire swarm [28]. Traditional controller design methods, so-called manually-designed controllers, use heuristic approaches to specify the individual controllers. However, the difficulty of doing this increases as the complexity of the task and environment increases [29]. To alleviate the design challenges, automated design methods have been proposed to enable individual agents to learn optimal control rules without external intervention [30]. To date, these techniques have utilised traditional, local inter-agent communication strategies, and whilst showing promising results in realistic environments [31], they have not exploited the potential benefits offered through the use of pheromone-based techniques.

Motivated by the scalability and adaptability of pheromone-based communication and its benefits for real-world applications, we propose a novel bio-inspired pheromone-based collision avoidance scheme for swarm systems. Moreover, to further enhance controller to be adaptive in complex and dynamic environment, we propose a DRL-based controller for the individual agents in a pheromone-based swarm system. This paper makes three major contributions as follows:

- We propose a bio-inspired PhERS (Pheromone for Every Robot Swarm) framework for a collision avoidance scheme in swarm systems. The framework is validated using three traffic scenarios in a realistic simulated environment (Gazebo). Through experiments, the controllers using this framework showed higher flexibility than the traditional centralised control method.
- We propose a DRL-based controller for pheromone-based swarm systems. The DRL-based controller is designed to provide collision avoidance and navigation in pheromone-based swarm systems. Comparisons show that the proposed controller performs greater than the traditional centralised controller, NH-ORCA [10], and the manually-tuned controller.
- A novel bio-inspired sampling strategy for experience replay buffer, Highlight Experience Replay (HLER), is proposed and integrated with the Deep Deterministic Policy Gradient algorithm (DDPG). The obtained results from this work showed the proposed HLER sampling strategy outperformed the Prioritized Experience Replay (PER) [32] sampling strategy in terms of training speed reducing to average 27% of training time with PER in three experimental stages. We further increased the training speed of HLER by integrating the Gaussian noise on parameters of a neural network (NN) to incentivise exploration for finding a better policy rapidly [33].

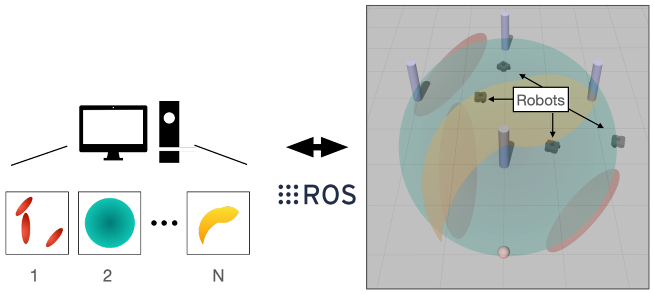


Fig. 1. An overview of PhERS framework. Multiple pheromone grids (1,2,...,N) are computed in the main PhERS controller and integrated with the environment, which retrieves and updates pheromone intensity values requested as a ROS message.

We believe that this research encourages pheromone-based swarm systems and use of DRL-based controller design methods to be used in a broader range of tasks and in more diverse and complex environments, such as those found in agriculture and terrestrial exploration as well as autonomous vehicles.

II. Artificial Pheromone Framework

We propose the PhERS (Pheromone for Every Robot Swarm) framework, which provides an augmented environment where virtual pheromones can be utilised in real and simulated environments for diverse applications including collision avoidance systems in autonomous vehicles. Figure 1 illustrates the general overview of the proposed framework. Within the framework, virtual pheromones are managed inside each of the virtual grids, which act as fields of pheromones having different characteristics. The framework is able to incorporate a varying number of virtual grids and hence a variety of virtual pheromones can be defined, which helps to simulate complex pheromone-based behaviours. For example, if repellent and attractive pheromones are utilised simultaneously, then the complex trail traffic management systems of ants can be replicated.

Figure 2 illustrates the proposed architecture of the PhERS framework. It consists of three parts: i) main PhERS controller, ii) data storage and iii) communication network. The main PhERS controller orchestrates all the pheromone grids (Phero-Grid) in the system. When the main PhERS controller is initialised, it creates the specified number of Phero-Grids, containing the individual characteristics requested. After initialisation, the controller updates the Phero-grids at each time-step, based on the spatio-temporal development model of each pheromone. The data storage is used to store the data contained within the Phero-grids, which represents the intensity of each of the pheromones, in the main PhERS controller at the specific time-step. The stored Phero-grid data

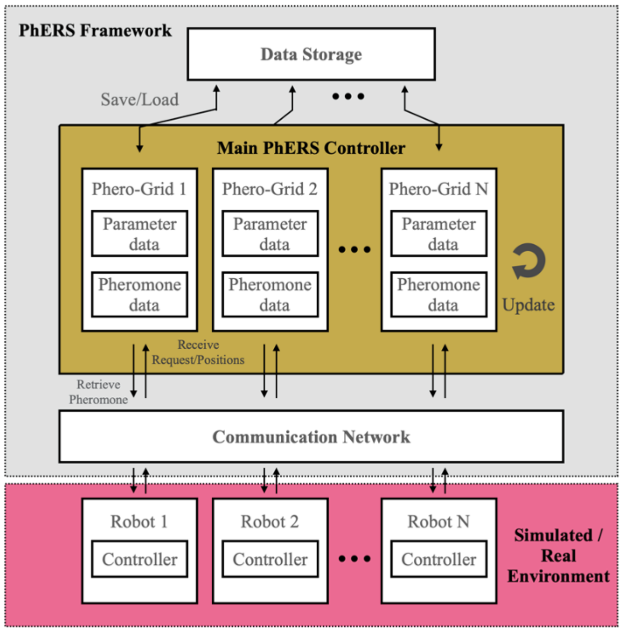


Fig. 2. Architecture of PhERS framework. The framework consists of (i) data storage, (ii) main controller and (iii) communication network. Phero-Grid represents an entity that stores parameter settings and pheromone data.

can be retrieved when requested. The communication network is used to manage communication between the main PhERS controller and the agents. When the agents read pheromone values, the positional information of the agents is sent to the main PhERS controller via the communication network using Robotic Operating System (ROS) messages. The main PhERS controller then retrieves the requested pheromone data from the agent through the communication network. Likewise, the communication network sends pheromone information from the agents to the main PhERS controller so that the released pheromone data is applied to the Phero-grids. A basic version of PhERS was tested for a simple environment scenario in [34].

In this work, the framework was implemented on a PC and with virtual pheromones created using a popular simulated environment (Gazebo). For future deployments on autonomous vehicles, it is feasible that the framework could be implemented on a cloud server. The virtual pheromone grids that are applied to the environment could be generated and updated in the cloud system using real-time communication with the vehicles. Using future generation communications, such as 5G, and high-performance cloud servers, the application of virtual pheromones to a large number of fast moving autonomous vehicles, with minimal delays is feasible [35], [36]. Moreover, the virtual pheromones can be generated and managed by the traffic infrastructure. The development of sensing technology and vehicle-to-infrastructure communication systems enabled to propose smart roads [37]. With the smart road technologies, the virtual pheromones can be implemented and managed by local smart traffic infrastructures in a decentralised manner rather than using a centralised server such as a cloud system.

There are several advantages for using PhERS for collision avoidance compared with other more traditional collision avoidance methods. For the traditional collision avoidance using centralised control, our pheromone-based communication method is more flexible to any increase in environmental complexity. The PhERS framework provides decentralised control and so can manage more effectively when the number of agents is large and/or varying, i.e. it has high scalability. Moreover, for the traditional collision avoidance using decentralised control, and using different sensory devices, e.g. LiDAR and camera, the PhERS-based collision avoidance system has the benefit that it can represent future collision hazards that cannot be detected using the sensory devices used when applying the traditional methods. For example, by marking the trail of a vehicle using a slow volatile pheromone, the vehicle can ensure there is a safe distance from other following vehicles.

Although the PhERS framework is able to replace the traditional collision avoidance methods using individual sensory devices such as cameras and LiDARs, it is more beneficial to be used as a higher level system for the traditional collision avoidance. The higher level system can be defined as traffic level system while the traditional individual level system can be named as vehicle level system. When used together with traditional sensory devices, it can cover more immediate collision avoidance behaviours that require minimal delay at a vehicle level and the PhERS-based system can be used to cover more complex situations at a traffic level. By synthesising the two methods, the vehicles can prevent collisions when the traditional sensory devices are damaged or malfunctioning.

The virtual pheromones in the PhERS framework share the same spatio-temporal update model. The pheromone grid, managed by the main PhERS controller, is composed of a 2D matrix, represented as $\mathbf{\Phi}$, that stores the pheromone concentration at any given position in the Cartesian coordinate system. Each cell of the grid represents a corresponding discrete cell in the real environment. Equation (1) describes the spatio-temporal update rule of the virtual pheromones.

$$\begin{aligned}\mathbf{\Phi}^{k+1}(x,y) = &-\mathbf{u}\cdot\nabla\mathbf{\Phi}^{k}(x,y) - \frac{\ln(2)}{e_{\Phi}}\mathbf{\Phi}^{k}(x,y)\\ &+\kappa\triangle\mathbf{\Phi}^{k}(x,y) + \iota(x,y) \quad ,\end{aligned} \tag{1}$$

where $\mathbf{\Phi}^{k+1}(x,y)$ is the pheromone concentration at the discrete time $k+1$, $\mathbf{\Phi}^{k}(x,y)$ is the pheromone concentration at position (x,y) at discrete time k. This equation is derived from the Navier-Stokes equation, describing fluid flow [38]. In Equation (1), each term on the right hand side represents phenomena that influence the pheromone concentration in the real world, which are: i) the effect of wind, ii) evaporation, iii) diffusion and iv) injection. The following list defines the parameters of each phenomena.

- Wind vector, $\mathbf{u}$, characterises the direction and velocity of the wind that acts on the released pheromone.
- Evaporation half-life, e_{Φ}, determines the time taken for the pheromone concentration to be halved.
- Diffusion constant, κ, defines the rate of diffusion in a given time period.

- Injection function, ι, defines the shape and intensity of the injected pheromone at a specific time step, an example of which is given in Equation (2).

The following equation describes the injection function used in this work. We assume that the injected pheromone has a circular shape emanating from the injection point of the agent.

$$\iota_i(x,y) = \begin{cases} s_\Phi, & \text{if } \sqrt{(x-x_r)^2+(y-y_r)^2} \leq r_\Phi \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where s_Φ represents the injection rate of the pheromone at a given time step, (x_r, y_r) is the Cartesian coordinates of the robot or agent, and r_Φ is the radius of injection.

III. Deep Reinforcement Learning based Controller

DRL-based control can be applied to individual agents in a swarm to automatically identify appropriate controller rules. As it is difficult to manually design controllers in a complex and dynamic scenario, DRL-based controller is proposed to overcome the limitation of manually designed controllers. In this section, we introduce the background of DRL and explain the proposed DRL algorithm that was used in our experiments.

A. Background

Reinforcement learning (RL) is an automated approach for finding the best solution to a task by maximising a numerical reward signal [39]. In RL, a subject that performs a task is defined as an agent. An agent interacts with the environment and obtains transition samples. Based on the current state information, an agent decides what action it should take and in response the environment returns a numerical reward signal and information regarding the next state. This process involves the transfer of information relating to the transition from one state to another, as well as action and reward data and hence is referred to as a transition sample. Using the transition samples, RL can be applied to find the best solution for an agent to maximise total cumulative reward, i.e. an optimal policy, where policy refers to the mapping between states and actions.

When the number of possible states and actions for an agent in an environment is relatively small, an agent's policy can be represented as a table that maps states and actions exhaustively. However, when states and actions have continuous values or the numbers are very large, it is not feasible to represent this information in such a table. In this case, deep neural networks can be used as a function approximator. The approach of using deep neural networks for reinforcement learning problem is referred to as DRL [40].

Current DRL algorithms can be divided into three branches depending on which functions are being approximated by the neural networks, which are used to derive the optimal policy. The first approach, referred to as policy-based method, uses the neural network to approximate the policy function and then the subsequent policy network is optimised [41], [42]. The second approach referred to as value based method, trains a value network to derive an optimal policy [40]. In this approach the value network indicates how good it is for an agent to be at a particular state. Using the value information, the policy can be driven in a way that leads the agent to the states with the highest values. The third branch is actor-critic method [43]. This method trains both policy and value networks and uses them both to derive an optimal policy. The policy and value networks are referred to as actor and critic networks as they play the roles of actor (take actions) and critic (evaluate the actions given states).

DDPG [44] is an actor-critic based DRL algorithm that can be used for continuous state and action space environments, shown to achieve high levels of performance in continuous control tasks in both simulated and real-world applications [13]. As an important characteristic, DDPG uses an experience replay buffer to store and reuse transition samples to improve sample efficiency. Furthermore, through selecting transition samples randomly from the buffer, we can ensure the training data are not correlated, through time, as consecutive transition samples will be. Since correlated data can lead to failure in the training of the neural network, the use of an experience replay buffer is essential [40].

Several research works have proposed diverse sampling strategies improving a basic uniform sampling for experience replay buffer to improve learning performance, e.g. hindsight experience replay [45], attentive experience replay [46] and prioritised experience replay [32]. The sampling strategies vary depending on the optimisation objectives. For example, hindsight experience replay is specialised for multi-goal learning scenarios. Prioritised Experience Replay (PER) [32] is one of the most popular sampling strategy used widely to improve performance and training time. This strategy measures the importance of transitions using the magnitude of temporal difference error (TD-error) to effective neural network model update. The probability of sampling transition i is defined as:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \tag{3}$$

where α is a probability constant. Higher priority leads to a higher probability of that sample being used during training. Higher sample efficiency and performance has been reported using PER, compared to uniform sampling strategies. Due to its sample efficiency, this sampling strategy is used with DDPG in several continuous control domains including robotics [13], [47].

B. Proposed Strategy

In this work, we propose a new sampling strategy using the concept of prioritising the samples. In the proposed method, transition samples that have higher absolute value of rewards than most of the samples are prioritised. As these transitions can be considered the "highlights" of the episode, the proposed method is referred to as HighLight Experience Replay (HLER). HLER is inspired from the fact that human brains can recall highlight events more effectively than neutral events [48]. For example, in a study using brain imaging, it was found that images with negative emotions were retrieved more effectively than images that produced neutral emotions [49]. Similarly, increasing brain activity was observed during the retrieval of positive contextual information, compared with neutral information [50]. The expectation with using HLER was that it would aid the neural network to be trained more effectively on samples associated with high absolute reward. While PER focuses on selecting samples with high TD-error for faster neural network optimisation, HLER focuses on selecting samples with high absolute reward to learn the highlight event faster. Our hypothesis is that, HLER is more efficient sampling strategy for our scenario, collision avoidance of autonomous vehicle, than the PER sampling strategy as a collision can be regarded as a highlight.

The proposed HLER strategy assigns a priority to each transition sample in a replay buffer, collected from K actors. The priority of the i^{th} sample, p_i is assigned as follows:

$$p_i = \begin{cases} k_{HL} \cdot p_{base}, & |r_t| \geq r_{HL} \\ p_{base}, & \text{otherwise} \end{cases}, \tag{4}$$

where p_{base} is the default priority, k_{HL} is a coefficient that is applied to samples that are considered highlights, and r_{HL} is a reward value to determine whether the sample is considered a highlight or not.

When p_i is specified to be equal to $k_{HL} \cdot p_{base}$, the priority of the previous l number of samples, $p_{i-l+1:i}$, are also assigned values of $k_{HL} \cdot p_{base}$. This defines the previous l samples as the moments contributing to the highlight sample. The sampling probability of the i^{th} sample in HLER sampling strategy follows (3) as PER.

Compared to PER, the HLER sampling strategy is considered to be more effective at sampling the important transition data that contains the type of behaviour that the neural network needs to learn, e.g. collision avoidance. For tasks where there is a clear distinction between important and neutral behaviours, HLER can accelerate training speed and lead to higher cumulative reward during an episode, i.e. movement from an initial to a terminal state.

To further improve the exploration capability of the DRL algorithm, noise was added to the neural network parameters, as suggested using the "noisy network" approach [33]. The concept of noisy networks is that by adding Gaussian noise to the weight and bias values in each linear layer of NN, stochastic features are incorporated. A linear layer of a neural network with p input and q output can be represented as follows:

$$y = wx + b, \tag{5}$$

where $x \in \mathbb{R}^p$ is the layer output, $w \in \mathbb{R}^{p \times q}$ represents the weight matrix and the bias is denoted by $b \in \mathbb{R}^q$. However, the linear layer of a noisy neural network can be characterised as follows:

$$y := (\mu^w + \sigma^w \odot \epsilon^w)x + \mu^b + \sigma^b \odot \epsilon^b, \tag{6}$$

where $\mu^w \in \mathbb{R}^{q \times p}, \mu^b \in \mathbb{R}^q, \sigma^w \in \mathbb{R}^{q \times p}$, $\sigma^b \in \mathbb{R}^q$ are the learnable parameters, ϵ^w and ϵ^b are the adaptive Gaussian noise values that are added to the weight and bias of the layer and $\odot$ is an element-wise multiplication operator. A noisy linear layer was applied to the actor network, to help improve the exploration ability of the agent. Unlike ϵ-greedy [39], another popular exploration method, the noisy network approach generates state-dependent noise, which helps reduce undesirable exploration in the states that do not require exploration (e.g. states where the next state causes a large penalty). Moreover, the noisy network approach can be used for different DRL problems as the parameter noise are adapted over training, while the fixed Gaussian parameter space noise approach [51] requires manual fine-tuning of the noise parameters. For these reasons, we chose the noisy network approach for exploration in our work to effectively learn collision avoidance behaviour which requires adaptive state-dependent exploration. For example, a high degree of exploration in the early stage is required for effective collision avoidance and low degree of exploration is needed after discovery of the optimal policy on the states near collision states.

The DDPG algorithm with the HLER sampling strategy

Algorithm 1: DDPG algorithm with HLER sampling strategy and noisy network.

```
1  Randomly initialise critic neural network with normal
     linear layers, $Q(s,a|\theta^Q)$ and actor neural network
     with noisy linear layers $\pi(s|\theta^\pi)$ with weights $\theta^Q$ and
     $\theta^\pi$
2  Initialise target network $Q'$ and $\pi'$ with weights $\theta^{Q'}$
     $\leftarrow \theta^Q$, $\theta^{\pi'} \leftarrow \theta^\pi$
3  Initialise replay buffer $R$
4  for episode = 1, ..., M do
5      Initialise the states $s_t = s_1$
6      for t = 1, ..., T do
7          Run $K$ actors and collect transition samples $D_t$
             = $(s_t, a_t, r_t, s_{t+1})$ into a replay buffer
8          if $|r_t| \geq r_{HL}$ then
9              Assign priority $k_{HL} \cdot p_{base}$ for $D_{t-l:t}$
10         else
11             Assign priority $p_{base}$ for $D_t$
12         end
13         if t = 0 mod $t_{train}$ then
14             Sample $N$ transitions $(s_i, a_i, r_i, s_{i+1})$ from
                 a replay buffer with the sampling
                 probability $P(i)$
15             Set $y_i = r_i + \gamma Q'(s_i, \pi'(s_{i+1}|\theta^{\pi'})|\theta^{Q'})$
16             Set weighted updates for networks:
                 $\omega_i = (\frac{1}{B} \cdot \frac{1}{P(i)})^\beta$
17             Update critic network by minimizing the
                 loss: $L_Q = \frac{1}{N}\omega \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
18             Update actor network using the sampled
                 policy gradient: $\nabla_{\theta^\pi} J \approx$
                 $\frac{1}{N}\omega \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\pi(s_i)}$
19                   $\cdot \nabla_{\theta^\pi} \pi(s_i|\theta^\pi)|_{s=s_i}$
20         end
21         if t = 0 mod $t_{target}$ then
22             Update the target networks:
                 $\theta^{Q'} \leftarrow \theta^Q$,$\theta^{\pi'} \leftarrow \theta^\pi$
23         end
24     end
25 end
```

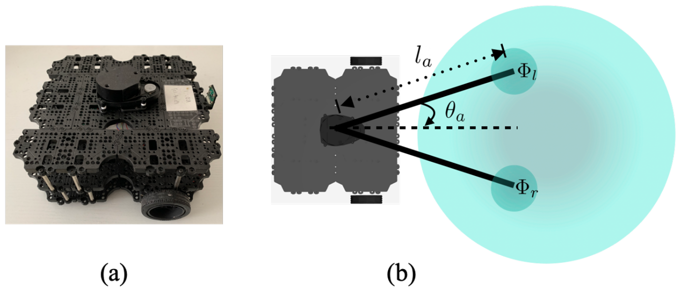


Fig. 3. The photo of Turtlebot3 Waffle Pi mobile robot, (a), and an illustration of the two virtual antennae are attached on Turtlebot3 Waffle Pi in the Gazebo simulated environment, (b). l_a is the length of antennae, α is the tilt angle, and Φ_l, Φ_r are the pheromone concentrations measured by the left and right antenna respectively.

technique and noisy network is summarised in Algorithm 1. When the algorithm is executed, it initialises the actor and critic neural networks, the target network and the replay buffer. The algorithm iterates for M training episodes and in each episode, a total of T steps of transition samples are collected from the K number of actors. Sample priorities are assigned using the HLER sampling strategy. In every t_{train} time step, the N number of transition samples are selected based on P_i and TD-target and y_i is assigned. In line 16, ω determines the degree of the weights of the neural network are updated, where β is a constant value and B is the total batch size. In every t_{target} in every episode, the target actor and critic networks are updated.

IV. Experiments

In this section, we introduce the structure and concepts of the experiments. The two goals of the experiments are to compare pheromone-based collision avoidance approaches to the traditional multi-agent collision avoidance method and to evaluate DRL-based controller compared to manually-designed controller integrated with pheromone-based communication. The following sections describe i) the specific experimental scenario, ii) traditional multi-agent collision avoidance method, iii) the conventional manually-tuned controller, iv) set-up of the DRL-based controller, and v) the metrics used to evaluate the performance of the controllers. Through the experiments, the suitability of the PhERS framework is also validated. Compared to our previous work [34] that first validated the PhERS framework, the experiments conducted in this study include more complex environments, use of a more advanced DRL-based controllers and an additional metric for more elaborate analysis so that the benefits of the proposed framework and the improved version of DRL-based controller could be further investigated.

A. Experimental Scenario

We have designed an experimental scenario with three stages that realise simple collision avoidance scenarios for autonomous vehicles in the real-world with different levels of complexity. All the experimental stages were constructed using the Gazebo simulator which offers a realistic physics engine. Simulated differential driven mobile robots, *Turtlebot3*, which are general-purpose wheeled robots developed for education, research and product prototyping, were deployed as the autonomous vehicles, shown in Fig. 3 (a). Fig. 4 illustrates the experimental stages and each stage is explained as follows.

1) Stage 1– Single Robot in a Static Scenario: In this scenario, we deployed one robot and four static cylindrical obstacles. This stage is modeled from the situation when an autonomous vehicle is surrounded by static obstacles in traffic. In the initialisation phase, the robot is placed at the centre (0,0) in the 2D Cartesian coordinates of the arena and four obstacles are placed at positions $[(2,0),(-2,0),(0,2),(0,-2)]$ m respectively. The goal for the navigation system is to move the robot to a random position 4 m from the origin (0,0). Fig. 4 (a) illustrates this stage.

2) Stage 2– Multi-robot in a Dynamic Scenario: In this scenario, we deployed two robots. The robots avoid colliding with each other while navigating to the initial position of the other robot. This stage mimics the situation when two autonomous vehicles are driving towards each other and a collision must be avoided. In the initialisation phase, the robots are placed at a distance of 5 m apart from each other. Fig. 4 (b) depicts this stage.

3) Stage 3– Multi-robot in a Complex Dynamic Scenario: In this scenario, we deployed four robots and one static obstacle. This stage recreates the situation where there are four autonomous vehicles at a roundabout and there are no fixed traffic rules. As in Stage 2, the goal for the robots in this stage is to navigate to the positions of the robot they are initially facing, and avoid colliding with other robots and the static obstacle. In the initialisation phase, the robots were placed at the positions $[(2.5,0),(-2.5,0),(0,2.5),(0,-2.5)]$ m respectively in the 2D Cartesian coordinates of the arena and the static obstacle was placed at the centre (0,0) of the arena. Fig. 4 (c) shows this stage.

To enable pheromone-based collision avoidance, we designed two different types of artificial pheromone: i) a non-volatile pheromone for static collision avoidance and ii) a highly volatile pheromone for dynamic collision avoidance. The concept of the two types of pheromone is inspired by alarm pheromones that exist in nature and cause individuals, e.g. ants, to become aggressive and fight or fearful and run away [17], [52]. In each set of experiments, non-volatile pheromones are released around static obstacles in the initialisation phase. For the introduction of the two

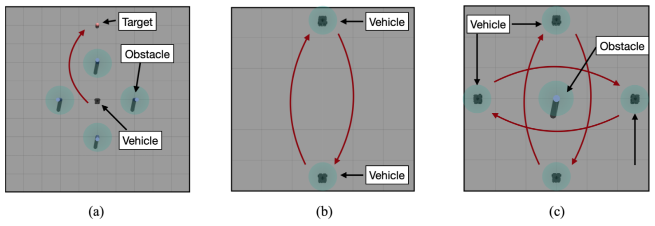


Fig. 4. Three experimental stages. (a), (b) and (c) illustrate Stage 1, 2 and 3 respectively. The cyan coloured shaded circles around the robots and obstacles illustrate the area where pheromones are injected and the red arrows show the arbitrary trajectory of the robots on a mission.

pheromones: the non-volatile pheromone is released with a circular shape of radius 0.5 m without any evaporation and diffusion, i.e. $e_\Phi = 0$ s and $\kappa = 0$, and the volatile pheromone is released in a circular shape with a radius of 0.3 m, and a very short evaporation half-life of $e_\Phi = 0.5$ s. The radius and the evaporation rate were selected empirically so that each type of pheromone worked effectively for collision avoidance. The choice of e_Φ for the volatile pheromone was inspired by [53], where highly volatile repellent pheromone was used to avoid collision with walls during navigation behaviours.

Experiments in each stage were repeated 100 times with both DRL-based controller and the baseline manually-tuned controller. For experiments with the DRL-based controller, the fully trained controller was utilised, following training in a separate training stage. In the training stage, the DRL-based controller was trained for all three test stages using the proposed DRL algorithm, as well as two baseline DRL algorithms, so that the capabilities of the proposed DRL algorithm could be determined.

B. Traditional Multi-agent Collision Avoidance

Non-Holonomic Optimal Reciprocal Collision Avoidnace (NH-ORCA) [10], [11] is a centralised multi-robot collision avoidance and navigation method. In this method, the position and velocity of the agents are shared to generate the velocities for the next sample time to ensure collision-free motion. Compared to the initial approach, ORCA [9], NH-ORCA applies non-holonomic feature of mobile robots to generate the optimal velocity to avoid collisions. As Turtlebot3, the robotic platform simulated in this work, is a non-holonomic wheeled-mobile robot, NH-ORCA offers effective collision avoidance performance for our experimental setup. Furthermore, since it guarantees successful collision avoidance when the position and velocities of the agents in a swarm is given, we chose this traditional algorithm as a baseline algorithm to compare our proposed pheromone-based collision avoidance strategy. Further details of the NH-ORCA algorithm are available in [11].

C. Manually-tuned Controller

The manually-tuned controller was proposed as the baseline to compare the performance of DRL-based controller and was designed to perform navigational tasks whilst avoiding collisions. A flowchart for the manually-tuned controller is provided in Fig. 5. When the robot begins operation, it measures pheromone concentration from the tip positions of its virtual antennae. Figure 3 illustrates how the virtual antennae were attached to the robot to sense the virtual pheromones. Both left and right antennae have lengths of 0.45 m, i.e. l_a = 0.45 m, and each antenna was tilted by θ_a, $-\theta_a$ respectively from the heading vector of the robot. With real vehicles, the same type of virtual antennae could be implemented in the cyber-physical space in each vehicle or in the cloud system, which manages all the virtual pheromone information. If the pheromone was implemented with physical materials, the physical antenna could be attached to the front side of the vehicles.

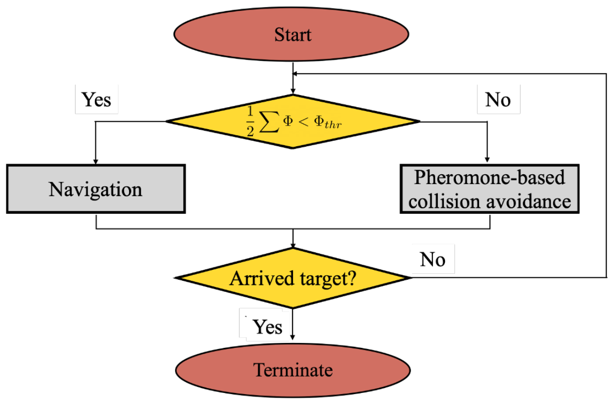

Start
Yes
No
$\frac{1}{2}\sum\Phi < \Phi_{thr}$
Navigation
Pheromone-based collision avoidance
Arrived target?
No
Yes
Terminate

Fig. 5. A flowchart of the manually-tuned controller.

The pheromone concentrations read from the left and right antennae are defined as Φ_l, Φ_r respectively. If the sum of the pheromone concentration from the two antennae, $\sum\Phi$, is higher than the pheromone concentration threshold, Φ_{thr}, the robot sets the speed of its motors to ensure it avoids a collision. Otherwise, the robot sets the speed of motors to perform navigation to the target regardless of pheromone concentration. This process is repeated until the robot arrives at the goal.

The design of the pheromone-based collision avoidance technique was inspired from [53], [54]. In those works, pheromone-based foraging and aggregation capability was applied to a mobile robot swarm using two artificial pheromone inputs. By modifying the controller used in those works, the pheromone-based collision avoidance model used in this work was defined to be:

$$\begin{aligned} &\beta = \beta_{const} - \frac{\Phi_l + \Phi_r}{2} \\ &w_l = \beta + \alpha(\Phi_l - \Phi_r), \quad w_r = \beta + \alpha(\Phi_r - \Phi_l)\ , \end{aligned} \tag{7}$$

where β is a bias velocity, β_{const} is a bias velocity constant, Φ_l, Φ_r are the pheromone intensity measurements from the left and right antenna respectively, α is a sensitivity gain applied to the difference between pheromone concentrations measured by the left and right antennae, and w_l, w_r are the left and right wheel velocities for the mobile robot respectively. To find the optimal values of β_{const} and α, that provide the best performance, we conducted preliminary experiments with a wide range of different parameters before the main experiments. During the preliminary experiments, different values of the parameters were assigned within the range $(0.6 \leq \beta_{const} \leq 1.4, 0.6 \leq \alpha \leq 1.4)$ with a step size of 0.1. Then, the controller was tested over 20 repeats for each experimental stage and the parameter set with the best performance was chosen for the main evaluation experiment. The results from the preliminary experiments are presented in Appendix A. The parameter sets for each stage that resulted in the highest performance metrics, described further in Section D, were chosen as the best parameter set for that stage. In the preliminary experiments, the range of parameters was chosen empirically as a range that the controller yields successful collision avoidance behaviours. With parameters outside of this range, the vehicle showed unrealistic backward movement or was unsuccessful in avoiding collisions.

The navigation behaviour of the robot is defined in Equation (8). When the value of the pheromone measured by the two antennae was below a pheromone threshold value, Φ_{thr}, it performed navigational tasks.

$$\begin{aligned}
&\phi = \text{atan2}\left(\frac{y_{targ}-y}{x_{targ}-x}\right), \quad \psi = \phi - \theta, \\
&v = v_{const}, \\
&\omega = \min\left(1, \max\left(-1, \omega_{coef}\,\text{atan2}\left(\frac{\sin(\psi)}{\cos(\psi)}\right)\right)\right),
\end{aligned} \tag{8}$$

where (x_{targ}, y_{targ}) represents the 2D Cartesian coordinates of the target, θ is an angular distance between the local coordinate frame of the robot and the global coordinate frame, v_{const} is a constant linear velocity, and ω_{coef} is an angular velocity coefficient. Using this equation, the robot determines (v, ω), linear and angular velocities. In the experiments, v_{const} = 0.5 m/s, and ω_{coef}= 10, which were empirically chosen to generate minimum fluctuations. Table I lists the parameters and corresponding values used for the experiments.

D. DRL-based Controller Setup

Here we provide the implementation details of the DRL-based controller that was used in the experiments. The four important design specifications are introduced as: i) observation space, ii) action space, iii) reward design and iv) actor and critic neural network structure. Using the describe design specifications, we trained the DRL-based controller using three different algorithms: 1) PER sampling strategy, 2) HLER sampling strategy and 3) HLER sampling strategy with a noisy network (HLER+noisy).

1) Observation Space: During the experiments observations were made of the pheromone intensity and gradient at the tips of the virtual antennae (total of 4 measurements). The current values of the pheromone concentrations, Φ_l, Φ_r, and the difference in pheromone concentrations between the current and previous time step, $\Delta\Phi_l, \Delta\Phi_r$, were measured at the tip of the virtual antennae. Taking the difference of pheromone concentration as observation inputs was inspired by how insects respond to pheromone gradients in nature, through chemotactic behaviour [55].

The absolute value of pheromone concentration was normalised in the range between $[0-1]$ as normalisation tends to lead to more effective training in DRL. For navigation, four different inputs were taken as observation inputs. The first two were the distance between the robot and the navigation goal in polar coordinate system, (d, θ) and the other two were the linear and angular velocities of the robot at the previous time step. By giving the previous linear and angular velocities as the inputs, the robot can learn the relationship between the velocities in two continuous time steps. In total, the DRL controller takes 8 observational inputs, each with a continuous range, i.e. $o_t \in (8 \times 1)$.

2) Action Space: We designed the DRL-based controller to output two values: i) translational velocity, v, and ii) rotational velocity, w, i.e. $a = [v, w]$. The range of velocities were limited to $v \in (0, 1)$ m/s and $w \in (-\frac{\pi}{2}, \frac{\pi}{2})$ rad/s to reflect the motion constraints of the robot.

3) Reward Design: One of the most important aspects to consider when designing the DRL-based controller was the reward functions. Appropriate values for the reward function will be more likely to produce desirable behaviours from the DRL-based controller. In the experiments, the reward function was designed in such a way that the DRL-based controller learned collision avoidance and navigation behaviour effectively.

There were five types of reward functions that were applied in the experiments. The reward functions were designed to enable the learning of i) pheromone-based collision avoidance, ii) navigation and iii) smoother trajectory. Equation (9) describes the reward functions used in the experiments.

$$\begin{aligned}
&r = r_c + r_g + r_p + r_v + r_w \\
&r_g = \begin{cases} R_g, & \text{if arrived goal} \\ 0, & \text{otherwise} \end{cases}, \quad r_c = \begin{cases} R_c, & \text{if collision} \\ 0, & \text{otherwise} \end{cases} \\
&r_d = \begin{cases} ad, & d > 0 \\ -ad, & \text{otherwise} \end{cases}, \quad r_v = \begin{cases} R_v, & \text{if } v < v_{min} \\ 0, & \text{otherwise} \end{cases}, \\
&r_w = \begin{cases} R_w, & \text{if} |w| > w_{max} \\ 0, & \text{otherwise} \end{cases}.
\end{aligned} \tag{9}$$

The total reward, r, was the sum of a series of rewards / penalties associated with reaching the goal, r_g, being involved in a collision, r_c, progress towards the goal position, r_d (specified to be equal to a step length, d, multiplied by a tunable factor, a, which was set empirically to 4.0), a velocity penalty, r_v and an angular velocity penalty, r_w. The parameter values used in the experiments are provided in Table I. However, these values can be changed for different experimental setups and tasks.

TABLE I
PARAMETERS AND VALUES

Parameters	Values	Parameters	Values
v_{const}	0.5	a	4.0
ω_{coef}	10	v_{min}	0.2 m/s
β_{const}	0.6 - 1.4	ω_{max}	0.8 rad/s
α	0.6 - 1.4	R_v	-1
R_g	100	R_ω	-1
R_c	-100	p_{base}	0.5
k_{HL}	2.0	r_{HL}	20
t_{out}	60 (s)		

4) Neural Network Architecture: The architecture of the neural network used for the experiments is shown in Fig. 6. The actor network, consisted of input and output layers, which are observations and actions, and 3 fully connected layers with 512, 512, 512 neurons followed by a rectified linear activation function (ReLU) nonlinearities between the input and output layers. Linear and angular velocities were

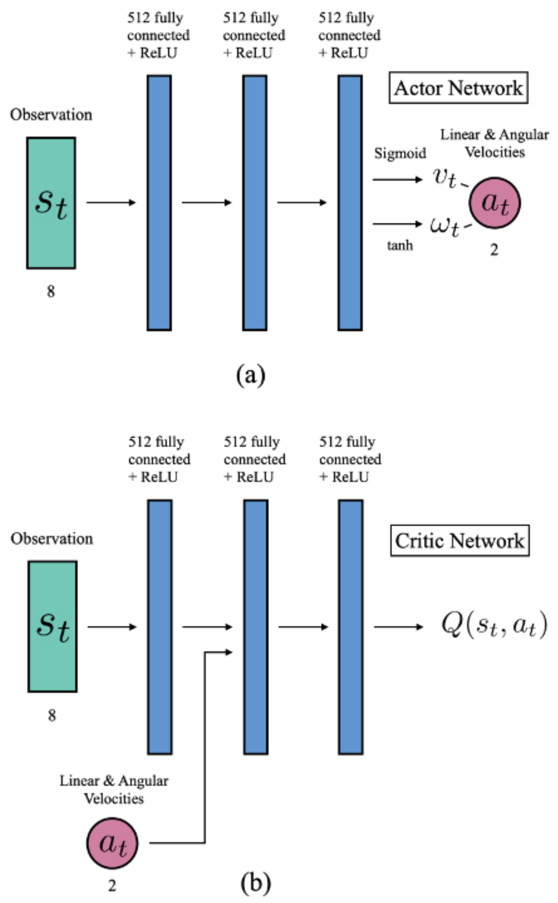

512 fully connected + ReLU
512 fully connected + ReLU
512 fully connected + ReLU
Actor Network
Observation
s_t
8
Sigmoid
Linear & Angular Velocities
v_t
ω_t
a_t
tanh
2
(a)
512 fully connected + ReLU
512 fully connected + ReLU
512 fully connected + ReLU
Critic Network
Observation
s_t
8
$Q(s_t, a_t)$
Linear & Angular Velocities
a_t
2
(b)

Fig. 6. Neural network architecture for the DRL-based controller. (a) and (b) represent the network architectures for the actor and critic networks respectively.

connected to sigmoid and tanh activation functions to limit the range to ($0 \leq v_t \leq 1$ and $-1 \leq \omega_t \leq 1$).

The critic network, consisted of input and output layers, which were observations and state-action values, and 3 fully connected layers with the same number of parameters as the actor network. Unlike the actor network, after the first layer, actions were concatenated and fed into the second layer.

E. Metrics

Three metrics were applied to evaluate the performance of the controllers for the experiments: i) success rate, ii) completion time, and iii) trajectory efficiency.

- Success rate, ρ_s, is the percentage of the successful runs, i.e. those runs where no collisions occurred and the completion time was below the threshold, t_{out}.
- Completion time, t_{comp}, represents the time taken for all the robots to reach the targets without collision and within t_{out}.
- Trajectory efficiency, η_t, denotes the average of the ratio of the Euclidean distance between the start and end points of the robots to the actual distance the robots traveled during each successful run.

V. Results & Discussion

In this section, we first show the obtained results using the traditional centralised NH-ORCA controller. We then show the performance of manually-tuned controllers. The experimental results with the DRL-based controller are then introduced and the results using the three different sampling strategies are analysed and compared with each other. Figure 7 illustrates the experimental results for all stages. We then compare the traditional centralised NH-ORCA controller and the manually-tuned and DRL-based controller using the virtual pheromone. After that, further discussion of the results follow. The clip of experiments are provided in the supplementary video.

A. Traditional Centralised Controller

In Fig. 7, the experimental results using the NH-ORCA controller in the three stages are shown, together with the three metrics. The numerical values of the metrics are listed in Table II. According to the results, the success rate of the experiments with NH-ORCA in all three stages was 100 %, which shows the high stability of the centralised controller when the position and velocity information are provided to the centralised controller. In contrast to the stable high success rate across all the experimental stages, the completion time was greatly increased and the trajectory efficiency considerably declined in Stage 3. This phenomenon demonstrates that the traditional centralised controller in a complex environment is not effective for finding an optimal strategy while it ensures the stability. In other words, the centralised controller exhibits low flexibility for the increasing complexity.

B. Manually-tuned controller

From the preliminary experiments, the best parameter sets were chosen for each stage. Further details on the choice of parameter sets are illustrated in Appendix A. Figure 7 shows the experimental results for each of the three stages, with numerical values for these results presented in Table II. The result showed that the manually-tuned controller is not effective for the complex environment. In Stage 1 and 2, the success rate of the manually-tuned controller was 99% and 100%. In contrast, in Stage 3, the success rate was 84%. The drop in success rate implies that the performance of manually-tuned controller decreases as the complexity of the task and environment increases.

C. DRL-based controller

Here, the experimental results of DRL-based controller with three different DRL algorithms (PER, HLER, HLER+noisy) are presented. First, we compare the training performance of three DRL algorithms in three stages to show the advantage of our proposed DRL algorithm over the baselines. Second, we compare the experimental results of the three algorithms in each stage.

Figure 8 illustrates the average reward per episode over the training phase in each stage and Table II and Fig. 7 show the results that were obtained with the proposed DRL-based algorithm and other two baselines.

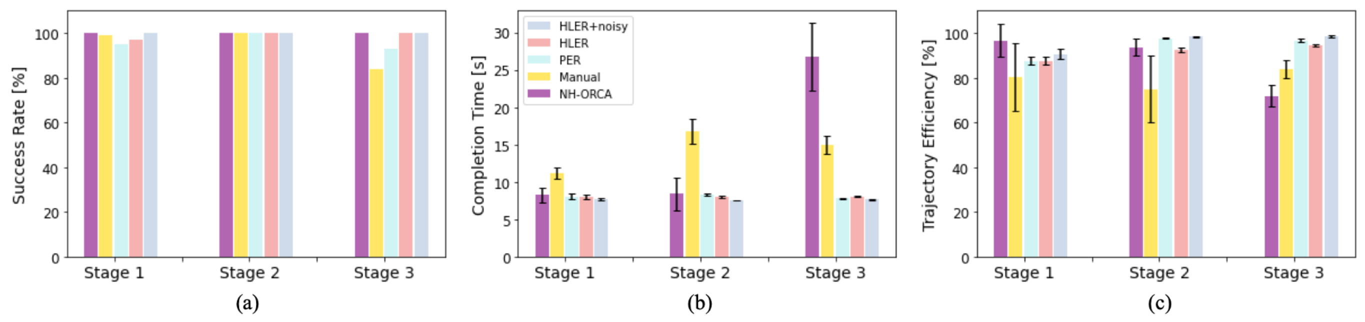


Fig. 7. The bar plots show the experimental results with the traditional controller with NH-ORCA, the manually-tuned controller (HT), and DRL-based controller with three different training strategies, PER, HLER and HLER with noisy network in three experimental stages. (a), (b), and (c) illustrates the success rate, completion time and trajectory efficiency in Stage 1, 2 and 3 respectively. The error bars represent standard deviation of each metric.

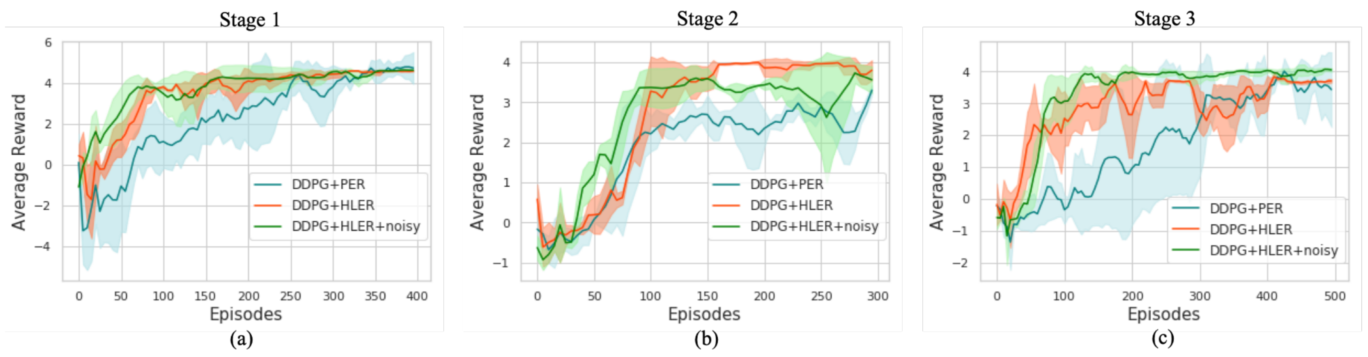


Fig. 8. Average reward per episode during training with PER, HLER and HLER with noisy network. (a), (b) and (c) illustrate the change in average reward over training in Stage 1, 2 and 3 respectively. The bold line and shading represents the mean and standard deviation of the reward over 3 different runs with different random seeds respectively.

TABLE II
EXPERIMENTAL RESULTS OF THE CONTROLLERS IN THE THREE STAGES.

Stage	Controller	ρ_s	t_{comp}	η_t
Stage 1	NH-ORCA	100%	8.25 ± 0.95	0.9666 ± 0.0738
	Manual	99%	11.16 ± 0.75	0.8036 ± 0.1527
	PER	95%	8.04 ± 0.36	0.8765 ± 0.0196
	HLER	97%	8.06 ± 0.30	0.8762 ± 0.0186
	HLER+noisy	100%	7.68 ± 0.17	0.9068 ± 0.0220
Stage 2	NH-ORCA	100%	8.39 ± 2.20	0.9367 ± 0.0381
	Manual	100%	16.74 ± 1.68	0.7486 ± 0.1485
	PER	100%	8.34 ± 0.15	0.9762 ± 0.0030
	HLER	100%	7.96 ± 0.11	0.9234 ± 0.0097
	HLER+noisy	100%	7.54 ± 0.06	0.9837 ± 0.0021
Stage 3	NH-ORCA	100%	26.81 ± 4.54	0.7187 ± 0.0477
	Manual	84%	14.95 ± 1.16	0.8394 ± 0.0387
	PER	93%	7.81 ± 0.08	0.9672 ± 0.0065
	HLER	100%	8.01 ± 0.07	0.9440 ± 0.0048
	HLER+noisy	100%	7.63 ± 0.06	0.9850 ± 0.0032

1) Training Efficiency: Training with each sampling strategy in each stage was repeated 3 times using different random seeds to vary the initialisation conditions. In all experiments, it was observed that the HLER sampling strategy accelerated the training speed compared to PER sampling strategy. The noisy network was also found to improve the training speed. To compare the training time quantitatively, the HLER with noisy network required 25%, 27% and 30% of training time of PER sampling strategy until reaching the converged value in Stage 1, Stage 2 and Stage 3 respectively, which average is 27% over the stages. These results are expected as we designed HLER for faster training by prioritising the highlighted transition samples and implemented the noisy network for better exploration, leading to faster searching of optimal policies in all experimental stages.

2) Performance Evaluation: In Stage 1, the DRL-based controller with PER and HLER sampling strategy exhibited similar performance with regards to the three metrics. However, when used with a noisy network, the DRL-based controller with HLER resulted in higher performance in all three metrics.

In Stage 2, similar to Stage 1, there was no significant difference in performance of the controllers using PER and HLER. Whilst trajectory efficiency was higher with PER than HLER, the completion time was faster with HLER than PER. This is because using the DRL-based controller trained with HLER generated slightly longer trajectory than PER to further ensure the collision does not happen. However, the controllers that utilised the noisy networks outperformed the other controllers for all metrics.

In Stage 3, the controller with the PER sampling strategy outperformed the controller with HLER in terms of completion time and trajectory efficiency, while the success rate was lower. This is because the controller with PER found the policy that minimised the completion time, but unfortunately also lowered the success rate. However, when HLER was used, the success rate was maximised to 100% because the HLER sampling strategy incentivises the controller to learn to avoid collisions at the highest priority. Furthermore, the controllers that used the noisy network approach yielded greater performance than those with normal layers. This result aligns with the results from Stage 1 and 2, which support the noisy network improved exploration of the agent to find the best policy, which ensures both safety and performance.

D. Comparison between the traditional centralised controller and decentralised pheromone-based controllers

Here we compare the traditional centralised NH-ORCA controller with the manually-tuned and DRL-based controllers with pheromone-based collision avoidance strategy, manually-tuned and DRL-based controllers. These comparisons demonstrate the difference between the centralised approach and the pheromone-based decentralised approach.

Comparing the manually-tuned controller with the pheromone-based communication to the NH-ORCA controller, it demonstrated lower success rate in Stage 3. In terms of the completion time and trajectory efficiency, the NH-ORCA controller showed higher performance than the manually-tuned controller with pheromoned-based communication in Stages 1 and 2. However, the completion time and trajectory efficiency of the manually-tuned controller outperformed the NH-ORCA controller in Stage 3. This result suggests that the manually-tuned controller with the decentralised pheromone-based communication approach leads to more effective behaviours than the centralised NH-ORCA controller in a complex environment, despite lower success rate due to the limitation of the manually-tuned approach.

Comparing the DRL-based controller with the pheromone-based communication to the NH-ORCA controller, it showed equally high success rates over all the three experimental stages. Furthermore, the DRL-based controller greatly outperformed the NH-ORCA controller in terms of the completion time and trajectory efficiency in all three stages, especially in Stage 3. This suggest that the DRL-based controller with pheromone-based communication has a similar degree of stability with the centralised controller, but greater flexibility and effectiveness in the complex environment.

Overall, the centralised controller can ensure the stability of collision avoidance and navigation behaviour in diverse environments; however, the performance greatly decreases when the complexity of environments and tasks increase. On the other hand, the pheromone-based collision avoidance ensures high flexibility with varying complexity of the environments and tasks, although the stability and the effectiveness of collision avoidance and navigation behaviour depends on the type of the controller. The detailed comparison between the manually-tuned and DRL-based controllers that used the pheromone-based approach is presented in the next section.

E. Comparison between the manually-tuned and DRL-based controllers

Here we compare the manually-tuned controller and DRL-based controller used with the pheromone-based approach. For DRL-based controller, the controller trained with our proposed DRL algorithm (HLER+noisy) was used as it exhibited the greatest performance among the DRL algorithms used for the experiments.

In terms of the success rate, the manually-tuned controller scored 99% and 100% in Stage 1 and 2, but only scored 84% in Stage 3. This result shows that the capabilities of the manually-tuned controller are reduced when the complexity of the environment increases. On the other hand, the DRL-based controller showed consistent performance regardless of the complexity of the environments, scoring 100% for all the three stages.

With regards to the completion time, the DRL-based controller considerably outperformed the manually-tuned controller in all three stages. For autonomous vehicle application, the faster completion time with the DRL-based controller is important as it means that the DRL-based controller can reduce travel time, greatly increasing the satisfaction of users.

Similar to the comparison of the completion time, the DRL-based controller outperformed the manually-tuned controller in terms of the trajectory efficiency in all three stages, which resulted from the ability of the DRL-based controller to optimise its performance. Trajectory efficiency is particularly important in real-world applications as with a larger number of vehicles in a real-world situation, the lower trajectory efficiency is likely to cause greater congestion and inefficient movement of vehicles, which may also increase the chance of collisions.

Furthermore, the DRL-based controller showed it had greater robustness than the manually-tuned controller, with lower standard deviations for both completion time and trajectory efficiency for all three stages. This result shows that performance of the manually-tuned controller with the best parameter set can notably change if the initial condition change even slightly, suggesting that the manually-tuned controller can be undesirable in real-world environments, while the DRL-based controller should be more effective in real-world environments.

Finally, the use of the DRL-based controller is beneficial in practice as it automates the optimisation process whilst a relatively laborious tuning process is required for the manually-tuned controller. This is of considerable importance in swarm systems as it can be extremely difficult to tune the controller manually with the real-world complexity [30].

Overall, the DRL-based controller used for the pheromone-based collision avoidance outperformed the manually-tuned controller in all aspects including stability, flexibility, robustness and the ease of tuning process. Summing up with the comparison with the centralised controller, the DRL-based controller with the pheromone-based approach ensure high flexibility from the advantage of the pheromone-based approach and high stability and performance due to optimisation capability of DRL.

F. Discussion

1) Trajectory Analysis: To further investigate the results of the pheromone-based collision avoidance strategy, the

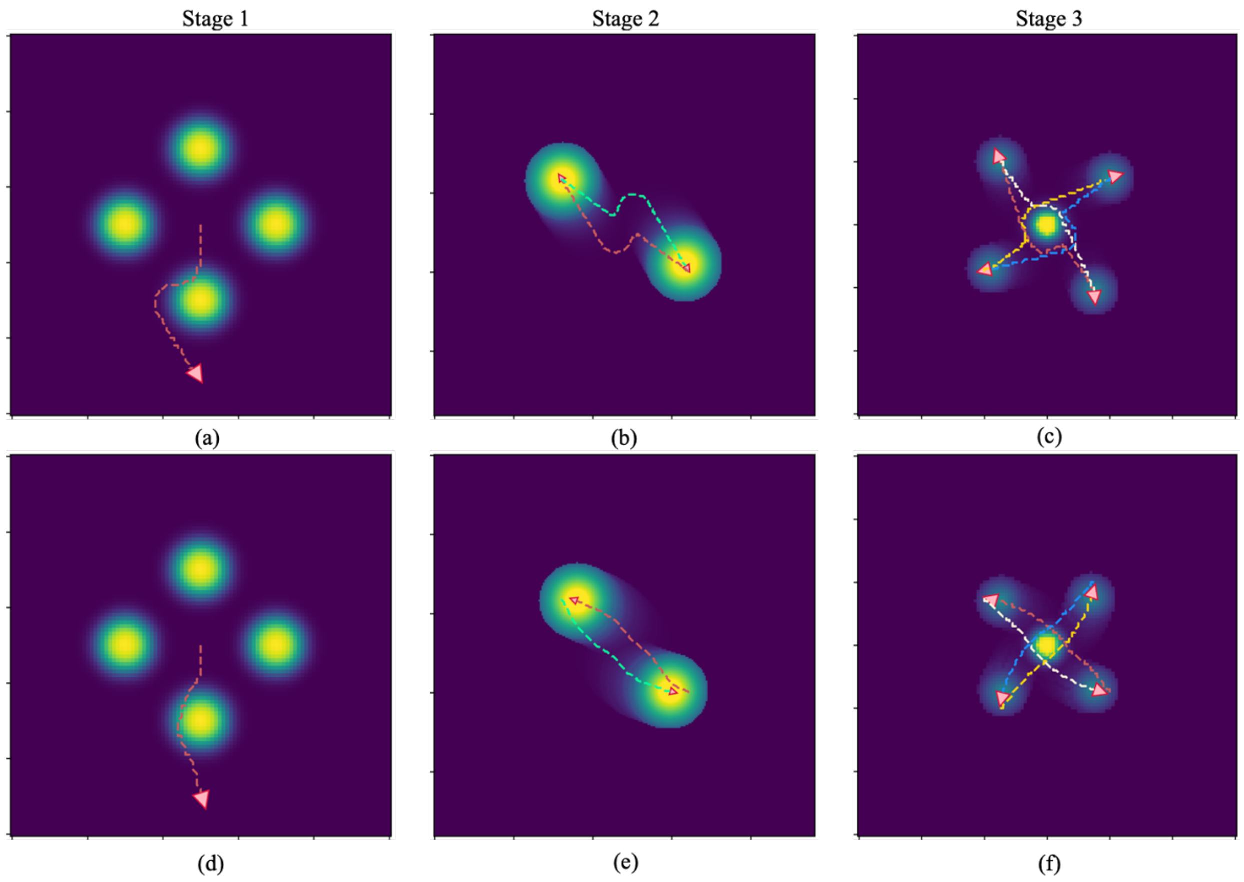


Fig. 9. Trajectories of the robots in the three experimental stages. (a), (b), (c) are the trajectories of the robots with the manually-tuned controller in Stage 1, 2, 3 respectively, and (d), (e), (f) are the trajectories of the robots with the DRL-based controller (DDPG + HLER + noisy network) in Stage 1, 2, 3 respectively. The triangles represent the final position and orientation of the robots. The closer to yellow the colour on the map is, the higher the concentration of pheromone is at that location.

trajectory data of the robots with the two controllers (manually-tuned controller & DRL-based controller with DDPG + HLER + noisy network) for each stage was analysed. Figure 9 illustrates the trajectory of the robots using two different pheromone-based controllers in each setting in all three stages. In general, the manually-tuned controller exhibited very sensitive repulsive behaviour. When the robots approached an obstacle, the controller rapidly altered its trajectory causing it to move away from the obstacle rapidly. This pattern in trajectory is shown in every stage. In contrast, the trajectory of the robots that used the DRL-based controller had a much smoother response. Without any rapid changes in direction, the robots using the DRL-based control were able to achieve high levels of navigation performance, whilst actively avoiding collisions. This observation highlights that the DRL-based controller is able to balance the two different requirements for it in a way that maximises its performance relative to the goal defined in the design phase. In a real-world application, a smoother response is highly desirable as it will help minimise collisions and improve driver/passenger satisfaction.

Moreover, in our experiments we found that the trajectory of the robots when using the DRL-based controller in Stage 3, shown in Fig. 9 (f) shows similar trajectories as robots used in other studies that performed collision avoidance and navigation using laser proximity sensors [56]. The similarity suggests that the use of virtual pheromones results in similar behaviour to that obtained when using more traditional approaches, such as proximity sensors.

2) Pheromone Characteristics: Although the experimental results demonstrated that the DRL-based controller outperformed the traditional manually-tuned controller, it is still unclear if the DRL-based controller is optimal. In [21], the importance of choosing appropriate parameters for the artificial pheromones, and the effect that these parameters have on the performance of the swarm robotic system were demonstrated. In this work, we did not conduct exhaustive tests on identifying optimal parameters for the virtual pheromone, such as the evaporation half-life, diffusion rate, shape and the gradient of the injected pheromone. This is a limitation of the study reported in this work and further analysis of the technique, particularly in regard to the optimal parameter selection in more complex environments, is the subject of the future works.

VI. Conclusion

This paper has demonstrated that pheromone-based communication systems can be used as an alternative approach for ensuring that autonomous vehicles avoid collisions while operating in traffic. In this work, we validated pheromone-based collision avoidance for autonomous vehicles in a simulated environment with a newly proposed artificial pheromone framework, which we refer to as PhERS. Furthermore, we have demonstrated that deep reinforcement learning can improve the performance of the pheromone-based system, both in terms of collision avoidance and the speed by which the robots are able to move to target locations within an environment. Through comparisons we were able to show that the pheromone-based communication system has able to cope much better with complex environments than the traditional centralised controller. Also, the performance of the proposed DRL-based controller exceeded that of the manually-tuned controller, with respect to stability, effectiveness, robustness and ease of tuning process. Furthermore, our proposed DRL algorithm was able to outperform baseline DRL algorithm by accelerating training speed and improving performance. Notably, the training time of the HLER sampling strategy accompanied with noisy network required average 27% of the PER over three experimental stages. The results suggest that our proposed DRL-based controller has a role to play in the development of pheromone-based swarm systems in the real-world, as their features, which include adaptability in scenarios with different complexities and a reduction in the time taken to tune the controller, make them an attractive method compared with alternative approaches, particularly when trying to generate the required behaviour in a swarm of autonomous vehicles. Our future work will include the development of pheromone-based communication systems for large numbers ($>$100) of real autonomous vehicles and performing tasks with greater complexity (e.g. collective navigation) in dynamic and complex environments. We believe this research showed the potential of pheromone-based communication as an alternative communication approach for autonomous vehicles and DRL-based approach to design effective controllers with pheromone-based swarm systems.

References

[1] P. Mallozzi, P. Pelliccione, A. Knauss, C. Berger, and N. Mohammadiha, *Autonomous Vehicles: State of the Art, Future Trends, and Challenges*. Cham: Springer International Publishing, 2019, pp. 347–367.

[2] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1243–1274, 2019.

[3] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A survey of driving safety with sensing, vehicular communications, and artificial intelligence-based collision avoidance," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–22, 2021.

[4] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2017.

[5] L. Tang, S. Dian, G. Gu, K. Zhou, S. Wang, and X. Feng, "A novel potential field method for obstacle avoidance and path planning of mobile robot," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 9, 2010, pp. 633–637.

[6] A. Nakamura, Y. C. Liu, and B. G. Kim, "Short-Term Multi-Vehicle Trajectory Planning for Collision Avoidance," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9253–9264, 2020.

[7] S. Shin, D. Ahn, Y. Baek, and H. Lee, "Adaptive aeb control strategy for collision avoidance including rear vehicles*," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2872–2878.

[8] J. Hu, P. Bhowmick, F. Arvin, A. Lanzon, and B. Lennox, "Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 977–984, 2020.

[9] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–19.

[10] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems: The 10th International Symposium*. Springer Berlin Heidelberg, 2013, pp. 203–216.

[11] J. Alonso-Mora, P. Beardsley, and R. Siegwart, "Cooperative collision avoidance for nonholonomic robots," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 404–420, 2018.

[12] C. Hu, F. Arvin, C. Xiong, and S. Yue, "Bio-inspired embedded vision system for autonomous micro-robots: The lgmd case," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 3, pp. 241–254, 2017.

[13] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.

[14] Y. Yuan, R. Tasik, S. S. Adhatarao, Y. Yuan, Z. Liu, and X. Fu, "Race: Reinforced cooperative autonomous vehicle collision avoidance," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9279–9291, 2020.

[15] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 285–292, 2017.

[16] M. Schranz, G. A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. Şekercioğlu, and M. Sende, "Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends," *Swarm and Evolutionary Computation*, vol. 60, p. 100762, 2021.

[17] B. Hölldobler, E. O. Wilson *et al.*, *The superorganism: the beauty, elegance, and strangeness of insect societies*. WW Norton & Company, 2009.

[18] T. Wyatt, *Pheromones and Animal Behavior: Chemical Signals and Signatures*. Cambridge University Press, 2014.

[19] T. J. Czaczkes and J. Heinze, "Ants adjust their pheromone deposition to a changing environment and their probability of making errors," *Proceedings of the Royal Society B: Biological Sciences*, vol. 282, no. 1810, 2015.

[20] A. Denny, J. Wright, and B. Grief, "Foraging efficiency in the wood ant, formica rufa: Is time of the essence in trail following?" *Animal Behaviour*, vol. 62, pp. 139–146, 2001.

[21] A. L. Alfeo, E. C. Ferrer, Y. L. Carrillo, A. Grignard, L. A. Pastor, D. T. Sleeper, M. G. Cimino, B. Lepri, G. Vaglini, K. Larson, M. Dorigo, and A. S. Pentland, "Urban swarms: A new approach for autonomous waste management," *IEEE International Conference on Robotics and Automation*, pp. 4233–4240, 2019.

[22] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[23] K. L. Soon, J. M.-Y. Lim, and R. Parthiban, "Coordinated traffic light control in cooperative green vehicle routing for pheromone-based multi-agent systems," *Applied Soft Computing*, vol. 81, p. 105486, 2019.

[24] A. Campo, Á. Gutiérrez, S. Nouyan, C. Pinciroli, V. Longchamp, S. Garnier, and M. Dorigo, "Artificial pheromone for path selection by a foraging swarm of robots," *Biological Cybernetics*, vol. 103, no. 5, pp. 339–352, 2010.

[25] G. Valentini, A. Antoun, M. Trabattoni, B. Wiandt, Y. Tamura, E. Hocquard, V. Trianni, and M. Dorigo, "Kilogrid: a novel experimental environment for the kilobot robot," *Swarm Intelligence*, vol. 12, no. 3, pp. 245–266, Sep 2018.

[26] M. Salman, D. Garzón Ramos, K. Hasselmann, and M. Birattari, "Phormica: Photochromic Pheromone Release and Detection System for Stigmergic Coordination in Robot Swarms," *Frontiers in Robotics and AI*, vol. 7, no. December, p. 195, 2020.

[27] S. Na, Y. Qiu, A. E. Turgut, J. Ulrich, T. Krajník, S. Yue, B. Lennox, and F. Arvin, "Bio-inspired artificial pheromone system for swarm robotics applications," *Adaptive Behavior*, vol. 4, no. 29, pp. 395–415, 2020.

[28] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," *Science robotics*, vol. 5, no. 49, December 2020.

[29] M. Birattari, A. Ligot, D. Bozhinoski, M. Brambilla, G. Francesca, L. Garattoni, D. Garzón Ramos, K. Hasselmann, M. Kegeleirs,

J. Kuckling, F. Pagnozzi, A. Roli, M. Salman, and T. Stützle, "Automatic off-line design of robot swarms: A manifesto," *Frontiers in Robotics and AI*, vol. 6, 2019.

[30] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[31] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, S. M. Oliveira, and A. L. Christensen, "Evolution of collective behaviors for a real swarm of aquatic surface robots," *PLoS ONE*, vol. 11, no. 3, pp. 1–25, 2016.

[32] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv:1511.05952[cs.LG]*, 2016.

[33] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," *arXiv:1706.10295[cs.LG]*, 2019.

[34] S. Na, H. Niu, B. Lennox, and F. Arvin, "Universal artificial pheromone framework with deep reinforcement learning for robotic systems," in *6th International Conference on Control and Robotics Engineering*. IEEE, 2021, pp. 28–32.

[35] K. Kuru, "Planning the Future of Smart Cities with Swarms of Fully Autonomous Unmanned Aerial Vehicles Using a Novel Framework," *IEEE Access*, vol. 9, pp. 6571–6595, 2021.

[36] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[37] Y. Huang, L. Wang, Y. Hou, W. Zhang, and Y. Zhang, "A prototype iot based wireless sensor network for traffic information monitoring," *International journal of pavement research and technology*, vol. 11, no. 2, pp. 146–152, 2018.

[38] J. H. Ferziger, M. Perić, and R. L. Street, *Computational Methods for Fluid Dynamics*, 4th ed. Springer International Publishing, 2020.

[39] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[41] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1889–1897.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347[cs.LG]*, 2017.

[43] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1928–1937.

[44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971[cs.LG]*, 2015.

[45] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," *arXiv:1707.01495[cs.LG]*, 2018.

[46] P. Sun, W. Zhou, and H. Li, "Attentive experience replay," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 5900–5907, Apr. 2020.

[47] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv:1707.08817[cs.AI]*, 2018.

[48] F. Dolcos, K. S. LaBar, and R. Cabeza, "Remembering one year later: Role of the amygdala and the medial temporal lobe memory system in retrieving emotional memories," *Proceedings of the National Academy of Sciences*, vol. 102, no. 7, pp. 2626–2631, 2005.

[49] J. A. Bisby, A. J. Horner, L. D. Hørlyck, and N. Burgess, "Opposing effects of negative emotion on amygdalar and hippocampal memory for items and associations," *Social cognitive and affective neuroscience*, vol. 11, no. 6, pp. 981–990, Jun 2016.

[50] S. Erk, S. Martin, and H. Walter, "Emotional context during encoding of neutral items modulates brain activation not only during encoding but also during recognition," *NeuroImage*, vol. 26, no. 3, pp. 829–838, 2005.

[51] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," in *International Conference on Learning Representations*, 2018.

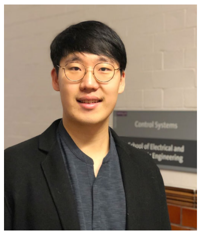
[52] F. Fossum, J.-M. Montanier, and P. C. Haddow, "Repellent pheromones for effective swarm robot search in unknown environments," in *2014 IEEE Symposium on Swarm Intelligence*, 2014, pp. 1–8.

[53] F. Arvin, T. Krajník, A. E. Turgut, and S. Yue, "COSΦ: artificial pheromone system for robotic swarms research," in *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2015, pp. 407–412.

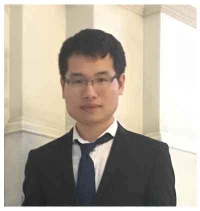
[54] S. Na, M. Raoufi, A. E. Turgut, T. Krajník, and F. Arvin, "Extended artificial pheromone system for swarm robotic applications," *Artificial Life Conference Proceedings*, no. 31, pp. 608–615, 2019.

[55] E. Bonabeau, G. Theraulaz, J. L. Deneubourg, N. R. Franks, O. Rafelsberger, J. L. Joly, and S. Blanco, "A model for the emergence of pillars, walls and royal chambers in termite nests," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 353, no. 1375, pp. 1561–1576, 1998.

[56] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.



Seongin Na (Student Member, IEEE) is a PhD candidate in robotics at the Swarm & Computational Intelligence Laboratory (SwaCIL) at the University of Manchester. He was awarded a President's Doctoral Scholar Award for his PhD study from the University of Manchester. Prior to his PhD study, he received a B.Eng in Mechatronic Engineering from the University of Manchester, United Kingdom in June 2019. His research interest includes swarm robotics behaviours and deep reinforcement learning. He is one of the main researchers of the EU H2020-FET RoboRoyale project.



Hanlin Niu (Member, IEEE) received B.Eng degree in Mechanics from Tianjin University in 2012 and Ph.D degree in Aeronautical Engineering from Cranfield University in 2018. From July 2017 to January 2020, he worked as a Research Associate in Robotics at Cardiff University. Since January 2020, he has been working as a Research Associate in Robotics at the University of Manchester. His research interests include tele-operation of robotic arm using virtual reality and digital-twin technology, path planning, path following, collision avoidance, deep reinforcement learning of unmanned surface vehicle, unmanned aerial vehicle, and mobile robot. He joined H2020-FET RoboRoyale project in 2021 as a key researcher working on machine learning and micro-robotics.

Barry Lennox (Senior Member, IEEE) is a Professor of applied control and nuclear engineering decommissioning and holds a Royal Academy of Engineering Chair in Emerging Technologies. He is a Director of the Robotics and Artificial Intelligence for Nuclear (RAIN) Robotics Hub and Research Director of the Dalton Cumbrian Facility. He is a Fellow of the Royal Academy of Engineering, Fellow of the IET and InstMC, and a Chartered Engineer. He is an expert in applied control and its use in robotics and process operations and has considerable experience in transferring leading edge technology into industry.

Farshad Arvin (Senior Member, IEEE) is an Associate Professor in Robotics at the Department of Electrical & Electronic Engineering in the University of Manchester. He received his BSc degree in Computer Hardware Engineering in 2004, MSc degree in Computer Systems Engineering in 2010, and a PhD in Computer Science in 2015. His research interests include swarm robotics and autonomous systems. Farshad joined the University of Manchester in July 2015 as a Post-Doctoral Researcher at the Department of Electrical and Electronic Engineering. He was a Research Assistant at the Computational Intelligence Laboratory (CIL) in the University of Lincoln, UK (2012 to 2015). He was awarded a Marie Curie-Skłodowska Fellowship to be involved in the FP7-EYE2E and LIVCODE EU projects during his PhD study. He visited several leading institutes including Artificial Life Laboratory in University of Graz, Austria in 2018, the Italian Institute of Technology (iit) in Genoa, Italy in 2017, Institute of Rehabilitation and Medical Robotics in Huazhong University of Science and Technology (HUST), Wuhan, China in 2014, and the Institute of Microelectronics at Tsinghua University in Beijing, in 2012 and 2013.

Farshad is the founding director of Swarm & Computation Intelligence Laboratory (SwaCIL) formed in 2018. He is coordinating several research projects including a large EU project H2020-FET RoboRoyale and he is a PI in H2020-FET Robocoenosis.

APPENDIX

A. Preliminary experiments for parameter choice of manually-tuned controller

As mentioned in Section IV, we conducted preliminary experiments to find a suitable parameter values for each experimental stage. Each parameter set of β_{const} and α in the range of $(0.6 \leq \beta_{const} \leq 1.4, 0.6 \leq \alpha \leq 1.4)$ with the step size of 0.1 was tested 20 times for each experimental stage. The results with three performance metrics in each stage are illustrated in Fig. 10, 11 and 12.

In Stage 1, it is found that with the low β_{const} ($\beta_{const} \leq 0.9$) led successful collision avoidance. However, with the higher β_{const} and lower α, the drop in the success rate is observed. Since higher β_{const} increases the speed of forward movement, the chance of collision rises when the robot faces the obstacle. With regards to the average completion time, the higher β_{const} resulted in a lower completion time. It is interesting to see that the higher α led the increased completion time when β_{const} is high ($\beta_{const} \geq 1.0$) while it led a lower completion time when β_{const} is low ($\beta_{const} < 1.0$). When β_{const} is high, the high α led adverse turning with the longer travel; therefore, it leads greater completion time. Conversely, when β_{const} is low, high α helps the robot escape from the collision avoidance situation with the adverse turning.

For trajectory efficiency, the lower β_{const} and lower α leads higher trajectory efficiency in general. It is because higher β_{const} leads a lower forward movement; therefore, if the robot is out of the right trajectory, the deviation increases. Also, the higher α leads more adverse turning behaviour during collision avoidance, thereby lower trajectory efficiency. However, there is also a trade-off between β_{const} and α, that leads the result deviates from the general trend.

When we chose the best parameter set, we first consider the success rate and then average completion time and trajectory efficiency. By considering all three performance metrics, the parameter set of $\beta_{const} = 0.9, \alpha = 0.7$ is chosen.

In Stage 2, with regards to success rate, it is seen that when β_{const} is greater than 0.9, there is almost no success regardless of α. However, when β_{const} is less than or equal to 0.9, the success rate is 100% or very close to 100% regardless of α. This dramatic difference in success rate happened since when β_{const} is greater than 0.9, the forward speed after detecting pheromone was too fast and the robots were not able to escape from collision.

In terms of completion time, it is seen that the lower α led a lower completion time regardless of β_{const}. It is because the higher α causes more adverse turning behaviour, it took more time to recover their optimal navigation strategy after collision avoidance behaviour. However, there are also outliers in the results that do not follow the general trend. In some cases, after the robots successfully complete first collision avoidance, the robots encounter again. Hence the completion time increased.

With regards to trajectory efficiency, there is similar trend with the completion time. In general, higher α led higher trajectory efficiency. However, there are outliers that violate the general trend. It is caused by re-encountering between robots as discussed with completion time.

When we chose the best parameter set for Stage 2, we first consider the success rate and then average completion time as the second priority and trajectory efficiency. By considering all three performance metrics, the parameter set of $\beta_{const} = 0.7, \alpha = 0.8$ is chosen.

In Stage 3, in terms of success rate, it is difficult to derive the general rule as in Stage 1 and 2. The parameter sets that generated highest success rates are: i) $\beta_{const} = 0.6, \alpha = 1.1$, ii) $\beta_{const} = 0.6, \alpha = 1.3$, and iii) $\beta_{const} = 0.7, \alpha = 1.3$. Since the complexity of the mission in Stage 3 increased from Stage 1 and 2, it is difficult to find the linear relationships between the parameter sets and the success rates. This results support the idea that the manually designed controller cannot be effectively used for the tasks with the real-world complexity.

As seen in success rate, it is also difficult to find a clear trend between the parameters and completion time, but some general trend that the higher α led a lower completion time when $\beta_{const} \in (0.6, 0.7)$. Likewise, only a general trend that higher α led higher trajectory efficiency is found.

By comparing the three parameter sets that resulted the highest success rates, we chose the parameter set with $\beta_{const} = 0.7, \alpha = 1.3$ for the experiments as it resulted the fastest completion time, which is the second priority.

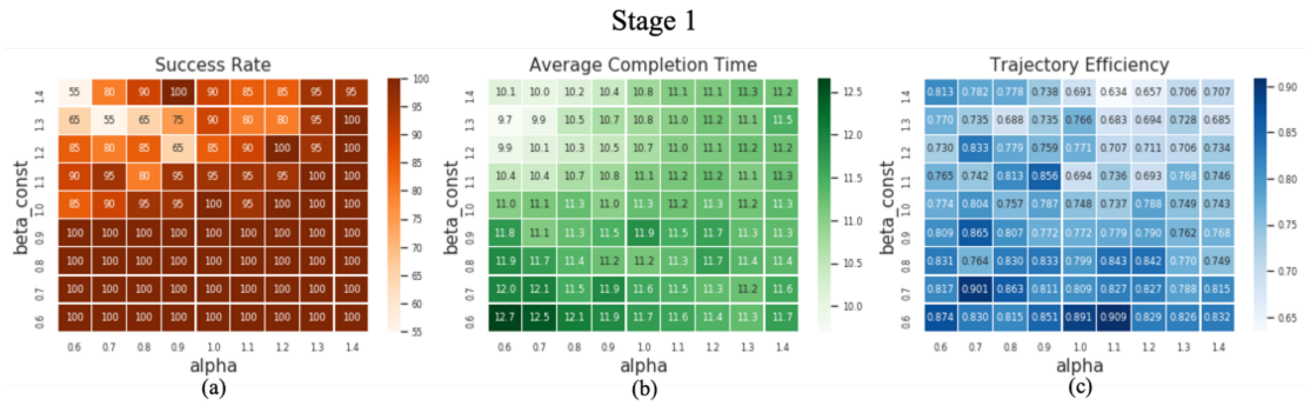


Fig. 10. Evaluation results of the hand-tuned controller with different parameter sets in Stage 1. (a), (b), (c) illustrates success rate, average completion time and trajectory efficiency respectively.

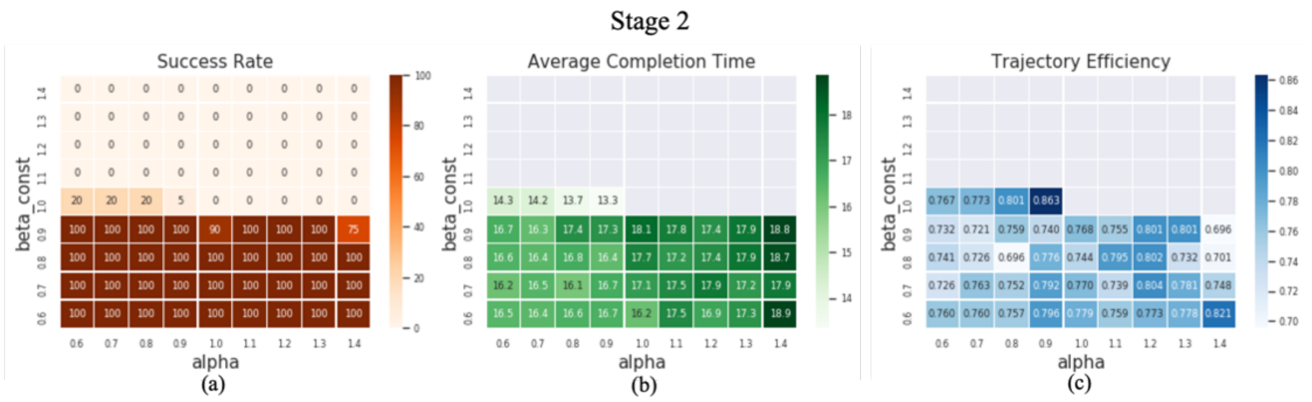


Fig. 11. Evaluation results of the hand-tuned controller with different parameter sets in Stage 2. (a), (b), (c) illustrates success rate, average completion time and trajectory efficiency respectively.

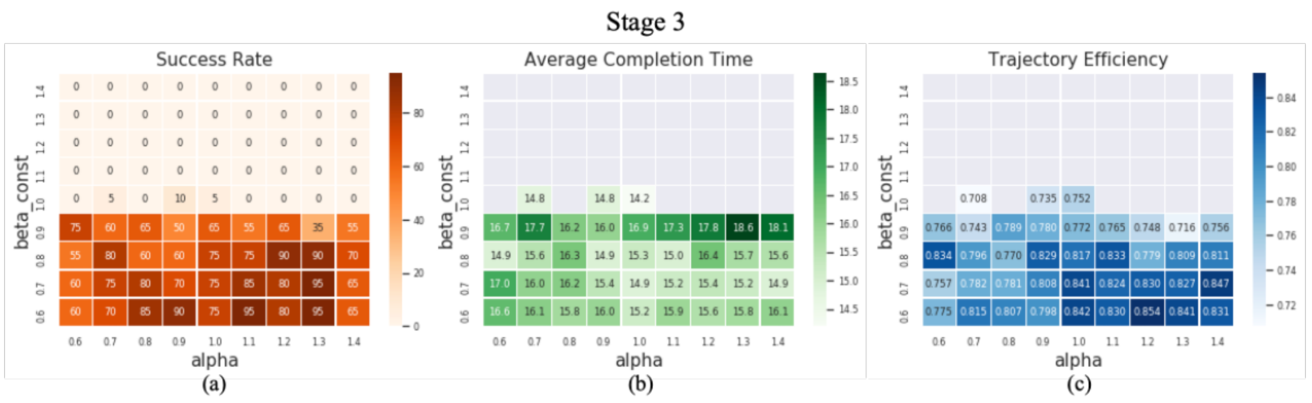


Fig. 12. Evaluation results of the hand-tuned controller with different parameter sets in Stage 3. (a), (b), (c) illustrates success rate, average completion time and trajectory efficiency respectively.